

Maschinelles Lernen

Analyzing Breast Ultrasound Images

Marcel Moczarski
marcel.moczarski@tu-dortmund.de

Abgabe: 15.08.2021

TU Dortmund – Fakultät Physik

Inhaltsverzeichnis

1	Einleitung	3
2	Datensatz	3
3	CNN-Architektur	4
3.1	U-Net	4
3.2	Attention-Recurrent-Residual-U-Net	6
4	Modellarchitekturen, Optimierungsparameter und Übertraining	8
5	Segmentierung von Ultraschallbildern von Brustkrebs	10
6	Alternativmethode: Canny-Edge-Detektor	12
7	Zusammenfassung	13
	Literatur	14

1 Einleitung

Anwendungen des maschinellen Lernens im medizinischen Bereich sind aufgrund der gestiegenen Rechenkapazitäten in den vergangenen Jahren populärer geworden. Insbesondere die Verwendungen von Algorithmen zur Klassifikation und Detektion von Bildbereichen, bspw. Zellen oder Tumore, erweist sich als wertvolle Ergänzung zum menschlichen Augenmaß und kann dieses in einigen Fällen sogar ersetzen.

Die Erkennung von Tumoren erfolgte dabei bisher durch subjektive Einschätzungen der Ärzte die in Röntgen-, MRT- oder Ultraschallbildern manuell nach Anomalien im Gewebe oder nach bestimmten Charakteristiken suchten, die auf Tumorgewebe hinweisen könnten. Neben der Klassifikation, d.h. die Frage ob ein Tumor vorliegt oder nicht, ist die Segmentierung des Tumors innerhalb des Bildes ein weiteres wichtiges Hilfswerkzeug für die Medizin. Bei dieser Art der Segmentierung wird eine binäre Maske erstellt, welche jeden Pixel als Tumorgewebe oder kein Tumorgewebe klassifiziert. Diese erstellte Maske gibt somit Aufschluss über die Lage, Größe und ggfs. Struktur des Tumors, was besonders hilfreich in Bezug auf weitere Therapiemöglichkeiten ist.

Die traditionelle Art der Klassifikation mittels Feature-Engineering, d.h. einer manuellen Auswahl von wichtigen Features, wurde dabei von einer Convolutional-Neural-Network [CNN]-Architektur abgelöst. Anders als bei den vorher verwendeten Algorithmen lernt ein CNN anhand der rohen Eingabedaten (Pixel) Features zu extrahieren, so dass keine Vorverarbeitungsschritte erforderlich sind, um bestimmte Features extrahieren zu können. Neben der Bilddimension ist eine weitere wichtige Größe dieser CNNs die Anzahl der Filter mit denen Faltungsoperationen, oder auch Convolutions, auf den Rohdaten ausgeführt werden. Je tiefer das CNN ist, d.h. je mehr Filter verwendet werden, desto mehr Features können potentiell extrahiert werden. Jedoch steigt mit jedem Filter auch die Anzahl der mathematischen Operation, wodurch sich die Laufzeit der Algorithmen deutlich erhöhen kann.

In diesem Projekt wird basierend auf einem CNN eine Segmentierung von Brustkrebsgewebe auf Ultraschallbildern durchgeführt. Dies geschieht anhand zweier verschiedener Neural-Network-Modelle, bei denen eine Hyper-Parameteroptimierung vorgenommen wird. Anschließend wird die Performance beider Modelle mittels Jaccard-Index bestimmt und so das bessere Modell ermittelt. Als Alternativmethode wird ein Canny-Edge-Detektor verwendet der mittels Kantenerkennung Krebstumore erkennt.

Die Unterteilung in normal, gutartig oder bösartig ist innerhalb des Projekts redundant, da sich die Aufgabe auf die Erstellung der Segmentierungsmasken beschränkt.

2 Datensatz

Der in diesem Projekt verwendete Datensatz "Breast Ultrasound Images Dataset - Breast ultrasound images for classification, detection and segmentation" [1] enthält insgesamt 1578 Bilder, davon 780 Ultraschallbilder und 798 Masken¹, von insgesamt 680 Patientinnen und stammt aus dem Jahr 2018. Die Durchschnittsgröße der Bilder beträgt

¹Befinden sich getrennte Segmente innerhalb eines Bildes, gibt es für jedes Segment eine einzelne Maske

500 × 500 Pixel und der Datensatz ist in die drei Untergruppen "normal, gutartig und bösartig" unterteilt. In Abb. 1 sind jeweils Beispiele für die Ultraschallbilder mit den dazugehörigen Masken abgebildet.

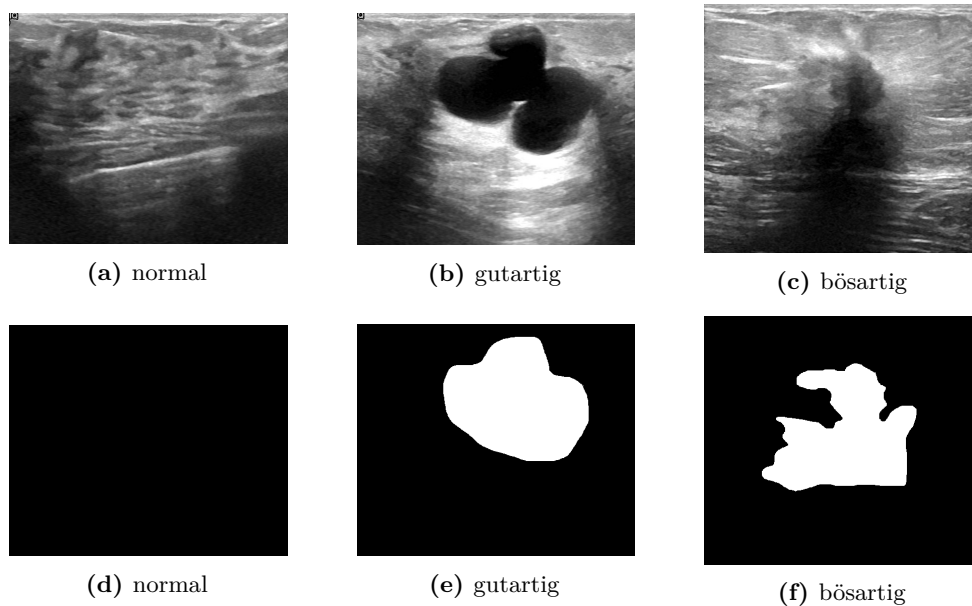


Abbildung 1: Ultraschallbilder der drei Kategorien normal, gutartig und bösartig mit den dazugehörigen Masken.

3 CNN-Architektur

Typischerweise ist die Aufgabe eines CNNs die Objekt-Klassifikation, d.h. es wird anhand eines Bildes eine Vorhersage in Bezug auf die Klassenzugehörigkeit eines Objekts getroffen. Davon abweichend wird bei einer Segmentierung eine Lokalisierung durchgeführt, d.h. dass eine Klassenvorhersage für jeden Pixel getroffen wird.

3.1 U-Net

Für eine Segmentierungsaufgabe zeigt sich, dass eine bestimmte Realisierung eines Fully-CNN², ein sogenanntes U-Net [2], besonders gute Ergebnisse erzielt. In Abb. 2 ist ein solches U-Net abgebildet. Es besteht aus einem Encoder-Path und einem Decoder-Path, wobei der Encoder-Path der typischen Architektur eines CNN folgt. Jede Ebene enthält einen Convolution-Block bestehend aus zwei 3×3 Faltungsoperationen, jeweils gefolgt von einer Rectified-Linear [ReLU]-Aktivierungsfunktion und anschließend Down-Sampling durch eine 2×2 -Max-Pooling-Operation. Die Kernel-Größe 3×3 entspricht dabei der Fläche der Filtermaske, auch Receptive-Field genannt, und repräsentiert die Kontextfläche

²Ein Neuronales Netzwerk, welches nur aus Convolutional-Layern besteht und keine Fully-Connected-Layers enthält.

aus der bei jeder Convolution-Operation Informationen gewonnen wird. Der Encoder-Path kodiert Informationen also der Art, dass die Faltungsoperationen wichtige Features erkennen und die wichtigsten Merkmale dieser Features durch das Down-Sampling extrahiert werden. Nach jedem Convolution-Block reduziert sich die Auflösung des Bildes, behält jedoch die wichtigsten Merkmale bei. Bei gleichbleibender Kernel-Größe wächst damit nach jeder Down-Sampling-Operation das Receptive-Field des Netzwerks, wodurch komplexere Features extrahiert werden können. Um diese komplexeren Features speichern zu können wird die Anzahl der Filter in jeder Ebene sukzessiv erhöht.

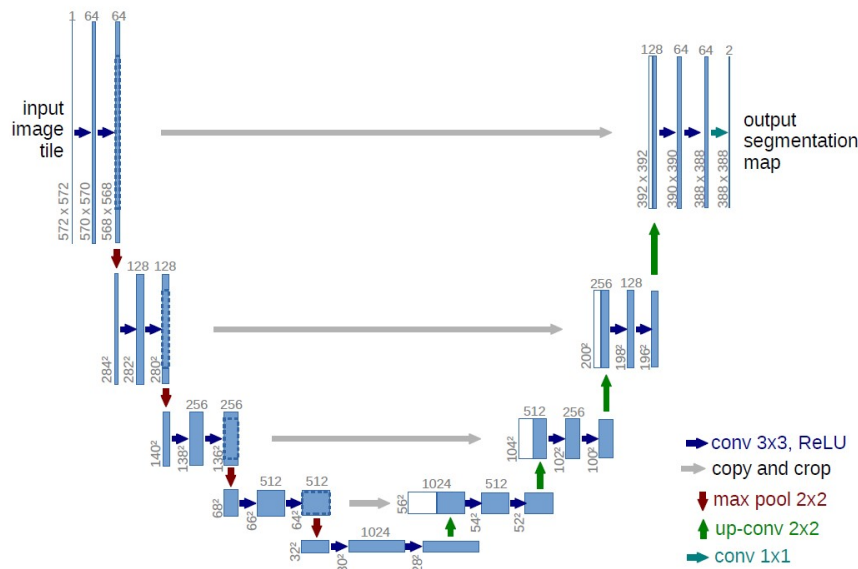


Abbildung 2: U-Net Architektur [2].

Aufgrund des wachsenden Receptive-Fields verliert das Netzwerk jedoch mit jeder hinzugefügten Ebene Informationen darüber, wo diese Features lokalisiert sind. Der symmetrische Decoder-Path gewinnt diese Informationen durch Up-Sampling zurück, so dass die Auflösung des Bildes im Decoder-Path wieder steigt. Für das Up-Sampling werden Transposed-Convolution-Operationen verwendet, sowie Long-Skip-Connections³ die Convolution-Blöcke des Encoder-Path mit Transposed-Convolution-Blöcken des Decoder-Path verbindet. Dieses Hintereinanderreihen von Filtern fügt dem Decoder-Path die verlorenen Ortsinformationen der Features hinzu, die durch weitere Convolutional-Operationen pro Ebene extrahiert werden können. Nach Durchlauf durch das U-Net lernt das Netzwerk somit, welche Features wo lokalisiert sind. In einem letzten Schritt führt eine Sigmoid-Aktivierungsfunktion dazu, dass die Ausgabe des U-Nets einem Zahlenwert zwischen null und eins entspricht.

³Verbindungen von Operations-Blöcken A und C , wobei A mit B und B mit C verbunden ist.

3.2 Attention-Recurrent-Residual-U-Net

Die Attention-Recurrent-Residual-U-Net [AR2-U-Net]-Architektur ist eine Erweiterung des normalen U-Nets aus Kap. 3.1, bestehend aus einem R2-U-Net dem sogenannte Attention-Gates hinzugefügt werden. In Abb. 3 ist ein R2-U-Net gemäß [3] abgebildet. Diese Architektur nutzt die in Abb. 4 gezeigten Residual-Blöcke, sowie die in Abb. 5 gezeigten Recurrent-Blöcke.

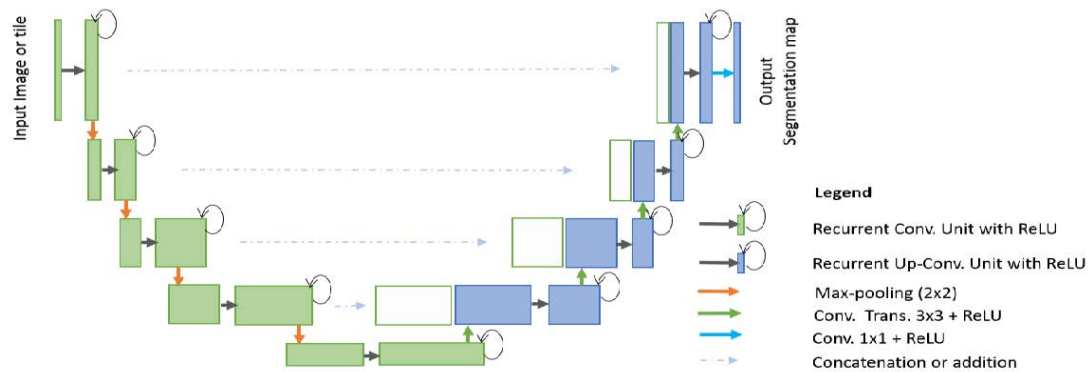


Abbildung 3: R2-U-Net Architektur [3].

Die Residual-Blöcke verfügen, ähnlich zum vorherigen Kapitel, über Short-Skip-Connections die bei besonders tiefen Netzwerken⁴ verhindern, dass der Gradient der Loss-Funktion insignifikant klein wird. Durch Verkettung der Gewichte des Netzwerks werden die Gewichte der verschiedenen Ebenen optimiert, so dass bei der Backpropagation, also der rückwärtsgerichteten Anpassung der Gewichte zur Minimierung der Loss-Funktion, kleine Änderungen im letzten Layer mit kleinen Änderungen im vorletzten Layer und so weiter, multipliziert werden. Somit erfahren die Gewichte nur minimale Änderungen, so dass sich die Loss-Funktion insignifikant ändert. Bei einer Segmentierungsaufgabe führt ein zu tiefes Netzwerk zusätzlich zu einem Verlust der Generalisierungsfähigkeit des Netzwerks, da Informationen aus früheren Ebenen die nur wenig semantische Information enthalten, eine starke Abstraktion der Informationen aus diesen Ebenen zur Folge haben. Die Skip-Connections verhindern also den Verlust der Generalisierungsfähigkeit des Netzwerks. Die Recurrent-Blöcke besitzen Feedback-Connections, durch den Pfeil in Abb. [img:recblock] dargestellt, die mehrere Male hintereinander ausgeführt werden können. Diese Feedback-Connections führen zu einer verbesserten Feature-Representation bei Segmentierungsaufgaben.

⁴Ein Netzwerk mit vielen Ebenen.

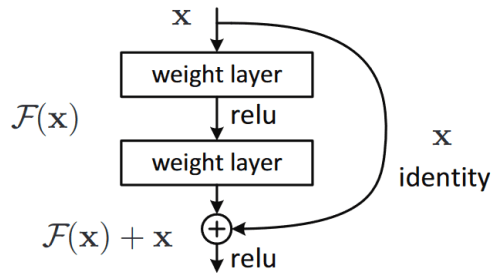


Abbildung 4: Skip-Connection-Block [4]

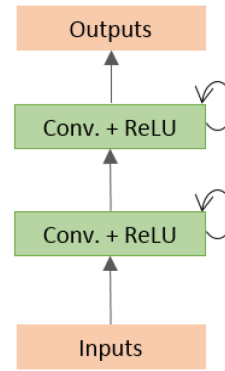


Abbildung 5: Recurrent-Block [3]

Das Hinzufügen von Attention-Gates, in Abb. 6 gezeigt, zur R2-U-Net-Architektur führt während des Trainings dazu, dass nur die relevanten Bereiche eines Bildes betrachtet werden. Im Kontext der U-Net-Architektur haben die Attention-Gates somit die Aufgabe alle redundanten Informationen aus dem Encoder-Path zu entfernen, die andernfalls durch Long-Skip-Connections zum Decoder-Path hinzugefügt werden. Ein Attention-Block kombiniert gemäß Abb. 6 die Ausgabe eines Decoder-Path-Blocks, welcher eine kleinere Dimension und bessere Feature-Repräsentation besitzt, mit der Ausgabe eines Encoder-Blocks, welcher durch eine Long-Skip-Connections repräsentiert wird. Durch Summation dieser Blöcke verkleinern sich diejenigen Gewichte, die redundante Informationen enthalten, wohingegen in beiden Blöcken übereinstimmende Gewichte vergrößert werden. Durch Anwendung einer ReLU-Aktivierungsfunktion und anschließender Sigmoid-Aktivierungsfunktion ergibt sich eine reine Gewichtsmatrix, deren Einträge nahe eins wichtige Features und Einträge nahe null unwichtige Features repräsentiert. Durch Up-Sampling und Multiplikation zur originalen Matrix werden Features entsprechend ihrer Relevanz skaliert. Abb. 7 zeigt den schematischen Aufbau des AR2-U-Nets nach Ref.[5], wobei sich die Blöcke jeder Ebene aus R2-Blöcken zusammensetzen.

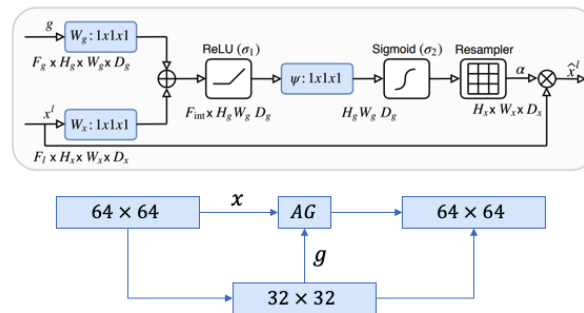


Abbildung 6: Attention-Block [6]

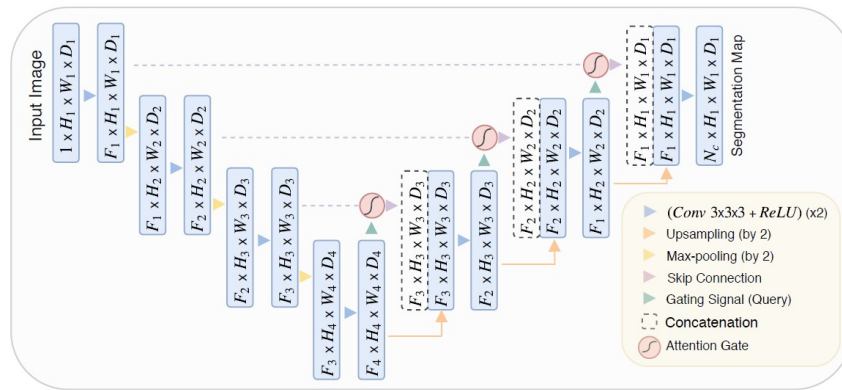


Abbildung 7: AR2-U-Net-Architektur[5].

4 Modellarchitekturen, Optimierungsparameter und Übertraining

Gemäß des vorherigen Kapitels wird jeweils ein U-Net-Modell und ein AR2-U-Net-Modell in Keras [7] implementiert. Aufgrund der beschränkten zur Verfügung stehenden Rechenkapazitäten wird sich bei der Parameteroptimierung auf die Anzahl der Ebenen beschränkt. Weiterhin werden die Bilder auf eine Größe von 128×128 Pixel verkleinert, um ebenfalls Rechenkapazität zu sparen. Da der Datensatz aus Greyscale-Bildern besteht werden zwei Bildkanäle entfernt und als Pre-Processing Schritt die Pixelwerte auf einen Zahlenbereich zwischen null und eins skaliert.

Insgesamt werden Modelle für $n \in [2, 3, 4, 5, 6]$ Ebenen erstellt, wobei sich bei jeder hinzugefügten Ebene die Bilddimension gemäß Abb. 2 halbiert und sich die Anzahl der Filter verdoppelt. Für jedes n wird mit einer Anzahl von $l \in [4, 8, 16, 32, 64]$ Filtern gestartet, wobei für $n = 6$ aufgrund mangelnden Arbeitsspeichers kein Modell für $l = 64$ erstellt wird.

In einem ersten Schritt wird der Datensatz in einen Test- und einen Trainingsdatensatz aufgeteilt, mit einer Testdatensatzgröße von 33 %, sowie einem Stratified-Sampling, was gleiche Klassenproportionen innerhalb der verschiedenen Datensätze garantiert. Als weitere Optimierung werden in einem nächsten Schritt durch Data-Augmentation zusätzliche Trainingssamples erstellt, da sich dadurch signifikante Performance-Verbesserungen auf dem Testdatensatz zeigten. Dazu werden die Trainingsdaten jeweils horizontal als auch vertikal gespiegelt, so dass sich die Menge der Trainingssamples von 522 Bildern auf 1566 Bildern erhöht.

Weiterhin zeigte sich eine minimale Verbesserung der Performance des U-Nets durch Hinzufügen von Dropout-Layern, wobei im Encoder- als auch Decoder-Path jeweils nach der ersten Convolution-Operation der Dropout-Layer eingefügt wird. Startend mit einem Dropout-Wert von 0,1 wird nach jeder Zweiten Ebene dieser Parameter um 0,1 erhöht. Beim AR2-U-Net zeigte sich keine solche Performance-Änderung durch Dropout-Layer, wohingegen bei diesem Modell Batchnormalization eine Verbesserung zeigte.

Weiterhin werden die Modelle mit dem Callback Early-Stopping trainiert, um Übertraining zu verhindern. Es wird ein Early-Stopping-Parameter von 20 Epochen gewählt und die Trainingsdauer auf insgesamt 400 Epochen gesetzt. Ein zusätzlicher Callback ModelCheckpoint garantiert, dass nur das Modell mit dem geringsten Loss-Wert gespeichert wird. Beide Modelle werden mit dem Optimizer Adam kompiliert, der bereits eine dynamische Anpassung der Lernrate vornimmt. Zusätzlich wird ein Callback verwendet, der die Lernrate um den Faktor 0,1 verkleinert, sollte sich die Loss-Funktion nach 10 Epochen nicht verringert haben.

Als Metrik wird anstelle der Accuracy der Jaccard-Index verwendet, da die Accuracy im Falle der Segmentierung nur eine geringe Aussagekraft besitzt ⁵. Der Jaccard-Index J zweier Mengen A und B ist definiert über

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (4.1)$$

und lässt sich piktografisch wie in Abb. 8 gezeigt darstellen. Der Index gibt das Verhältnis der Überschneidungsmenge zur Vereinigungsmenge beider Mengen an. Je näher dieser Index an eins liegt, desto genauer ist die Vorhersage des Modells in Bezug auf die Groundtruth-Maske. Das Negative dieses Index lässt sich daher ebenfalls als Loss-Funktion definieren, die die Netzwerke während des Trainings zu minimieren versuchen.

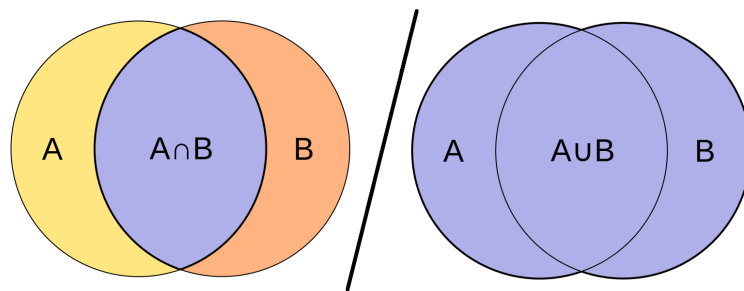


Abbildung 8: Grafische Darstellung des Jaccard-Index [8].

Aufgrund mangelnder Rechenkapazität wird ebenfalls von einer Cross-Validierung abgesehen, da die Rechenzeit um den Faktor der Anzahl an Cross-Validierungen ansteigen würde. Der Trainingsdatensatz wird lediglich einmal in einen Trainings- und einen Validierungsdatensatz, der aus 20 % des Datensatzes besteht, aufgeteilt. Für die Ausgaben der Sigmoid-Aktivierungsfunktion wird ein Cut-Off von 0,5 gewählt, so dass alles oberhalb des Cut-Offs als Tumorgewebe angesehen wird und alles unterhalb als normales Gewebe.

⁵Bspw. wird auf einem gutartigen Ultraschallbild kein Tumorgewebe vorhergesagt, die Groundtruth zeigt jedoch einen kleinen Tumor an. Trotz der falschen Vorhersage wird eine hohe Genauigkeit berechnet, da ein Großteil des "schwarzen" Hintergrunds korrekt erkannt wird.

5 Segmentierung von Ultraschallbildern von Brustkrebs

Die Ergebnisse der Gridsearch sind in Abb. 9 dargestellt. Die x -Achse zeigt die Anzahl der Filter für die jeweiligen n an und auf der y -Achse sind die Metrik-Werte der jeweils besten Modelle eingetragen, die auf dem Testdatensatz ausgewertet wurden. Dabei ergibt sich beim U-Net ein maximaler Wert von $J_{\text{U-Net}} = 0,583$ für $n = 4$ und $l = 16$ und beim AR2-U-Net ein maximaler Wert von $J_{\text{AR2-U-Net}} = 0,571$ für $n = 4$ und $l = 8$.

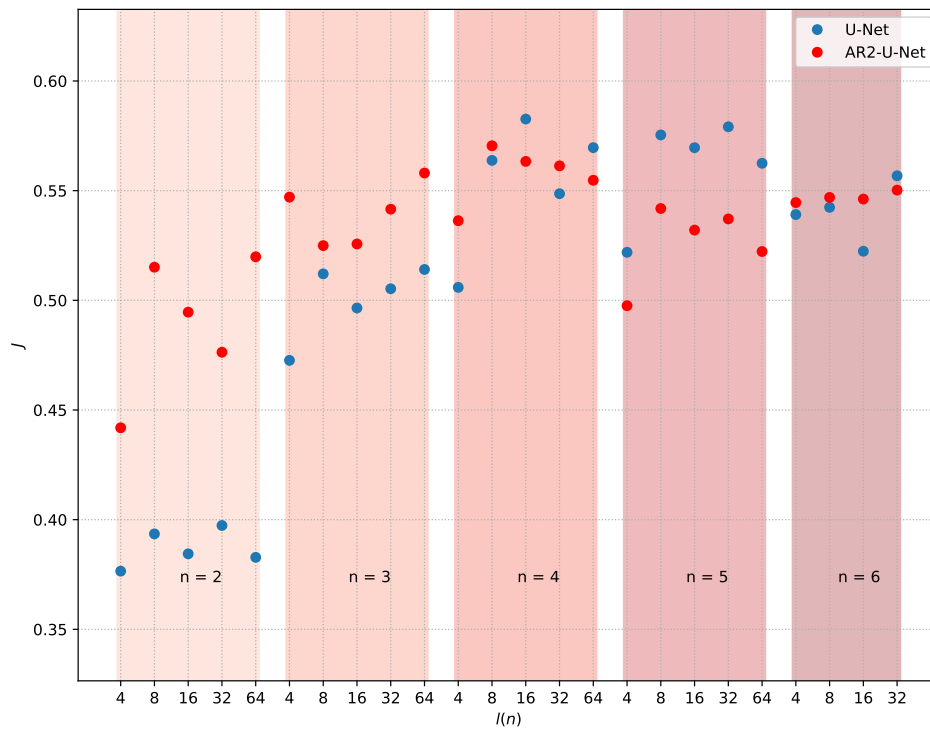


Abbildung 9: Vergleich des U-Nets und AR2-U-Nets auf dem Testdatensatz

In Abb. 10 sind die Loss-Kurven und Metrik-Werte des Trainings- und Validierungsdatensatzes des entsprechenden U-Net-Modells abgebildet. Es ist erkennbar, dass sich jeweils Plateaus bei der Loss-Kurve und dem Jaccard-Index des Validierungssatzes ergeben.

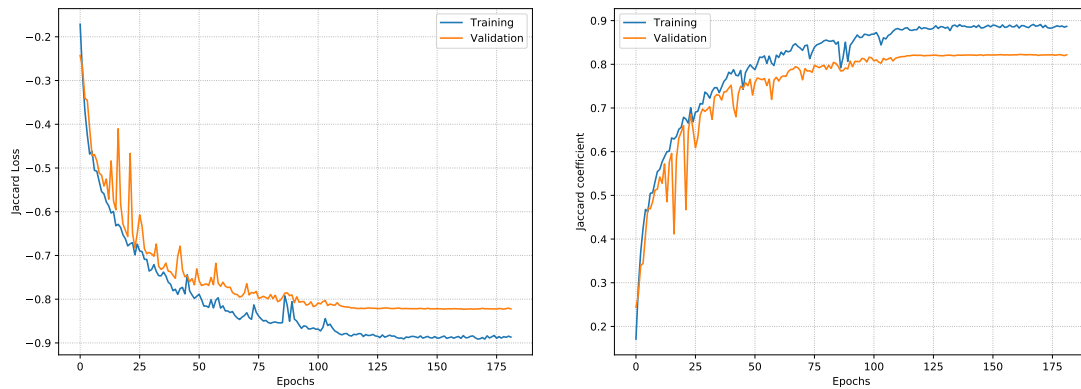


Abbildung 10: Loss-Kurven und Jaccard-Index des besten U-Net-Modells.

In Abb. 11 sind die Loss-Kurven und Metrik-Werte des Trainings- und Validierungsdatensatzes des besten AR2-U-Net-Modells abgebildet. Es ist ebenfalls zu erkennen, dass kein Übertraining auftritt und sich jeweils Plateaus ergeben.

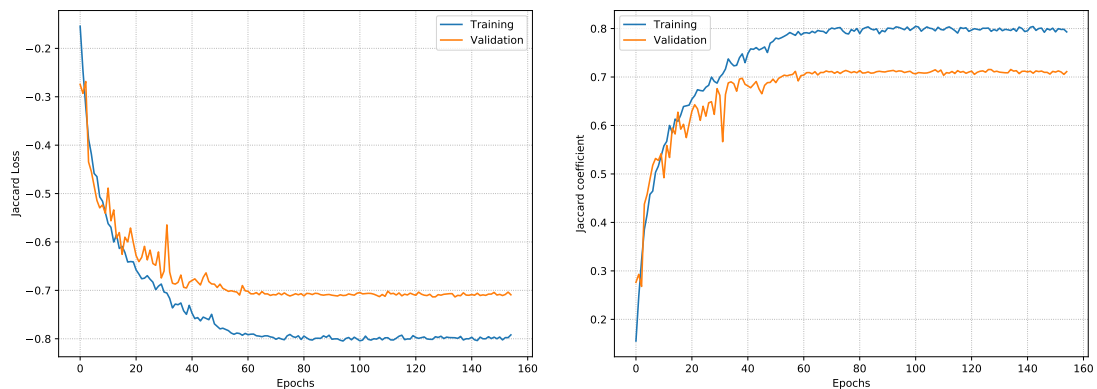


Abbildung 11: Loss-Kurven und Jaccard-Index des besten AR2-U-Net-Modells.

Insgesamt ist die Tendenz erkennbar, dass für $n = 4$ die optimale Anzahl an Ebenen erreicht wird. Das U-Net schneidet minimal besser ab als das AR2-U-Net, wobei die Unterschiede zwischen den Modellen klein sind. Für $n = 2$ und $n = 3$ zeigt sich, dass das komplexere AR2-U-Net-Modell deutlich bessere Werte erzielt, wohingen für $n = 5$ der umgekehrte Fall eintritt. Hier zeigen sich beim U-Net ähnliche Jaccard-Index-Werte wie für $n = 4$. Weiterhin ist erkennbar, dass die maximalen Metrik-Werte auf dem Validierungssatz deutlich besser sind, als auf dem Testdatensatz. Ein Grund dafür wird die Verwendung von augmentierten Trainingsdaten sein. Somit ähnelt der Validierungsdatensatz dem Trainingsdatensatz mehr, als dem Testdatensatz. In Abb. 12

sind die Vorhersagen beider Modelle gegen die Groundtruth und das Ultraschallbild aufgetragen.

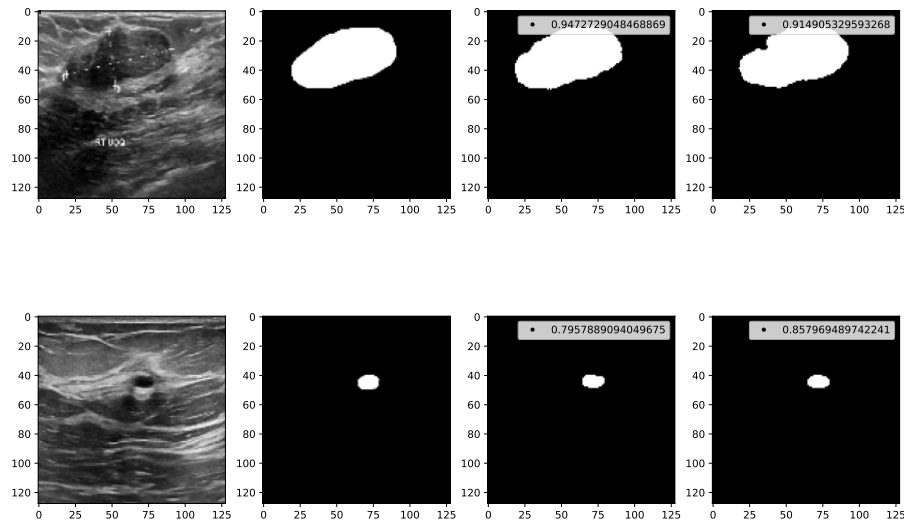


Abbildung 12: Vergleich der Vorhersagen beider Modelle für zwei zufällig gewählte Ultraschallbilder. Für die Vorhersagen der Modelle ist der Jaccard-Index eingetragen.

6 Alternativmethode: Canny-Edge-Detektor

Eine Alternativmethode zu der Nutzung von CNNs ist die Verwendung eines Canny-Edge [CE]-Detektors. Der CE-Detektor reduziert in einem ersten Schritt Bildrauschen durch Verwendung eines Gaußschen Filters. Anschließend werden alle potentiell erkennbaren Kanten auf eine Pixelbreite von eins reduziert. Anschließend werden nur diejenigen Pixel behalten, die einen bestimmten Schwellenwert überschreiten. Ein großer Unterschied zwischen dem CE-Detektor und den CNN-Architekturen ist, dass der CE-Algorithmus direkt auf den Ultraschallbildern angewendet werden kann ohne dass ein Modell trainiert werden muss. Der CE-Detektor wird mit Hilfe der Bibliothek [9] berechnet.

In Abb. 13 ist für ein zufällig gewähltes Ultraschallbild ein Vergleich zwischen der Vorhersage des CE-Detektors und der Vorhersage des U-Nets abgebildet. Ein Vergleich verschiedener Metriken erscheint nicht als sinnvoll, da die Ausgaben beider Modelle unterschiedliche Eigenschaften aufweisen. Ein quantitativer Vergleich zeigt jedoch, worin die Stärken der CNN-Modelle liegen. Die erstellten Segmentierungsmasken weisen eine

höhere Aussagekraft auf, da bei dem CE-Detektor nach wie vor entschieden werden muss, welcher Bereich zum Tumor gehören und bei diesem zusätzlich noch Fine-Tuning der Parameter durchgeführt werden muss.

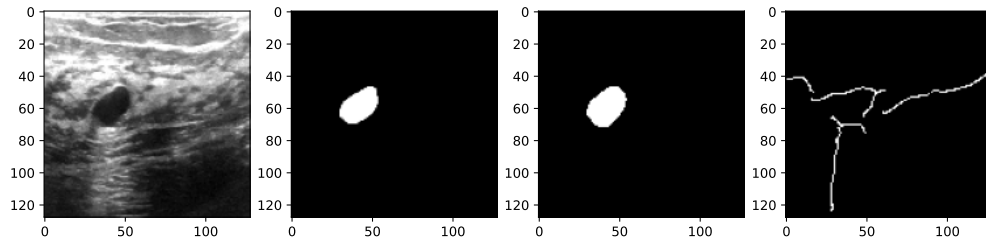


Abbildung 13: Vergleich der Ausgabe des CE-Detektors zur Ausgabe des U-Nets.

7 Zusammenfassung

Es wurden jeweils zwei verschiedene CNN-Architekturen implementiert und durch eine Gridsearch in Bezug auf die Anzahl der Ebenen optimiert. Es zeigt sich, dass die U-Net-Architektur die besten Jaccard-Werte aufweist, wobei die Unterschiede zwischen den Modellen für $n = 4$ sehr gering ausfallen und ohne eine Cross-Validierung nicht genau entschieden werden kann, ob ein Modell tatsächlich bessere Ergebnisse liefert. Jedoch ist eine Tendenz für beide Modelle erkennbar, dass jeweils mehr oder weniger Ebenen als $n = 4$ keine Verbesserung in Bezug auf den Jaccard-Index erbringen. Ein Vergleich der Anzahl an Parametern der besten Modelle ergibt, dass das U-Net 1941777 trainierbare Parameter enthält und das AR2-U-Net 1700317. Somit zeigt sich, dass das AR2-U-Net mit weniger Parametern eine ähnliche Vorhersagekraft aufweist als das U-Net.

Ein Vergleich mit dem CE-Detektor erscheint wenig sinnvoll, da der CE-Detektor lediglich Umrisse detektiert, wohingegen die CNN-Modelle komplette Masken erstellen. Allerdings gibt es keine vergleichbare, oder bessere, Alternativmethodik zur Segmentierung.

Der Vergleich beider CNN-Modelle erweist sich als vielversprechend, da bisher lediglich die Anzahl an Ebenen optimiert wurde und beide Modelle bereits eine sehr gute Vorhersagekraft besitzen. In weiteren Arbeitsschritten können weitere Hyperparameter optimiert werden, um den Jaccard-Index dieser Modelle zu verbessern.

Literatur

- [1] Walid Al-Dhabyani u. a. „Dataset of breast ultrasound images“. In: *Data in Brief* 28 (2020), S. 104863. ISSN: 2352-3409. DOI: <https://doi.org/10.1016/j.dib.2019.104863>. URL: <https://www.sciencedirect.com/science/article/pii/S2352340919312181>.
- [2] Olaf Ronneberger, Philipp Fischer und Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].
- [3] Md Zahangir Alom u. a. *Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation*. 2018. arXiv: 1802.06955 [cs.CV].
- [4] *Intuitive Explanation of Skip Connections in Deep Learning*. <https://theaisummer.com/skip-connections/>. Accessed: 2021-08-15.
- [5] Ozan Oktay u. a. *Attention U-Net: Learning Where to Look for the Pancreas*. 2018. arXiv: 1804.03999 [cs.CV].
- [6] *A detailed explanation of the Attention U-Net*. <https://towardsdatascience.com/a-detailed-explanation-of-the-attention-u-net-b371a5590831>. Accessed: 2021-08-15.
- [7] François Chollet u. a. *Keras*. <https://keras.io>. 2015.
- [8] *Jaccard-Koeffizient*. <https://de.wikipedia.org/wiki/Jaccard-Koeffizient>. Accessed: 2021-08-15.
- [9] Stéfan van der Walt u. a. „scikit-image: image processing in Python“. In: *PeerJ* 2 (Juni 2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453. URL: <https://doi.org/10.7717/peerj.453>.