

Hochschule Merseburg  
Sommersemester 24



# Masterarbeit

Optimierung automatisierter Analysen von  
Ultraschallaufnahmen mittel Bild-Segmentierung am  
Fallbeispiel des Breast Ultrasound Images Dataset und der  
U-Net-Architektur

Bearbeiter:	Timo Bundt	Erster Betreuer:	Dr. Johannes Schwerdt
Matrikelnr.:	26636	Zweiter Betreuer:	Prof. Dr. Thomas Meier
Fach:	Automatisierungstechnik und Informatik	Datum:	27. November 2024

## Fachbereich Ingenieur- und Naturwissenschaften

Dr.-Ing. Johannes Schwerdt  
Eberhard-Leibnitz-Straße 2  
06217 Merseburg  
[johannes.schwerdt@hs-merseburg.de](mailto:johannes.schwerdt@hs-merseburg.de)

### Aufgabenstellung für die Masterarbeit

**Titel:** Optimierung automatisierter Analysen von Ultraschallaufnahmen mittel Bild-Segmentierung am Fallbeispiel des Breast Ultrasound Images Dataset und der U-Net-Architektur

**Bearbeiter:** Timo Bundt

**Themenstellender Hochschullehrer:** Dr.-Ing. Johannes Schwerdt

**2. Hochschulischer Betreuer:** Prof. Dr. Thomas Meier

#### Beschreibung der Aufgabenstellung:

1. Recherche und Beschreibung der verbreitetsten Methoden und Daten-Sätze zur automatisierten Bildsegmentierung mit Fokus auf der Krebsfrüherkennung: Recherche und Analyse der Erkennungsraten und Bewertung der Vor- und Nachteile
2. Ermitteln des Einflusses von modellunabhängigen Faktoren, bspw. verschiedener Vorverarbeitungsmethoden wie Bildgrößennormierung, unterschiedlicher Bildgrößen oder Kontrast- und Helligkeitseinstellungen, auf die Genauigkeit und Robustheit der ausgewählten Segmentierungsmethoden
3. Ermitteln des Einflusses modellabhängiger Faktoren, bspw. Tiefe und Breite der Netzwerkarchitektur, verschiedene Verlustfunktionen, Optimierungsverfahren oder Regularisierungstechniken wie Dropout und Batch Normalization, auf die Genauigkeit und Robustheit der Bildsegmentierung mittels der U-Net-Architekturen
4. Vergleich der Genauigkeit und Robustheit verschiedener U-Net Modifikationen unter Berücksichtigung der besten modellunabhängigen Faktoren und heuristisch bestimmter modellabhängigen Faktoren aus den vorherigen Analysen

abzugebende Exemplare: 2 + pdf-Datei

Prof. Dr. rer. nat. habil. Andreas Spillner  
Vorsitzender des Prüfungsausschusses

Dr.-Ing. Johannes Schwerdt  
Themenstellender Hochschullehrer

# Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit eigenständig und ohne fremde Hilfe verfasst und keine anderen Hilfsmittel als die angegebenen verwendet habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe.

Merseburg, 27. November 2024

---

**Timo Bundt**

# Abkürzungen

**BUSI** Breast Ultrasound Image

**BUSIS** Breast Ultrasound Image Segmentation

**CLAHE** Contrast Limited Adaptive Histogram Equalization

**CNN** Convolutional Neural Network

**CT** Computertomografie

**DDSM** Digital Database for Screening Mammography

**FFNN** Feedforward Neural Network

**FN** False Negative

**FP** False Positive

**KI** Künstliche Intelligenz

**MRT** Magnetresonanztomographie

**OpenCV** Open Source Computer Vision

**RMS** Root Mean Square

**SEER** Surveillance Epidemiology and End Results

**SGD** stochastische Gradientenabstieg

**TN** True Negative

**TP** True Positive

**TPU** Tensor Processing Unit

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Brustkrebs und Relevanz der Krebsfrüherkennung . . . . .	1
1.2	Forschungsfragen . . . . .	3
1.3	Gliederung der Arbeit . . . . .	4
<b>2</b>	<b>Grundlagen</b>	<b>5</b>
2.1	Mammasonographie . . . . .	5
2.2	Künstliche Intelligenz . . . . .	6
2.3	Computervision . . . . .	7
2.4	Darstellung eines künstlichen Neurons . . . . .	7
2.5	Aktivierungsfunktionen . . . . .	8
2.6	Neuronale Netze . . . . .	9
	2.6.1 Perzeptron . . . . .	11
	2.6.2 Feedforward Neural Networks . . . . .	11
2.7	Architektur des U-Net . . . . .	12
2.8	Confusion-Matrix . . . . .	15
2.9	Evaluierungsmetriken . . . . .	15
2.10	Loss-Funktionen . . . . .	17
2.11	Optimizer . . . . .	18
2.12	Hyperparameter . . . . .	19
	2.12.1 Netztiefe . . . . .	20
	2.12.2 Startfilteranzahl . . . . .	20
	2.12.3 Lernrate . . . . .	20
	2.12.4 Batch-Größe . . . . .	20
	2.12.5 Anzahl der Epochen . . . . .	21
	2.12.6 Dropout . . . . .	21
	2.12.7 Optimierung der Hyperparameter . . . . .	21
2.13	Training . . . . .	22
	2.13.1 Datenaufteilung . . . . .	22
	2.13.2 Iteratives Training . . . . .	23
	2.13.3 Validierung während des Trainings . . . . .	23
	2.13.4 Testphase . . . . .	23
2.14	Anzahl der Epochen und Early Stopping . . . . .	23
2.15	Skalierung . . . . .	24
2.16	Rauschreduzierung . . . . .	25
2.17	Bildkontrast . . . . .	26
<b>3</b>	<b>Frameworks und Datensatz</b>	<b>28</b>
3.1	TensorFlow . . . . .	28
3.2	Keras . . . . .	29
3.3	TensorBoard . . . . .	30
3.4	OpenCV . . . . .	31
3.5	Datensatz . . . . .	31
	3.5.1 BUSI-Datensatz . . . . .	31
	3.5.2 BUS Image Segmentation . . . . .	32

3.5.3	Digital Database for Screening Mammography . . . . .	33
<b>4</b>	<b>Forschungsstand</b>	<b>33</b>
4.1	U-Net++ . . . . .	33
4.2	Attention U-Net . . . . .	34
4.3	ResU-Net . . . . .	35
4.4	Dense U-Net . . . . .	36
4.5	3D U-Net . . . . .	37
4.6	Attention ResU-Net . . . . .	38
4.7	Ergebnisse . . . . .	39
<b>5</b>	<b>Methodik und Umsetzung</b>	<b>39</b>
5.1	Experimenteller Aufbau . . . . .	39
5.1.1	Verwendetes U-Net . . . . .	40
5.1.2	Implementierungsdetails . . . . .	40
5.2	Modellunabhängige Faktoren . . . . .	42
5.2.1	Bildskalierung . . . . .	42
5.2.2	Entrauschen . . . . .	43
5.2.3	Bildkontrast . . . . .	45
5.2.4	Gamma und Helligkeit . . . . .	47
5.2.5	Bildgröße . . . . .	48
5.2.6	Zusammenfassung . . . . .	49
5.3	Modellabhängige Faktoren . . . . .	50
5.3.1	Verlustfunktionen . . . . .	50
5.3.2	Batchgröße . . . . .	51
5.3.3	Dropout . . . . .	52
5.3.4	Aktivierung . . . . .	53
5.3.5	Optimizer . . . . .	53
5.3.6	Netztiefe und Breite . . . . .	54
5.3.7	Zusammenfassung . . . . .	56
5.4	Vergleich verschiedener U-Net Architekturen . . . . .	57
<b>6</b>	<b>Fazit</b>	<b>59</b>
6.1	Limitationen . . . . .	60
	<b>Literaturverzeichnis</b>	<b>63</b>

# 1 Einleitung

## 1.1 Brustkrebs und Relevanz der Krebsfrüherkennung

Der Begriff Krebs beschreibt bösartige Tumore, die durch unkontrolliertes Zellwachstum gekennzeichnet sind.[1] Diese Tumore dringen in umliegendes Gewebe ein und können Metastasen bilden. Im Gegensatz dazu sind gutartige Tumore nicht krebsartig und in der Regel nicht lebensbedrohlich. Sie entstehen zwar ebenfalls durch abnormales Zellwachstum, breiten sich jedoch nicht in umliegendes Gewebe aus, gutartige Tumore bleiben auf ihren Ursprung beschränkt und sind daher meist weniger gefährlich.[2], [3]

Krebs gehört weltweit zu den häufigsten Todesursachen und stellt daher eine der größten gesundheitlichen Herausforderungen des 21. Jahrhunderts dar. Allein im Jahr 2020 wurden etwa 10 Millionen krebsbedingte Todesfälle verzeichnet.[4]

Besorgniserregend ist dabei die steigende Tendenz der Neuerkrankungen. In Deutschland beispielsweise wurden 2018 etwa 498.000 neue Krebsfälle registriert, wobei Prognosen bis 2030 einen weiteren Anstieg um 23 % aufgrund des demografischen Wandels vorhersagen.[5]

Eine schwedische Untersuchung stellte fest, dass Brustkrebsdiagnosen zwischen 1989 und 2013 um 40 % gestiegen sind und erklärte einen Teil dieses Anstiegs ebenfalls durch die alternde Bevölkerung, da das Alter ein wichtiger Risikofaktor für Brustkrebs ist. Als weiterer Faktor, wird die Einführung der digitalen Mammographie mit hoher Erkennungsrate sowie der verstärkte Einsatz von Ultraschall und Magnetresonanztomographie (MRT) zur Brustuntersuchung und eine generell erhöhte Bereitschaft der Vorsorge genannt.[6]

Unter den verschiedenen Krebsarten nimmt Brustkrebs eine besondere Stellung ein. Mit weltweit 2,3 Millionen Fällen ist dieser die häufigste Krebsart, von der zu etwa 99 % Frauen betroffen sind.[7]–[9] Ungefähr 13.1 % aller Frauen werden innerhalb ihres Lebens mit Brustkrebs diagnostiziert werden.[10]

Der Zeitpunkt der Diagnose hat dabei einen großen Einfluss auf die Prognose der Betroffenen. Diese wird dabei als relatives 5-Jahres-Überleben<sup>1</sup> angegeben. Zwar hat sich dieses mit der Zeit stark verbessert. So betrug es für Frauen die zwischen 1975 und 1979 diagnostiziert wurden noch 74.6 % und für Frauen die 2006 diagnostiziert wurden bereits 90.6 %.[12] Nichtsdestotrotz sinkt die Überlebensrate mit jedem fortgeschrittenen Stadium erheblich:

---

<sup>1</sup> Ein relatives 5-Jahres-Überleben von 90 % bedeutet, dass fünf Jahr nach einer Krebsdiagnose der Anteil der Überlebenden 90 % des Anteils beträgt, der in der Gesamtbevölkerung gleichen Alters und Geschlechts im gleichen Zeitraum beobachtet wird.[11]

Stadium	Beschreibung[13]	5-Jahres-Überlebensrate (%) [14]–[16]
I	Tumor klein, lokalisiert	90–100
II	Tumor größer oder auf nahe Lymphknoten begrenzt	80–90
III	Starke lokale Ausbreitung	60–70
IV	Fernmetastasen	< 30

Tabelle 1: Relative 5-Jahres-Überlebensraten von Brustkrebs nach Stadien.

Den Daten der Surveillance Epidemiology and End Results (SEER) Datenbank nach sind 63 % der Brustkrebsfälle bei der Diagnose noch lokal und hauptsächlich im Stadium 1. mit wenigen Fällen im Stadium 2. 28 % der Brustkrebsfälle sind regionale Fälle mit wenigen im Stadium 2 und den meisten im Stadium 3.[10]

Dies unterstreicht die Bedeutung der Früherkennung für den Behandlungserfolg. Dabei kommen moderne bildgebende Verfahren wie MRTs, Computertomografien (CTs) und Ultraschall zum Einsatz, die eine präzise Visualisierung von Tumoren ermöglichen und Größe, Lage und Ausbreitung von bestimmen können.

Allerdings stellt die manuelle Auswertung der medizinischen Bilddaten das Personal vor nicht unerhebliche Herausforderungen. Die unvermeidliche Subjektivität sowie potenzielle Ermüdungseffekte bei der Auswertung der Bilder, aber auch die voraussichtlich weiter steigenden Fallzahlen zeigt, dass neue automatisierte Lösungs- und Unterstützungsansätze hier ein großes Potenzial besitzen.

Der Künstliche Intelligenz (KI)-Boom der letzten Jahre hat das Bewusstsein für die Möglichkeiten von KI in verschiedenen Sektoren massiv erhöht. Im medizinischen Bereich haben KI und Bildsegmentierung das Potenzial, die Untersuchung und Diagnostik mit Ultraschallgeräten deutlich zu verbessern. Sie können dabei sowohl die Effizienz als auch die Genauigkeit der Früherkennung und Überwachung von Brustkrebs steigern. KI-gestützte Algorithmen könnten Tumoren oder andere pathologische Veränderungen aus Ultraschallbildern identifizieren, indem sie Unterschiede im Gewebe erkennen. Dies reduziert die Abhängigkeit von subjektiven Einschätzungen und kann somit die Fehlerquote minimieren.

Durch eine präzisere Bildanalyse könnten selbst kleine Tumoren erkannt werden, die mit herkömmlichen Methoden gegebenenfalls unentdeckt blieben. Dies erhöht die Chancen auf eine erfolgreiche Behandlung. Darüber hinaus könnten zukünftig Möglicherweise sogar Anzeichen von Erkrankungen erkannt werden, bevor diese für einen Menschen sichtbar werden.[17]

Zusätzlich kann eine KI in diesem Einsatzgebiet eine objektivere Analyse bieten, die die Konsistenz der Diagnose zwischen verschiedenem Fachpersonal und Kliniken sicherstellen könnte. Aber auch als Lern- und Entscheidungsunterstützung für weniger erfahrene Personen dienen. Dadurch werden Fehldiagnosen reduziert und die Qualität der Versorgung verbessert.

Der Einsatz von KI-gestützter Bildsegmentierung ermöglicht eine automatische Annotation und Markierung von auffälligen Bereichen in Echtzeit. Dies könnte Fachpersonal entlasten, da manuelle Markierungen entfallen und die Zeit für die Analyse erheblich verkürzt wird. Weiter gedacht könnten Ultraschallgeräte mit integrierten KI-Systemen ausgestattet werden, die während der Untersuchung live Hinweise auf Auffälligkeiten geben und so unmittelbar auf verdächtige Regionen aufmerksam machen.

Ein vielversprechender Ansatz innerhalb der KI-gestützten Bildsegmentierung ist der Einsatz von Convolutional Neural Networks (CNNs) wie den U-Net-Architekturen. Diese haben sich aufgrund ihrer Fähigkeit, präzise Segmentierungen auch bei komplexen medizinischen Bilddaten zu ermöglichen, besonders bewährt. Ihre einfache Anpassbarkeit und Effizienz machen sie zu einer der bevorzugten Methoden in der Forschung und praktischen Anwendung, insbesondere bei der Analyse von Ultraschallbildern.[18], [19]

## 1.2 Forschungsfragen

Ziel der Arbeit ist es, eine nachvollziehbare Empfehlung zur optimalen Konfiguration eines U-Net-Modells für die Bildsegmentierung zur Erkennung von Tumoren zu entwickeln. Für die Untersuchung wird der Breast Ultrasound Image (BUSI)-Datensatz genutzt.

Dabei sollen folgende Themen und Fragen behandelt werden:

- **Analyse gängiger Methoden und Datensätze zur Bildsegmentierung in der Krebsfrüherkennung**

Welche Methoden und Datensätze sind in der aktuellen Forschung zur automatisierten Bildsegmentierung mit Fokus auf die Krebsfrüherkennung etabliert?

- **Einfluss modellunabhängiger Faktoren**

Inwiefern beeinflussen modellunabhängige Vorverarbeitungsmethoden die Segmentierungsgenauigkeit und Robustheit? Dies umfasst die Wirkung unter-

schiedlicher Techniken zur Bildskalierung, sowie Einstellungen zu Kontrast und Helligkeit.

- **Einfluss modellabhängiger Faktoren**

Wie beeinflussen modellabhängige Parameter wie Tiefe und Breite der U-Net-Architektur die Genauigkeit und Robustheit der Segmentierung?

- **Vergleich und Bewertung verschiedener U-Net-Modifikationen**

Wie schneiden verschiedene Modifikationen oder Erweiterungen der U-Net-Architektur unter Berücksichtigung der optimalen Vorverarbeitungsmethoden sowie der besten modellabhängigen Faktoren in Bezug auf Genauigkeit und Robustheit ab?

Spezifische Herausforderungen und Anforderungen, die sich aus den Besonderheiten des Datensatzes oder generell medizinischer Daten ergeben werden nicht gesondert betrachtet.

So enthält der Datensatz beispielsweise Bilder, mit unterschiedlichen Größen oder Bildausschnitten, die und unter verschiedenen Aufnahmebedingungen erstellt wurden. Die Ultraschallbilder können von variabler Qualität sein und möglicherweise Rauschen und Anmerkungen beinhalten, die die Segmentierungsgenauigkeit beeinträchtigen können.

Auch die ungleiche Verteilung der Tumortypen im BUSI-Datensatz stellt eine Herausforderung für das Modell dar.

### 1.3 Gliederung der Arbeit

Diese Arbeit gliedert sich in sechs Hauptkapitel, die die Forschungsfrage systematisch bearbeiten und die Ergebnisse der Untersuchung präsentieren:

- **Kapitel 1: Einleitung**

Dieses Kapitel führt in die Thematik der Brustkrebsfrüherkennung ein, erläutert die Relevanz der Krebsdiagnostik und skizziert die Forschungsfrage sowie die Zielstellung der Arbeit.

- **Kapitel 2: Grundlagen**

In diesem Kapitel werden die theoretischen Grundlagen der Arbeit behandelt, darunter die Funktionsweise neuronaler Netze, Bildverarbeitungsmethoden und die wichtigsten Metriken zur Evaluierung von Modellen.

- **Kapitel 3: Frameworks und Datensatz**

Es werden kurz die verwendeten Frameworks und der genutzte BUSI-Datensatz vorgestellt.

- **Kapitel 4: Stand der Forschung**

Dieses Kapitel gibt einen Überblick über verschiedene Ansätze der Bildsegmentierung mit dem BUSI-Datensatz. Es werden verschiedene Modellansätze und ihre Performance verglichen.

- **Kapitel 5: Methodik und Umsetzung**

Der experimentelle Aufbau, Preprocessing-Schritte, Modellarchitekturen und Evaluationsmethoden, werden in diesem Kapitel beschrieben.

- **Kapitel 6: Fazit und Ausblick**

In diesem Kapitel werden die wichtigsten Erkenntnisse der Arbeit zusammengefasst. Es werden die Ergebnisse der Experimente präsentiert und diskutiert, um Rückschlüsse auf die Effektivität der untersuchten Methoden zu ziehen.

Abschließend wird ein Ausblick auf mögliche zukünftige Forschungsrichtungen gegeben.

## 2 Grundlagen

### 2.1 Mammasonographie

Die Mammasonographie, auch als Brustultraschall oder Brustsonographie bekannt, ist eine diagnostische Methode in der Krebsfrüherkennung, insbesondere als Ergänzung zur Mammographie. Es handelt sich um eine nicht-invasive Technik, bei der Ultraschallwellen genutzt werden, um detaillierte Bilder von Gewebe, Zysten, Tumoren und anderen Veränderungen in der Brust zu erzeugen.[20], [21] Dies ermöglichen es, Tumore im Körper zu visualisieren und ihre Größe sowie Lage zu bestimmen.

Die Untersuchung erfolgt durch einen speziellen Schallkopf, der hochfrequente Schallwellen in das Brustgewebe sendet. Diese Wellen werden von verschiedenen Gewebestrukturen reflektiert und anschließend vom Schallkopf wieder aufgenommen. Ein Computer verarbeitet die Signale und erzeugt detaillierte Echtzeitbilder. Ein Vorteil der Mammasonographie ist, dass sie strahlenfrei ist und keinerlei Belastung für den

Körper darstellt. Die Untersuchung ist schmerzfrei und wird in Rückenlage durchgeführt.[20]

Die Mammasonographie eignet sich besonders für Frauen mit dichtem Brustgewebe<sup>2</sup>, das in der Mammographie schwer analysierbar ist. Sie spielt auch bei jüngeren Frauen, Frauen mit Brustimplantaten und in der Untersuchung von Männern[24], [25] eine wichtige Rolle. Die Methode wird zur Früherkennung von Brustkrebs, zur Überwachung bekannter Läsionen sowie zur Diagnostik von Zysten, Fibroadenomen und entzündlichen Veränderungen eingesetzt.[20], [21]

Insbesondere bei unklaren Befunden in der Mammographie liefert der Brustultraschall zusätzliche Informationen. Während die Mammographie als Standardmethode in der Brustkrebs-Früherkennung bei Frauen über 50 Jahren gilt, ergänzt die Mammasonographie diese, indem sie Veränderungen sichtbar macht, die auf einer Mammographie nicht erkennbar sind und verbessert damit die Genauigkeit der Diagnose.[10]

## 2.2 Künstliche Intelligenz

KI bezeichnet den Bereich der Informatik, der sich mit der Entwicklung von Systemen befasst, die Aufgaben lösen können, die normalerweise menschliche Intelligenz erfordern. Dazu gehören Bereiche wie maschinelles Lernen, Sprachverarbeitung und Bildverarbeitung. Eine grundlegende Definition von KI beschreibt sie als die Fähigkeit eines Systems, Daten zu analysieren, Muster zu erkennen und darauf basierend Entscheidungen zu treffen oder Aktionen auszuführen.[26]

Die Entwicklung von KI hat erhebliche Fortschritte erzielt, insbesondere durch Techniken wie Deep Learning und neuronale Netzwerke. Diese Methoden ermöglichen es Maschinen, aus großen Datenmengen zu lernen und komplexe Probleme zu lösen.[27] Neben den vielen Vorteilen, wie der Automatisierung von Prozessen und der damit verbundenen Verbesserung der Effizienz, gibt es jedoch auch Herausforderungen im Zusammenhang mit ethischen Fragen, wie etwa der möglichen Diskriminierung durch voreingenommene Daten[28], [29], der Verletzung von Urheberrechten[30] beim Zusammenstellen der Daten für das Training der Modelle oder die Wiederherstellung von Trainingsdaten aus dem Modell.[31] Aber auch der generelle Einsatz von KI für bestimmte Zwecke.[32]–[34]

---

<sup>2</sup> Ein dichtes Brustgewebe bedeutet, dass in der Brust mehr Drüsen- und Bindegewebe als Fettgewebe vorhanden ist. Dies kann die Krebsfrüherkennung erschweren, da sowohl Tumoren als auch dichtes Gewebe auf Mammogrammen weiß erscheinen, was die Unterscheidung erschwert.[22], [23]

## 2.3 Computervision

Computervision ist ein interdisziplinäres Feld der KI, mit dem Ziel, visuelle Informationen zu verstehen, zu interpretieren und zu analysieren. Es soll Computern die Fähigkeit gegeben werden, Bilder ähnlich wie der Mensch zu erfassen und zu verarbeiten.[35], [36] Durch Fortschritte in der KI-Entwicklung wird der Einsatz auch in der medizinischen Bildverarbeitung zunehmend Erfolgsversprechender. Insbesondere die Anwendung von Techniken der Computervision hat die Fähigkeit verbessert, komplexe visuelle Muster in medizinischen Bilddaten zu erkennen und zu analysieren.[18], [37]

## 2.4 Darstellung eines künstlichen Neurons

Ein künstliches Neuron ist der Grundbaustein künstlicher neuronaler Netze und ahmt die Funktionsweise biologischer Neuronen nach. Es nimmt Eingabewerte entgegen, verarbeitet diese durch eine gewichtete Summe und eine Aktivierungsfunktion und gibt anschließend einen Ausgabewert zurück.[38]

Die mathematische Berechnung der Ausgabe eines Neurons lautet:

$$y = \sigma \left( \sum_{i=1}^n w_i x_i + b \right) \quad (1)$$

[27]

Die Hauptbestandteile eines Neurons sind:

- **Eingaben ( $x_i$ ):**  
Daten, die in das Neuron eingespeist werden (z. B. Pixelwerte eines Bildes).
- **Gewichte ( $w_i$ ):**  
Jeder Eingang  $x_i$  wird mit einem Gewicht  $w_i$  multipliziert, welches die Bedeutung dieses Eingangswerts bestimmt.
- **Bias ( $b$ ):**  
Ein konstanter Wert, der der Funktion zusätzliche Flexibilität verleiht und Verschiebungen ermöglicht.
- **Aktivierungsfunktion ( $\sigma$ ):**  
Eine nichtlineare Funktion, die auf die gewichtete Summe angewendet wird, um die Ausgabe des Neurons zu berechnen.[38], [39]

## 2.5 Aktivierungsfunktionen

Aktivierungsfunktionen werden eingesetzt, da sie es den Netzwerken ermöglichen, Nichtlinearitäten in den Daten zu lernen und zu modellieren. Ohne Aktivierungsfunktionen würden neuronale Netze nur lineare Transformationen durchführen können, was ihre Fähigkeit, komplexe Zusammenhänge zu erfassen, stark einschränken würde. Diese Funktionen stellen sicher, dass das Netzwerk in der Lage ist, komplexe Muster zu erkennen und zu verarbeiten, indem sie die gewichteten Summen der Eingabewerte auf eine nichtlineare Weise transformieren.[39]

Beispiele für gängige Aktivierungsfunktionen sind:

**Sigmoid[27] :**

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

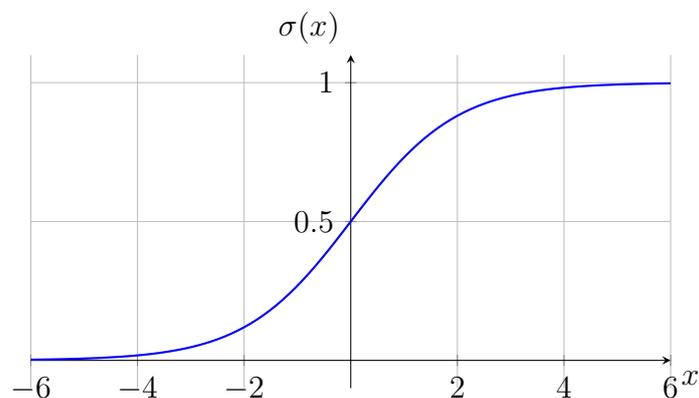


Abbildung 1: Sigmoid Funktion

**ReLU (Rectified Linear Unit)[40]:**

$$\sigma(x) = \max(0, x) \quad (3)$$

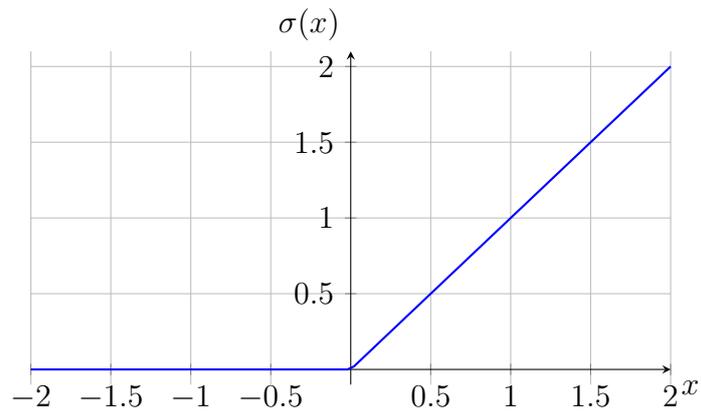


Abbildung 2: ReLU Funktion

**Tanh[41]:**

$$\sigma(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

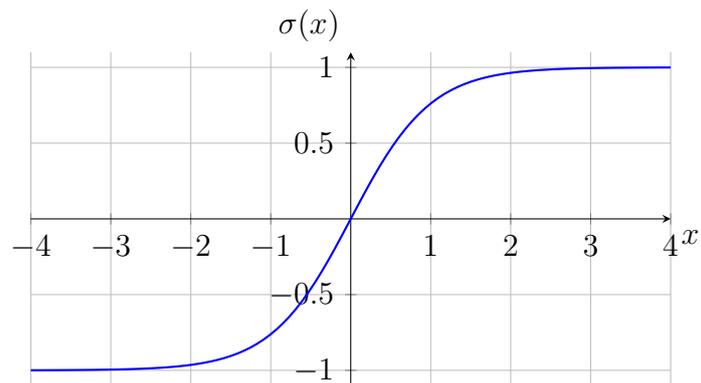


Abbildung 3: Tanh Funktion

## 2.6 Neuronale Netze

Neuronale Netze sind komplexe Berechnungsmodelle, die von der Struktur und Funktionsweise biologischer Neuronen inspiriert sind. Sie bestehen aus einer Vielzahl miteinander verbundener künstlicher Neuronen, die Informationen verarbeiten und weiterleiten. Diese Architektur ermöglicht es neuronalen Netzen, aus großen Datenmengen zu lernen und komplexe Probleme zu lösen.[39]

Die Leistungsfähigkeit neuronaler Netze basiert auf ihrer Fähigkeit, komplexe nicht-lineare Beziehungen zwischen Eingabe- und Ausgabedaten zu modellieren.

Sie bestehen aus miteinander verbundenen Neuronen oder Knoten, die in Schichten organisiert sind.[27], [39]

Diese Schichten sind:

- **Eingabeschicht:**

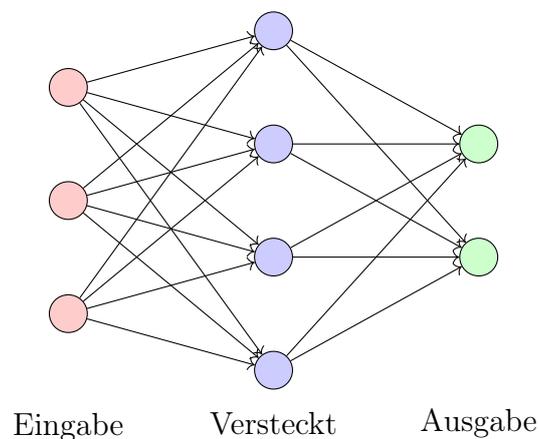
Diese Schicht empfängt die Eingangsdaten, die das Netzwerk verarbeiten soll. Jedes Neuron in dieser Schicht repräsentiert einen einzelnen Eingabewert.

- **Versteckte Schichten:**

Diese Schichten führen Berechnungen an den Eingabedaten durch. Die Verbindungen zwischen Neuronen haben sogenannte Gewichte, die bestimmen, wie stark der Wert eines Neurons auf das nächste Neuron übertragen wird. In jedem Neuron findet eine Aktivierung statt, die von einer Aktivierungsfunktion gesteuert wird.

- **Ausgabeschicht:**

Diese Schicht gibt das endgültige Ergebnis des Netzwerks zurück. In Klassifizierungsproblemen könnte dies eine Wahrscheinlichkeit für jede mögliche Klasse sein, während es in Regressionsproblemen ein numerischer Wert ist.[27], [39]



Während des Trainingsprozesses lernt das neuronale Netzwerk, indem es die Gewichte der Verbindungen so anpasst, dass der Fehler zwischen der tatsächlichen und der vorhergesagten Ausgabe minimiert wird. Dieser Prozess basiert auf Optimierungsalgorithmen wie dem Gradientenabstieg welche in dem Unterkapitel Optimizer genauer betrachtet werden.[39]

Neuronale Netzwerke werden häufig für komplexe Probleme wie Bilderkennung, Sprachverarbeitung und Zeitreihenprognosen verwendet. Dank ihrer Fähigkeit, nicht-lineare Muster in Daten zu erkennen, eignen sie sich besonders gut für die Analyse großer und unstrukturierter Datensätze wie Bilder, Audiodateien oder Textdokumente.[27]

### 2.6.1 Perzeptron

Um die Funktionsweise von KI zu verstehen, ist es sinnvoll, mit einem einfachen Netzwerk zu beginnen. Das Perzeptron ist ein grundlegendes Konzept der KI und eine der einfachsten Formen eines neuronalen Netzes. In seiner Grundform besteht es aus einem einzelnen Neuron. Diese können jedoch auch in verschiedenen Konstellationen miteinander verknüpft werden, um komplexere Netzwerke zu bilden.[42]

Die Eingabedaten  $x_1, x_2, \dots, x_n$  werden verarbeitet, indem sie mit Gewichten  $w_1, w_2, \dots, w_n$  multipliziert werden. Anschließend wird ein Bias  $w_0$  hinzugefügt. Das Ergebnis wird dann durch eine Aktivierungsfunktion  $f$  geleitet:

$$y = \sigma \left( w_0 + \sum_{i=1}^n w_i x_i \right) \quad (5)$$

[43]

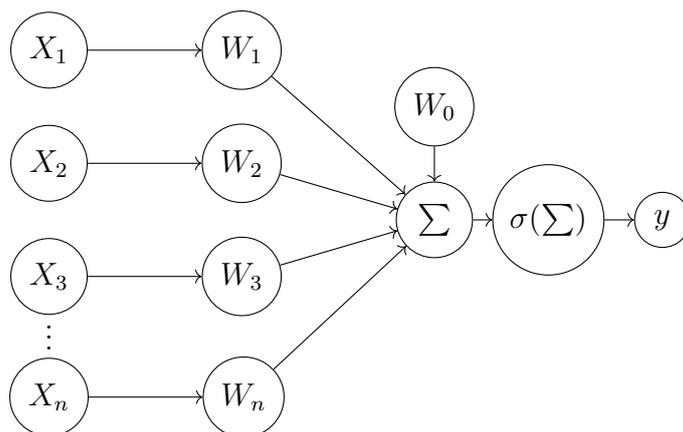


Abbildung 4: Darstellung eines künstlichen Neurons. Nach[44]

### 2.6.2 Feedforward Neural Networks

Feedforward Neural Networks (FFNNs) erweitern das Konzept des Perzeptrons, indem sie mehrere Neuronen und Schichten verwenden. Diese Netzwerke bestehen aus einer Eingabeschicht, einer oder mehreren verdeckten Schichten und einer Ausgabeschicht. Jedes Neuron in einer Schicht berechnet die gewichtete Summe seiner Eingaben, fügt einen Bias hinzu und leitet das Ergebnis durch eine Aktivierungsfunktion, um nichtlineare Beziehungen zwischen den Eingaben zu modellieren. Die Daten werden in einer Vorwärtsrichtung durch das Netzwerk propagiert, ohne Rückkopplungsschleifen[27], [39], [45], [46]:

## 2.7 Architektur des U-Net

Das U-Net, eine 2015 von Wissenschaftlern der Universität Freiburg speziell für medizinische Bildgebung entwickelte Netzwerkarchitektur, hat sich dabei als besonders vielversprechender Ansatz erwiesen auf den innerhalb der Arbeit weiter eingegangen werden soll. Seine U-förmige Struktur ermöglicht es, sowohl feine Details als auch den größeren Kontext in medizinischen Bildern zu erfassen und präzise Segmentierungen von Tumoren vorzunehmen.

Das U-Net-Modell ist besonders gut geeignet für Segmentierungsaufgaben in der biomedizinischen Bildverarbeitung. Es basiert auf einer Encoder-Decoder-Architektur mit charakteristischen „Skip-Connections“, die zwischen dem kontrahierenden und dem expandierenden Pfad verlaufen und so den Informationsfluss über die Netzwerkebenen hinweg verbessern.[19], [47]

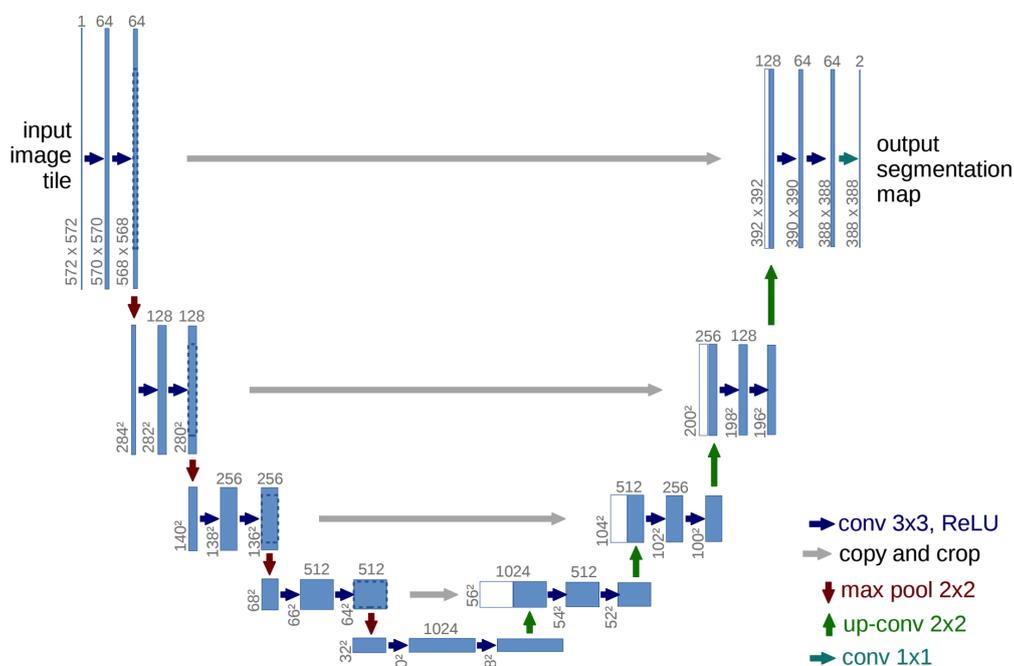


Abbildung 5: U-Net aus [47]

Das U-Net besteht aus zwei Pfaden:

- **Kontraktionspfad (Encoder):**

Der kontrahierende Pfad ist ein sequentieller Aufbau aus Konvolutionsschichten, gefolgt von Max-Pooling-Operationen. Dieser Pfad extrahiert die Merkmale der Eingabebilder und reduziert dabei schrittweise die räumlichen Dimensionen, um eine kompakte Darstellung der Bildinhalte zu erzeugen.[47]

- **Konvolutionsblöcke:**
  - \*  **$3 \times 3$  Konvolutionsschichten[39]:**
    - Sie erfassen Details aus einer sehr begrenzten Umgebung, was zu präzisen Merkmalen wie Kanten und Texturen führt.
    - Zwei aufeinanderfolgende Schichten erhöhen die Nichtlinearität und ermöglichen das Lernen komplexerer Merkmale
  - \* **Batch-Normalisierung[48], [49]:**
    - Batch-Normalisierung kann das Training von einem neuronalen Netzes erheblich beschleunigen und zu einer höheren Leistung führen.
    - Die Aktivierungen jeder Schicht werden während des Trainings für jede Batch mit ihrem Mittelwert und ihrer Standardabweichung normalisiert, wobei die globalen Statistiken anhand dieser Werte aktualisiert werden.
    - Wirkt gegen Überanpassung und verbessert die Generalisierung
  - \* **Aktivierungsfunktion:**
    - Führt Nichtlinearität in das Netzwerk ein
    - Ermöglicht das Lernen komplexer, nichtlinearer Zusammenhänge
  - \* **Spatial Dropout[50]:**
    - Deaktiviert zufällig ganze Feature-Maps
    - Verhindert die Überanpassung an räumliche Strukturen
    - Besonders effektiv bei stark korrelierten benachbarten Pixeln
- **Max-Pooling ( $2 \times 2$ )[51]:**
  - \* Reduziert die räumlichen Dimensionen um den Faktor 2 in Höhe und Breite
  - \* Versucht die wichtigsten Informationen zu erhalten
  - \* Macht die Merkmalsextraktion robuster gegenüber kleinen Verschiebungen
- **Verdopplung der Filteranzahl[39], [47]:**
  - \* Kompensiert den Informationsverlust durch Pooling
  - \* Mit mehr Filtern kann das Netzwerk komplexere und abstraktere Merkmale lernen

- \* Die unteren Schichten lernen grundlegende Merkmale wie Kanten und Texturen, während die oberen Schichten komplexere Muster und Objekte erfassen.

- **Expansionspfad (Decoder):**

Der expandierende Pfad rekonstruiert die räumliche Auflösung der Merkmale durch Up-Convolutions und verwendet Skip-Connections, die hochauflösende Informationen aus dem Encoder direkt in den Decoder einspeisen. Dies ermöglicht eine präzisere Lokalisierung der Objekte im Bild.[47]

- **Transponierte Konvolution[52]:**

- \* Lernt das Upsampling der Feature-Maps
- \* Verdoppelt die räumlichen Dimensionen: Feature-Maps werden schrittweise auf die ursprüngliche Bildauflösung hochskaliert

- **Skip-Connections[53]:**

- \* Verbinden korrespondierende Ebenen von Encoder und Decoder
- \* Kombinieren hochaufgelöste Details aus frühen Schichten mit semantischen Informationen aus späten Schichten
- \* Beschleunigen die Konvergenz
- \* Ermöglichen präzise Lokalisierung von Objektgrenzen

- **Analoge Konvolutionsblöcke:**

- \* Verarbeiten die kombinierten Feature-Maps aus Upsampling und Skip-Connection
- \* Verfeinern die räumlichen Details, die während des kontrahierenden Pfades abstrahiert wurden, für die Segmentierung

- **Halbierung der Filteranzahl[47]:**

- \* Reduziert die Komplexität des Modells schrittweise
- \* Fokussiert die Verarbeitung auf die wesentlichen Merkmale für die Segmentierung
- \* Spiegelt die Filteranzahl des kontrahierenden Pfads

iese Architektur ermöglicht es dem Netzwerk, sowohl lokale Details als auch globale Kontextinformationen zu berücksichtigen. Durch die Kombination aus Dimensionsreduktion und -expansion sowie den Skip-Connections wird ein effektiver Informationsfluss zwischen den verschiedenen Auflösungsstufen geschaffen, wodurch eine

ausgewogene Balance zwischen Detailgenauigkeit und der Erfassung des größeren Kontextes erreicht wird.[54]

## 2.8 Confusion-Matrix

Die Confusion-Matrix ist ein wichtiges Werkzeug zur Bewertung der Leistung eines Klassifikations- oder Segmentierungsmodells. Sie besteht aus einer tabellarischen Darstellung, die zeigt, wie gut ein Modell zwischen Klassen unterscheidet, und ermöglicht die Ableitung wichtiger Metriken wie Genauigkeit, Präzision, Sensitivität und Spezifität.[55]

		Vorhersage		
		Positiv	Negativ	
Tatsächlich	Positiv	True Positive (TP)	False Negative (FN)	Recall
	Negativ	False Positive (FP)	True Negative (TN)	
		Precision		

Tabelle 2: Schematische Darstellung der Confusion-Matrix für ein binäres Klassifikationsproblem mit den Metriken Recall und Precision.

- **True Positive (TP):** Korrekt als positiv klassifizierte Fälle.
- **False Positive (FP):** Fälschlicherweise als positiv klassifizierte Fälle.
- **True Negative (TN):** Korrekt als negativ klassifizierte Fälle.
- **False Negative (FN):** Fälschlicherweise als negativ klassifizierte Fälle.

## 2.9 Evaluierungsmetriken

Für die Entwicklung und Bewertung von Segmentierungsmodellen sind Qualität und Robustheit der Segmentierungsergebnisse entscheidend. Qualität bezieht sich hierbei auf die Präzision und Genauigkeit, mit der das Modell Tumorstrukturen erkennt und abgrenzt, während Robustheit die Konsistenz und Zuverlässigkeit der Segmentierung unter verschiedenen Bedingungen und Datensätzen beschreibt. Geeignete Leistungsmetriken sind essenziell, um diese beiden Aspekte systematisch zu bewerten. Sie ermöglichen den Vergleich unterschiedlicher Modelle und Konfigurationen und helfen dabei, den Fortschritt bei der Optimierung der Modelle nachvollziehbar zu dokumentieren.[39], [47], [56]

Die in der Bildsegmentierung verbreiteten Metriken zur Bewertung der Qualität, basierend auf den Einträgen der Confusion-Matrix:

1. **Genauigkeit (Accuracy):**

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

Der Anteil korrekt klassifizierter Pixel im Verhältnis zur Gesamtanzahl der Pixel.[56]

2. **Präzision (Precision):**

$$\text{Precision} = \frac{TP}{TP + FP} \quad (7)$$

Der Anteil korrekt klassifizierter positiver Pixel im Verhältnis zur Gesamtanzahl der als positiv klassifizierten Pixel.[56]

3. **Sensitivität (Recall):**

$$\text{Recall} = \frac{TP}{TP + FN} \quad (8)$$

Misst die Fähigkeit des Modells, positive Fälle korrekt zu erkennen.

Der Anteil korrekt klassifizierter positiver Pixel im Verhältnis zur Gesamtanzahl der tatsächlich positiven Pixel.[56]

4. **Spezifität (True Negative Rate):**

$$\text{Specificity} = \frac{TN}{TN + FP} \quad (9)$$

Der Anteil korrekt klassifizierter negativer Pixel im Verhältnis zur Gesamtanzahl der tatsächlich negativen Pixel.[56]

5. **F1-Score:**

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

Das harmonische Mittel aus Präzision und Sensitivität und mathematisch äquivalent zu Dice.[57]

6. **Dice-Koeffizient:** Der Dice-Koeffizient kann als

$$\frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (11)$$

berechnet werden.[58], [59]

7. **Intersection over Union (IoU):** Auch Jaccard-Koeffizient. Das Verhältnis der Überschneidungsfläche zwischen der vorhergesagten und der tatsächlichen Segmentierung und der zur vereinigten Fläche, berechnet als[60]

$$\frac{TP}{TP + FP + FN} \quad (12)$$

## 2.10 Loss-Funktionen

Loss-Funktionen messen den Unterschied zwischen den Modellvorhersagen und den tatsächlich vorliegenden Werten. Der Optimierungsprozess versucht, diese Abweichung zu minimieren, sodass das Modell die zugrunde liegende Struktur der Daten erlernt.[39]

- **Kreuzentropie-Loss**

Der Kreuzentropie-Loss ist eine weit verbreitete Loss-Funktion für Klassifikationsaufgaben und bildet auch bei der Segmentierung eine Grundlage. Die Funktion misst die Differenz zwischen den vorhergesagten und den tatsächlichen Klassenwahrscheinlichkeiten und ist besonders effektiv, wenn die Klassen eindeutig trennbar sind.

Die Formel des Kreuzentropie-Loss lautet:

$$L_{CE} = - \sum_i y_i \log(\hat{y}_i) \quad (13)$$

wobei  $y_i$  der wahre Labelwert (0 oder 1) und  $\hat{y}_i$  die vom Modell vorhergesagte Wahrscheinlichkeit für die Klasse ist. Der Kreuzentropie-Loss minimiert somit die Differenz zwischen den wahren Labels und den Vorhersagewerten und stellt sicher, dass die Klassenzuordnung möglichst präzise ist.[61]

- **Dice Loss**

Der Dice-Koeffizienten-Loss basiert auf dem Dice-Koeffizienten, der die Überlappung zwischen der vorhergesagten und der tatsächlichen Segmentierung quantifiziert.

Die Formel des Dice-Loss lautet:

$$L_{Dice} = 1 - \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (14)$$

wobei X und Y die vorhergesagten und tatsächlichen Segmentierungsmasken darstellen. Ein niedriger Dice-Loss-Wert weist auf eine hohe Überlappung hin,

was bedeutet, dass das Modell die gewünschten Strukturen im Bild gut erfasst.[61]

- **IoU-Loss** Basiert auf dem IoU Der IoU-Loss, auch bekannt als Jaccard-Loss misst das Verhältnis der Schnittmenge zur Vereinigungsmenge von Vorhersage und tatsächlichen Werten.

Die Formel für den Jaccard-Loss lautet:

$$L_{Jaccard} = 1 - \frac{TP}{TP + FP + FN} \quad (15)$$

wobei X und Y die vorhergesagte und tatsächliche Segmentierung darstellen. Ein niedriger IoU-Loss-Wert bedeutet eine hohe Übereinstimmung, was ideal für die Segmentierung komplexer und unregelmäßiger Strukturen ist.

## 2.11 Optimizer

Die Optimierung neuronaler Netzwerke ist ein zentraler Bestandteil des Trainingsprozesses, da sie bestimmt, wie gut ein Modell Muster und Zusammenhänge aus den Daten erlernen kann. Ein Optimierungsalgorithmus passt die Gewichte des Modells so an, dass der Verlust, also den unterschied zwischen den Modellvorhersagen und den tatsächlichen Daten, minimiert wird.[39]

- **Gradientenabstieg als Basisstrategie**

Der Gradientenabstieg ist die grundlegendste Optimierungsmethode, bei der die Gewichte in Richtung des negativen Gradienten des Verlusts angepasst werden. Das Ziel ist es, eine optimale Lösung zu finden, die entweder ein Minimum des Verlusts erreicht oder ihm möglichst nahekommt. Der Gradientenabstieg kann in verschiedenen Varianten implementiert werden.[62]

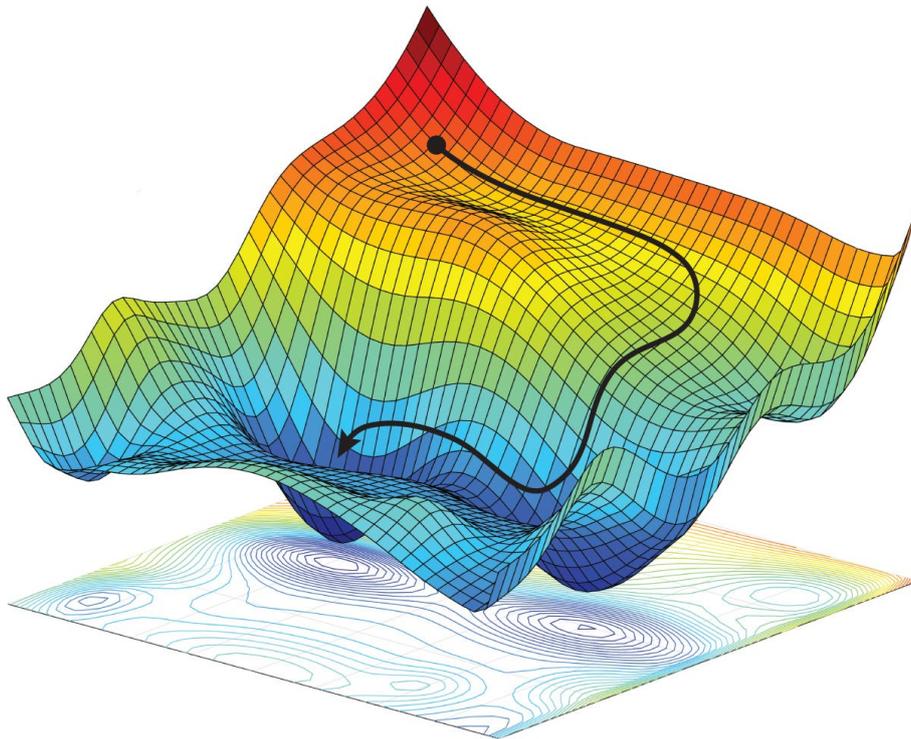


Abbildung 6: Gradientenabstieg aus [63]

- **Stochastischer Gradientenabstieg**

Eine gängige Methode zur Beschleunigung des Gradientenabstiegs ist der stochastische Gradientenabstieg (SGD), bei dem der Gradientenabstieg auf einzelnen zufällig ausgewählten Datenpunkten oder kleinen Datenmengen berechnet wird. Der Vorteil von SGD liegt in seiner Effizienz und Skalierbarkeit, da große Datenmengen verarbeitet werden können, ohne dass alle Datenpunkte gleichzeitig berücksichtigt werden müssen. Jedoch führt diese Methode zu stärkeren Schwankungen während des Trainings, was zu Instabilität führen kann.[62]

- **Der Adam-Optimizer (Adaptive Moment Estimation)**

Der Adam-Optimizer kombiniert die Vorteile des SGD mit Mechanismen zur Anpassung der Lernrate, um eine stabilere und schnellere Konvergenz zu ermöglichen. Er verwendet adaptives Lernverhalten, das den Gradienten während des Trainings dynamisch skaliert, indem Mittelwert und die Varianz der Gradienten berücksichtigt werden.[62], [64]

## 2.12 Hyperparameter

Hyperparameter sind Parameter, die im Gegensatz zu Modellparametern, die während des Trainings durch die Optimierung angepasst werden können, fest bleiben

und daher vor Beginn des Trainings festgelegt werden müssen.[39]

### **2.12.1 Netztiefe**

Die Netztiefe bezieht sich auf die Anzahl der Schichten in einem neuronalen Netzwerk. Eine höhere Netztiefe kann dazu beitragen, komplexere Merkmale aus den Eingabedaten zu extrahieren, was zu einer besseren Modellleistung führen kann. Allerdings kann eine zu große Tiefe auch zu Problemen wie Überanpassung oder längeren Trainingszeiten führen. Ein ausgewogenes Verhältnis zwischen Tiefe und Komplexität ist daher entscheidend. In der Praxis werden häufig Netzwerke mit einer mittleren Tiefe verwendet, die ausreichende Lernkapazität bieten, ohne die Trainingszeiten unnötig zu verlängern.[65]

### **2.12.2 Startfilteranzahl**

Die Startfilteranzahl bezieht sich auf die Anzahl der Filter in der ersten Faltungsschicht eines CNNs. Zu Beginn sind in der Regel relativ wenige Filter im Netzwerk vorhanden, die eine Vielzahl grundlegender Merkmale wie Kanten und Texturen extrahieren. Eine höhere Anzahl von Startfiltern ermöglicht es dem Netzwerk, vielfältigere Merkmale zu erfassen, was besonders bei komplexen Bildverarbeitungsaufgaben nützlich sein kann. Jedoch muss auch hier eine Balance gefunden werden, da eine zu hohe Anzahl an Startfiltern die Berechnungsanforderungen erheblich steigern kann.[66]

### **2.12.3 Lernrate**

Die Lernrate bestimmt die Schrittgröße, mit der die Gewichte des Modells während des Trainings angepasst werden. Eine zu hohe Lernrate kann zu instabilem Training und schlechter Konvergenz führen, während eine zu niedrige Lernrate den Lernprozess verlangsamen kann.[67]

### **2.12.4 Batch-Größe**

Die Batch-Größe gibt die Anzahl der Datenpunkte an, die pro Schritt durch das Modell geleitet werden. Kleinere Batch-Größen bieten eine detailliertere Aktualisierung der Modellgewichte, sind aber rechenintensiv und können zu verrauschteren Gradientenabschätzungen führen, da die Gewichtsanzpassung auf Basis weniger Datenpunkte erfolgt, wodurch stärkere Schwankungen in der Gradientenrichtung entstehen können. Größere Batch-Größen verbessern die Effizienz, können jedoch die Generalisierungsfähigkeit beeinträchtigen.[67]

### 2.12.5 Anzahl der Epochen

Die Anzahl der Epochen gibt an, wie oft das gesamte Trainingsset durchlaufen wird. Mehr Epochen führen zu gründlicherem Training, aber auch zu einem Risiko der Überanpassung, insbesondere bei zu wenigen Daten.[67]

### 2.12.6 Dropout

Der Dropout ist eine regulierende Technik, die häufig in neuronalen Netzwerken eingesetzt wird, um eine Überanpassung zu verhindern. Dabei werden während des Trainings zufällig eine bestimmte Anzahl von Neuronen “ausgeschaltet”, was bedeutet, dass diese während einer Trainingsiteration keine Informationen weitergeben. Dies hilft, das Modell zu generalisieren und verhindert, dass es zu stark an die Trainingsdaten angepasst wird.[39]

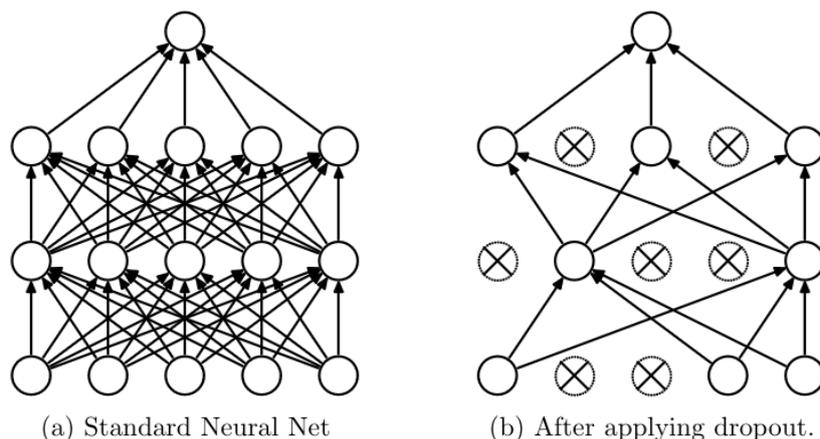


Abbildung 7: Veranschaulichung des Dropout aus [50]

### 2.12.7 Optimierung der Hyperparameter

Die optimale Einstellung von Hyperparametern ist nicht einfach und erfordert oft eine experimentelle Annäherung. Häufige Methoden zur Auswahl der besten Kombinationen sind:

- **Grid Search:**

Eine systematische Methode, bei der alle möglichen Kombinationen ausprobiert werden. Grid Search liefert umfassende Ergebnisse, kann jedoch bei vielen Hyperparametern rechnerisch aufwendig sein.[68]

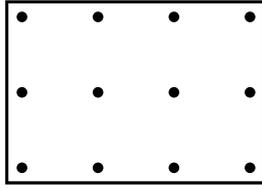


Abbildung 8: Grid Search nach [39]

- **Random Search:**

Statt jede mögliche Kombination zu testen, wählt Random Search zufällige Werte innerhalb vorgegebener Bereiche. Diese Methode ist oft effizienter und kann gute Ergebnisse bei geringerem Rechenaufwand liefern.[68]

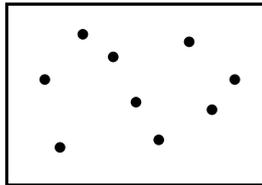


Abbildung 9: Random Search nach [39]

- **Bayesianische Optimierung:**

Ein probabilistischer Ansatz, der frühere Ergebnisse nutzt, um die nächste Kombination zu wählen. Bayesianische Optimierung ist oft effizient und besonders hilfreich bei hochdimensionalen Hyperparameterräumen.[68]

## 2.13 Training

Der Trainingsprozess eines neuronalen Netzes folgt in der Regel einem iterativen Ansatz, um die Gewichte des Modells so zu optimieren, dass der Fehler zwischen den vorhergesagten und den tatsächlichen Ausgaben minimiert wird. Dieser Prozess besteht aus den folgenden Schritten:

### 2.13.1 Datenaufteilung

Die vorhandenen Daten werden typischerweise in drei Datensätze aufgeteilt[39]:

- **Trainingsdaten:** Der größte Anteil der Daten (z. B. 70–80 %) wird verwendet, um die Modellparameter zu trainieren.
- **Validierungsdaten:** Ein kleiner Anteil der Daten (z. B. 10–20 %) wird verwendet, um die Leistung des Modells während des Trainings zu bewerten.

- **Testdaten:** Der verbleibende Anteil (z. B. 10–20 %) dient dazu, die Leistung des Modells nach dem Training zu bewerten.

### 2.13.2 Iteratives Training

Das Training erfolgt über eine festgelegte Anzahl von Iterationen, die als **Epochen** bezeichnet werden. Während jeder Epoche wird der gesamte Trainingsdatensatz durchlaufen, wobei folgende Schritte ausgeführt werden[39]:

1. **Vorhersage (Forward Pass):** Die Eingabedaten werden durch das Modell geleitet, und die Ausgabe wird berechnet.
2. **Fehlerberechnung:** Der Unterschied zwischen der Vorhersage des Modells und den tatsächlichen Werten wird durch eine Verlustfunktion berechnet.
3. **Rückwärtspropagation (Backpropagation):** Der Gradientenabstieg wird verwendet, um die Gewichte des Modells in Richtung eines geringeren Verlustes anzupassen.

### 2.13.3 Validierung während des Trainings

Nach jeder Epoche wird das Modell auf den Validierungsdaten getestet. Dadurch kann festgestellt werden, ob das Modell überanpasst (Overfitting) oder unteranpasst (Underfitting) ist. Ein typisches Zeichen für Overfitting ist eine stetige Verbesserung des Verlustes auf den Trainingsdaten, während der Validierungsverlust steigt.[69]

### 2.13.4 Testphase

Nach Abschluss des Trainings wird das Modell auf den zuvor nicht gesehenen Testdaten evaluiert. Dies liefert eine objektive Messung der Generalisierungsfähigkeit des Modells.

## 2.14 Anzahl der Epochen und Early Stopping

Die Anzahl der Epochen, die während des Trainings eines Modells durchlaufen werden, haben einen erheblichen Einfluss auf dessen Leistung. Ein zu langes Training kann zu Overfitting führen, bei dem das Modell die Trainingsdaten ‘auswendig lernt’ und seine Generalisierungsfähigkeit auf unbekannte Daten verliert. Ein zu kurzes Training hingegen kann dazu führen, dass das Modell nicht ausreichend trainiert wird und eine suboptimale Leistung zeigt.

Um Overfitting zu vermeiden, wird häufig der Ansatz des Early Stopping verwendet. Dabei wird das Training beendet, sobald der Validierungsverlust über mehrere aufeinanderfolgende Epochen nicht mehr sinkt. Ein zusätzlicher Parameter, der sogenannte Patience-Wert, definiert die Anzahl der Epochen, über die keine Verbesserung beobachtet wird, bevor das Training gestoppt wird. Dadurch wird sichergestellt, dass kleinere Schwankungen im Validierungsverlust nicht zu einem verfrühten Abbruch führen. [39], [70]

## 2.15 Skalierung

Skalierung ist die Anpassung der Bildgröße und hat einen direkten Einfluss auf die Leistung des Modells. Eine Verkleinerung der Bilder kann die Rechenzeit und die Speicherkosten reduzieren, kann jedoch die Genauigkeit reduzieren, insbesondere bei der Erkennung kleinerer Strukturen.[18], [71]

Die am häufigsten verwendeten Methoden sind[71]:

- **Nearest Neighbor Interpolation:** Bei dieser Methode wird der Wert eines neuen Pixels durch den Wert des nächstliegenden Originalpixels bestimmt. Sie ist sehr schnell, aber die Bildqualität leidet besonders bei Vergrößerungen unter Pixelartefakten.[35], [72]
- **Bilineare Interpolation:** Diese Methode nutzt die vier nächsten Pixel, um den neuen Wert eines Pixels zu berechnen. Die Methode liefert glattere Ergebnisse als die Nearest Neighbor-Methode und wird häufig für kleinere Skalierungen verwendet.[35], [72]
- **Bikubische Interpolation:** Diese Methode verwendet eine gewichtete Summe der 16 nächstgelegenen Pixel, um einen neuen Pixelwert zu bestimmen. Sie führt zu glatteren Übergängen, ist jedoch langsamer als bilineare Interpolation. Sie eignet sich besonders gut für Vergrößerungen.[35], [72]
- **Lanczos-Interpolation:** Lanczos verwendet eine Fensterfunktion (Lanczos-Kernel[73]), um besonders hochwertige Skalierungen zu erreichen, indem es über eine größere Anzahl an Nachbarpixeln interpoliert. Diese Methode ist besonders für starke Vergrößerungen oder Verkleinerungen geeignet, allerdings rechenintensiver.[35], [74]

## 2.16 Rauschreduzierung

Entrauschen oder Rauschunterdrückung ist eine wichtige Technik in der Bearbeitung von Bilddaten. Bildaufnahmeverfahren, wie etwa Aufnahmen von einem Ultraschallgerät, sind anfällig für verschiedene Arten von Rauschen, das die Bildqualität und damit die Genauigkeit von Segmentierungsmodellen beeinträchtigen kann.

Das Rauschen in Bilddaten kann durch verschiedene Faktoren verursacht werden, darunter die Bildaufnahmehardware, die Bildkompression und externe Störungen. Rauschartefakte wie Salz-und-Pfeffer-Rauschen, Gaussian Rauschen oder Poisson Rauschen sind weit verbreitet.[75] Solche Störungen können die Segmentierung von Tumoren oder anderen relevanten Strukturen erschweren, indem sie die Unterscheidung zwischen Hintergrund und Vordergrund ungenau machen.[76], [77]

Durch den Einsatz von Entrauschungsverfahren kann die Qualität eines Bildes unter Umständen erheblich verbessert werden, wodurch das Modell stabilere und präzisere Ergebnisse liefert.[78]

Es gibt verschiedene Ansätze, um das Rauschen zu reduzieren:

- **Medianfilter:** Der Medianfilter ist eine nichtlineare Methode, die häufig zur Reduzierung von Salz-und-Pfeffer-Rauschen eingesetzt wird. Bei dieser Technik wird jedes Pixel durch den Medianwert seiner benachbarten Pixel ersetzt, was hilft, extreme Ausreißer zu entfernen[75], [76], [79]
- **Gauß-Filter:** Der Gaussian Blur ist eine Weichzeichnungsmethode, die auf einer Gaußschen Verteilung basiert. Der Filter verwendet eine Gaußsche Verteilung, um die Pixelwerte zu gewichten.[79] Diese Methode ist besonders wirksam bei der Beseitigung von Gaußischem Rauschen, das durch eine normale Verteilung der Rauschwerte gekennzeichnet ist.[80]
- **Median Filter**

Der Median Filter ersetzt jeden Pixelwert durch den Medianwert seiner benachbarten Pixel. Diese Methode ist besonders effektiv bei der Entfernung von Impulsrauschen, da der Median von extremen Werten wie Ausreißern weniger beeinflusst wird als der Durchschnitt.[80]
- **Non-Local Means Denoising**

Der Non-Local Means Denoising-Algorithmus vergleicht Pixel aus verschiedenen Bildbereichen, um ähnliche Strukturen zu finden und das Rauschen zu

reduzieren, während die Details bewahrt werden. Diese Methode ist besonders nützlich bei der Reduktion von Rauschen bei gleichzeitigem Erhalt von Bilddetails.[81]

- **Bilateral Filter**

Der bilaterale Filter ist eine fortgeschrittene Methode zur Rauschunterdrückung, die sowohl den räumlichen Abstand als auch die Intensitätsähnlichkeit berücksichtigt. Dadurch wird das Rauschen entfernt, während Kanten und Details im Bild bewahrt werden. Diese Methode ist besonders nützlich, wenn feine Details erhalten bleiben müssen.[80]

- **Neuronale Netzwerke für Entrauschen:** Moderne Deep-Learning-Ansätze, wie Autoencoder und spezialisierte CNN, können trainiert werden, um Rauschen in Bildern zu reduzieren. Diese Netzwerke lernen, Rauschmuster zu identifizieren und die saubere Version des Bildes zu rekonstruieren.[76], [82]

## 2.17 Bildkontrast

Kontrast beschreibt die Unterschiede in Helligkeit zwischen verschiedenen Bildbereichen und ist ein wichtiges Kriterium für die visuelle Wahrnehmung und die Erkennbarkeit von Details in Bildern. Hoher Kontrast zeichnet sich durch deutliche Hell-Dunkel-Unterschiede aus, während ein niedriger Kontrast durch sanfte Übergänge zwischen Helligkeitswerten charakterisiert ist.[83]

Der Bildkontrast spielt eine entscheidende Rolle in der Bildverarbeitung, da er die Erkennbarkeit von Strukturen und Details in Aufnahmen erheblich beeinflusst.[78]

Mathematisch lässt sich der Bildkontrast zum Beispiel durch den Michelson-Kontrast definieren. Der Michelson-Kontrast ist eine weit verbreitete Methode, um den Kontrast von Mustern, insbesondere periodischen Reizen wie Gittermustern, zu bewerten.

$$C = \frac{I_{max} - I_{min}}{I_{max} + I_{min}} \quad (16)$$

wobei:

- $I_{max}$  den maximalen Intensitätswert im Bild und
- $I_{min}$  den minimalen Intensitätswert im Bild beschreibt.

Dieser Ansatz misst den relativen Unterschied zwischen den hellsten und dunkelsten Bereichen eines Musters.

Anders arbeitet zum Beispiel der Root Mean Square (RMS)-Kontrast. Der RMS-Kontrast ist ein Maß für die Variabilität der Helligkeit in einem Bild, das die gesamte Helligkeitsverteilung berücksichtigt. Er wird berechnet als die Standardabweichung der Pixelhelligkeitswerte:

$$C_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (L_i - \bar{L})^2} \quad (17)$$

wobei  $L_i$  den Helligkeitswert eines Pixels,  $\bar{L}$  die durchschnittliche Helligkeit und  $N$  die Anzahl der Pixel darstellt.

RMS-Kontrast wird häufig in der Analyse natürlicher Szenen verwendet, da er globale Helligkeitsunterschiede berücksichtigt und dadurch für komplexe Bildinhalte geeignet ist. Anwendungen umfassen die Effizienzbewertung von Bildverarbeitungsalgorithmen und die Optimierung der Bildqualität.[83]

## Methoden der Kontrastanreicherung

Um den Kontrast medizinischer Bilder zu verbessern und die Segmentierung zu erleichtern, stehen verschiedene Techniken zur Verfügung.

### Histogramm-Ausgleich

Der Begriff Histogramm bezieht sich auf die Verteilung der Pixelwerte, und der Ausgleich sorgt dafür, dass diese Verteilung flacher wird, wodurch mehr Helligkeitsstufen sichtbar werden. Es wird der Kontrast eines Bildes verbessert, indem die Helligkeitswerte über das gesamte verfügbare Intensitätsspektrum besser verteilt werden. Der Histogramm-Ausgleich kann mathematisch wie folgt beschrieben werden:

$$g(x, y) = \text{round} \left( \frac{L-1}{MN} \sum_{i=0}^{f(x,y)} n_i \right) \quad (18)$$

wobei:

- $g(x, y)$  die Ausgabeintensität an der Bildposition  $(x, y)$  ist,
- $L$  die Anzahl der Intensitätsstufen (typisch 256),
- $M \times N$  die Bildgröße,

- $f(x, y)$  die Eingangsintensität an Position  $(x, y)$  und
- $n_i$  die Häufigkeit der Intensität  $i$  ist.

Durch diese Umverteilung werden Bilddetails in dunklen und hellen Bereichen gleichermaßen hervorgehoben.[83]

### **Adaptive Methoden zur Kontrastverbesserung**

Ein wesentlicher Nachteil des globalen Histogramm-Ausgleichs ist, dass der Kontrast gleichmäßig anpasst und dabei lokale Bilddetails möglicherweise übersehen werden. Eine Alternative ist die Contrast Limited Adaptive Histogram Equalization (CLAHE), bei der das Bild in kleine Regionen unterteilt wird und in jeder Region ein Histogramm-Ausgleich durchgeführt wird. Dies Vermeidet Überbetonung heller Bereiche und verbessert den Kontrast in dunklen und hellen Bereichen gleichzeitig.[84], [85]

### **Gamma-Korrektur**

Eine weitere Methode zur Kontrastanpassung ist die Gamma-Korrektur. Hierbei wird jeder Pixelwert des Bildes potenziert, wodurch dunklere oder hellere Bildbereiche betont werden können. Die Transformation erfolgt durch folgende Gleichung:

$$I_{out} = I_{in}^{\gamma} \quad (19)$$

Hierbei steuert der Wert  $\gamma$  die Helligkeit und den Kontrast des Bildes. Werte kleiner als 1 erhöhen den Kontrast in dunklen Bereichen, während Werte größer als 1 den Kontrast in hellen Bereichen betonen. Somit erfordert die Gamma-Korrektur eine feine Abstimmung, da ein unpassender Gamma-Wert Details auch maskieren kann und das Verfahren ist weniger effektiv für Bilder mit großem Helligkeitsspektrum.[86]

## **3 Frameworks und Datensatz**

### **3.1 TensorFlow**

TensorFlow ist ein Open-Source-Framework von Google, das für maschinelles Lernen und tiefes Lernen entwickelt wurde. Es wurde ursprünglich von dem Google-Brain-Team erstellt und ist eines der am häufigsten genutzten Tools in der Forschung und Industrie.[87]–[89]

## Wichtige Eigenschaften von TensorFlow[87]:

- **Flexibilität:**

TensorFlow ermöglicht die Erstellung und Ausführung von Modellen für verschiedene Aufgaben wie Klassifikation, Regression oder Bild- und Spracherkennung.

- **Plattformübergreifend:**

Kann auf verschiedenen Plattformen ausgeführt werden, darunter CPU, GPU und Tensor Processing Units (TPUs), sowie Anwendungen auf Desktop, Cloud und Mobile.

- **Sprachen:**

TensorFlow unterstützt mehrere Programmiersprachen, darunter Python, C++, JavaScript und Swift.

## 3.2 Keras

Keras ist eine High-Level-Bibliothek, die ursprünglich unabhängig von TensorFlow entwickelt wurde, seit 2017 jedoch ein Bestandteil von TensorFlow ist. Keras dient als Schnittstelle, die die Erstellung neuronaler Netze wesentlich vereinfacht.

Es bietet eine intuitive API, die es ermöglicht, Modelle effizient und unkompliziert zu erstellen und zu trainieren. Gleichzeitig stellt es fortgeschrittenen Nutzern Erweiterungsmöglichkeiten zur Verfügung, um spezifische Anforderungen umzusetzen.

Keras ist modular aufgebaut, was bedeutet, dass essenzielle Komponenten wie Schichten, Optimierer und Aktivierungsfunktionen als Bausteine genutzt werden können, um komplexe Modelle zu erstellen. Darüber hinaus bietet Keras eine Vielzahl von integrierten Funktionen, die die Entwicklung beschleunigen. Dazu gehören vorgefertigte Loss-Funktionen, Optimierer sowie Callback-Funktionen wie Early Stopping, die den Trainingsprozess optimieren und überwachen.[90], [91]

Kriterium	TensorFlow	Keras
Level	Low-Level API	High-Level API
Flexibilität	Sehr flexibel, aber komplex	Einfach, weniger Flexibilität
Komplexität	Erfordert tiefgehende Kenntnisse	Benutzerfreundlich
Integration	Unterstützt Keras als Bestandteil	Funktioniert innerhalb von TensorFlow

Tabelle 3: TensorFlow und Keras[91]

### 3.3 TensorBoard

TensorBoard ist ein Visualisierungstool, das in Verbindung mit TensorFlow verwendet wird, um den Trainingsprozess von neuronalen Netzen zu visualisieren und zu überwachen. Es bietet eine interaktive Benutzeroberfläche, um Metriken, Modelle und andere wichtige Aspekte des Trainingsprozesses zu analysieren.

#### Funktionen von TensorBoard

Eine Auswahl der Funktionen die TensorBoard bietet:

- **Visualisierung von Metriken:**

Trainings- und Validierungsmetriken wie zum Beispiel Verlust und Genauigkeit können in Echtzeit angezeigt werden.

- **Modellgraphen:**

TensorBoard stellt den Graphen des neuronalen Netzes dar, sodass die Architektur und Verbindungen überprüft werden können.

- **Histogramme:**

Verteilung und Änderungen der Modellparameter können als Histogramme angezeigt werden, um die Modellanpassung zu analysieren.[92]

#### Integration von TensorBoard

Die Nutzung von TensorBoard ist sehr unkompliziert gestaltet. Um TensorBoard in den Trainingsprozess zu integrieren, sind lediglich folgende Schritte erforderlich:

- **Einrichten eines Log-Directories:**

Ein Verzeichnis wird erstellt, in dem TensorBoard die Trainingsdaten speichert.

- **Hinzufügen von Callbacks:**

Während des Trainingsprozesses wird ein TensorBoard-Callback verwendet, um Metriken und Modelle in das Log-Verzeichnis zu schreiben.

- **Starten des TensorBoard-Servers:**

TensorBoard kann lokal oder in der Cloud gestartet werden, um die Visualisierungen über den Browser aufzurufen.

```

1     from tensorflow.keras.callbacks import TensorBoard
2
3     # Log-Verzeichnis festlegen
4     log_dir = "logs/fit/" + datetime.datetime.now().strftime("%Y%m%d-%H%M%S")
5
6     # TensorBoard-Callback erstellen
7     tensorboard_callback = TensorBoard(log_dir=log_dir, histogram_freq=1)
8
9     # Modelltraining mit TensorBoard
10    model.fit(
11        x_train,
12        y_train,
13        validation_data=(x_val, y_val),
14        epochs=10,
15        callbacks=[tensorboard_callback])

```

Abbildung 10: Integration von TensorBoard. Nach [93]

## 3.4 OpenCV

Open Source Computer Vision (OpenCV) ist eine weit verbreitete Open-Source-Bibliothek, die eine Vielzahl von Werkzeugen und Funktionen für die Bildverarbeitung und Computer Vision bietet. Ursprünglich von Intel entwickelt, wird OpenCV mittlerweile in vielen Forschungs- und Anwendungsbereichen eingesetzt.

Für die Bildsegmentierung mit U-Net-Modellen bietet OpenCV wichtige Vorverarbeitungsfunktionen wie das Laden, Skalieren und Normalisieren von Bildern sowie Techniken zur Bildaugmentierung.[35]

## 3.5 Datensatz

### 3.5.1 BUSI-Datensatz

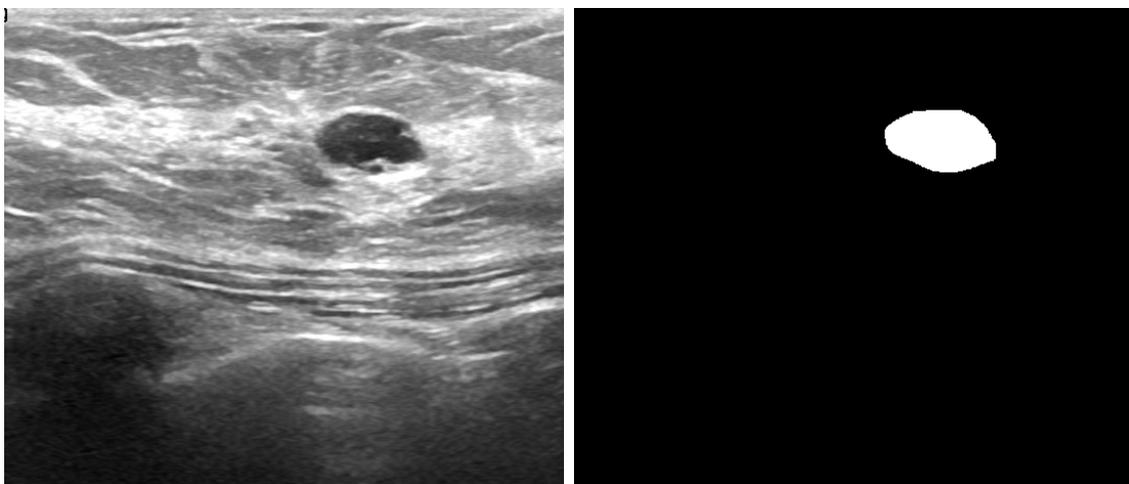
Für die Bewertung und das Training von Bildsegmentierungsmodellen in der Krebserkennung in Ultraschallaufnahmen ist der BUSI-Datensatz eine der zentralen Ressourcen. Dieser Datensatz stellt eine breite Auswahl an Ultraschallbildern zur Verfügung, die eine Grundlage für die automatisierte Klassifikation und Segmentierung in der Brustkrebsdiagnose bieten.[94]

Der BUSI-Datensatz ist ein öffentlich verfügbarer Datensatz, der zur Klassifikation und Segmentierung von Brustultraschallbildern genutzt wird. Der Datensatz besteht aus 780 Ultraschallbildern von 600 Patientinnen in einem Alter zwischen 20 und 75 Jahren die in drei Klassen aufgeteilt sind[95]:

Kategorie	n	Beschreibung
Normal	133	Bilder von gesunden Brustgeweben ohne Anzeichen von Tumoren.
Benign	437	Bilder von gutartigen Tumoren.
Malignant	210	Bilder von bösartigen Tumoren (Krebs).

Tabelle 4: Einteilung der Bilder im Datensatz.

Zusätzlich zu den Bilddaten stellt der BUSI-Datensatz Segmentierungsmasken bereit, die den Bereich eines Tumors im Bild markieren und zum Training des Modells und der Kontrolle der Segmentierungsergebnisse genutzt werden.



(a) Begniner Tumor in Ultraschallaufnahme

(b) Maske der Aufnahme

Abbildung 11: Vergleich zwischen Ultraschallaufnahme und der zugehörigen Maske

### 3.5.2 BUS Image Segmentation

Der Breast Ultrasound Image Segmentation (BUSIS)-Datensatz umfasst 562 Ultraschallbilder, die durch die Zusammenarbeit von Wissenschaftlern und Forschern mehrerer Institutionen mit unterschiedlichen Ultraschallgeräten erstellt wurden.

Die Generierung der Masken erfolgte in einem standardisierten Prozess durch vier Radiologen. Drei Radiologen markierten die Tumorgrenzen unabhängig voneinander, während ein erfahrener Radiologe die Ergebnisse überprüfte und gegebenenfalls anpasst.

Der BUSIS-Datensatz diente als Benchmark zur Quantifizierung und zum Vergleich der Leistung von 14 Segmentierungsmethoden. Das standard U-Net erreichte in diesem Test eine Dice-Koeffizienten von 0.9.[96] Der Datensatz ist allerdings nicht öffentlich zugänglich sondern der Zugriff muss angefragt werden.

### 3.5.3 Digital Database for Screening Mammography

Häufig stößt man im Zusammenhang von Bildsegmentierung und Brustkrebs noch auf die Digital Database for Screening Mammography (DDSM).

Die DDSM ist eine Sammlung von 2.620 Mammographie-Bildern, die speziell für die Forschung in der medizinischen Bildverarbeitung und die Entwicklung computergestützter Diagnoseverfahren erstellt wurde. Diese Aufnahmen unterschieden sich jedoch von Ultraschallaufnahmen weshalb die DDSM für Brustkrebssegmentierung in Ultraschallbildern weniger hilfreich ist.

## 4 Forschungsstand

In den letzten Jahren hat die automatisierte Segmentierung medizinischer Bilder durch den Einsatz von Deep-Learning-Methoden erhebliche Fortschritte gemacht. Die kontinuierliche Verbesserung von Netzwerkarchitekturen hat dazu beigetragen, dass die Bildverarbeitung auch in medizinischen Anwendungen immer präziser und robuster wird. Ein bedeutender Meilenstein in der medizinischen Bildsegmentierung war die Einführung des U-Net-Modells durch Ronneberger et al.[47] im Jahr 2015.[18], [19], [97] Diese Architektur wurde speziell für die Segmentierung biomedizinischer Bilder entwickelt und zeichnet sich durch eine symmetrische, U-förmige Struktur aus. Ein Hauptvorteil des U-Net ist seine Fähigkeit, hochaufgelöste Segmentierungsergebnisse selbst bei kleinen Datensätzen zu erzielen.

Seit seiner Einführung hat es zahlreiche Erweiterungen und Abwandlungen erfahren, die die Architektur an verschiedene Anwendungen und Anforderungen angepasst haben. Hier sind einige Weiterentwicklungen:

### 4.1 U-Net++

Das U-Net++ führt eine Reihe von Zwischenschichten (Nested U-Nets) zwischen den Encoder- und Decoder-Pfaden ein. Diese Zwischenschichten verbessern die Informationsweitergabe und ermöglichen eine feinere Segmentierung. Das U-Net++ verwendet außerdem ein überarbeitetes Skip-Verbindungssystem, um die Lernfähigkeit zu steigern.

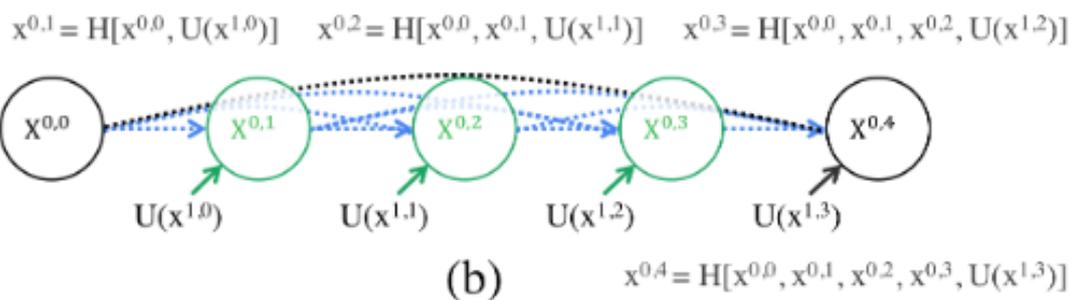
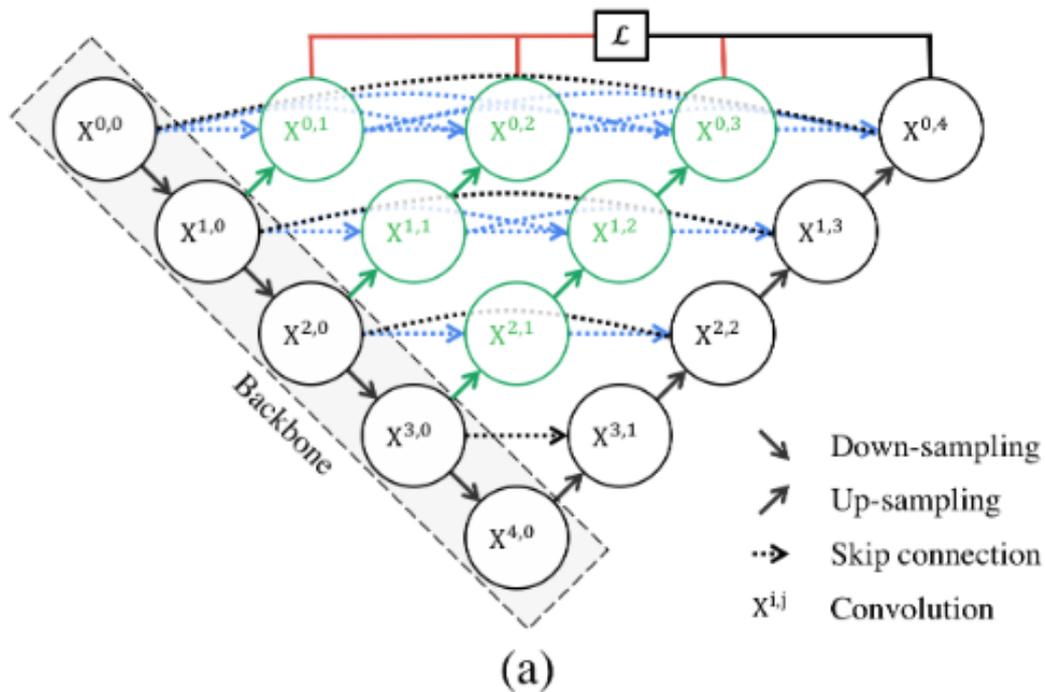


Abbildung 12: Aus [19]

## 4.2 Attention U-Net

Diese Variante integriert Aufmerksamkeitsmodule (Attention Modules) in die Architektur, die es dem Modell ermöglichen, relevantere Bildregionen stärker zu gewichten. Dieses Netz ist besonders nützlich in Szenarien, in denen der Kontrast zwischen Ziel- und Hintergrundregionen gering ist, wie es bei Ultraschallbildern der Fall sein kann.

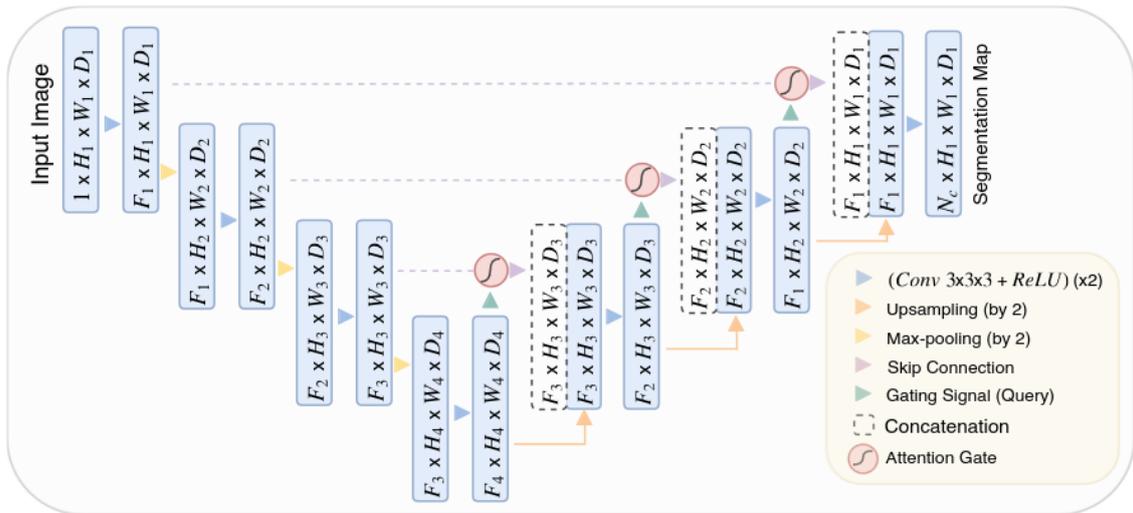


Abbildung 13: Attention u-Net. Aus [98]

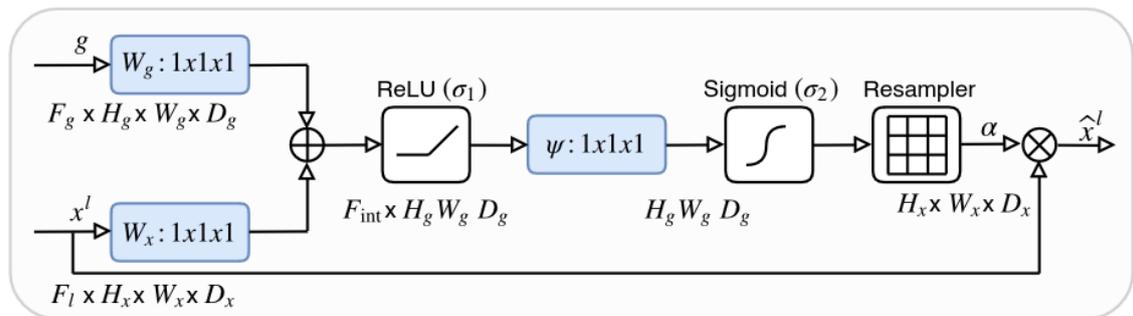


Abbildung 14: Attention Gate. Aus [98]

### 4.3 ResU-Net

Das ResU-Net kombiniert das U-Net mit Residual-Blöcken.

Die Verwendung von Residual-Blöcken macht ResU-Net besonders geeignet für die Verarbeitung komplexer Bilddaten mit fein strukturierten Mustern, da sie helfen, wichtige Merkmale auch in tieferen Schichten der Netzarchitektur zu bewahren. [99], [100]

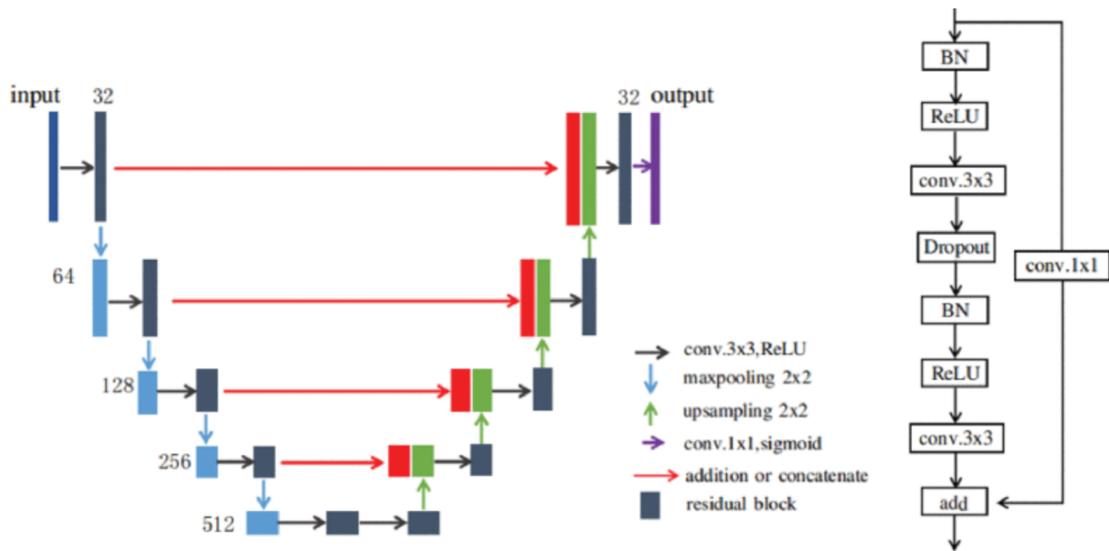


Abbildung 15: ResU-Net und Residual Block. Aus [101]

#### 4.4 Dense U-Net

Diese Variante nutzt DenseNet-Blöcke, um die Wiederverwendung von Merkmalen zu maximieren. Diese dichte Konnektivität verbessert die Effizienz und Genauigkeit, da sie eine stärkere Informationsweitergabe zwischen Schichten ermöglicht.

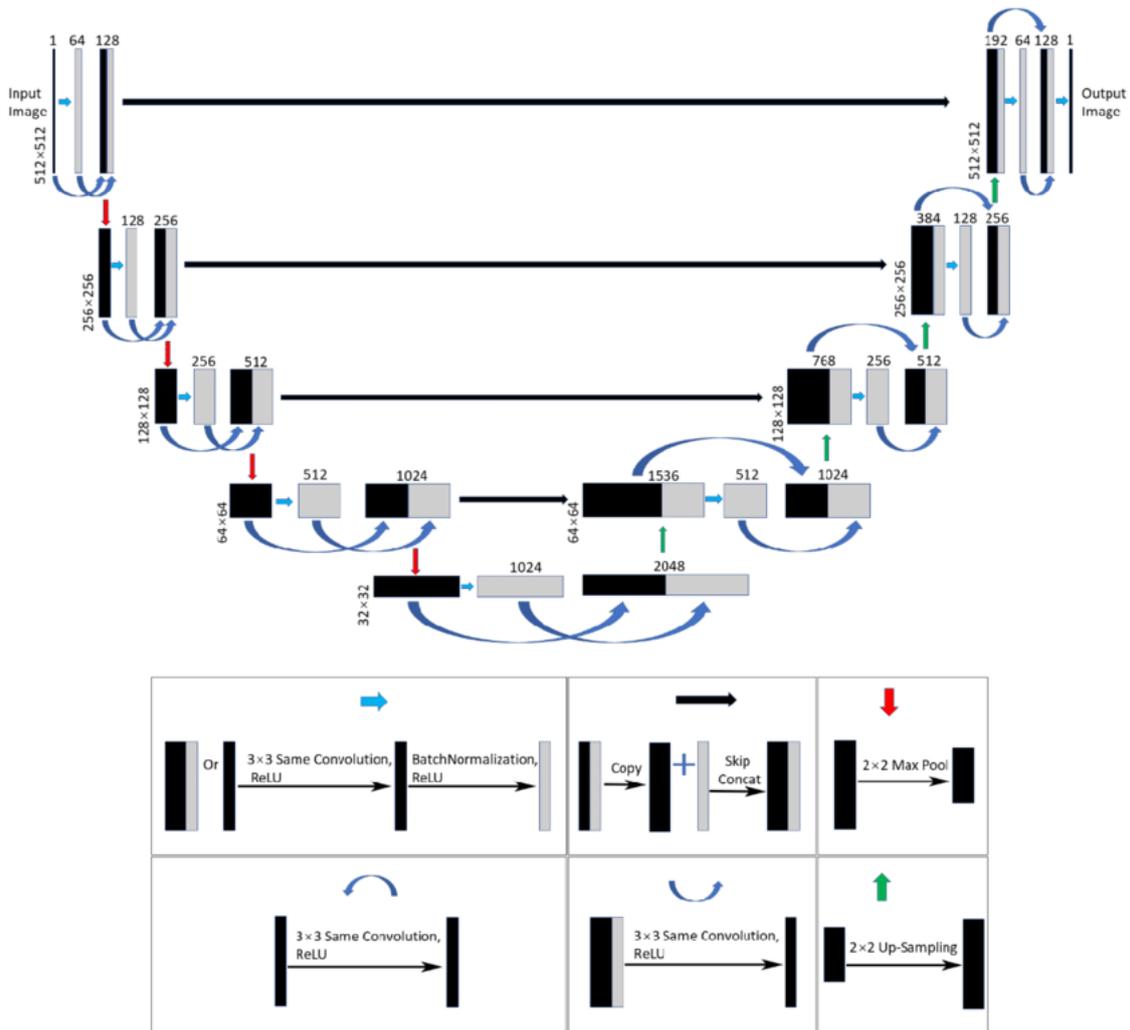


Abbildung 16: Dense U-Net. Aus [102]

## 4.5 3D U-Net

Für Anwendungen wie die Segmentierung volumetrischer Daten (z. B. in der Radiologie) wurde U-Net in 3D erweitert. 3D U-Net verarbeitet räumliche Informationen in drei Dimensionen und ist daher besonders gut für Aufgaben geeignet, bei denen die Kontextinformationen in allen Raumdimensionen relevant sind.

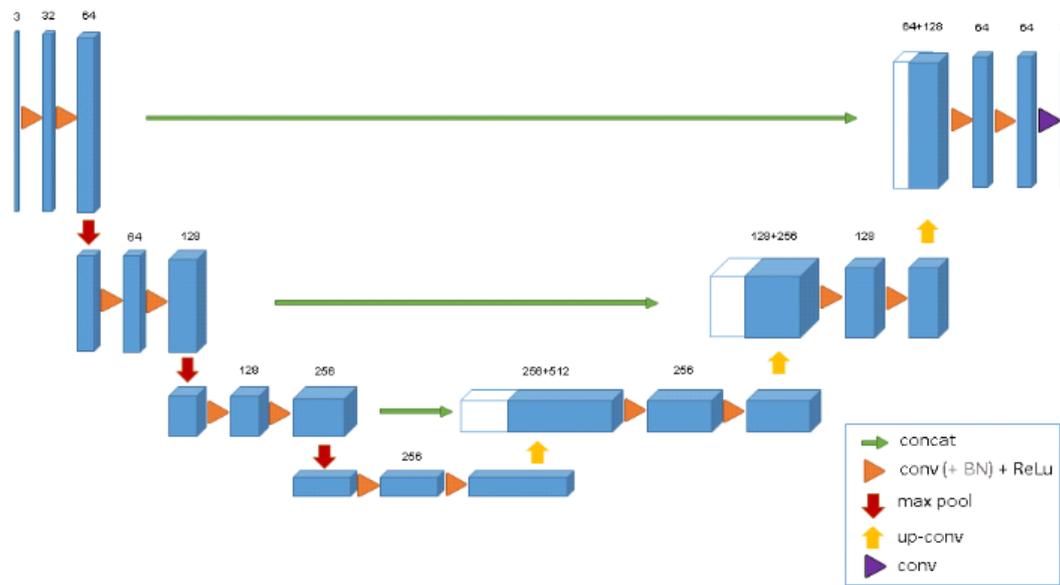


Abbildung 17: 3D U-Net. Aus [49]

## 4.6 Attention ResU-Net

Diese Architektur kombiniert Residual-Blöcke und Attentiongates. Dadurch wird nicht nur die Modellstabilität erhöht, sondern auch die Fähigkeit, spezifische Bereiche eines Bildes hervorzuheben, deutlich verbessert.

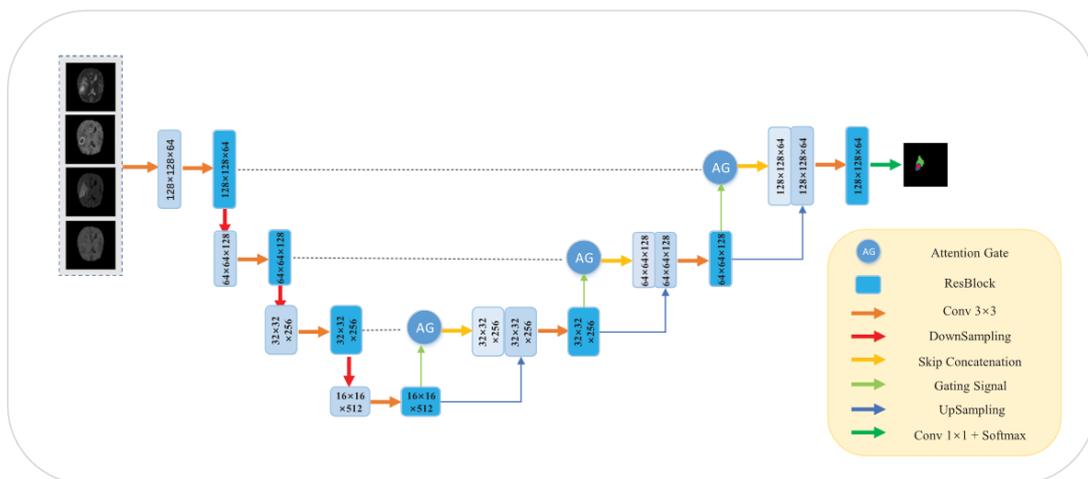
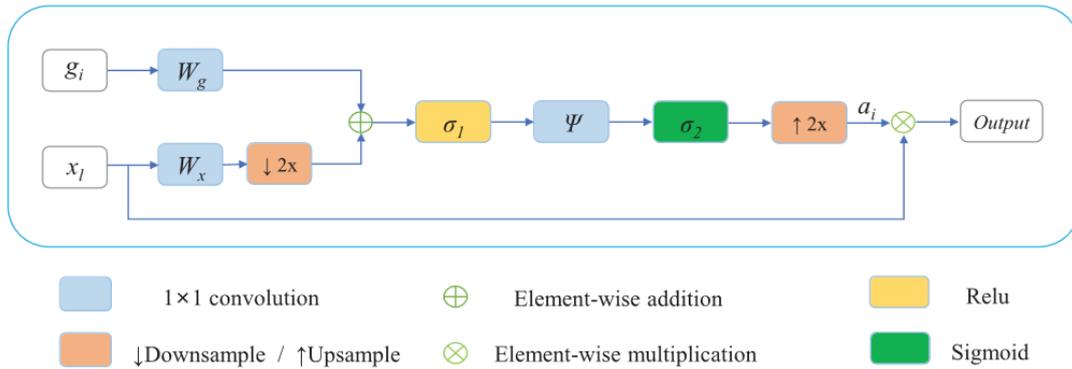


Abbildung 18: Attention ResU-Net. Aus [103]



**FIGURE 2.** The basic schematic of attention gate. The  $g_i$  and  $x_l$  are represented as the gating signal vector and the feature map of the layer  $l$ , respectively. The  $\sigma_1$  and  $\sigma_2$  denote the Relu function and Sigmoid function.  $W_x$ ,  $W_g$  and  $\psi$  are linear transformations. The  $\alpha_i$  indicates the attention coefficient.

Abbildung 19: Attention Gate. Aus [103]

## 4.7 Ergebnisse

Die Ergebnisse der verschiedenen Netze schwanken über unterschiedlichen Publikationen stark. Dies liegt an kleinen Unterschieden im Netz oder dem Umgang mit den Daten. Daher wurden die Ergebnisse hier als ein Bereich in folgender Tabelle zusammengefasst.

Modell	Architektur	Dice-Koeffizient (%)	Quellen
U-Net	Encoder-Decoder mit Skip-Connections	75-89	[104]–[107]
Attention U-Net	U-Net + Aufmerksamkeitsmodule	85-92	[104]–[106], [108]
ResU-Net	U-Net + ResNet-Module	77-85	[107]
SegNet	Encoder-Decoder ohne Skip-Connections	81	[109]
U-Net++	Erweiterte U-Net Architektur	77-85	[106], [107]
U-Net3+	kombiniert Features aus allen Encoder-Ebenen mit jeder Decoder-Ebene	77-88	[105], [107]
TransUNet	Transformer-basierte U-Net Variante	91	[105]

# 5 Methodik und Umsetzung

## 5.1 Experimenteller Aufbau

Im Rahmen dieser Arbeit wurde ein experimenteller Aufbau entwickelt, mit dem die Leistung verschiedener U-Net Modellen bei der Tumorerkennung in Ultraschallbildern ausgewertet werden kann. Ziel ist es, die Effektivität verschiedener Bildverarbeitungsmethoden und Modelle für die Brustkrebsfrüherkennung zu untersuchen und deren Leistung anhand des Dice-Koeffizient und IoU bewertet.

Der experimentelle Aufbau wird in den folgenden Unterkapiteln beschrieben, wobei zunächst die Auswahl der Modellarchitekturen und die Preprocessing-Schritte erläutert werden.

### 5.1.1 Verwendetes U-Net

Für die Untersuchungen wurde ein U-Net verwendet, dessen Architektur sich an der ursprünglichen U-Net Implementierung orientiert.

Kategorie	Parameter	Wert
Modellarchitektur	Modellname	<code>unet</code>
	Anzahl U-Net-Blöcke	5
	Initializer	<code>he_normal</code>
	Dropout-Rate	0.5
	Basisanzahl Filter	32
	Endanzahl Filter	1
Bildparameter	Bildgröße	(64, 64)
	Bildkanäle	1
	Anzahl Klassen	2
	Skalierungsmethode	<code>bicubic</code>
	Denoising	<code>none</code>
	CLAHE	<code>false</code>
Trainingsparameter	Epochen	2000
	Batch-Größe	32
	Validierungssplit	0.2
	Testgröße	0.2
	Zufallszustand	42
	Optimierer	<code>adam</code>
	Verlustfunktion	<code>dice</code>
	Metriken	<i>Accuracy, Precision, Recall, AUC, Dice, IoU, usw.</i>
Callback-Parameter	Nur bestes Modell speichern	<code>true</code>
	Model Checkpoint Verbose	1
	Model Verbose	2
	Patience	500

Tabelle 5: Übersicht der Standard-Modellparameter

### 5.1.2 Implementierungsdetails

Für das Pythonskript, welches den Testablauf steuert wurde Python 3.12 zusammen mit TensorFlow 2.18.0 und Keras 3.6.0 genutzt.

Die folgende Tabelle gibt eine Übersicht über die Hauptmodule des Programms, ihre Funktionalitäten und ihre Hauptaufgaben:

Tabelle 6: Übersicht der Programmmodule und deren Funktionalitäten.

Modul	Beschreibung
<b>Datenvorbereitung</b>	Lädt Datensätze, führt Normalisierung und Vorverarbeitungen durch. Die Vorbereitung gewährleistet, dass die Daten für das Modelltraining geeignet sind. Es ist wichtig zu beachten, dass Vorverarbeitungsaufgaben wie die Kontrastanpassung oder das Entrauschen von Bildern vor der Skalierung durchgeführt werden. Viele der angewendeten Methoden basieren auf umliegenden Pixeln. Beispielsweise verwendet die CLAHE einen Bereich von 8x8 Pixeln, um den Kontrast anzupassen. Nach der Skalierung könnte diese Methode möglicherweise nicht mehr sinnvoll angewendet werden.
<b>Modellarchitektur</b>	Definiert die Architektur des Modells, basierend auf einer modifizierten U-Net-Struktur.
<b>Training</b>	Implementiert den Trainingsprozess.
<b>Evaluation</b>	Analysiert die Ergebnisse des Modells, berechnet Metriken wie Dice-Koeffizient und IoU, und generiert Visualisierungen der Segmentierungsergebnisse.
<b>Ausgabe und Speicherung</b>	Speichert die modellierten Ergebnisse, wie segmentierte Bilder und Metriken, und erstellt eine abschließende Dokumentation des Prozesses.

Der Testlauf kann über ein in der Config.py definiertes Test-Array angepasst werden. Innerhalb dieses Arrays werden die Parameter als Tupel übergeben, wobei der erste Eintrag den zu ändernden Parameternamen angibt und der zweite eine Liste der zu testenden Werte enthält. Aus diesen Angaben werden alle Permutationen der Konfigurationen generiert und anschließend nacheinander getestet.

```

1 self.test_array = [
2     ("img_size", [(512, 512)]),
3     ("filters_base", [64, 128]),
4     ("num_unet_blocks", [5, 6]),
5 ]

```

Die Vorhersageergebnisse des Modells werden nach Abschluss des Trainings in zwei JSON-Dateien gespeichert:

- **Full:** Enthält die Metriken aller Vorhersagen, darunter Dice-Koeffizient, IoU

(Intersection over Union) und Pixel Accuracy.

- **Summary:** Enthält statistische Kennzahlen wie `mean`, `median`, `std`, `min`, `max`, `q1` und `q3`.

Zusätzlich werden verschiedene Informationen in einem mit TensorBoard abrufbaren Format bereitgestellt, um eine detaillierte Analyse der Trainings- und Validierungsergebnisse zu ermöglichen.

## 5.2 Modellunabhängige Faktoren

Im folgenden Abschnitt werden verschiedene vom Modell unabhängiger Faktoren analysiert und deren Auswirkungen auf die Segmentierungsgenauigkeit untersucht.

### 5.2.1 Bildskalierung

Die Verkleinerung eines Bildes kann zu einer Verschlechterung der Bildqualität durch einen Verlust von Details führen, was wiederum die Segmentierungsergebnisse beeinträchtigen könnte.

Daher ist es wichtig die Auswirkung der verschiedenen Skalierungsmethoden auf die Segmentationsergebnisse zu überprüfen.

Es wurden verschiedene Skalierungsmethoden verwendet, um die Eingabedaten in unterschiedlichem Maße zu verkleinern. Anschließend wurde das U-Net mit den skalierten Daten trainiert. Am Ende wurden die Resultate der unterschiedlichen Trainingsdurchläufe miteinander verglichen.

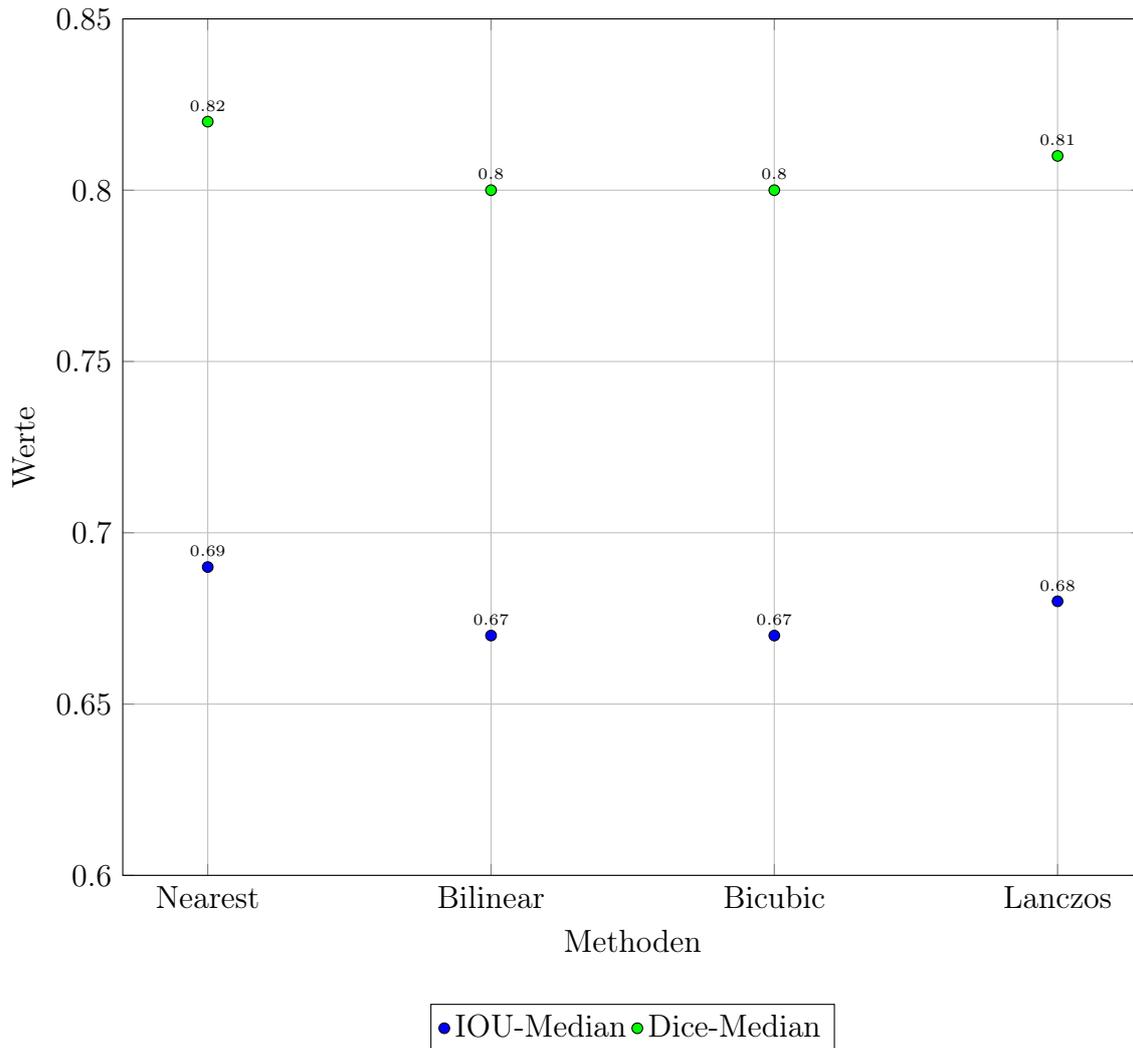


Abbildung 20: Vergleich der Dice-Median und IOU-Median Werte für verschiedene Methoden bei einer Verkleinerungen auf 32x32 Pixel

Es zeigt sich, dass bei einer Skalierung auf 32x32 Pixel kein großer Unterschied zwischen den Methoden besteht. Bei weniger stark skalierten Bildern wird dieser Unterschied immer kleiner, da die Skalierungsmethode immer weniger relevant wird.

### 5.2.2 Entrauschen

Es wurde innerhalb der Arbeiten im Kapitel Stand der Forschung öfter erwähnt, dass die Modelle unterschiedliche Leistungen zeigen:

- Bei gutartigen (benigne) Tumoren: Höhere Genauigkeit aufgrund klarer Grenzen
- Bei bösartigen (maligne) Tumoren: Deutlich schwierigere Segmentierung
  - Mehr Rauschen[109]

- Unklarere Tumorformen
- Geringere Segmentierungsgenauigkeit

[105]

Daher könnte ein Entrauschen die Qualität der Bilder verbessern und eine verbesserte Bildqualität kann die Fähigkeit des Modells erhöhen, wichtige Bildmerkmale zu erkennen, was zu einer höheren Genauigkeit und robusteren Segmentierungen führen könnte.

Die Leistung der Entrauschungsmethode hängt von der Art des Rauschens und den spezifischen Anforderungen der Segmentierungsaufgabe ab. Eine zu starke Entrauschung kann möglicherweise zu einem Verlust von wichtigen Bilddetails führen, während unzureichendes Entrauschen unnötiges Rauschen im Bild lässt.

Das Entrauschen wurden in Kombination mit den zuvor Besprochenen Skalierungsmethoden getestet. Hierbei ergab sich bei einer Skalierung zu  $16 \times 16$  Pixeln ein interessanter Effekt. Bei dieser Bildgröße scheiden der bilaterale Filter und die nearest Skalierung in jeder Kombination im Verhältnis zu den anderen Methoden gut ab und führt gemeinsam zu dem Modell mit dem höchsten Dice-Koeffizienten.

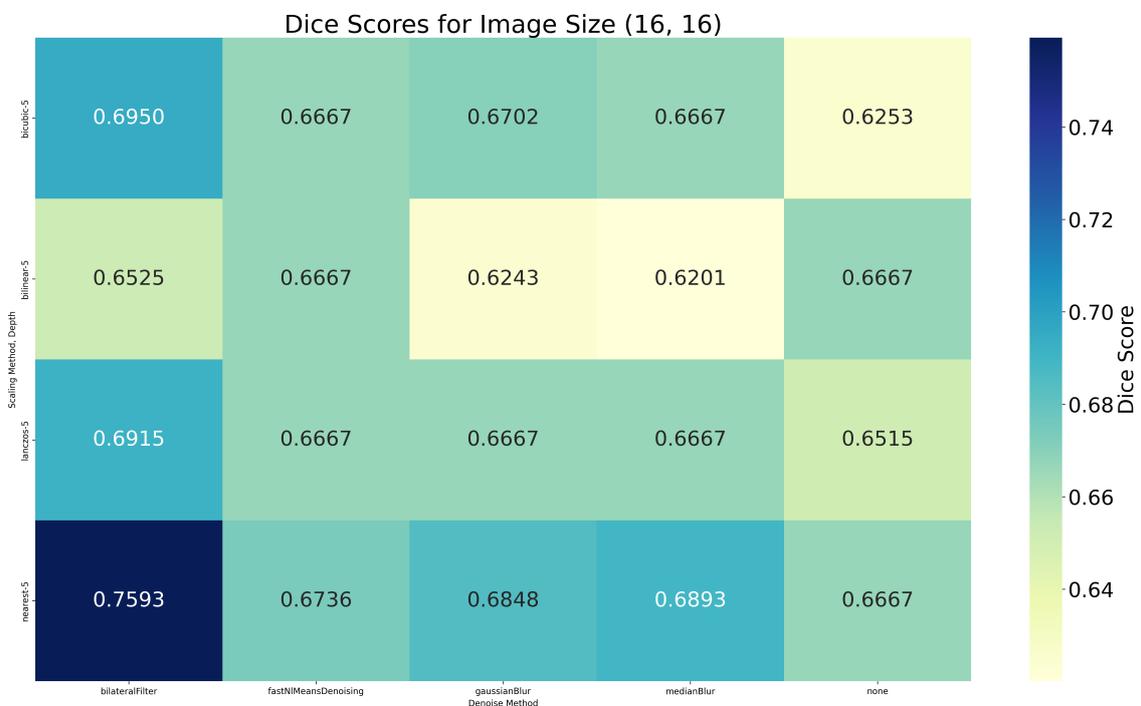


Abbildung 21: Dice-Koeffizient bei Denoise und Skalierungs kombinationen.

Allerdings lässt dieses Ergebnis sich nicht auf größere Bilder übertragen.

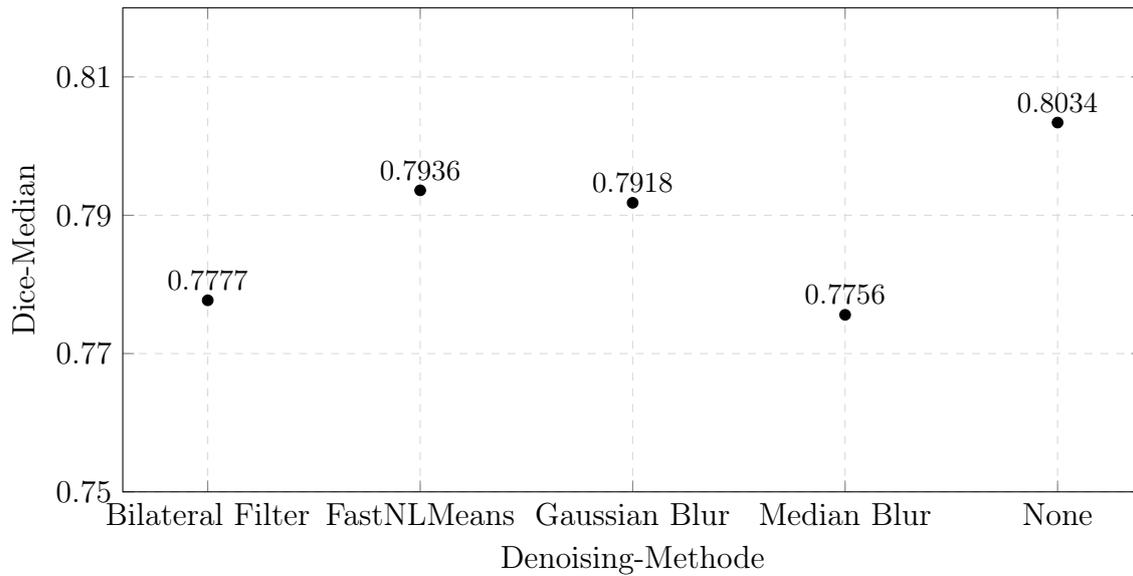


Abbildung 22: Scatterplot der Dice-Median-Werte für verschiedene Denoising-Methoden mit Scaling Nearest(Bildgröße: 128x128).

Nichtsdestotrotz unterstreichen diese Ergebnisse die Relevanz der beiden Techniken, insbesondere bei der Arbeit mit stark skalierten Bildern. Wird hingegen nicht mit stark skalierten Bildern gearbeitet, scheint das Entrauschen nicht nur wenig Nutzen zu bringen, sondern kann die Leistung des Modells sogar beeinträchtigen.

### 5.2.3 Bildkontrast

Eine Erhöhung des Kontrast könnte zu einer besseren Erkennbarkeit von Strukturen führen und so den Dice-Koeffizienten des Modells erhöhen.

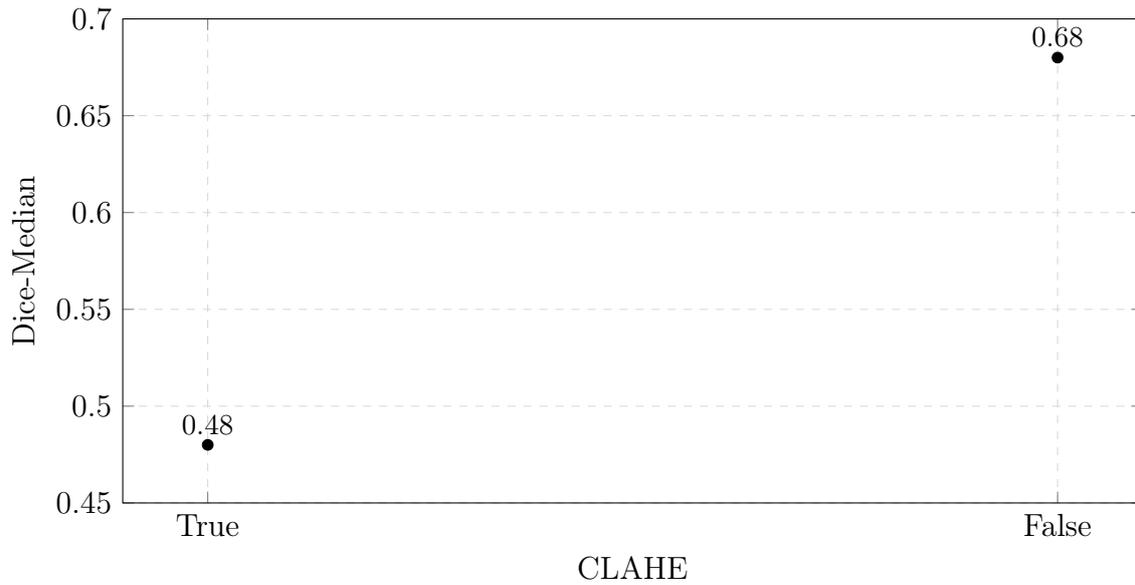


Abbildung 23: Scatterplot der Dice-Median-Werte für die Anwendung von CLAHE (Bildgröße: 16x16).

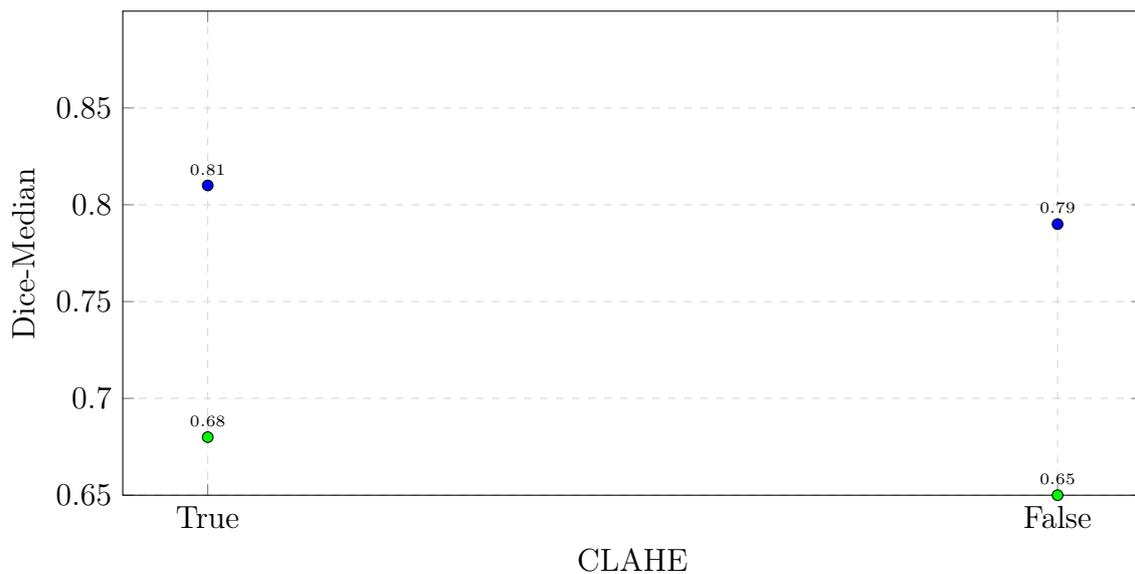


Abbildung 24: Scatterplot der Dice-Median-Werte für die Anwendung von CLAHE (Bildgröße: 64x64).

Die Ergebnisse zeigen, dass selbst adaptive Methoden zur Kontrasterhöhung, wie die CLAHE, die Leistung der Modelle negativ beeinflussen können. Daher sollte eine Kontrastanpassung mit Bedacht und vorzugsweise manuell bei der Zusammenstellung oder Qualitätssicherung des Datensatzes erfolgen.

### 5.2.4 Gamma und Helligkeit

Die Anpassung von Helligkeit und Gamma hat das Potenzial, die Bildmerkmale so zu verändern, dass sie besser vom Modell erkannt werden. Eine präzisere Erkennung von Strukturen führt tendenziell zu einer höheren Segmentierungsleistung.

Es wurden verschiedene Kombinationen von Werten getestet. Hierbei ist der Offset ein fester Wert, der zur Helligkeit jedes Pixels addiert wird und der Multiplikator skaliert die Helligkeit der Pixelwerte des Bildes. Gamma-Korrektur kann Kontraste in bestimmten Helligkeitsbereichen betonen. Niedrigere Gamma-Werte betonen dunklere Töne, höhere Gamma-Werte verstärken hellere Töne.

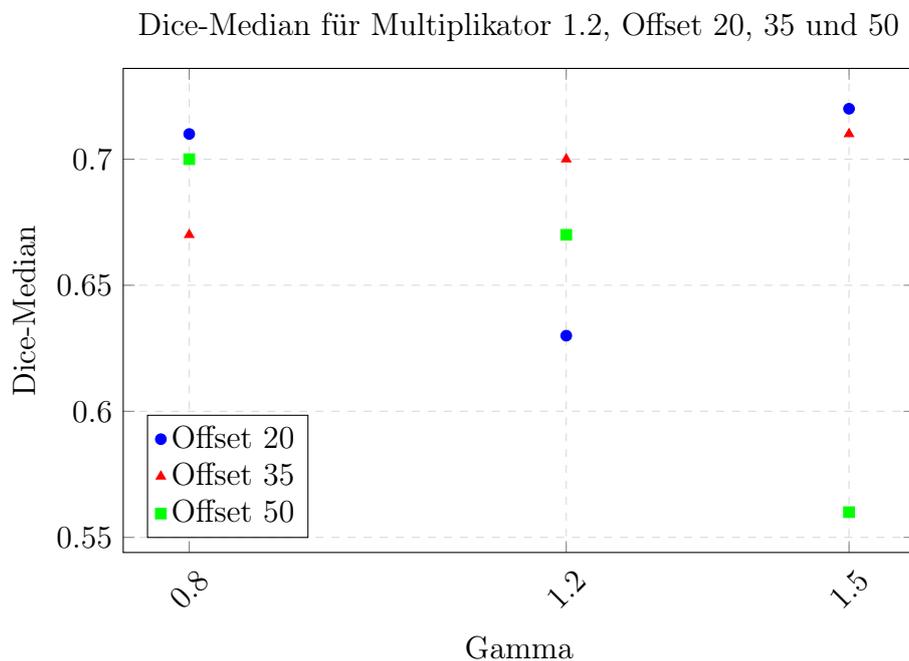


Abbildung 25: Dice-Median für verschiedene Helligkeitsmultiplikatoren und Offsets bei Gamma-Werten 0.8, 1.2 und 1.5. (64x64)

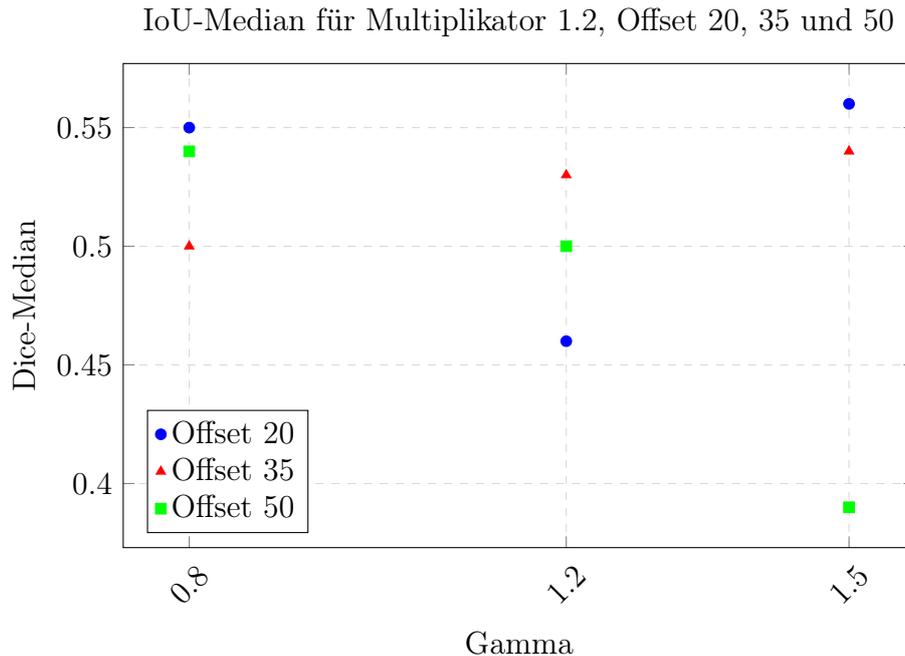


Abbildung 26: IoU-Median für verschiedene Helligkeitsmultiplikatoren und Offsets bei Gamma-Werten 0.8, 1.2 und 1.5. (64x64)

Diese Daten deuten darauf hin, dass ein zu hoher Gamma-Wert in Kombination mit einer Helligkeitsanpassung zu einem Verlust an Details führen könnte, was sich negativ auf die Segmentierung auswirkt.

Basierend auf den Daten scheint die Kombination aus Multiplikator 1.50 und Offset 35 mit Gamma 0.8 die beste Leistung zu bieten, da der höchste Dice-Median-Wert erreicht wurde. Dies deutet darauf hin, dass diese Kombination von Bildhelligkeit und Kontrastanpassung für die Segmentierung am günstigsten ist.

Die besten Ergebnisse wurden bei einer moderaten Helligkeitsanpassung und niedrigen Gamma-Werten erzielt. Zu starke Kontrastanpassungen (hohe Gamma-Werte oder hohe Offsets) scheinen die Leistung der Modelle zu beeinträchtigen, was bedeutet, dass eine feine Abstimmung der Helligkeitseinstellungen wichtig ist, um die optimale Segmentierungsleistung zu erreichen.

### 5.2.5 Bildgröße

Als Modelunabhängiger Faktor soll die Bildgröße betrachtet werden. Größere Bilder enthalten in der Regel detailliertere Informationen und könnten daher die Qualität eines Modells verbessern. Auf der anderen Seite erfordern größere Bilder mehr Rechenleistung und Speicher, was zu längeren Trainingszeiten führen kann. Das Ziel dieser Untersuchung ist es, den Einfluss verschiedener Bildgrößen auf die Modellqualität zu evaluieren.

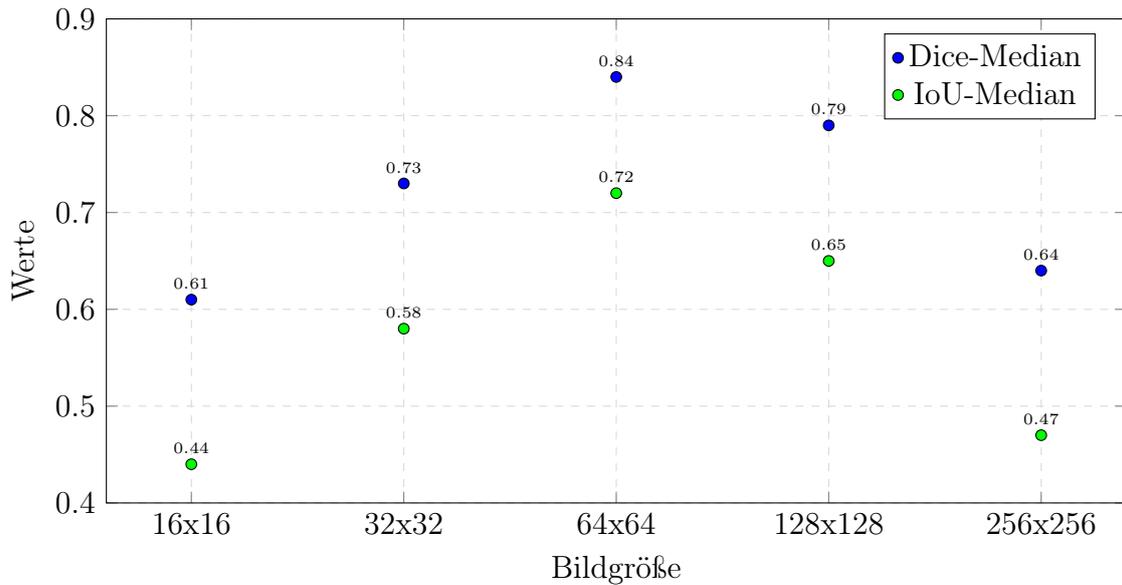


Abbildung 27: Vergleich der Dice- und IoU-Median-Werte für unterschiedliche Bildgrößen.

Die Verbesserung der Segmentierungsleistung bis zu einer gewissen Bildgröße ist zu erwarten, da mit zunehmender Bildgröße mehr Informationen im Bild enthalten sind, die vom Netzwerk genutzt werden können. Bei einer Bildgröße von bis zu 64x64 Pixel steigt die Leistung, da das Modell mehr Details aus den größeren Bildern extrahieren kann.

Ab einer Bildgröße von 128x128 Pixel scheinen jedoch die zusätzlichen Informationen die Kapazität des U-Net-Modells zu übersteigen, was zu einer Verschlechterung der Segmentierungsleistung führt. Es ist wahrscheinlich, dass ein tieferes oder breiteres Netzwerk, das mehr Parameter und Kapazitäten bietet, in der Lage wäre, mit diesen größeren Bildgrößen besser umzugehen und somit die Leistung weiter zu steigern. Ein größeres oder komplexeres Modell könnte die Segmentierungsleistung bei größeren Bildern verbessern, indem es in der Lage ist, mehr Details zu verarbeiten und gleichzeitig besser zu generalisieren. In diesem Zusammenhang wäre es sinnvoll, zusätzlich zur Bildgröße auch die Netzwerkarchitektur anzupassen, um die bestmögliche Leistung für unterschiedliche Bildgrößen zu erzielen.

### 5.2.6 Zusammenfassung

Zusammenfassend lässt sich sagen, dass keine der getesteten Vorverarbeitungsmethoden einen signifikanten Einfluss auf die Modelleistung hatte. Es sollte jedoch darauf geachtet werden, dass die Größe der verwendeten Bilder zur Architektur des U-Net-Modells passt, damit alle relevanten Informationen im Bild effizient vom

Netzwerk genutzt werden können. Es ist entscheidend, dass die Bildgröße so gewählt wird, dass sie genug Details enthält, um die Modellleistung zu maximieren, ohne das Modell mit zu vielen unnötigen Informationen zu überlasten.

## 5.3 Modellabhängige Faktoren

In diesem Unterkapitel werden die verschiedenen modellabhängigen Aspekte analysiert und ihre Auswirkungen auf die Segmentierungsqualität anhand des Dice-Koeffizient bewertet.

### 5.3.1 Verlustfunktionen

Die Verlustfunktion trägt dazu bei, dass das Modell die relevanten Muster und Strukturen in den Eingabedaten korrekt lernt, was zu einer höheren Genauigkeit und besseren Performance auf neuen, ungesehenen Daten führt.

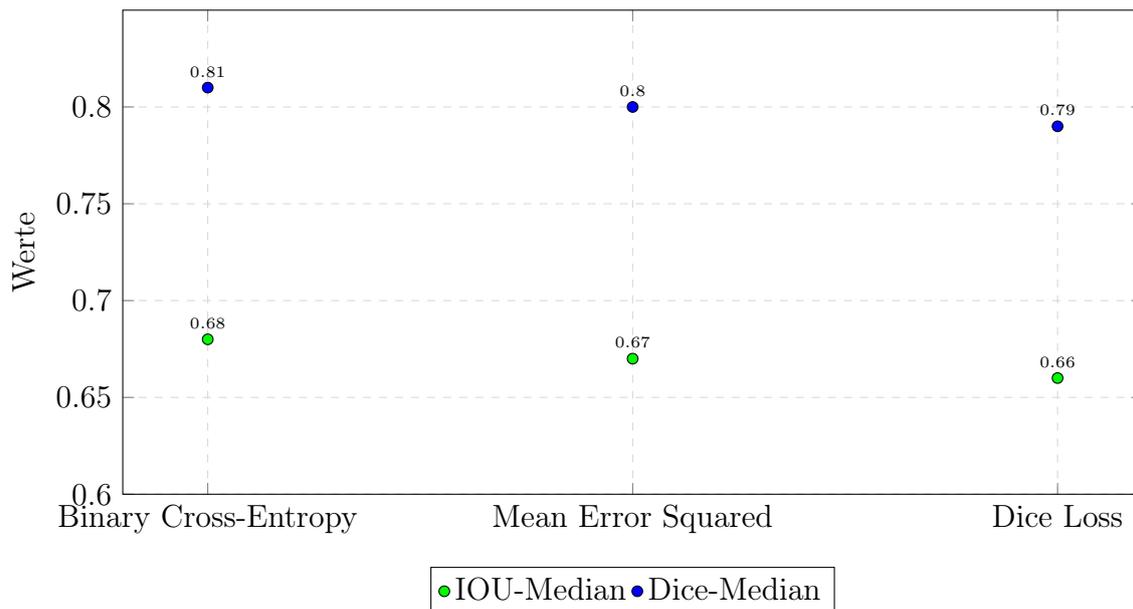


Abbildung 28: Vergleich der Metriken für Binary Cross-Entropy und Dice Loss (Dice-Median und IoU-Median).

Da die Unterschiede zwischen den getesteten Verlustfunktionen minimal sind, lässt sich keine eindeutige Empfehlung abgeben.

### 5.3.2 Batchgröße

Die Batchgröße beeinflussen direkt die Trainingszeit eines Modells. Kleine Batchgrößen benötigen mehr Epochen, da sie pro Epoche weniger Daten verarbeiten, während größere Batchgrößen die Verarbeitung pro Schritt beschleunigen, aber auch mehr Speicher erfordern. Es ist entscheidend, eine Balance zwischen Geschwindigkeit und Speicherbedarf zu finden.

Batchgrößen können sich auf die Modellgenauigkeit auswirken. Zu kleine Batchgrößen führen oft zu ungenauen Gradienten, während zu große Batchgrößen den Gradientenabstieg weniger effektiv machen können. Das Ziel ist es, die Batchgröße so zu wählen, dass sie zu einer guten Modellgeneralisation führt.[39]

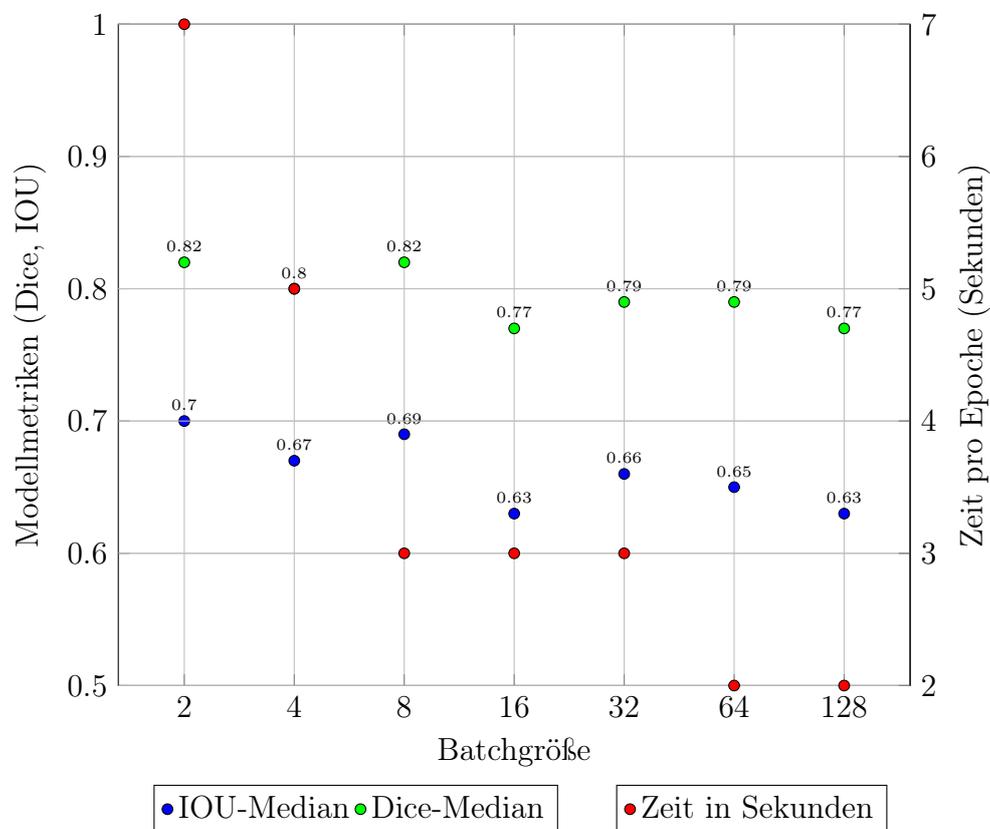


Abbildung 29: Vergleich der Batchgröße, Zeit pro Epoche und Modellmetriken (Dice-Median, IOU-Median).

Die Grafik zeigt, dass der Dice-Median-Wert bei Batchgrößen zwischen 2 und 8 relativ hoch bleibt, mit einem leichten Rückgang bei größeren Batchgrößen (32, 64, 128). Besonders bei einer Batchgröße von 16 sinkt der Dice-Median relativ stark (von 0.82 auf 0.77). Die Zeit pro Epoche sinkt mit zunehmender Batchgröße (von 7 Sekunden bei Batchgröße 2 auf 2 Sekunden bei Batchgröße 128), was zeigt, dass größere Batchgrößen zu einer schnelleren Berechnung führen, jedoch mit einem potenziellen

Verlust in der Modelleleistung.

Die Daten zeigen, dass kleinere Batchgrößen die besten Ergebnisse in Bezug auf Dice-Median und IoU-Median liefern, während größere Batchgrößen ( $> 32$ ) zwar schneller sind, aber die Modelleleistung verschlechtern können. Es scheint sinnvoll zu sein, eine Batchgröße von 8 zu wählen, um eine gute Balance zwischen Trainingsgeschwindigkeit und Modellgenauigkeit zu erreichen.

### 5.3.3 Dropout

Die Wahl der richtigen Dropout-Rate ist wichtig, da ein zu hoher Wert das Modell möglicherweise daran hindert, ausreichend aus den Trainingsdaten zu lernen, was zu einer schlechten Leistung führen kann. Andererseits kann ein zu niedriger Wert das Modell anfällig für Überanpassung machen.

Deshalb ist es sinnvoll, die Auswirkungen unterschiedlicher Dropout-Werte auf die Modelleleistung zu testen. Dies ermöglicht es, den optimalen Wert zu finden, der eine Balance zwischen einer guten Generalisierungsfähigkeit und der Fähigkeit, von den Trainingsdaten zu lernen, bietet.

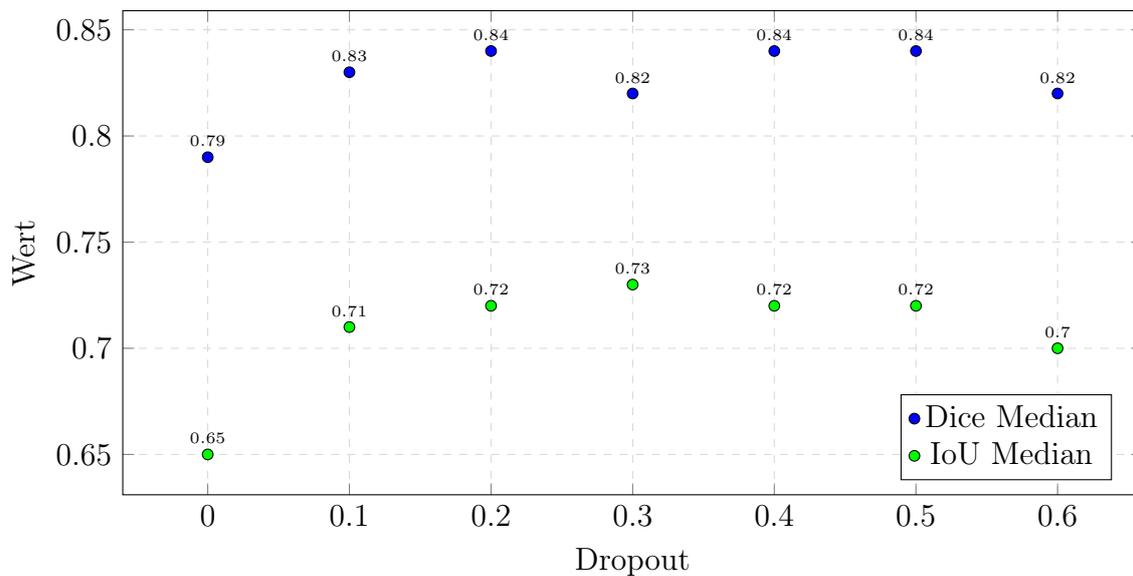


Abbildung 30: Scatterplot der Median-Werte von Dice und IoU in Abhängigkeit vom Dropout-Wert.

Die Daten deuten darauf hin, dass Werten für den Dropout zwischen 0,2 und 0,5 eine gute Wahl sein könnte, da die Leistung hier stabil hoch ist.

### 5.3.4 Aktivierung

Es soll betrachtet werden, wie sich unterschiedliche Aktivierungsfunktionen auf die Leistung eines Modells auswirken.

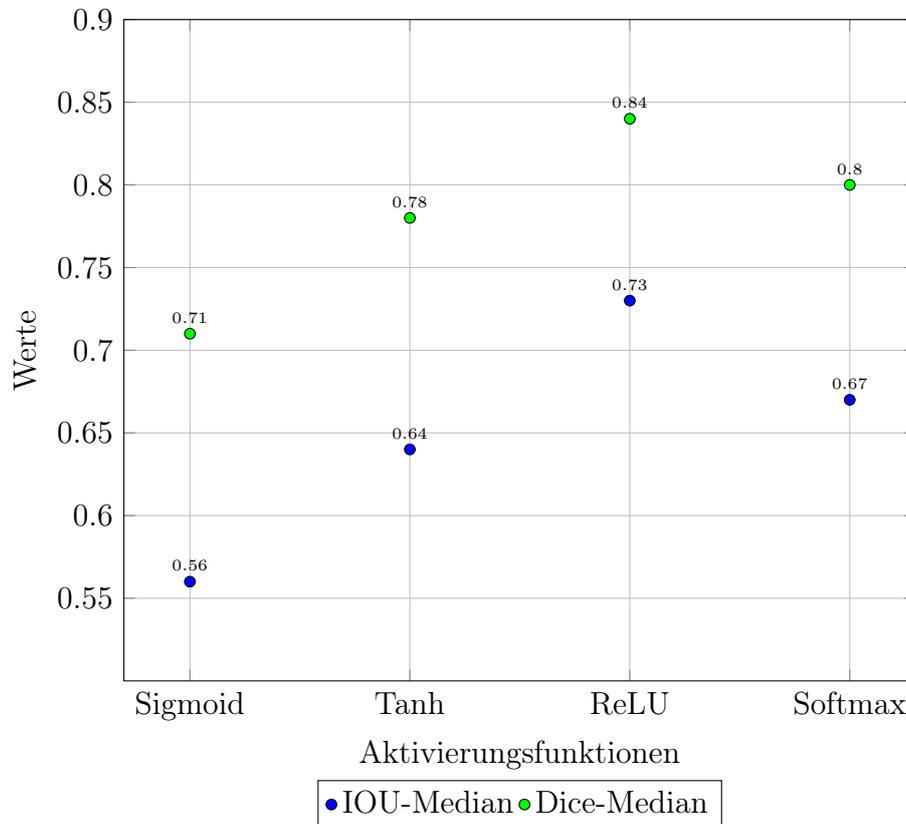


Abbildung 31: Vergleich der Dice-Median und IOU-Median Werte für verschiedene Aktivierungsfunktionen.

In diesem Test zeigt die ReLU-Aktivierungsfunktion die besten Ergebnisse. Auch die Tanh- und Softmax-Funktionen erzielen noch akzeptable Werte, während die Sigmoid-Funktion deutlich schlechter abschneidet, was darauf hinweist, dass sie in diesem Szenario möglicherweise weniger effektiv ist. ReLU scheint daher die beste Wahl zu sein, da sie bei beiden Metriken eine hohe Leistung zeigt.

### 5.3.5 Optimizer

Es ist davon auszugehen, dass Adam im Vergleich zum SGD besser abschneidet, da dieser einen adaptiven Lernratenansatz verwendet und vergangene Gradienten zur Verbesserung der Konvergenz nutzt.

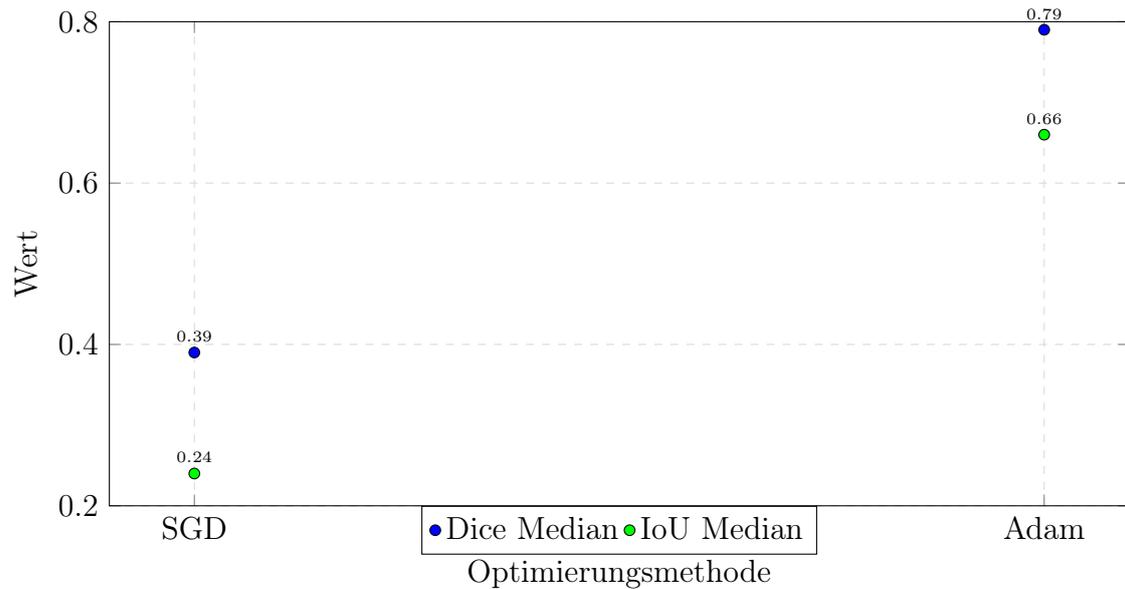


Abbildung 32: Vergleich der Metriken Dice-Median und IoU-Median für die Optimierungsmethoden SGD und Adam mit Bildgröße 32x32 und 2000 Epochen für SGD, 659 Epochen für Adam.

Der Adam-Optimizer zeigte im Vergleich zum SGD eine deutlich schnellere Konvergenz und erzielte daher nach 2000 Epochen eine bessere Leistung. Während SGD die vollen 2000 Epochen benötigte und sich weiterhin verbesserte, war Adam bereits nach etwa 700 Epochen nahezu konvergiert und hatte das Training abgeschlossen.

### 5.3.6 Netztiefe und Breite

Der Test ließ ein U-Net mit allen Permutationen aus verschiedenen Startfiltern (4, 8, 16, 32, 64) und unterschiedlichen Anzahlen an Netzschichten (3,4,5,6) trainieren. Diese Grafiken wurden bei einer Bildgröße von  $(32 \times 32)$  Pixeln erstellt.

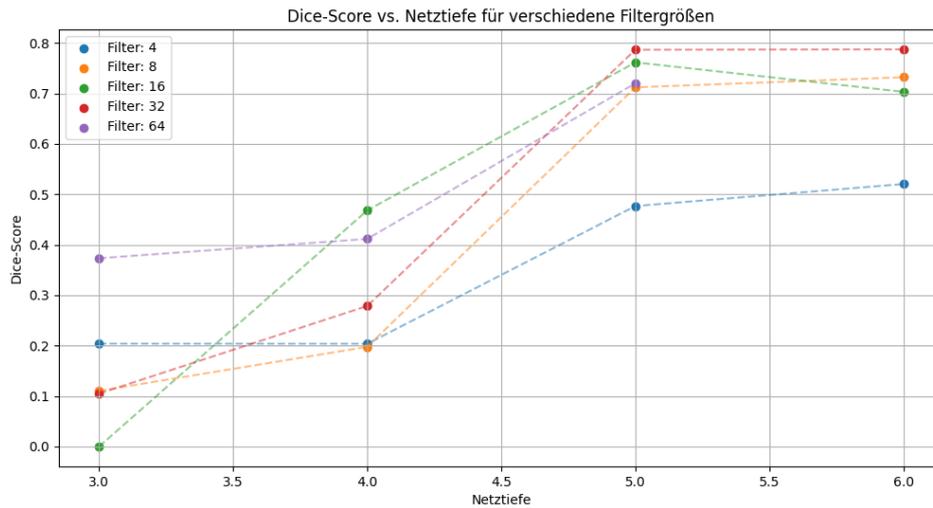


Abbildung 33: Dice-Median der verschiedenen Startfilteranzahlen bei unterschiedlich tiefen Netzen

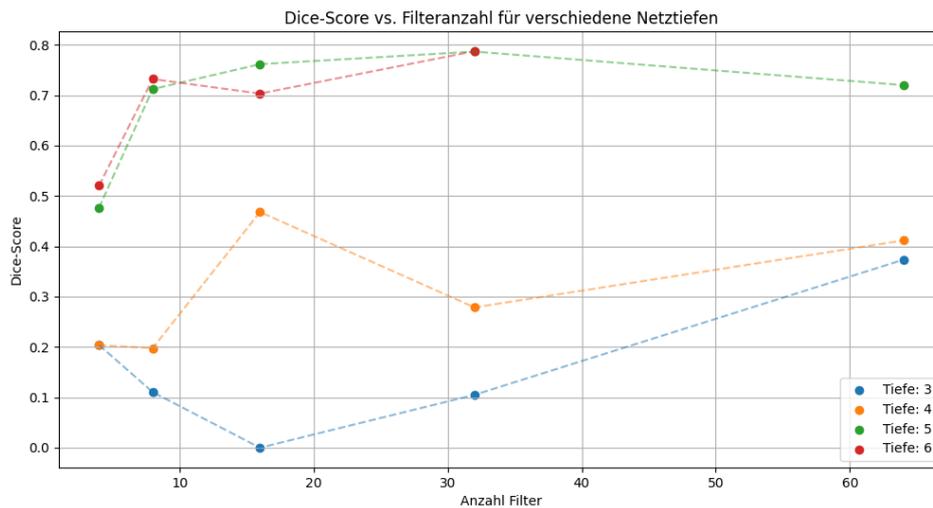


Abbildung 34: Dice-Median der verschiedenen Netztiefen bei unterschiedlichen Startfilteranzahlen

Es zeigt sich, dass das Modell mit einer höheren Netzwerktiefe, also mehr Schichten, von einer besseren Leistung profitiert. Dies lässt darauf schließen, dass tiefere Netzwerke in der Lage sind, komplexere und detailliertere Merkmale im Bild zu erfassen, was zu einer präziseren Segmentierung führt. Die Anzahl der Basisfilter, die zu Beginn des Netzwerks verwendet werden, hatte hingegen einen weniger signifikanten Einfluss auf die Leistung. Lediglich das Netz mit nur 4 Startfiltern bleibt deutlich unter der Leistung der anderen Netze bei gleicher Netztiefe. Dies deutet

darauf hin, dass die Tiefe des Netzwerks für die Leistungssteigerung wichtiger ist als die anfängliche Anzahl der Filter.

Es ist wäre interessant zu sehen, ob das Plateau ab einer Tiefe von 5 sich weiter fortsetzen würde. Dies könnte darauf hindeuten, dass das Netzwerk ab einer Tiefe von 5 bereits in der Lage ist, alle relevanten Informationen eines Bildes dieser Größe zu verarbeiten. In diesem Fall würden zusätzliche Schichten keine signifikante Verbesserung der Modellleistung mehr bringen, da die Fähigkeit des Netzwerks, die Bildmerkmale zu extrahieren und zu segmentieren, die Anforderungen überschreitet. Dies könnte auch darauf hinweisen, dass die weitere Erhöhung der Netzwerktiefe zu einer Überanpassung des Modells an die Trainingsdaten führen könnte, ohne dass ein echter Mehrwert in der Generalisierung oder der Genauigkeit auf neuen, ungeesehenen Daten erzielt wird.

### 5.3.7 Zusammenfassung

In den durchgeführten Tests zur Optimierung des U-Net wurde der Einfluss verschiedener Hyperparameter wie Verlustfunktion, Batchgröße, Dropout-Rate, Aktivierungsfunktionen und Optimierungsmethoden auf die Modellleistung untersucht. Es zeigte sich, dass der Optimierer eine entscheidende Rolle in der Trainingsdynamik spielt. Der Vergleich zwischen SGD und Adam zeigte, dass Adam mit adaptiven Lernraten eine bessere Performance in kürzerer Trainingszeit erreicht.

Die Batchgröße hatte ebenfalls einen Einfluss auf die Modellmetriken, wobei kleinere Batchgrößen bessere IoU- und Dice-Werte aufwiesen, während größere Batchgrößen zu einer schlechteren Modellleistung führten. Dies könnte auf die bessere Generalisierung durch kleinere Batchgrößen und die damit verbundene hohe Variabilität in den Gradientenupdates hindeuten.

Beim Dropout zeigte sich, dass moderate Dropout-Werte zu besseren Ergebnissen führten, während höhere Werte die Leistung des Modells negativ beeinflussten. Dies bestätigt die Notwendigkeit, Dropout gezielt zu testen und anzupassen, um Overfitting zu vermeiden und die Modellgenauigkeit zu maximieren.

Auch die Wahl der Aktivierungsfunktion hatte einen deutlichen Einfluss auf die Leistung des Modells. ReLU zeigte die besten Ergebnisse.

Besonders hat sich jedoch gezeigt, dass die Tiefe des Netzes eine besondere Bedeutung hat und eng mit der Größe des Bildmaterials abgestimmt sein sollte. Ein tieferes Netzwerk ist in der Lage, komplexere und abstraktere Merkmale zu extrahieren, was bei größeren Bildern mit mehr Details und Variationen von Vorteil ist. Bei kleineren Bildern hingegen, die weniger komplexe Strukturen enthalten, könnte eine geringere Netzwerktiefe ausreichen, um eine gute Leistung zu erzielen. Eine zu tiefe Architek-

tur bei kleinen Bildern könnte hingegen unnötige Rechenressourcen beanspruchen und zu Überanpassung führen, ohne signifikante Vorteile in der Modelleistung zu bieten.

Zusammenfassend lässt sich sagen, dass die sorgfältige Auswahl von Hyperparametern einen erheblichen Einfluss auf die Modelleistung haben kann.

## **5.4 Vergleich verschiedener U-Net Architekturen**

In diesem Abschnitt werden verschiedene erweiterungen der U-Net-Architekturen miteinander verglichen, die jeweils auf Basis der zuvor gefundenen optimalen Hyperparameter trainiert wurden. Ziel dieses Vergleichs ist es, die Leistung verschiedener Varianten des U-Net zu bewerten und festzustellen, ob es deutliche Unterschiede gibt.

Verglichen werden die Modelle mit folgender Konfiguration.

Kategorie	Parameter	Wert
Modellarchitektur	Anzahl U-Net-Blöcke	5
	Initializer	he_normal
	Dropout-Rate	0.5
	Basisanzahl Filter	32
Bildparameter	Bildgröße	(32, 32)
	Skalierungsmethode	bicubic
	Denoising	none
	CLAHE	false
Trainingsparameter	Epochen	2000
	Batch-Größe	16
	Validierungssplit	0.2
	Testgröße	0.2
	Zufallszustand	42
	Optimierer	adam
	Verlustfunktion	dice
Callback-Parameter	Patience	500

Tabelle 7: Übersicht der Standard-Modellparameter

- **U-Net++:**

Das U-Net++ wurde dem aus [19] nachgebildet. Das erstellte U-Net++ hatte das Problem nur sehr langsam zu konvergieren. Der angegebene Wert ist nach 2000 Epochen.

- **Residual U-Net:**

Das Residual U-Net Das Residual U-Net wurde dem aus [98] nachgebildet.

- **Attention U-Net:**

Das Attention U-Net wurde dem aus [98] nachgebildet. Es konnten aber leider nicht mehr innerhalb der Zeit alle Fehler behoben werden um einen Testdurchlauf zu machen.

Die Ergebnisse der verschiedenen U-Net-Architekturen sind in der folgenden Tabelle zusammengefasst.

Architektur	Dice	IOU
U-Net Standard	0.81	0.68
U-Net++	0.64	0.45
Residual U-Net	0.78	0.64
Attention U-Net	0	0

Tabelle 8: Vergleich der U-Net Architekturen

Im Ergebnis zeigt das standard U-Net die besten Leistungen. Es ist jedoch anzunehmen, dass mit zusätzlicher Bearbeitungszeit und kleinen Anpassungen die Fehler in den anderen getesteten Netzwerken identifiziert und behoben werden könnten, wodurch deren Leistung ebenfalls verbessert werden könnte.

## 6 Fazit

Ziel dieser Arbeit war es, eine fundierte Empfehlung zur optimalen Konfiguration eines U-Net-Modells für die Bildsegmentierung zur Erkennung von Tumoren zu entwickeln. Die Untersuchung des BUSI-Datensatzes und die Analyse verschiedener Vorverarbeitungs- und Modellparameter haben gezeigt, dass die Wahl der richtigen Bildgröße, Architektur- und Hyperparametereinstellungen eine wesentliche Rolle für die Modellgenauigkeit spielen.

In Bezug auf die Vorverarbeitung zeigte sich, dass keine der getesteten Methoden, wie Bildskalierung, Kontrast- oder Helligkeitsanpassung, einen großen Einfluss auf die Modellleistung hatte. Vielmehr kommt es auf die richtige Anpassung der Bildgröße an, die es dem Modell ermöglicht, relevante Informationen effizient zu extrahieren, ohne mit unnötigen Details überladen zu werden.

Die Tests zur Optimierung des U-Net-Modells verdeutlichten den Einfluss von Hyperparametern wie der Wahl des Optimierers, der Batchgröße, der Dropout-Rate, der Aktivierungsfunktion und der Verlustfunktion. Besonders der Vergleich zwischen den Optimierern zeigte, dass Adam mit adaptiven Lernraten eine bessere Performance bei kürzerem Training erreichte. Kleinere Batchgrößen verbesserten die Generalisierung und führten zu besseren Modellmetriken, während größere Batchgrößen die Leistung verringerten. Bei der Dropout-Rate ergab sich, dass moderate Werte zu

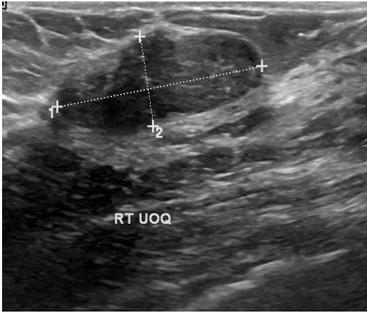
besseren Ergebnissen führten, während höhere Dropout-Werte das Modell negativ beeinflussten.

Ein weiterer entscheidender Faktor war die Wahl der Aktivierungsfunktion. ReLU zeigte sich als die effektivste Funktion für das U-Net-Modell. Besonders hervorzuheben ist jedoch, dass die Tiefe des Netzwerks eng mit der Bildgröße abgestimmt sein sollte. Während tiefere Netzwerke bei größeren Bildern mit mehr Details von Vorteil sind, kann eine zu hohe Netzwerktiefe bei kleineren Bildern unnötige Rechenressourcen beanspruchen und zu einer Überanpassung führen, ohne die Modellleistung signifikant zu steigern.

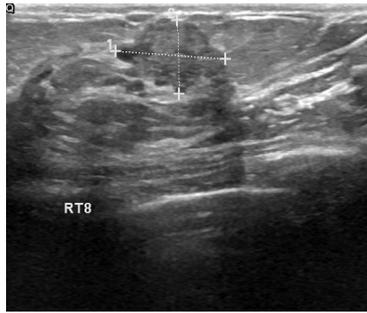
Abschließend lässt sich sagen, dass die sorgfältige Auswahl und Anpassung von Hyperparametern sowie die Abstimmung der Netzwerktiefe auf die Bildgröße maßgeblich dazu beitragen, die Modellleistung zu maximieren. Diese Ergebnisse bieten wertvolle Einsichten für die zukünftige Entwicklung von U-Net-Modellen für die Tumorerkennung und Bildsegmentierung.

## 6.1 Limitationen

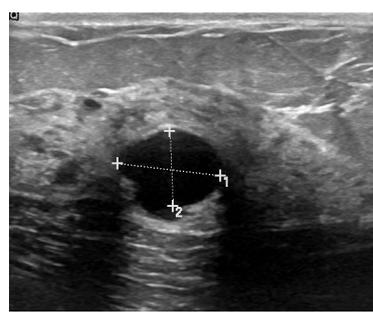
Eine wesentliche Einschränkung dieser Arbeit betrifft die Datengrundlage. Der verwendete BUSI-Datensatz[94], [95] weist, wie durch Pawłowska et al. aufgezeigt[110], erhebliche Inkonsistenzen auf. So enthält der Datensatz Duplikate (19%), falsch klassifizierte Bilder (>2%) sowie Aufnahmen außerhalb des Brustgewebes (>8%).



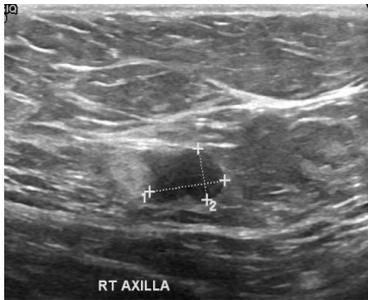
(a) benign (200)



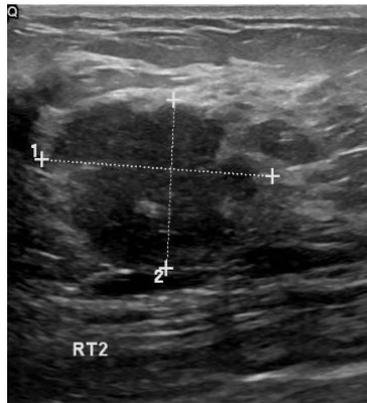
(b) benign (204)



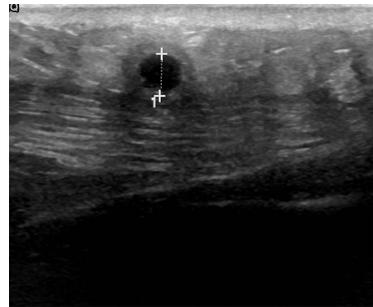
(c) benign (22)



(d) malignant (12)



(e) malignant (15)



(f) malignant (145)

Abbildung 35: Anmerkungen im Datensatz

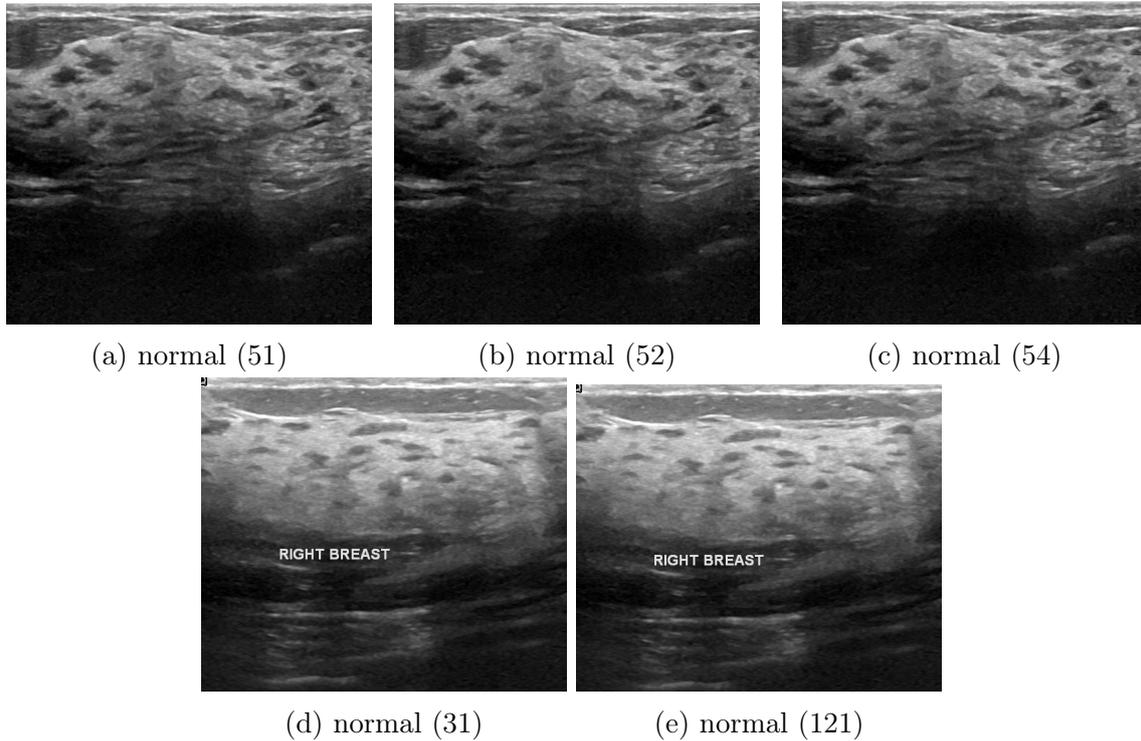


Abbildung 36: Beispiele für Duplikate

Diese Qualitätsprobleme beeinträchtigen zwar die Modelleistung sollte aber nur einen geringen Einfluss auf die Generalisierbarkeit der Ergebnisse haben. Es ist zu beachten, dass diese Herausforderungen nicht nur die vorliegende Arbeit betreffen, sondern auch zahlreiche andere Studien, die denselben Datensatz verwendet haben. In zukünftigen Arbeiten sollte daher auf eine gründlichere Bereinigung des Datensatzes geachtet werden, um die Auswirkungen solcher Inkonsistenzen zu minimieren und die Qualität der Ergebnisse zu verbessern.

Eine Überarbeitung des Datensatzes, die diese Ungenauigkeiten bereinigt, wäre für zukünftige Forschungsarbeiten sicherlich von großem Wert.

## Literaturverzeichnis

- [1] N. C. Institute, *What Is Cancer?* <https://www.cancer.gov/about-cancer/understanding/what-is-cancer>, Aufgerufen am 29.10.2024, 2024.
- [2] C. T. C. of America, *What's the difference? Benign vs. malignant tumors*, <https://www.cancercenter.com/community/blog/2023/01/whats-the-difference-benign-vs-malignant-tumors>, Aufgerufen am 29.10.2024, 2023.
- [3] H. Sung, J. Ferlay, R. L. Siegel u. a., „Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries,“ *CA: A Cancer Journal for Clinicians*, 2021.
- [4] W. H. Organization, *Cancer Fact Sheet*, <https://www.who.int/news-room/fact-sheets/detail/cancer>, Aufgerufen am 29.10.2024, 2024.
- [5] Krebsinformationsdienst, *Immer mehr Menschen erkranken an Krebs*, <https://www.krebsinformationsdienst.de/aktuelles/detail/immer-mehr-menschen-erkranken-an-krebs>, Aufgerufen am 29.10.2024, 2024.
- [6] A. E. Nordenskjöld, H. Fohlin, L. G. Arnesson u. a., „Breast cancer survival trends in different stages and age groups – a population-based study 1989–2013,“ *Acta Oncologica*, Jg. 58, Nr. 1, S. 45–51, 2018. DOI: 10.1080/0284186X.2018.1532601. Adresse: <https://doi.org/10.1080/0284186X.2018.1532601>.
- [7] Krebsinformationsdienst, *Krebszahlen - Daten und Fakten*, <https://www.krebsinformationsdienst.de/forschung/krebszahlen>, Aufgerufen am 29.10.2024, 2024.
- [8] D. Hanahan, „Rethinking the war on cancer,“ *The Lancet*, 2015.
- [9] D. Krebsgesellschaft, *Krebshäufigkeit - die aktuellen Daten*, <https://www.krebsgesellschaft.de/onko-internetportal/basis-informationen-krebs/allgemeine-informationen/krebshaeufigkeit-die-aktuellen-daten.html>, Aufgerufen am 29.10.2024, 2024.
- [10] N. C. Institute, *Breast Cancer: SEER Cancer Stat Facts*, Zugriff am 23. November 2024, 2024. Adresse: <https://seer.cancer.gov/statfacts/html/breast.html>.

- [11] Krebsdaten.de, *Überlebensraten bei Krebs: Methodische Grundlagen und Ergebnisse*, Zugriff am 23. November 2024, 2024. Adresse: [https://www.krebsdaten.de/Krebs/DE/Content/Methoden/Ueberlebensraten/ueberlebensraten\\_node.html](https://www.krebsdaten.de/Krebs/DE/Content/Methoden/Ueberlebensraten/ueberlebensraten_node.html).
- [12] L. Chen, H. M. Linden, B. O. Anderson und C. I. Li, „Trends in 5-year survival rates among breast cancer patients by hormone receptor status and stage,“ *Breast Cancer Research and Treatment*, Jg. 147, Nr. 3, S. 609–616, 2014, ISSN: 1573-7217. DOI: 10.1007/s10549-014-3112-6. Adresse: <https://doi.org/10.1007/s10549-014-3112-6>.
- [13] Krebsinformationsdienst, *Stadieneinteilung bei Brustkrebs: Wachstumsverhalten und molekulare Subtypen*, Zugriff am 23. November 2024, 2024. Adresse: <https://www.krebsinformationsdienst.de/brustkrebs/stadieneinteilung>.
- [14] A. M. Ugnat, L. Xie, J. Morriss, R. Semenciw und Y. Mao, „Survival of women with breast cancer in Ottawa, Canada: variation with age, stage, histology, grade and treatment,“ *British Journal of Cancer*, Jg. 90, Nr. 6, S. 1138–1143, März 2004, ISSN: 1532-1827. DOI: 10.1038/sj.bjc.6601662. Adresse: <https://doi.org/10.1038/sj.bjc.6601662>.
- [15] I. Chitapanarux, P. Sripan, A. Somwangprasert u. a., „Stage-Specific Survival Rate of Breast Cancer Patients in Northern Thailand in Accordance with Two Different Staging Systems,“ *Asian Pacific Journal of Cancer Prevention*, Jg. 20, Nr. 9, S. 2699–2706, Sep. 2019. DOI: 10.31557/APJCP.2019.20.9.2699. Adresse: <https://doi.org/10.31557/APJCP.2019.20.9.2699>.
- [16] L. Mangone, F. Marinelli, I. Bisceglia, M. B. Braghiroli, A. Damato und C. Pinto, „Five-year relative survival by stage of breast and colon cancers in northern Italy,“ *Frontiers in Oncology*, Jg. 12, 2022, ISSN: 2234-943X. DOI: 10.3389/fonc.2022.982461. Adresse: <https://www.frontiersin.org/journals/oncology/articles/10.3389/fonc.2022.982461>.
- [17] H. M. School. „AI Predicts Future Pancreatic Cancer.“ Accessed: 2024-11-24. (Mai 2023), Adresse: <https://hms.harvard.edu/news/ai-predicts-future-pancreatic-cancer>.
- [18] G. Litjens, T. Kooi, B. E. Bejnordi u. a., „A survey on deep learning in medical image analysis,“ *Medical Image Analysis*, Jg. 42, S. 60–88, 2017, ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2017.07.005>. Adresse: <https://www.sciencedirect.com/science/article/pii/S1361841517301135>.

- [19] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh und J. Liang, „UNet++: A Nested U-Net Architecture for Medical Image Segmentation,“ in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, D. Stoyanov, Z. Taylor, G. Carneiro u. a., Hrsg., Cham: Springer International Publishing, 2018, S. 3–11, ISBN: 978-3-030-00889-5.
- [20] M. Galanski und K. Lackner, „Prinzipien der bildgebenden Diagnostik in der Onkologie,“ in *Kompendium Internistische Onkologie: Standards in Diagnostik und Therapie*, H.-J. Schmoll, K. Höffken und K. Possinger, Hrsg. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, S. 429–448, ISBN: 978-3-540-31303-8. DOI: 10.1007/3-540-31303-6\_21. Adresse: [https://doi.org/10.1007/3-540-31303-6\\_21](https://doi.org/10.1007/3-540-31303-6_21).
- [21] D. Krebsgesellschaft, *Ultraschall in der Krebsdiagnostik*, Zugriff am 23. November 2024, 2024. Adresse: <https://www.krebsgesellschaft.de/onko-internetportal/basis-informationen-krebs/diagnosemethoden/ultraschall-sonografie.html>.
- [22] P. Lundberg, M. F. Forsgren, J. Tellman, J. Kihlberg, A. Rzepecka und C. Dabrosin, „Breast density is strongly associated with multiparametric magnetic resonance imaging biomarkers and pro-tumorigenic proteins in situ,“ *British Journal of Cancer*, Jg. 127, Nr. 11, S. 2025–2033, Nov. 2022, ISSN: 1532-1827. DOI: 10.1038/s41416-022-01976-3. Adresse: <https://doi.org/10.1038/s41416-022-01976-3>.
- [23] N. F. Boyd, H. Guo, L. J. Martin u. a., „Mammographic Density and the Risk and Detection of Breast Cancer,“ *New England Journal of Medicine*, Jg. 356, Nr. 3, S. 227–236, 2007. DOI: 10.1056/NEJMoa062790. eprint: <https://www.nejm.org/doi/pdf/10.1056/NEJMoa062790>. Adresse: <https://www.nejm.org/doi/full/10.1056/NEJMoa062790>.
- [24] V. P. Jackson und R. L. Gilmor, „Male breast carcinoma and gynecomastia: comparison of mammography with sonography,“ *Radiology*, Jg. 149, Nr. 2, S. 533–536, 1983. DOI: 10.1148/radiology.149.2.6622700. Adresse: <https://doi.org/10.1148/radiology.149.2.6622700>.
- [25] R. A. Stewart, D. C. Howlett und F. J. Hearn, „Pictorial review: the imaging features of male breast disease,“ *Clinical radiology*, Jg. 52, Nr. 10, S. 739–744, 1997. DOI: 10.1016/s0009-9260(97)80151-1. Adresse: [https://doi.org/10.1016/s0009-9260\(97\)80151-1](https://doi.org/10.1016/s0009-9260(97)80151-1).

- [26] S. Russell und P. Norvig, *Artificial Intelligence, Global Edition A Modern Approach*. Pearson Deutschland, 2021, S. 1168, ISBN: 9781292401133. Adresse: <https://elibrary.pearson.de/book/99.150005/9781292401171>.
- [27] Y. LeCun, Y. Bengio und G. Hinton, „Deep learning,“ *Nature*, Jg. 521, Nr. 7553, S. 436–444, Mai 2015, ISSN: 1476-4687. DOI: 10.1038/nature14539. Adresse: <https://doi.org/10.1038/nature14539>.
- [28] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman und A. Galstyan, „A Survey on Bias and Fairness in Machine Learning,“ *CoRR*, Jg. abs/1908.09635, 2019. arXiv: 1908.09635. Adresse: <http://arxiv.org/abs/1908.09635>.
- [29] „Ethical Issues Arising Due to Bias in Training A.I. Algorithms in Healthcare and Data Sharing as a Potential Solution,“ Jg. 1, 2023. DOI: 10.47289/AIEJ20200916. Adresse: <https://aiej.org/aiej/article/view/1>.
- [30] D. M. McIntosh, G. J. Suzuki und Y. Schaal, „AI Copyright Issues Loom Over Data and Content Usage,“ *ropesgray*, 2024, Accessed: 2024-11-24. Adresse: <https://www.ropesgray.com/en/insights/alerts/2024/04/ai-and-the-copyright-liability-overhang-a-brief-summary-of-the-current-state-of-ai-related>.
- [31] M. Veale, R. Binns und L. Edwards, „Algorithms that Remember: Model Inversion Attacks and Data Protection Law,“ *CoRR*, Jg. abs/1807.04644, 2018. arXiv: 1807.04644. Adresse: <http://arxiv.org/abs/1807.04644>.
- [32] L. Stark und J. Hutson, „Physiognomic Artificial Intelligence,“ *SSRN Electronic Journal*, 2021. Adresse: <https://api.semanticscholar.org/CorpusID:244221334>.
- [33] M. E. O’Connell, „Banning Autonomous Weapons: A Legal and Ethical Mandate,“ *Ethics; International Affairs*, Jg. 37, Nr. 3, S. 287–298, 2023. DOI: 10.1017/S0892679423000357.
- [34] P. (PwC), „Impact of Artificial Intelligence on Fraud and Scams,“ *PwC Research Report*, 2023. Adresse: <https://www.pwc.co.uk/forensic-services/assets/impact-of-ai-on-fraud-and-scams.pdf>.
- [35] G. Bradski und A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA: O’Reilly Media, 2008, ISBN: 978-0-596-51613-0.
- [36] T. S. Huang, „Computer Vision: Evolution And Promise,“ 1996. Adresse: <https://api.semanticscholar.org/CorpusID:67163270>.

- [37] A. Esteva, B. Kuprel, R. A. Novoa u. a., „Dermatologist-level classification of skin cancer with deep neural networks,“ *Nature*, Jg. 542, S. 115–118, 2017. DOI: 10.1038/nature21056. Adresse: <https://doi.org/10.1038/nature21056>.
- [38] J. Schmidhuber, „Deep learning in neural networks: An overview,“ *Neural Networks*, Jg. 61, S. 85–117, 2015, ISSN: 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2014.09.003>. Adresse: <https://www.sciencedirect.com/science/article/pii/S0893608014002135>.
- [39] I. Goodfellow, Y. Bengio und A. Courville, *Deep Learning*. MIT Press, 2016, <http://www.deeplearningbook.org>.
- [40] P. Ramachandran, B. Zoph und Q. V. Le, *Searching for Activation Functions*, 2017. arXiv: 1710.05941 [cs.NE]. Adresse: <https://arxiv.org/abs/1710.05941>.
- [41] K. Choo, E. Greplova, M. H. Fischer und T. Neupert, *Neuronale Netzwerke und überwachtetes Lernen*. Wiesbaden: Springer Fachmedien Wiesbaden, 2020, S. 28, ISBN: 978-3-658-32268-7. DOI: 10.1007/978-3-658-32268-7\_3. Adresse: [https://doi.org/10.1007/978-3-658-32268-7\\_3](https://doi.org/10.1007/978-3-658-32268-7_3).
- [42] „[ Front Matter ],“ in *Perceptrons: An Introduction to Computational Geometry*, The MIT Press, Sep. 2017, ISBN: 9780262343930. DOI: 10.7551/mitpress/11301.003.0023. eprint: <https://direct.mit.edu/book/chapter-pdf/2340558/f010005\9780262343930.pdf>. Adresse: <https://doi.org/10.7551/mitpress/11301.003.0023>.
- [43] F. Rosenblatt, „The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain,“ *Psychological Review*, Jg. 65, Nr. 6, S. 386–408, 1958. DOI: 10.1037/h0042519.
- [44] A. Parmezan, V. Alves de Souza und G. Batista, „Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model,“ *Information Sciences*, Jan. 2019. DOI: 10.1016/j.ins.2019.01.076.
- [45] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [46] M. Kipp, *Feedforward-Netze*, Zugriff am 22. November 2024, 2024. Adresse: <https://michaelkipp.de/deeplearning/FeedforwardNetze.html>.

- [47] O. Ronneberger, P. Fischer und T. Brox, „U-Net: Convolutional Networks for Biomedical Image Segmentation,“ in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells und A. F. Frangi, Hrsg., Cham: Springer International Publishing, 2015, S. 234–241, ISBN: 978-3-319-24574-4.
- [48] S. Ioffe und C. Szegedy, „Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,“ *CoRR*, Jg. abs/1502.03167, 2015. arXiv: 1502.03167. Adresse: <http://arxiv.org/abs/1502.03167>.
- [49] Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox und O. Ronneberger, „3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation,“ *CoRR*, Jg. abs/1606.06650, 2016. arXiv: 1606.06650. Adresse: <http://arxiv.org/abs/1606.06650>.
- [50] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever und R. Salakhutdinov, „Dropout: a simple way to prevent neural networks from overfitting,“ *J. Mach. Learn. Res.*, Jg. 15, Nr. 1, S. 1929–1958, Jan. 2014, ISSN: 1532-4435.
- [51] Y. Lecun, L. Bottou, Y. Bengio und P. Haffner, „Gradient-based learning applied to document recognition,“ *Proceedings of the IEEE*, Jg. 86, Nr. 11, S. 2278–2324, 1998. DOI: 10.1109/5.726791.
- [52] V. Dumoulin und F. Visin, *A guide to convolution arithmetic for deep learning*, 2018. arXiv: 1603.07285 [stat.ML]. Adresse: <https://arxiv.org/abs/1603.07285>.
- [53] M. Drozdal, E. Vorontsov, G. Chartrand, S. Kadoury und C. Pal, „The Importance of Skip Connections in Biomedical Image Segmentation,“ *CoRR*, Jg. abs/1608.04117, 2016. arXiv: 1608.04117. Adresse: <http://arxiv.org/abs/1608.04117>.
- [54] I. Dimitrovski, V. Spasev, S. Loshkovska und I. Kitanovski, „U-Net Ensemble for Enhanced Semantic Segmentation in Remote Sensing Imagery,“ *Remote Sensing*, Jg. 16, Nr. 12, 2024, ISSN: 2072-4292. DOI: 10.3390/rs16122077. Adresse: <https://www.mdpi.com/2072-4292/16/12/2077>.
- [55] S. V. Stehman, „Selecting and interpreting measures of thematic classification accuracy,“ *Remote Sensing of Environment*, Jg. 62, Nr. 1, S. 77–89, 1997, ISSN: 0034-4257. DOI: [https://doi.org/10.1016/S0034-4257\(97\)00083-7](https://doi.org/10.1016/S0034-4257(97)00083-7). Adresse: <https://www.sciencedirect.com/science/article/pii/S0034425797000837>.

- [56] A. A. Taha und A. Hanbury, „Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool“, *BMC Medical Imaging*, Jg. 15, Nr. 1, S. 29, 2015, ISSN: 1471-2342. DOI: 10.1186/s12880-015-0068-x. Adresse: <https://doi.org/10.1186/s12880-015-0068-x>.
- [57] P. Christen, D. J. Hand und N. Kirielle, „A Review of the F-Measure: Its History, Properties, Criticism, and Alternatives“, *ACM Comput. Surv.*, Jg. 56, Nr. 3, Okt. 2023, ISSN: 0360-0300. DOI: 10.1145/3606367. Adresse: <https://doi.org/10.1145/3606367>.
- [58] L. R. Dice, „Measures of the Amount of Ecologic Association Between Species“, *Ecology*, Jg. 26, Nr. 3, S. 297–302, 1945. DOI: <https://doi.org/10.2307/1932409>. eprint: <https://esajournals.onlinelibrary.wiley.com/doi/pdf/10.2307/1932409>. Adresse: <https://esajournals.onlinelibrary.wiley.com/doi/abs/10.2307/1932409>.
- [59] A. Carass, S. Roy, A. Gherman u. a., „Evaluating White Matter Lesion Segmentations with Refined Sørensen-Dice Analysis“, *Scientific Reports*, Jg. 10, Nr. 1, S. 8242, 2020, ISSN: 2045-2322. DOI: 10.1038/s41598-020-64803-w. Adresse: <https://doi.org/10.1038/s41598-020-64803-w>.
- [60] P. Jaccard, „Lois de distribution florale dans la zone alpine“, *Bulletin de la Société vaudoise des sciences naturelles*, Jg. 38, S. 69–130, Jan. 1902. DOI: 10.5169/seals-266762.
- [61] S. Jadon, „A survey of loss functions for semantic segmentation“, in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, 2020, S. 1–7. DOI: 10.1109/CIBCB48159.2020.9277638.
- [62] S. Ruder, „An overview of gradient descent optimization algorithms“, *CoRR*, Jg. abs/1609.04747, 2016. arXiv: 1609.04747. Adresse: <http://arxiv.org/abs/1609.04747>.
- [63] A. Amini, A. Soleimany, S. Karaman und D. Rus, „Spatial Uncertainty Sampling for End-to-End Control“, *CoRR*, Jg. abs/1805.04829, 2018. arXiv: 1805.04829. Adresse: <http://arxiv.org/abs/1805.04829>.
- [64] D. P. Kingma und J. Ba, *Adam: A Method for Stochastic Optimization*, 2017. arXiv: 1412.6980 [cs.LG]. Adresse: <https://arxiv.org/abs/1412.6980>.
- [65] K. He, X. Zhang, S. Ren und J. Sun, „Deep Residual Learning for Image Recognition“, *CoRR*, Jg. abs/1512.03385, 2015. arXiv: 1512.03385. Adresse: <http://arxiv.org/abs/1512.03385>.

- [66] D. Darwish, „Improving Techniques for Convolutional Neural Networks Performance,“ *European Journal of Electrical Engineering and Computer Science*, Jg. 8, Nr. 1, S. 1–16, 2024. DOI: 10.24018/ejece.2024.8.1.596. Adresse: <https://ejece.org/index.php/ejece/article/view/596>.
- [67] S. Li, P. Zhao, H. Zhang u. a., *Surge Phenomenon in Optimal Learning Rate and Batch Size Scaling*, 2024. arXiv: 2405.14578 [cs.LG]. Adresse: <https://arxiv.org/abs/2405.14578>.
- [68] J. Bergstra und Y. Bengio, „Random search for hyper-parameter optimization,“ *J. Mach. Learn. Res.*, Jg. 13, Nr. null, S. 281–305, Feb. 2012, ISSN: 1532-4435.
- [69] S. Salman und X. Liu, *Overfitting Mechanism and Avoidance in Deep Neural Networks*, 2019. arXiv: 1901.06566 [cs.LG]. Adresse: <https://arxiv.org/abs/1901.06566>.
- [70] *TensorFlow Early Stopping*, Accessed: 2024-11-25. Adresse: [https://www.tensorflow.org/api\\_docs/python/tf/keras/callbacks/EarlyStopping](https://www.tensorflow.org/api_docs/python/tf/keras/callbacks/EarlyStopping).
- [71] R. Gonzalez und R. Woods, *Digital Image Processing Global Edition*. Pearson Deutschland, 2017, S. 1024, ISBN: 9781292223049. Adresse: <https://elibrary.pearson.de/book/99.150005/9781292223070>.
- [72] S. Fadnavis, „Image Interpolation Techniques in Digital Image Processing: An Overview,“ *International Journal Of Engineering Research and Application*, Jg. 4, S. 2248–962 270, Nov. 2014.
- [73] „An iterative method for the solution of the eigenvalue problem of linear differential and integral,“ 1950. Adresse: <https://api.semanticscholar.org/CorpusID:478182>.
- [74] *Geometric transformations in OpenCV*, Accessed: 2024-11-25. Adresse: [https://docs.opencv.org/3.4/da/d6e/tutorial\\_py\\_geometric\\_transformations.html](https://docs.opencv.org/3.4/da/d6e/tutorial_py_geometric_transformations.html).
- [75] M. K. Jasim, „Image Noise Removal Techniques : A Comparative Analysis,“ *International Journal of Science and Applied Information Technology*, 2019. Adresse: <https://api.semanticscholar.org/CorpusID:213559589>.
- [76] A. Kaur und G. Dong, „A Complete Review on Image Denoising Techniques for Medical Images,“ *Neural Processing Letters*, Jg. 55, Nr. 6, S. 7807–7850, Dez. 2023, ISSN: 1573-773X. DOI: 10.1007/s11063-023-11286-1. Adresse: <https://doi.org/10.1007/s11063-023-11286-1>.

- [77] A. Kumar, T. Sharma, N. K. Verma, P. Sircar und S. Vasikarla, „Detection and Removal of Salt and Pepper Noise by Gaussian Membership Function and Guided Filter,“ in *2019 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)*, 2019, S. 1–9. DOI: 10.1109/AIPR47015.2019.9174579.
- [78] S. Kaur, J. Singla, Nikita und A. Singh, „Review on Medical Image Denoising Techniques,“ in *2021 International Conference on Innovative Practices in Technology and Management (ICIPTM)*, 2021, S. 61–66. DOI: 10.1109/ICIPTM52218.2021.9388367.
- [79] M. Mafi, H. Martin, M. Cabrerizo, J. Andrian, A. Barreto und M. Adjouadi, „A comprehensive survey on impulse and Gaussian denoising filters for digital images,“ *Signal Processing*, Jg. 157, S. 236–260, 2019, ISSN: 0165-1684. DOI: <https://doi.org/10.1016/j.sigpro.2018.12.006>. Adresse: <https://www.sciencedirect.com/science/article/pii/S0165168418303979>.
- [80] M. Werner, „Glättungsfilter, Rauschen und Verzerrungen,“ in *Digitale Bildverarbeitung: Grundkurs mit neuronalen Netzen und MATLAB®-Praktikum*. Wiesbaden: Springer Fachmedien Wiesbaden, 2021, S. 111–135, ISBN: 978-3-658-22185-0. DOI: 10.1007/978-3-658-22185-0\_5. Adresse: [https://doi.org/10.1007/978-3-658-22185-0\\_5](https://doi.org/10.1007/978-3-658-22185-0_5).
- [81] O. D. Team, *Image Denoising*, [https://docs.opencv.org/master/d5/d69/tutorial\\_py\\_non\\_local\\_means.html](https://docs.opencv.org/master/d5/d69/tutorial_py_non_local_means.html), OpenCV, 2024.
- [82] K. Zhang, W. Zuo, Y. Chen, D. Meng und L. Zhang, „Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising,“ *IEEE Transactions on Image Processing*, Jg. 26, Nr. 7, S. 3142–3155, 2017. DOI: 10.1109/TIP.2017.2662206.
- [83] D. G. Pelli und P. Bex, „Measuring contrast sensitivity,“ *Vision Research*, Jg. 90, S. 10–14, 2013, Testing Vision: From Laboratory Psychophysical Tests to Clinical Evaluation, ISSN: 0042-6989. DOI: <https://doi.org/10.1016/j.visres.2013.04.015>. Adresse: <https://www.sciencedirect.com/science/article/pii/S0042698913001132>.
- [84] K. Zuiderveld, „Contrast limited adaptive histogram equalization,“ in *Graphics Gems IV*. USA: Academic Press Professional, Inc., 1994, S. 474–485, ISBN: 0123361559.
- [85] G. Siracusano, A. La Corte, M. Gaeta, G. Cicero, M. Chiappini und G. Finocchio, „Pipeline for Advanced Contrast Enhancement (PACE) of Chest X-ray in Evaluating COVID-19 Patients by Combining Bidimensional Empirical

- Mode Decomposition and Contrast Limited Adaptive Histogram Equalization (CLAHE),“ *Sustainability*, Jg. 12, Nr. 20, 2020, ISSN: 2071-1050. DOI: 10.3390/su12208573. Adresse: <https://www.mdpi.com/2071-1050/12/20/8573>.
- [86] S. Rahman, M. M. Rahman, M. Abdullah-Al-Wadud, G. D. Al-Quaderi und M. Shoyaib, „An adaptive gamma correction for image enhancement,“ *EURASIP Journal on Image and Video Processing*, Jg. 2016, Nr. 1, S. 35, 2016, ISSN: 1687-5281. DOI: 10.1186/s13640-016-0138-1. Adresse: <https://doi.org/10.1186/s13640-016-0138-1>.
- [87] *TensorFlow on Open Hub*, Accessed: 2024-11-25. Adresse: <https://openhub.net/p/tensorflow>.
- [88] *Kaggle’s State of Machine Learning and Data Science 2021*, Accessed: 2024-11-25. Adresse: <https://storage.googleapis.com/kaggle-media/surveys/Kaggle’s%20State%20of%20Machine%20Learning%20and%20Data%20Science%202021.pdf>.
- [89] *Stack Overflow Developer Survey 2023*, Accessed: 2024-11-25. Adresse: <https://survey.stackoverflow.co/2023/#most-popular-technologies-misc-tech-prof>.
- [90] *keras*, Accessed: 2024-11-25. Adresse: <https://keras.io/about/>.
- [91] *TensorFlow and Keras*, Accessed: 2024-11-25. Adresse: <https://www.tensorflow.org/guide/keras>.
- [92] *TensorBoard: TensorFlow’s Visualization Toolkit*, Accessed: 2024-11-25. Adresse: <https://www.tensorflow.org/tensorboard>.
- [93] *Tensorboard Get Started*, Accessed: 2024-11-25. Adresse: [https://www.tensorflow.org/tensorboard/get\\_started](https://www.tensorflow.org/tensorboard/get_started).
- [94] W. Al-Dhabyani, M. Gomaa, H. Khaled und A. Fahmy, *BUSI: Dataset of Breast Ultrasound Images*, <https://doi.org/10.1016/j.dib.2019.104863>, Aufgerufen am 15.10.2024, 2020.
- [95] W. Al-Dhabyani, M. Gomaa, H. Khaled und A. Fahmy, „Dataset of breast ultrasound images,“ *Data in Brief*, Jg. 28, S. 104863, 2020, ISSN: 2352-3409. DOI: <https://doi.org/10.1016/j.dib.2019.104863>. Adresse: <https://www.sciencedirect.com/science/article/pii/S2352340919312181>.

- [96] Y. Zhang, M. Xian, H.-D. Cheng u. a., „BUSIS: A Benchmark for Breast Ultrasound Image Segmentation,“ *Healthcare*, Jg. 10, Nr. 4, 2022, ISSN: 2227-9032. DOI: 10.3390/healthcare10040729. Adresse: <https://www.mdpi.com/2227-9032/10/4/729>.
- [97] A. Esteva, A. Robicquet, B. Ramsundar u. a., „A guide to deep learning in healthcare,“ *Nature Medicine*, Jg. 25, S. 24–29, 2019. Adresse: <https://api.semanticscholar.org/CorpusID:205572964>.
- [98] O. Oktay, J. Schlemper, L. L. Folgoc u. a., „Attention U-Net: Learning Where to Look for the Pancreas,“ *CoRR*, Jg. abs/1804.03999, 2018. arXiv: 1804.03999. Adresse: <http://arxiv.org/abs/1804.03999>.
- [99] F. I. Diakogiannis, F. Waldner, P. Caccetta und C. Wu, „ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data,“ *CoRR*, Jg. abs/1904.00592, 2019. arXiv: 1904.00592. Adresse: <http://arxiv.org/abs/1904.00592>.
- [100] C. Szegedy, S. Ioffe und V. Vanhoucke, „Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,“ *CoRR*, Jg. abs/1602.07261, 2016. arXiv: 1602.07261. Adresse: <http://arxiv.org/abs/1602.07261>.
- [101] C. Kou, W. Li, Z. Yu und L. Yuan, „An Enhanced Residual U-Net for Microaneurysms and Exudates Segmentation in Fundus Images,“ *IEEE Access*, Jg. 8, S. 185 514–185 525, 2020. DOI: 10.1109/ACCESS.2020.3029117.
- [102] P. Chen, X. Su, M. Liu und W. Zhu, „Lensless Computational Imaging Technology Using Deep Convolutional Network,“ *Sensors*, Jg. 20, S. 2661, Mai 2020. DOI: 10.3390/s20092661.
- [103] J. Zhang, Z. Jiang, J. Dong, Y. Hou und B. Liu, „Attention Gate ResU-Net for Automatic MRI Brain Tumor Segmentation,“ *IEEE Access*, Jg. 8, S. 58 533–58 545, 2020. DOI: 10.1109/ACCESS.2020.2983075.
- [104] M. A. Farooq, Z. X. Gong, Y. Liu und M. Zubair, „Breast Cancer Detection from Ultrasound Images using Attention U-Net Model,“ *14th International Conference on Digital Image Processing (ICDIP 2022)*, 2022. DOI: 10.1117/12.2643599.
- [105] M. F. Dar und A. Ganivada, „EfficientU-Net: A Novel Deep Learning Method for Breast Tumor Segmentation and Classification in Ultrasound Images,“ *Neural Processing Letters*, Jg. 55, Nr. 8, S. 10 439–10 462, Dez. 2023, ISSN: 1573-773X. DOI: 10.1007/s11063-023-11333-x.

- [106] Q. Sang, Y. Hou, P. Qian und Q. Wu, „Self-supervised learning-leveraged boosting ultrasound image segmentation via mask reconstruction,“ *International Journal of Machine Learning and Cybernetics*, Jg. 15, Nr. 5, S. 2039–2048, Mai 2024, ISSN: 1868-808X. DOI: 10.1007/s13042-023-02014-1.
- [107] C. Zhang und J. Wang, „PIS-Net: Efficient Medical Image Segmentation Network with Multivariate Downsampling for Point-of-Care,“ *Entropy*, Jg. 26, Nr. 4, 2024, ISSN: 1099-4300. DOI: 10.3390/e26040284.
- [108] A. Sulaiman, V. Anand, S. Gupta u. a., „Attention based UNet model for breast cancer segmentation using BUSI dataset,“ *Scientific Reports*, Jg. 14, Nr. 1, S. 22 422, Sep. 2024, ISSN: 2045-2322. DOI: 10.1038/s41598-024-72712-5.
- [109] P. O. Vianna, R. Farias und W. C. A. Pereira, „Performance of the SegNet in the Segmentation of Breast Ultrasound Lesions,“ in *2021 Global Medical Engineering Physics Exchanges/Pan American Health Care Exchanges (GMEPE/PAHCE)*, IEEE, 2021, S. 1–4. DOI: 10.1109/GMEPE/PAHCE50215.2021.9434877.
- [110] A. Pawłowska, P. Karwat und N. Żółek, „Letter to the Editor,“ *Data in Brief*, Jg. 48, S. 109 247, 2023, ISSN: 2352-3409. DOI: 10.1016/j.dib.2023.109247. Adresse: <https://www.sciencedirect.com/science/article/pii/S2352340923003669>.