

# **Vergleich verschiedener Modell-Architekturen und Modalitäten zur Erkennung von Hochwasser mit Satellitendaten**

Bachelorarbeit

vorgelegt am 22.04.2022

Fakultät Wirtschaft

Wirtschaftsinformatik - Data Science

WWI19F

von

TOBIAS HÖLZER

Name des Betreuers in der Ausbildungsstätte: DHBW Stuttgart:

IBM  
Dr. Jonas Weiss  
Research

Prof. Dr. Holzweißig  
Studiendekan Wirtschaftsinformatik

Unterschrift des Betreuers

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>Tabellenverzeichnis</b>	<b>VI</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Zielsetzung und Ablauf der Arbeit . . . . .	2
1.2 Aufbau der Arbeit . . . . .	3
<b>2 Diskussion des aktuellen Stands der Forschung und Praxis</b>	<b>4</b>
2.1 Hochwasser: Eigenschaften, Ursachen, Ablauf und Auswirkungen . . . . .	4
2.1.1 Nomenklatur deutscher Begriffe zu Hochwasser . . . . .	5
2.1.2 Einordnung und Arten von Überschwemmungen . . . . .	5
2.1.3 Folgen und Auswirkungen von Überschwemmungen . . . . .	7
2.2 Datenerfassung von Überschwemmungen durch Satelliten . . . . .	7
2.2.1 Radar basierte Ansätze . . . . .	8
2.2.2 Optische Erfassung . . . . .	8
2.2.3 Bestehende annotierte Datensätze zur Hochwasser-Erkennung . . . . .	11
2.3 Aufgabe und Metriken der Bild-Segmentierung zur Hochwasser-Erkennung . . . . .	13
2.4 Statistische Modelle zur Erkennung von Überschwemmungen auf Satelliten . . . . .	17
2.4.1 Ansätze basierend auf Grenzwerten . . . . .	18
2.4.2 Entscheidungsbäume und Random-Forests . . . . .	20
2.4.3 Einsatz von Algorithmen der Cluster Erkennung . . . . .	22
2.5 Vorstellung und historische Entwicklung neuronaler Modelle der Bild-Segmentierung	23
2.5.1 Funktionsweise von Encoder-Decoder Netzwerke . . . . .	23
2.5.2 Historische Entwicklung von Encoder Modellen . . . . .	24
2.5.3 Decoder-Architekturen in der Bild-Segmentierung . . . . .	27
2.5.4 Methoden beim Training von neuronalen Modellen . . . . .	29
<b>3 Erläuterung und Durchführung der Experimente</b>	<b>31</b>
3.1 Nähere Untersuchung des Datensatzes . . . . .	31
3.1.1 Normalisierungs-Methoden . . . . .	33
3.1.2 Einstellung des Datensatzes zur Verwendung in Experimenten . . . . .	35
3.2 Methodik zur Evaluation . . . . .	35
3.3 Training zweier Vergleichsmodelle . . . . .	36
3.3.1 Statistisches SAR-Grenzwert Modell . . . . .	36
3.3.2 Naives Training eines U-Net . . . . .	38
3.4 Verschiedene Verlustfunktionen im Vergleich . . . . .	41
3.4.1 Verhalten von Verlustfunktionen in unausgeglichenen Datensätzen . . . . .	41
3.4.2 Anwendung ausgewählter Verlustfunktionen . . . . .	45
3.5 Einfluss der Verwendung verschiedener Modalitäten . . . . .	46
3.5.1 Spektroskopische Betrachtung des Datensatzes . . . . .	47
3.5.2 Training des U-Net mit ausgewählten Kombinationen von Modalitäten . . . . .	48
3.6 Optimierung eines neuronalen Modells . . . . .	50
3.6.1 Sequentielle Ermittlung der Architektur . . . . .	50

3.6.2	Statistische Hyperparameter-Suche . . . . .	53
3.7	Training eines Modells mit optimierten Hyperparametern . . . . .	54
<b>4</b>	<b>Evaluierung und Vergleich des Modells mit vergleichbaren Arbeiten</b>	<b>56</b>
4.1	Qualitative Evaluation der optimierten Modelle . . . . .	56
4.2	Quantitative Evaluation der optimierten Modelle . . . . .	57
4.3	Erfolgs der durchgeföhrten Experimente . . . . .	59
4.4	Vergleich mit ähnlichen Arbeiten aus der Forschung . . . . .	60
4.5	Möglichkeiten der weiteren Optimierung, Ausblick und Implikationen für die Wissenschaft . . . . .	62
<b>5</b>	<b>Fazit</b>	<b>64</b>
<b>Literaturverzeichnis</b>		<b>65</b>
<b>Appendix</b>		<b>73</b>

# Abkürzungsverzeichnis

<b>ACC</b>	Accuracy
<b>AWEI</b>	Automated Water Extraction Index
<b>CNN</b>	Convolutional Neural Network
<b>DNN</b>	Deep-Neural-Networks
<b>DPN</b>	Deep Parsing Network
<b>DSR</b>	Design-Science-Research
<b>ESA</b>	European Space Agency
<b>FCN</b>	Fully-Convolutional Neural Network
<b>FLOP</b>	Floating Point Operation
<b>FPN</b>	Feature Pyramid Network
<b>gcForest</b>	multi-Grained Cascade Forest
<b>IoU</b>	Intersection over Union
<b>JI</b>	Jaccard-Index
<b>JRC</b>	European Joint Research Centre
<b>LKA</b>	Large Kernel Attention
<b>KNN</b>	Künstliche Nueronale Netze
<b>MAC</b>	Multiply–Accumulate Operation
<b>mIoU</b>	Mean Intersection over Union
<b>MLF</b>	Multi-Layer Feed-Forward Neural Network
<b>NDVI</b>	Normalized Difference Vegetation Index
<b>NDWI</b>	Normalized Difference Water Index
<b>NIR</b>	Near Infrared
<b>NLP</b>	Natural Language Processing
<b>OA</b>	Overall Accuracy
<b>PPV</b>	Positive Predicted Value
<b>SAR</b>	Synthetic Aperture Radar
<b>SGD</b>	Stochastischer Gradienten Abstieg
<b>SMP</b>	Segmentation Models Pytorch
<b>SVM</b>	Support Vektor Machines

**SWIR** Short Wavelength Infrared

**TNR** True Negative Rate

**TPR** True Positive Rate

**VAN** Visual Attention Network

**ViT** Visual Transformer

**VGG** Visual Geometry Group Network

# Abbildungsverzeichnis

1	Fluviale, pluviale und Küsten-Flut . . . . .	6
2	Kombination verschiedener optischer Bänder . . . . .	11
3	Beispiel der Bild-Annotation von Rahnemoonfar u. a. 2020 . . . . .	12
4	Bild-Segmentierung anhand eines Beispiels . . . . .	13
5	Visualisierung der Intersection over Union (IoU) Metrik . . . . .	15
6	Bestimmung eines Grenzwert über Verteilung eines Datensatzes . . . . .	19
7	Beispiel einer Precision-Recall-Curve . . . . .	20
8	Deep Forest: gcForest Architektur . . . . .	21
9	Support-Vector-Machine . . . . .	22
10	Aufbau eines Convolutional Neural Network (CNN) . . . . .	24
11	Visual Attention Network (VAN) Architektur . . . . .	26
12	Fully-Convolutional Neural Network (FCN) . . . . .	27
13	U-Net Architektur . . . . .	28
14	Visualisierungen zu einer Satellitenaufnahme eines Gebiets in Bolivien . . . . .	31
15	Sen1Floods11 Datensatz: Verteilung der Klassen . . . . .	33
16	Lineares Modell: Resultate der Rastersuche . . . . .	37
17	Lineare Klassifikation anhand SAR-Verteilung . . . . .	37
18	Qualitative Visualisierung: U-Net Vergleichsmodell . . . . .	40
19	Verlustfunktionen: Experimentaufbau . . . . .	41
20	Verlustfunktionen: Ergebnisse des Experiments . . . . .	44
21	Verlustfunktionen: Ergebnisse des Trainings . . . . .	46
22	Spektroskopie des Sen1Flood11 Datensatzes . . . . .	47
23	Modalitäten: Ergebnisse des Trainings . . . . .	49
24	Architektur: Ergebnisse des Trainings . . . . .	52
25	Architektur: Ergebnisse der Rastersuche . . . . .	55
26	Qualitative Ergebnisse der optimierten Modelle . . . . .	56
27	Vergleich zwischen optimierten und unoptimierten Modellen . . . . .	59
28	Metriken mehrerer Trainings im Vergleich mit dem Verlust . . . . .	60
29	Quantitativer Vergleich mit anderen Arbeiten . . . . .	61
30	Visualisierungen zu einer normalisierten Satellitenaufnahme eines Gebiets in Bolivien	76
31	Qualitative Ergebnisse des Trainings zur Bewertung von Modalität-Kombinationen	77
32	Qualitative Ergebnisse des Trainings zur Bewertung von Verlustfunktionen . . . . .	78

# Tabellenverzeichnis

1	Übersicht der Sentinel-2 Bänder: Wellenlänge, Auflösung und Verwendung . . . . .	9
2	Sen1Floods11 Datensatz: Statistische Eigenschaften . . . . .	32
3	Vergleichsmodelle: Erfasste Metriken . . . . .	38
4	Training-Parameter: U-Net Vergleichsmodell . . . . .	39
5	Training-Parameter: Verlustfunktionen . . . . .	45
6	Training-Parameter: Modalitäten . . . . .	48
7	Training-Parameter: Architektur . . . . .	51
8	Training-Parameter: Rastersuche . . . . .	54
9	Training-Parameter: Optimierte Modelle . . . . .	55
10	Ergebnisse des finalen Trainings der ersten Architektur . . . . .	57
11	Ergebnisse des finalen Trainings der zweiten Architektur . . . . .	58
12	Sen1Floods11 Datensatz: Statistische Eigenschaften nach Normalisierung über die Standardabweichung . . . . .	74
13	Sen1Floods11 Datensatz: Statistische Eigenschaften nach MinMax-Normalisierung .	75
14	Sen1Floods11 Datensatz: Statistische Eigenschaften nach 3-Sigma-Normalisierung .	75

# 1 Einführung

Überschwemmungen durch Hochwasser gehören zu den bedrohlichsten Naturkatastrophen für die Gesellschaft. Forschende vermuten einen statistisch signifikanten Anstieg der Häufigkeit im Auftreten und einen Anstieg der Stärke von Hochwasser auf der Welt, verursacht durch die Klimaerwärmung.<sup>1</sup> Zurzeit können sie keine präziseren Aussagen über diesen Anstieg geben, da es keinen globalen, georeferenzierten Datensatz, welcher Überschwemmungen auf der ganzen Welt abbildet, gibt. Ein solcher Datensatz würde es weiterhin erlauben, die Entwicklung der Stärke und Häufigkeit von Hochwasser global genauer zu studieren, auch in Bezug auf geografische Regionen.<sup>2</sup> Auch könnte mit solch einem Datensatz eine globale Risiko-Karte erstellt werden, welche von Überschwemmungen besonders betroffene Landstriche identifiziert und es den Menschen in betroffenen Regionen erlaubt, sich besser auf Hochwasser vorzubereiten um zukünftige Überschwemmungen zu vermeiden. Um Menschen gezielt vor spezifischen Hochwasser und daraus resultierenden Überschwemmungen schützen zu können, müssen solche so frühzeitig wie möglich erkannt und vorhergesagt werden.<sup>3</sup> Zur Entwicklung und Evaluierung von Modellen zur Vorhersage von Hochwasser wird eine Grund-Wahrheit benötigt, eine Reihe von annotierten, historischen Daten.<sup>4</sup> Ein oben angesprochener Datensatz könnte als solche Grund-Wahrheit dienen. **Es gibt demnach drei konkrete Use-Cases für einen globalen, georeferenzierten Hochwasser-Datensatz:** (1) die historische Analyse der Entwicklung von Stärke und Häufigkeit von Hochwasser, (2) die Erstellung einer Hochwasser-Risiko-Karte und (3) das Dienen als Grund-Wahrheit für Modelle zur Vorhersage von Hochwasser.

Um einen solchen Datensatz zu erzeugen, ist es notwendig Hochwasser auf historischen Daten von Satelliten zu erkennen und automatisch zu klassifizieren. Ein Modell, welches einen solchen Datensatz mittels Satelliten-Daten erzeugt, könnte auch auf zukünftigen Satelliten-Daten Hochwasser automatisch erkennen und klassifizieren. Da Satellitendaten häufig als Bilder dargestellt werden können, eignet sich der Bild-Segmentierung-Ansatz für diese Aufgabe.<sup>5</sup> Bei dieser geht es darum, Bereiche auf Bildern zu erkennen und zu klassifizieren. In diesem Fall sollen mit Wasser bedeckte Landflächen von trockenen Landflächen separiert werden können. Zur Bild-Segmentierung wird häufig maschinelles Lernen eingesetzt, um Modelle zu trainieren, welche diese Aufgabe selbstständig lösen sollen.<sup>6</sup> Die automatische Lösung dieser Aufgabe gestaltet sich kompliziert, da sehr viele Faktoren bei Hochwasser Einfluss nehmen.<sup>7</sup> So ändert sich zum Beispiel die Farbe des Wassers je nach Region, Wetter und Tageszeit.<sup>8</sup> Daher sind bestehende Methoden, welche Hochwasser automatisch erkennen, noch unausgereift. Sie weisen eine unzureichende

---

<sup>1</sup>Vgl. auch im Folgenden Masson-Delmotte u. a. 2021, S. 19

<sup>2</sup>Vgl. Bonafilia u. a. 2020, S. 843; Vgl. auch im Folgenden Mateo-Garcia u. a. 2021, S. 1

<sup>3</sup>Vgl. Tsakiris 2014, S. 1361; Vgl. Hartmann/Juepner 2017, S. 143

<sup>4</sup>Vgl. Bates u. a. 2021, S. 1 f.

<sup>5</sup>Vgl. Khatami/Mountrakis/Stehman 2016, S. 89

<sup>6</sup>Vgl. Minaee u. a. 2021, S. 1

<sup>7</sup>Vgl. Bates u. a. 2021, S. 4f.

<sup>8</sup>Vgl. auch im Folgenden Bonafilia u. a. 2020, S. 841, 843

Genauigkeit auf, sind meist auf bestimmte Regionen optimiert und nicht generisch genug. Weiterhin zeigt sich in der Forschung keine einheitliche Methode zur Evaluierung solcher Modelle und daher auch keine Definition eines Verlusts durch eine fehlerhafte Erkennung. Dies ist jedoch relevant für das Training eines Modells mit maschinellem Lernen, da dabei das Modell auf einen definierten Zielwert – den Verlust – optimiert wird.<sup>9</sup>

## 1.1 Zielsetzung und Ablauf der Arbeit

Ziel dieser Arbeit ist es daher eine Methode zu entwickeln mit welcher ein Datensatz, auf welchem neben Wasser auch Hochwasser abgebildet ist, automatisch erzeugt werden kann. Dazu sollen neue Methoden entwickelt, respektive existierende Methoden optimiert und allenfalls kombiniert werden. Damit soll eine vergleichbare Genauigkeit und Zuverlässigkeit erreicht werden, wie sie die Methode der manuellen Annotierung besitzt, ohne dabei mehr Rechenkapazitäten zu gebrauchen als heutige automatische Methoden. Weiterhin soll diese Methode auf einem Modell basieren, welches mit Methoden des maschinellen Lernens trainiert wird, da diese Methode bei der Bild-Segmentierung bereits bei vielen Analysen von Satellitenbildern erfolgreich angewendet wurde.<sup>10</sup> Demnach lautet die zentrale Forschungsfrage dieser Arbeit: *Ist es mit heutigem Wissensstand möglich, eine ähnliche, wenn nicht sogar bessere Genauigkeit und Zuverlässigkeit bei der Erkennung von Hochwasser auf Satellitendaten als durch manuelle, menschliche Annotierung zu erzielen?*

In dieser Arbeit wird die Design-Science-Research (DSR) nach Gregor/Hevner 2013 Methode als übergeordnete Forschungsmethode angewandt. Diese erlaubt es mehrere präzise Forschungsmethoden für einzelne Analysen über verschiedene Iterationen innerhalb eines einheitlichen Konstrukt einzusetzen.<sup>11</sup> Dabei entsteht nach jeder Iteration jeweils ein Artefakt, welches in den folgenden Iterationen genutzt werden kann. Diese Arbeit beantwortet die zentrale Forschungsfrage über die folgenden sechs Iterationen:

In der ersten Iteration wird (1) eine *einheitliche Methodik zur Evaluierung* der Methoden erarbeitet. Dazu werden angewandte Evaluationen aus der Forschungsliteratur zur Bild-Segmentierung entnommen und die genutzten Metriken miteinander verglichen. Artefakt der Iteration ist ein Set an Metriken u. ä., welches nicht nur die Genauigkeit und Zuverlässigkeit in Bezug auf unterschiedliche geografische Regionen der Methode wiedergibt, sondern auch den Rechen- und Speicheraufwand der Methode aufzeigt.

Darauf werden (2) zwei *Basis-Methoden*, welche Vergleichsmodelle mit maschinellem Lernen trainieren, entwickelt und mit der in der ersten Iteration entwickelten Methodik evaluiert. Die erste Basis-Methode nutzt ein statistisches Modell, welches keine neuronalen Netze benutzt. So soll später zwischen neuronalen und statistischen Methoden verglichen werden können. Grundlage

---

<sup>9</sup>Vgl. Mitchell 1997, S. 7 f.

<sup>10</sup>Siehe McFeeters 1996, und weitere; Fretwell/Staniland/Forcada 2014; Augusteijn/Warrender 1998

<sup>11</sup>Vgl. auch im Folgenden Gregor/Hevner 2013

der zweiten Basis-Methode ist ein U-Net, eine Modell-Architektur, welche sehr erfolgreich in der Bild-Segmentierung eingesetzt wird.<sup>12</sup> Die Ergebnisse aus der Evaluation sowie die Implementierung der Methoden selbst sind das resultierende Artefakt der zweiten Iteration.

Anhand der Basis-Methode aus der zweiten Iteration wird (3) die Verwendung von verschiedenen *Verlustfunktionen* beim Trainieren von Modellen mit maschinellem Lernen studiert und deren Einfluss auf die Ergebnisse mit den entsprechenden Metriken aus der ersten Iteration evaluiert. So wird eine angepasste, optimale Verlustfunktion hergeleitet und evaluiert und stellt das Artefakt der dritten Iteration dar.

Weiterhin wird (4) die Nutzung von unterschiedlichen *Bändern / Modalitäten und Daten-Features* in Bezug auf den Einfluss auf die Ergebnisse mit den entsprechenden Metriken aus der ersten Iteration evaluiert. Die Ergebnisse aus dieser Evaluation und das Wissen über die Einflüsse der verschiedenen Bänder und ihrer Kombinationen stellt das Artefakt der vierten Iteration dar.

Basierend auf der optimalen Verlustfunktion und den besten Modalitäten, also den Artefakten aus der dritten und vierten Iteration, werden (5) verschiedene *Modell-Architekturen* angewandt und miteinander verglichen. Dazu werden mehrere Modelle der verschiedenen Architekturen trainiert und in einem Benchmark anhand der Metriken aus der ersten Iteration miteinander verglichen. So wird die für die Aufgabe der Hochwasser-Erkennung auf Satellitenbildern beste Modell-Architektur gefunden. Gleichzeitig werden über eine Rastersuche die verschiedenen Architekturen samt Hyper-Parameter verglichen. Die Implementierung der besten Architektur stellt das Artefakt der fünften Iteration dar.

In der letzten Iteration wird (6) ein Modell basierend auf der besten Modell-Architektur mit den besten Hyper-Parametern, der idealen Verlustfunktion und den besten geeigneten Modalitäten trainiert. Diese optimierte Methode stellt schließlich das Artefakt dieser Arbeit dar. Es wird abschließend genauer evaluiert und mit Methoden aus der Literatur verglichen.

## 1.2 Aufbau der Arbeit

In Kapitel 2 (Diskussion des aktuellen Stands der Forschung und Praxis) zunächst der theoretische Hintergrund von Hochwasser, Erdbeobachtung durch Satelliten sowie Methoden zur Bild-Segmentierung dargelegt. Hier wird auch der aktuelle Stand der Forschung näher diskutiert. Basierend auf den Erkenntnissen der Literatur werden die in der Zielsetzung beschriebenen Experimente durchgeführt, um genauere Einblicke in das Verhalten von neuronalen Modellen bei der Erkennung von Hochwasser auf Satellitendaten zu gewinnen. Die schlussendlich entwickelte Methode wird im Kapitel 4 (Evaluierung und Vergleich des Modells mit vergleichbaren Arbeiten) sowohl näher evaluiert als auch mit anderen Methoden aus der Forschung verglichen und diskutiert. Im Kapitel 5 (Fazit) wird schlussfolgernd die zentrale Fragestellung beantwortet.

---

<sup>12</sup>Vgl. Ronneberger/Fischer/Brox 2015, S. 237 f.

## 2 Diskussion des aktuellen Stands der Forschung und Praxis

Überschwemmungen durch Hochwasser sind natürliche Ereignisse, womit die Menschen seit jeher konfrontiert werden. Im 20. Jahrhundert waren Überschwemmungen für über 6 Millionen Tote verantwortlich.<sup>13</sup> Durch den Verlust ihres Zuhauses und der Angst vor solchen Ereignissen sind zwischen 1980 und 2009 schätzungsweise 4,5 Millionen Menschen geflohen, Überschwemmungen stellen daher auch eine bedeutende Fluchtursache dar.<sup>14</sup> Somit gehören Überschwemmungen zu den häufigsten und bedrohlichsten Naturkatastrophen der Welt. Forschende weltweit befassen sich mit Hochwasser, wie sie entstehen, wie man sie voraussagt und wie man Überschwemmungen verhindern kann. Es lassen sich drei Forschungsfelder die sich mit Fluten beschäftigen aus der Forschung ableiten: *Hochwasser-Detektion*, *Hochwasser-Vorhersage* und *Hochwasser-Management*.

Die *Hochwasser-Detektion*, auch Hochwasserrisiko-Ermittlung genannt, befasst sich mit dem Messen und der automatischen Erkennung von Hochwasser, zum Beispiel über Satellitendaten. Um Menschen gezielt vor den Auswirkungen von Hochwasser schützen zu können, beschäftigt sich die *Hochwasser-Vorhersage* mit der Vorhersage der Stärke und möglichen Auswirkungen von Hochwasser. Heutzutage versuchen wissenschaftliche Arbeiten zur Hochwasser-Vorhersage, zum Beispiel Brunner u. a. 2021, Trends in historischen Daten zu finden. Im Vordergrund heutiger Forschung steht somit weniger die Vorhersage des Ablaufs von spezifischen Hochwasser-Events, sondern vielmehr die Entwicklung von statistischen Aussagen über Hochwasser bezüglich z. B. Regionen oder Jahreszeiten. Die Modelle der Hochwasser-Vorhersage werden daher meist mit Daten der Hochwasser-Erkennung validiert. Das *Hochwasser-Management*, auch Hochwasserrisiko Management, nutzt die Vorhersagen und Erkennung von Hochwasser, um Menschen gezielt vor Hochwasser zu schützen.<sup>15</sup> Alle drei Forschungsfelder sind stark miteinander verknüpft, somit finden sich auch fach-übergreifende Arbeiten wie zum Beispiel die von Tsakiris 2014. Diese Arbeit fokussiert sich jedoch nur auf die Erkennung von Hochwasser.

### 2.1 Hochwasser: Eigenschaften, Ursachen, Ablauf und Auswirkungen

Um Modelle zu entwickeln, welche Hochwasser mithilfe von Satellitendaten erkennen können, muss erst eine Eingrenzung des Begriffes *Hochwasser* stattfinden. Deshalb werden im Folgenden die Eigenschaften zur Entstehung sowie die Konsequenzen von Hochwasser und Fluten erklärt.

---

<sup>13</sup>Vgl. Doocy u. a. 2013, S. 1

<sup>14</sup>Vgl. Doocy u. a. 2013, S. 7

<sup>15</sup>z.B. Yeo 2002

### 2.1.1 Nomenklatur deutscher Begriffe zu Hochwasser

Im deutschsprachigen Raum gibt es drei Wörter, welche meist als Synonyme für einander Verwendung finden: Hochwasser, Flut und Überschwemmung. In der Literatur lässt sich leider keine eindeutige Begriffsabgrenzung finden, daher grenzt diese Arbeit die drei Begriffe im Folgenden genauer ein. Alle drei Begriffe lassen sich ins Englische mit *Flood* übersetzen und sind auch im Deutschen schwer voneinander abzugrenzen. Im Deutschen können je nach Kontext jedoch unterschiedliche Sachverhalte gemeint sein: Der klassische Begriff Flut beschreibt den Anstieg des Meeresspiegels im Laufe des natürlichen Gezeitenspiels. In diesem Zusammenhang bedeutet der Begriff Hochwasser den höchsten Stand des Wassers in einer Gezeitenperiode. Da es sich bei den Gezeiten um ein regelmäßig wiederkehrendes Ereignis handelt, ist hierbei noch nicht von einer Naturkatastrophe die Rede. Im Kontext einer Naturkatastrophe wird der Begriff Hochwasser genutzt, wenn der Wasserstand signifikant höher ist als normalerweise. In diesem Fall ist es möglich, dass Wasser auf das umliegende Land übertritt – eine Überschwemmung aufgrund eines Hochwassers tritt auf. Häufig wird der Begriff Flut auch als Synonym für eine Überschwemmung genutzt. Diese Arbeit nutzt im Weiteren beide Begriffe und bezieht sich nur auf den Kontext der Naturkatastrophe.

In der Fachsprache wird zwischen verschiedenen Arten von Überschwemmungen unterschieden: *fluviale Flut*, *pluviale Flut* und *Küstenflut*.<sup>16</sup>

### 2.1.2 Einordnung und Arten von Überschwemmungen

Eine fluviale Flut entsteht durch ein starkes Hochwasser eines Binnengewässers.<sup>17</sup> In Abbildung 1 ist dies schematisch dargestellt. Durch starken Regen und Schneefall steigt der Wasserstand eines Flusses oder eines Sees an (Hochwasser). Übersteigt der Wasserstand nun die Höhe des Ufers, gelangt Wasser auf das umliegende Land – es tritt eine Überschwemmung auf. Durch eine plötzliche Lawine, einen Erdsturz oder ähnlichem können Flutwellen in Flüssen entstehen. Diese bewegen sich flussabwärts und können beim Zusammentreffen zweier Flüsse einander verstärken. So können auch durch starke Flutwellen plötzliche und heftige Überschwemmungen auftreten.

Abwassersysteme besitzen eine Limitierung bezüglich der ableitbaren Wassermenge. Wenn in urbanen Gebieten, welche durch eine hohe Bodenversiegelung gekennzeichnet sind, in kurzer Zeit viel Wasser eintritt, können Abwassersysteme an ihre Grenzen kommen und nicht mehr genug Wasser ableiten. Eine so entstehende Flut wird pluviale Flut genannt.<sup>19</sup> Eine pluviale Flut ist ebenfalls in Abbildung 1 schematisch dargestellt. Eine weitere Möglichkeit zur Entstehung von pluviale Fluten basiert auf derselben Logik: Die Wasseraufnahme-Kapazität des Bodens ist begrenzt. Sie besitzt also eine ähnliche Limitierung wie menschliche Abwassersysteme. Wenn nun viel Wasser durch Regen oder von höheren Gebieten eintritt und die Erde das Wasser nicht mehr

<sup>16</sup>Übersetzt aus den englischen Begriffen fluvial-, pluvial- und coastal-flood.

<sup>17</sup>Vgl. auch im Folgenden Chen u. a. 2010, S. 1492

<sup>18</sup>Entnommen aus: Zurich o.J.

<sup>19</sup>Vgl. Chen u. a. 2010, S. 1492

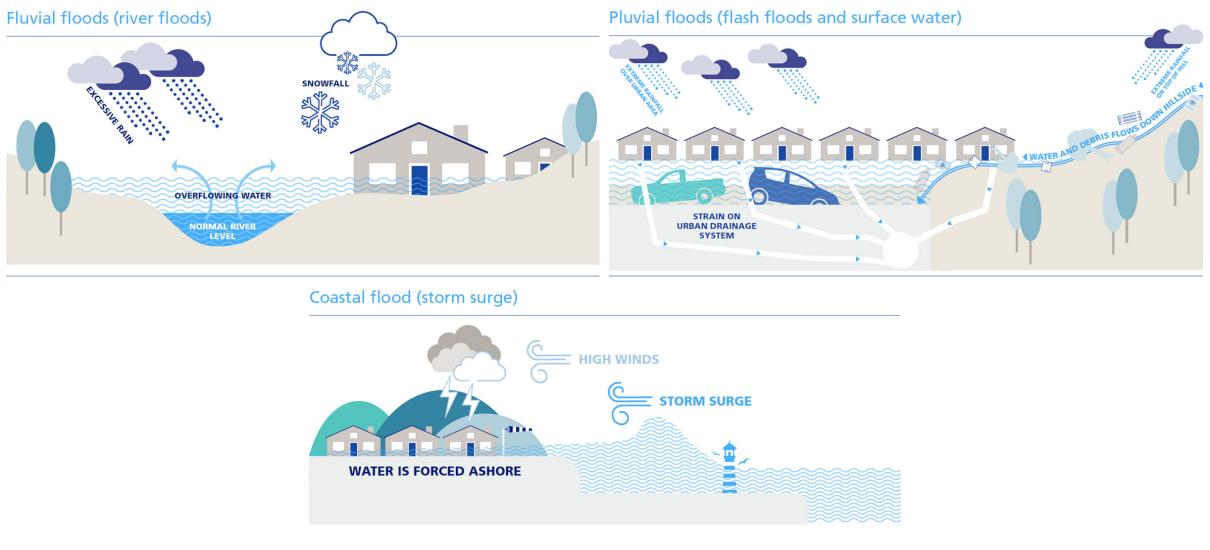


Fig. 1: Grafische Erklärung der drei Flut-arten. Oben links: Bei einer fluvialen Flut sorgt ein starkes Hochwasser für ein Übertreten des Flusses auf umliegendes Land. Oben rechts: Wenn auf versiegeltem Land die Abwassersysteme überlastet sind oder die Erde mit Wasser gesättigt ist, kann ein-kommendes Wasser durch zum Beispiel Starkregen nicht mehr abfließen und staut sich in Tälern an. Eine pluviale Flut tritt auf. Unten mittig: Durch Sturmfluten oder Tsunamis erzeugte Wellen übersteigen den Küstenschutz wie zum Beispiel Dünen oder Dämme. Wasser tritt auf das Küstengebiet über. (Fluviale, pluviale und Küsten-Flut)<sup>18</sup>

aufnehmen kann, sammelt sich das Wasser auf der Oberfläche an und fließt in das nächste Tal ab. Pluviale Fluten können demnach auch in Regionen ohne Gewässer auftreten.

In manchen Regionen treten pluviale Überschwemmungen durch Hochwasser regelmäßig und im Einklang mit dem dortigen ökologischen Kreislauf auf. Zum Beispiel entstehen durch den El Niño Effekt Pazifischen Ozean Zyklonen.<sup>20</sup> Diese können unter entsprechenden Wetterbedingungen, wie Temperatur, Hoch-/Tiefdruckgebiete oder Kältefronten an den Küsten Australiens starken Regen auf dem Festland auslösen. Diese Gebiete werden aufgrund der geringen Wasseraufnahmekapazität des Bodens im Laufe eines Tages langsam überschwemmt.<sup>21</sup> Doch können diese Überschwemmungen in manchen Jahren stärker ausfallen, wodurch mehr Gebiete überschwemmt und somit Menschen stark beeinflussen werden.<sup>22</sup>

Unter einer Küstenflut wird eine Überschwemmung auf ein Küstengebiet verstanden. Küstenfluten entstehen entweder durch Sturmfluten, Tsunamis, durch ein durch die Gezeiten bedingtes Hochwasser oder durch die Kombination beider, wie ebenfalls in Abbildung 1 dargestellt. Bei Sturmfluten treffen mehrere große Wellen auf die Küste und der Meeresspiegel steigt stark an. Wenn die Wellen, die den Küstenschutz wie Dünen oder Deiche brechen lassen oder der Wasserstand den Küstenschutz übersteigt wird, das umliegende Küstengebiet überflutet. Eine weitere

<sup>20</sup>Vgl. auch im Folgenden Yeo 2002, S. 177

<sup>21</sup>Vgl. Bates u. a. 2021, S. 5 ff.

<sup>22</sup>Vgl. Czymzik u. a. 2013, S. 97f.

Entstehungsmöglichkeit von Küstenfluten sind Tsunamis. Auch hier ist die Ursache für einen Bruch des Küstenschutzes eine Welle. Diese ist meist jedoch deutlich stärker und größer als bei Sturmfluten und bringt somit mehr Zerstörungspotential mit sich. In flachen Küstengebieten kann sogar eine durch die Gezeiten auftretende Flut Teile des Landes überfluten. Durch den Klimawandel bedingt steigt auch der Meeresspiegel im Mittel immer weiter an.<sup>23</sup> In Zukunft werden demnach vor allem flache Küstengebiete von häufiger auftretenden Überschwemmungen betroffen.

### 2.1.3 Folgen und Auswirkungen von Überschwemmungen

Durch Verdunstung und unterirdischen Abfluss des Wassers wird das Land langsam wieder von einer Überschwemmung befreit. Dabei hinterlassen Überschwemmungen manchmal einen Abdruck, eine beobachtbare Signatur, in den Erdschichten zurück, welcher auch noch Millionen Jahre später zu erkennen ist. So konnten die Forschenden Jaramillo u. a. 2017 zwei kurzzeitige Überschwemmungen in einem Amazonas-Gebiet nachweisen.<sup>24</sup>

Durch Überschwemmungen entstehen starke wirtschaftliche und finanzielle Schäden durch den Verlust von Infrastruktur und Wohngebieten, so werden exemplarischen Fluten aus dem 20. Jahrhundert in Australien ein wirtschaftlicher Schaden von mehreren Millionen Dollar zugewiesen.<sup>25</sup> Doch der größere Schaden entsteht durch den Verlust von Menschenleben, durch Verletzungen und durch den Verlust der Heimat von den betroffenen Menschen. **Schäden durch Fluten sind somit subjektiv und können daher nur schwer quantitativ verglichen werden.**

## 2.2 Datenerfassung von Überschwemmungen durch Satelliten

Auch andere natürliche Ereignisse wie zum Beispiel Vulkanausbrüche, Erdbeben oder starke Hitzewellen können Menschen schaden. Um solche Ereignisse besser verstehen zu können beobachten Menschen die Erde, seit dem Zeitalter des Weltraumflugs auch aus dem Weltraum. Satelliten-daten bilden eine vielversprechende Grundlage zur Kartografierung der Erde. Durch sie ist eine Analyse der geologischen und atmosphärischen Veränderungen unseres Planeten möglich. Seit dem Start des ersten Satelliten Sputnik am 4. Oktober 1957 entwickelte sich die Messtechnik von Satelliten erheblich weiter.<sup>26</sup> Heute kreisen viele hoch entwickelte Erdbeobachtungs-Satelliten um unseren Planeten, so nehmen Satelliten zum Beispiel wie die Landsat Satelliten kontinuierlich seit den 70er Jahren Bilder von unserem Planeten auf.<sup>27</sup> Die Copernicus Mission der European Space Agency (ESA) hat sich das Ziel gesetzt mit einer Reihe an Satelliten unseren Planeten langfristig zu observieren.<sup>28</sup>

<sup>23</sup>Vgl. auch im Folgenden Bell u. a. 2017, S. 74

<sup>24</sup>Vgl. Jaramillo u. a. 2017, S. 1

<sup>25</sup>Vgl. Smith 1981, S. 34; Vgl. Yeo 2002, S. 181

<sup>26</sup>Vgl. Madry 2013, S. 660 ff.

<sup>27</sup>Vgl. Short 1982, S. 3

<sup>28</sup>Vgl. Thépaut u. a. 2018, S. 1591

Mit der Copernicus Mission stellt die ESA täglich aktualisierte geografische, atmosphärische und ozeanische Daten der Wissenschaft zur Verfügung.<sup>29</sup> So sollen bessere Aussagen und Vorhersagen in Bereichen wie Klima, Wetter, Katastrophenschutz und Sicherheit getroffen werden können. Die Mission wird mit 6 verschiedenen Satellitenmissionen sowie weiteren Observatorien und Messstationen auf der Erde durchgeführt.<sup>30</sup> Zu den Satellitenmissionen gehören die Satellitenprogramme Sentinel 1 bis 6. Jedes Satellitenprogramm besteht aus mehreren Satelliten, meist zwei, welche auf derselben Umlaufbahn zeitversetzt Daten sammeln. Die Sentinel-1 Satelliten nutzen aktive Sensoren, d. h. sie senden Radarwellen aus und messen, wie viel davon von der Erdoberfläche reflektiert wird. Dazu nutzen sie die Synthetic Aperture Radar (SAR) Technologie. Das von der Erdoberfläche reflektierte Sonnenlicht, also elektromagnetische Wellen aus dem sichtbaren und nahe-sichtbaren, optischen Spektrum, wird von den passiven Sensoren Sentinel-2 Satelliten aufgezeichnet. Für die Aufgabe der Wasser-Erkennung über Satelliten sind vor allem auch die Sentinel-6 Satelliten sehr interessant. Diese Satelliten besitzen wie die Sentinel-1 Satelliten aktive Sensoren und zeichnen die Oberflächentopografie der Meere und anderen Wasser-Stellen mittels Radar und Mikrowellen Radiometer auf. Zum Zeitpunkt dieser Arbeit sind die Daten der Sentinel-6 Satelliten im Gegensatz zu den anderen vorgestellten Sentinel Satellitenprogrammen noch nicht zur Nutzung verfügbar.

### 2.2.1 Radar basierte Ansätze

Radardaten wie die der Sentinel-1 Daten werden von Satelliten meist über ein SAR erfasst. Dabei werden polarisierte Radarwellen vom Satelliten in Richtung Erdoberfläche ausgestrahlt von welcher diese reflektiert werden.<sup>31</sup> Der Satellit empfängt mit einer relativ kurzen zeitlichen Verzögerung diese reflektierten Radarwellen. Da der Satellit in dieser Zeit ein Stückchen weiter gereist ist, entsteht eine Winkelverschiebung zwischen dem Senden des Signals und dem Empfangen des Signals. Anhand dieser Winkelverschiebung der auftreffenden Wellen sowie der Stärke und Frequenz dieser können über verschiedene Methoden die Topografie der Erdoberfläche berechnet werden.<sup>32</sup> Durch wiederum andere Methoden können SAR Daten auch für andere Anwendungen verwendet werden, so nutzt zum Beispiel die Erkennung von Wasserflächen die Reflexionen der VV, VH, HV und HH Polarisierungen.<sup>33</sup> Radarwellen interferieren nicht mit Wolken, weshalb auf SAR-Satellitenaufnahmen keine Wolken zu sehen sind und somit diese einen entscheidenden Vorteil gegenüber optischen Messungen haben, bei welchen große Gebiete verdeckt bleiben.

### 2.2.2 Optische Erfassung

Sowohl die Landsat Satelliten als auch Sentinel-2 messen im optischen Spektrum, also das von der Erde reflektierte Sonnenlicht sowie menschengemachte Lichter. Das optische Spektrum selbst

<sup>29</sup>Vgl. auch im Folgenden Thépaut u. a. 2018, S. 1591

<sup>30</sup>Vgl. auch im Folgenden Thépaut u. a. 2018, S. 1591 f.

<sup>31</sup>Vgl. auch im Folgenden BROWN 1967, S. 217 f.

<sup>32</sup>Vgl. Moreira u. a. 2013, S. 15

<sup>33</sup>Vgl. auch im Folgenden Bonaflia u. a. 2020, S. 837

kann unterteilt werden in einzelne Wellenlängen-Bereiche, sogenannte Bänder. Materialien reflektieren Licht verschiedener Wellenlängen unterschiedlich stark. So können Materialien mit der sogenannten *Spektroskopie* über das von ihnen reflektierte Licht auch aus sehr großer Entfernung klassifiziert werden.<sup>34</sup> Diese Methode kann auf unterschiedlichste Weise genutzt werden, zum Beispiel konnten Fretwell/Staniland/Forcada 2014 über das *Coastal* Band „aus dem All die Population einer Gruppe aus Walen in einem argentinischen Gewässer bestimmen“.<sup>35</sup>

<b>Band</b>	<b>Name</b>	<b>Wellenlänge (nm)<sup>36</sup></b>	<b>Auflösung (m)</b>	<b>Verwendung<sup>37</sup></b>
B1	Coastal	442	60	Aerosole
B2	Blue	492	10	Aerosole, Landnutzung, Vegetation
B3	Green	560	10	
B4	Red	665	10	
B5	RedEdge-1	704	20	
B6	RedEdge-2	740	20	
B7	RedEdge-3	780-783	20	
B8	NIR	833	10	
B8a	Narrow NIR	864	20	
B9	Water Vapor	943-945	60	
B10	Cirrus	1373-1377	60	
B11	SWIR-1	1610-1614	20	
B12	SWIR-2	2185-2202	20	

Tab. 1: Erfasste optische Bänder der Sentinel-2 Satelliten. Zu jedem Band: umgangssprachlicher Name, Wellenlänge, Auflösung und Nutzungs-Beispiele. (Übersicht der Sentinel-2 Bänder: Wellenlänge, Auflösung und Verwendung)<sup>38</sup>

Sentinel 2 erfasst das optische Spektrum über 13 Bänder, welche in Tabelle 1 aufgelistet sind. Durch gezielte Verrechnung und Kombination verschiedener Bänder können weitere Einsichten gewonnen werden. So wird vor allem in der Landwirtschaft der Normalized Difference Vegetation Index (NDVI) (Gleichung 2.1) genutzt, welcher aus dem Band des roten Spektrums sowie dem Band des Near Infrared (NIR) Spektrums berechnet wird.<sup>39</sup> Die Kombination beider Bänder ist

<sup>34</sup>Vgl. Schmidt 2014

<sup>35</sup>Fretwell/Staniland/Forcada 2014, S. 1

<sup>36</sup>Zentrale Wellenlänge, gerundet. Aufgrund der unterschiedlichen Konfiguration der Satelliten Sentinel-2A und B werden bei manchen Bändern zwei Werte angegeben.

<sup>37</sup>Ausgewählte Verwendungen, weitere sind natürlich möglich.

<sup>38</sup>Entnommen und erweitert aus: Online 2015

<sup>39</sup>Vgl. auch im Folgenden Pettorelli 2013, S. 19 ff.

für die Landwirtschaft besonders interessant, da mit ihr kranke Pflanzen erkannt werden können. Da gesunde, also besonders grüne, Pflanzen wenig rotes Licht und sehr viel Licht im NIR Bereich reflektieren, kranke Pflanzen jedoch nur wenig Licht des NIR Bereichs reflektieren, sind solche Einsichten möglich. Durch die Normierung der Bänder miteinander befinden sich alle Werte zwischen -1 und 1, dabei zeigen alle Werte zwischen 0 und 1 Vegetation auf und zwischen 0 und -1 Wasser.

$$NDVI = \frac{NIR - Red}{NIR + Red} \quad (2.1)$$

Analog zum NDVI funktioniert der Normalized Difference Water Index (NDWI), welcher das anstelle des roten Spektrums das grüne Spektrum mit dem NIR Spektrum verrechnet (Gleichung 2.2).<sup>40</sup> Dieser soll vor allem die Veränderung von Wasserstellen über längere Zeiträume hinweg sichtbar machen.

$$NDWI = \frac{Green - NIR}{Green + NIR} \quad (2.2)$$

Ein weiterer Index zur Hervorhebung von Wasser auf Satellitendaten ist der Automated Water Extraction Index (AWEI) von Feyisa u. a. 2014. Ziel des AWEI ist es den Kontrast zwischen Wasser und anderen dunklen Stellen zu vergrößern.<sup>41</sup> Die Forschenden stellten zwei Formeln für diesen Index auf, mit  $AWEI_{nsh}$  einen welcher Nicht-Wasser Pixel eliminiert und mit  $AWEI_{sh}$  einen welcher zusätzlich Schatten eliminiert:

$$AWEI_{nhs} = 4 \cdot (Green - SWIR_1) - (0,25 \cdot NIR + 2,75 \cdot SWIR_2) \quad (2.3)$$

$$AWEI_{sh} = Blue + 2,5 \cdot Green - 1,5 \cdot (NIR + SWIR_1) - 0,25 \cdot SWIR_2 \quad (2.4)$$

Durch die Falschfarben-Darstellung von einzelnen Bändern oder einer Kombination von genau drei beliebigen Bändern können qualitative Untersuchungen an den Satellitendaten durchgeführt werden. So sollen Wasserstellen für Menschen optisch sichtbar gemacht werden. Abbildung 2 zeigt drei Beispiele mit verschiedenen Kombinationen und Verrechnungen der Bänder. Die Kombination der Bänder NIR (8), Short Wavelength Infrared (SWIR)-1 (11) und Rot (4) wird verwendet, um den Unterschied zwischen *Wasser* und *Land* Fläche deutlich sichtbar zu machen und wird daher im Folgenden *Land/Wasser* Kombination genannt.<sup>42</sup>

---

<sup>40</sup>Vgl. McFeeters 1996, S. 1427

<sup>41</sup>Vgl. Feyisa u. a. 2014, S. 27

<sup>42</sup>Vgl. Bonafilia u. a. 2020, S. 838

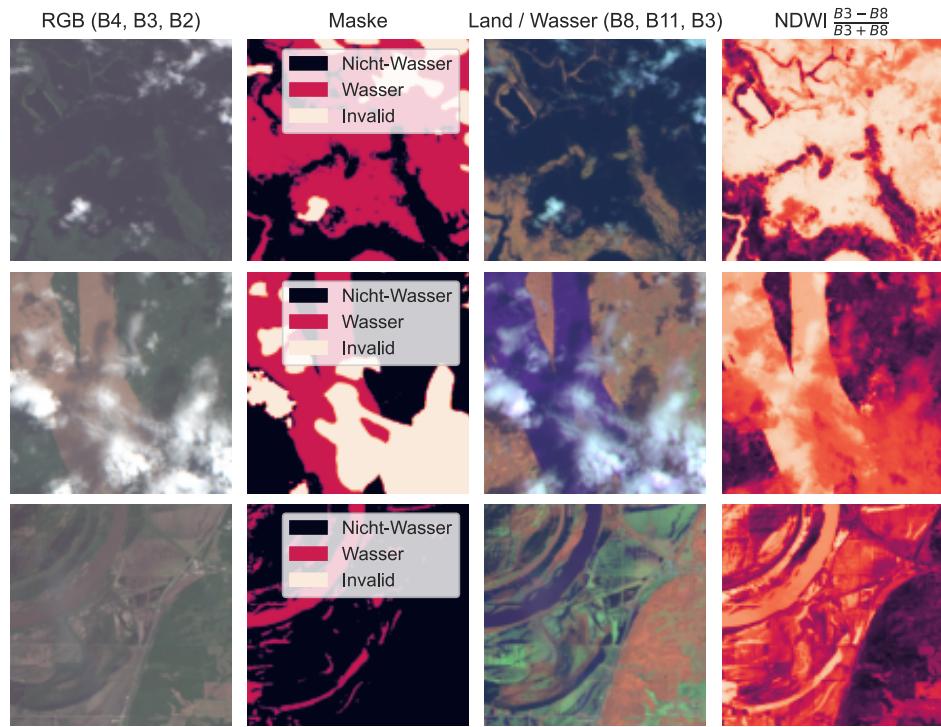


Fig. 2: Beispiele verschiedener Band-Kombinationen und Verrechnungen in Falschfarbendarstellung. Regionen von oben nach unten: Bolivien, Mekong, USA. Maske entnommen aus Bonafilia u. a. 2020. (Kombination verschiedener optischer Bänder)

Bonafilia u. a. 2020 nutzen zur Annotierung ihrer Daten neben der Land/Wasser Kombination eine weitere Falschfarben-Darstellung: SWIR-2 (12), NIR (8) und Rot (4).<sup>43</sup> Diese Kombination ist sehr ähnlich zur oberen Darstellung, aufgrund des Tausch der NIR und SWIR Bänder erscheint Wasser in dieser Darstellung jedoch Blau-Lila anstatt Blau-Grün.

### 2.2.3 Bestehende annotierte Datensätze zur Hochwasser-Erkennung

Dank Erdbeobachtungs-Satelliten wie jene des Copernicus Programms können für jeden Ort der Welt täglich verschiedene Daten abgerufen werden. Um Modelle zu evaluieren und mit maschinellem Lernen zu trainieren, benötigt es neben den durch zum Beispiel Copernicus zur Verfügung gestellten Daten auch Annotationen dieser. Abbildung 3 zeigt drei Beispiele solcher Annotationen.

Bonafilia u. a. 2020 stellten mit ihrer Publikation einen von Hand annotierten Datensatz zur Verfügung, welcher aus 11 Überschwemmung-Events besteht. Basierend auf diesem Datensatz

---

<sup>43</sup>Vgl. Bonafilia u. a. 2020, S. 840

generierten die Forschenden mit einem Modell automatisch einen größeren Datensatz mit mehr Annotationen, welche jedoch geringere Qualität aufweisen. Zur Annotation der Daten nutzten die Expert\*innen zwei Falschfarben-Darstellungen basierend der optischen Bänder der Sentinel-2 Satelliten ([R, G, B]: [B12, B8, B4] und [B8, B11, B4]) sowie das VH-Band der Sentinel-1 Satelliten.<sup>44</sup> Es wurden 446 sogenannter *Chips*, i.e. Bilder der Größe  $512 \times 512$  Pixel per Hand annotiert, worauf 4,385 *Chips* gleicher Größe automatisch annotiert wurden. Zu jedem Chip stellen sie neben den 2 SAR-Bändern *VV* und *VH* auch die 13 optischen Bänder der Sentinel 2 Satelliten zur Verfügung.

Ähnlich wurde der Datensatz *World Floods* von Mateo-Garcia u. a. 2021 entwickelt. Hierbei wurden 422 Karten automatisch durch einen automatischen Algorithmus annotiert und darauf von Menschen validiert.<sup>45</sup>

Pekel u. a. 2016 entwickelten einen großen Datensatz, um permanentes Wasser der Erde zu bestimmen und weiterhin Unterschiede in der Wassermenge zwischen 1983 und 2016 zu untersuchen. Dabei nutzten sie verschiedene Systeme, darunter ein Expert\*innen-System und ein von Expert\*innen entwickelter Entscheidungsbaum.<sup>46</sup> Es wurde für die gesamte Erde eine Karte mit permanentem Wasser angefertigt, diese wird in der Forschung auch als *European Joint Research Centre (JRC)-Permanent-Water* bezeichnet.<sup>47</sup>

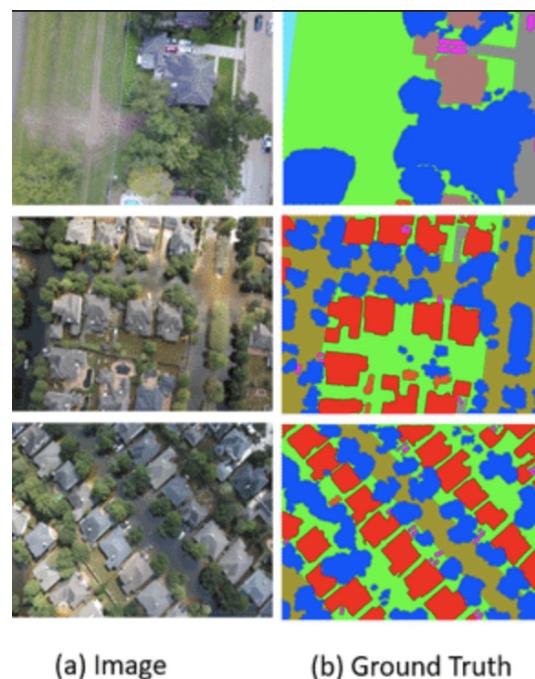


Fig. 3: Beispiel einer Multi-Klassen-Annotation von drei Drohnen-Aufnahmen. Links (a) Bild des sichtbaren Spektrums, rechts (b) Maske / Annotation. (Beispiel der Bild-Annotation von Rahnemoonfar u. a. 2020)<sup>48</sup>

<sup>44</sup>Vgl. auch im Folgenden Bonafilia u. a. 2020, S. 840

<sup>45</sup>Vgl. Mateo-Garcia u. a. 2021, S. 4

<sup>46</sup>Vgl. Pekel u. a. 2016, S. 424

<sup>47</sup>Vgl. Pekel u. a. 2016, S. 420; Vgl. Bonafilia u. a. 2020

<sup>48</sup>Entnommen aus: Rahnemoonfar u. a. 2020, S. 89651

Rahnemoonfar u. a. 2020 haben anstelle von Satellitendaten Drohnen-Aufnahmen für die Entwicklung ihrer Modelle genutzt und diese annotiert. Anstelle Flächen in *Wasser* und *Nicht-Wasser* Klassen zu unterteilen, klassifizierten sie direkt mehrere andere Objekte, wie zum Beispiel Häuser und Straßen.<sup>49</sup> Somit ist diese Form der Annotation hoch-auflösender, wie in Abbildung 3 verdeutlicht wird. Diese Form der Annotation ist jedoch auf sehr hochauflösende Bilder angewiesen, welche zurzeit noch nicht von Satelliten aufgenommen werden können.

Aus der Literatur somit geht hervor, dass es bereits unterschiedlichste Methoden zur Erkennung von Wasser auf Satellitendaten gibt und entsprechend bereits eine Reihe an Datensätzen mit annotierten Satellitenbildern existiert. **Wasser kann entweder über Radar, mit der Falschfarben-Darstellung verschiedener Kombinationen von optischen Bändern oder durch die Berechnung von Indexen von Experten auf Satellitendaten erkannt werden.**

## 2.3 Aufgabe und Metriken der Bild-Segmentierung zur Hochwasser-Erkennung

Das Ziel der Bild-Segmentierung ist es alle Pixel auf einem Bild einer Klasse zuzuweisen. Abbildung 4 zeigt eine solche Zuweisung von Klassen zu jedem Pixel des Bildes einer Situation im Straßenverkehr. Wenn lediglich eine Klasse auf Bildern erkannt, bzw. zwischen zwei Klassen unterschieden werden soll handelt es sich um eine binäre Bild-Segmentierung, welche jedoch den gleichen Mustern folgt wie eine Multi-Klassen-Bild-Segmentierung.



Fig. 4: Bild-Segmentierung einer gewöhnlichen Szene im Straßenverkehr. Oben Legende, Links RGB-Aufnahme einer Auto-Kamera, Mitte eine von Hand annotierte Maske, Rechts erkannte Maske eines Modells. (Bild-Segmentierung anhand eines Beispiels)<sup>50</sup>

Bei der automatischen Erkennung von Überschwemmungen auf Satellitendaten müssen die Pixel zwischen „Wasser“ und „Nicht-Wasser“ unterschieden werden. In Abbildung 2 sind Satellitenaufnahmen von Überschwemmungen zu erkennen, welche von Expert\*innen annotiert wurden. Diese

<sup>49</sup>Vgl. Rahnemoonfar u. a. 2020, S. 89646

<sup>50</sup>Entnommen aus: Jegou u. a. 2016, S. 1183

teilten jeden Pixel entweder der Klasse *Wasser* oder *Nicht-Wasser* ein. Weiterhin kennzeichneten sie *invalid* Pixel, wenn zum Beispiel Wolken das Bild verdecken. Diese Klassifizierung kann auch anders gestaltet werden, indem zum Beispiel bei Mateo-Garcia u. a. 2021 eine weitere Klasse „Wolken“ hinzugefügt und *Nicht-Wasser* schlicht als „Land“ gekennzeichnet wird.

Bei der Bewertung von Modellen zur Bild-Segmentierung kann je nach Anwendung eine Vielzahl an Metriken betrachtet werden. Die Bild-Segmentierung beschreibt ein mehrdimensionales Multi-Klassen-Problem: Jedem Pixel  $x$  eines zwei-dimensionalen Bildes soll eine Klasse  $c$  aus einer vordefinierten Menge an Klassen  $C$  zugewiesen werden, also  $F(x \in \{H \times W\}) - > c \in C$ . Für die Anwendung eines Modells können unterschiedliche Klassen verschieden priorisiert sein, heißt Fehler bei der Voraussage einer Klasse können schwerwiegender sein als bei einer anderen. Weiterhin stellt sich die Frage wie die Klassifizierung auf Pixel-Ebene bewertet werden soll: Es kann zum Beispiel wichtiger sein größere zusammenhängende Pixelgruppen zu erkennen als jeden Pixel an sich richtig zu klassifizieren, zum Beispiel wenn Bild-Rauschen unterdrückt werden soll. Dies gilt natürlich vice-versa. Ein weiteres Problem bei der gewissenhaften Beurteilung von Bild-Segmentierungs Algorithmen stellen unausgeglichene Datensätze dar, in denen eine oder mehrere Klassen über-, bzw. unterrepräsentiert sind. Daher gibt es eine Vielzahl an möglicher Metriken um Bild-Segmentierungs Algorithmen zu bewerten.

Eine Studie von Harouni/Baghmaleki 2020 ordnet verschiedene Metriken und Bewertungsansätze ein und gibt einen Überblick über mehr als 35 verschiedene Metriken. Eine Evaluierung sei in zwei Kategorien zu unterscheiden: subjektive und objektive Evaluierung.<sup>51</sup> Bei der subjektiven Evaluierung werden Personen oder Fachleute über den „Erfolg“ des Modells befragt, indem Beispiele gezeigt und diese von den Befragten beurteilt werden. Da dies, wie der Name bereits zum Ausdruck gibt, von den subjektiven Wahrnehmungen der Personen abhängig ist, handelt es sich um eine qualitative Analyse. Die objektive Evaluierung misst über „interne Beobachtungen“ und einer quantitativen Metrik den Erfolg eines Modells.<sup>52</sup> Diese können weiter unterschieden werden in statistische und Distanz-orientierte Metriken.<sup>53</sup> Im Folgenden werden ausgewählte statistische, quantitative Metriken näher beschrieben und erklärt.

Viele Arbeiten zur Bild-Segmentierung vergleichen ihre Methoden mit dem IoU, auch Jaccard-Index (JI) genannt.<sup>55</sup> IoU beschreibt die Überlappung der Zielpixel  $Y$  mit den vorausgesagten Pixel  $\hat{X}$ , Bild 5 zeigt dies exemplarisch. Aus der Mittlung des IoU aller Klassen  $C$  ergibt sich der Mean Intersection over Union (mIoU), welcher bei Multi-Klassen-Aufgaben als Zusammenfassung aller IoU dient und somit zu einer einzelnen Metrik reduziert.

---

<sup>51</sup>Vgl. Harouni/Baghmaleki 2020, S. 99 f.

<sup>52</sup>Harouni/Baghmaleki 2020, S. 99

<sup>53</sup>Vgl. Harouni/Baghmaleki 2020, S. 100

<sup>54</sup>Entnommen aus: Jordan 2018

<sup>55</sup>Siehe Bonafilia u. a. 2020; Zhang, H. u. a. 2018; Zhang, Z./Liu, Q./Wang 2018; Long/Shelhamer/Darrell 2015; Ronneberger/Fischer/Brox 2015

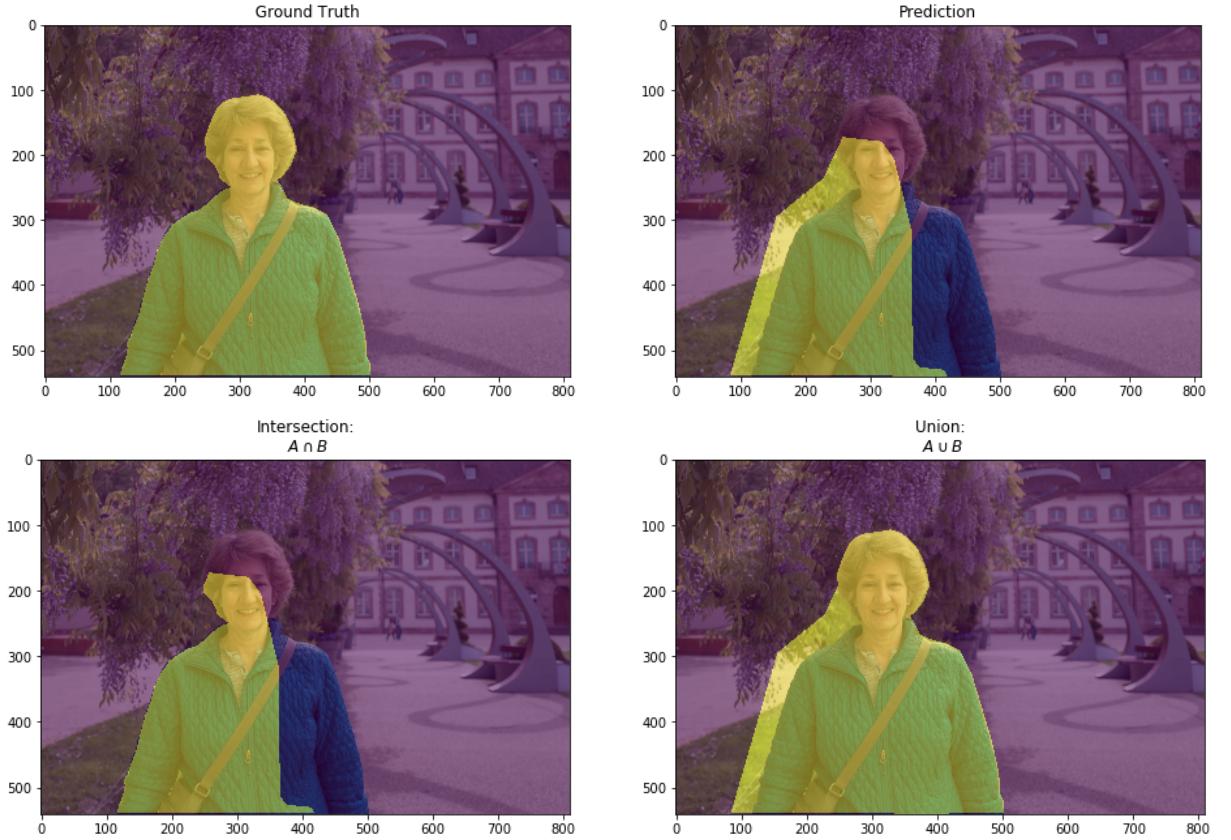


Fig. 5: Funktionsweise und optische Herleitung des IoU einer Klasse bei der Bild-Segmentierung. IoU ist schlussendlich das Verhältnis zwischen Schnittmenge (Intersection) und der Vereinigung (Union). (Visualisierung der IoU Metrik)<sup>54</sup>

$$IoU = \frac{Y \cap \hat{X}}{Y \cup \hat{X}}$$

$$mIoU = \frac{1}{|C|} \sum_{c \in C} IoU_c$$

Der mIoU beträgt bei Zufalls-Modellen 50%, versteckt bei einem unausgeglichenen Datensatz jedoch mögliche Bias einer Klasse. Dabei kann IoU Werte zwischen 0 und 1 annehmen. 0 bedeutet es gibt keine Übereinstimmung zwischen den vorausgesagten Pixeln und den Zielpixeln, 1 bedeutet eine perfekte Übereinstimmung. Da bei der Erkennung von Fluten auf Satellitenbildern meist mehr Land als Wasser zu erkennen ist, es also einen verhältnismäßig kleinen Anteil an *Wasser* Pixeln gibt wird hier meist nicht der gemittelte mIoU sondern lediglich der IoU der Klasse *Wasser* als Vergleichsmetrik herangezogen. So wird der Bias des unausgeglichenen Klassenverhältnisses berücksichtigt. Eine zum IoU ähnlich aufgebaute Metrik ist der Sørensen–Dice coefficient, eher bekannt als Dice-Coefficient, welche jedoch weniger Anwendung findet.

$$DSC = \frac{2|\hat{X} \cap Y|}{|\hat{X}| + |Y|}$$

Über die Konfusionsmatrix lassen sich eine Vielzahl an Metriken ableiten, die wichtigsten davon die Genauigkeit (Accuracy (ACC)), die Präzision (Positive Predicted Value (PPV)), die Sensitivität (True Positive Rate (TPR)) und die Spezifität (True Negative Rate (TNR)).<sup>56</sup>

$$\begin{aligned} ACC &= \frac{TP + TN}{TP + TN + FP + FN} \\ PPV &= \frac{TP}{TP + FP} \\ TPR &= \frac{TP}{TP + FN} \\ TNR &= \frac{TN}{TN + FP} \end{aligned}$$

Da jede dieser Metriken eine binäre Reduzierung der Konfusionsmatrix für jede Klasse benötigt, kann bei einem Multi-Klassen-Problem jede dieser Metriken entweder für jede Klasse einzeln, durch eine Makro- oder eine Mikro-Berechnung berechnet werden. Bei der Makro-Berechnung wird für jede Klasse einzeln die entsprechende Metrik berechnet um darauf diese Ergebnisse zu summieren und zu mitteln. Die Mikro-Berechnung verzichtet auf diesen Zwischenschritt und rechnet stattdessen alle Komponenten der Metrik vor der Berechnung der Metrik zusammen. Dadurch ergeben sich vor allem bei unausgeglichenen Datensätzen erhebliche Unterschiede: Ein Modell kann eine Vielzahl an Beispielen der überrepräsentierten Klasse falsch und die restlichen Beispiele richtig klassifizieren, wodurch sich eine große Makro-Genauigkeit und dafür aber eine geringe Mikro-Genauigkeit ergibt.

Die Mikro-Genauigkeit wird auch Overall Accuracy (OA) oder *Pixel-Accuracy* genannt und zeigt strikt auf wie viele Pixel richtig klassifiziert wurden. Aus der OA kann der Pixel-Fehler abgeleitet werden, welcher simpel mit  $PixelError = 1 - OA$  berechnet werden kann und aufzeigt wie viele Pixel falsch klassifiziert wurden. Aufgrund der Aufsummierung der Komponenten bei der Mikro-Berechnung gilt:  $ACC_{micro} = TPR_{micro} = PPV_{micro}$ .

Über die Makro-Präzision und die Makro-Sensitivität lassen sich Aussagen über den Erfolg des Modells bezüglich richtiger Vorhersagen der positiven Klasse jeder einzelnen Klasse treffen. Diese fokussieren sich auf den positiven Teil der Vorhersagen anstelle des Negativen. Anders ist dies bei der Spezifität, welche sich auf den negativen Anteil beziehen. Die Makro-Präzision zeigt mit welcher Wahrscheinlichkeit eine vom Modell erkannte Klasse eines Pixels tatsächlich richtig erkannt wurde. Wie viele Pixel klassenweise richtig erkannt wurden zeigt hingegen die Makro-Sensitivität auf. Ähnlich wie bei dem IoU kann auch die Sensitivität und die Präzision klassenweise angegeben werden.

---

<sup>56</sup>Vgl. Harouni/Baghmaleki 2020, S. 101

Neben statistischen Metriken zur Bestimmung der Effektivität eines Modells können Aufwandbezogene Metriken zur Ermittlung der Effizienz eines Modells genutzt werden. Bei maschinell lernfähigen Modelle gehören dazu zum Beispiel die Anzahl an trainierbaren Parametern, benötigte Trainingszeit, benötigte Trainingsbeispiele, benötigte Multiply–Accumulate Operation (MAC)s oder Floating Point Operation (FLOP)s pro einzelner Vorhersage und benötigter Speicheraufwand des Modells. So konzentrierten sich beispielsweise die Arbeiten von Krizhevsky/Sutskever/Hinton, G. E. 2012 Simonyan/Zisserman 2014 und He u. a. 2015 auf die Reduzierung des Rechenaufwands des Trainings solcher Modelle.

Die Anzahl der trainierbaren Parameter stellt meistens auch die Komplexität eines Modells dar, welche bei heutigen Modellen mit mehr als 1 Milliarde trainierbaren Parametern sehr groß sein kann.<sup>57</sup> Ein geringer Speicheraufwand geht meistens mit einer geringen Anzahl an Parametern einher. Der Speicheraufwand wirkt sich auf den benötigten Arbeitsspeicher aus, weshalb neben der Anzahl an Parametern auch der benötigte Arbeitsspeicher ermittelt wird. Die benötigte Trainingszeit und die benötigten Trainingsbeispiele sind eng miteinander verknüpft: Bei wenigen Beispielen muss das Modell während des Trainings sich diese häufiger anschauen. Eine geringe Trainingszeit hat positive Auswirkungen auf die Umwelt, da weniger Strom verbraucht wird. Weiterhin ist es möglich mehr Experimente durchzuführen, wodurch die Wissenschaft noch besser nachvollziehen kann wie ein Modell funktioniert. In der Praxis hingegen finden sich häufig nur wenige Trainingsbeispiele anhand dieser trainiert werden können, daher sind mit wenig Daten auskommende Modelle vorzuziehen. Der benötigte Rechenaufwand eines Modells zur Vorhersage eines Beispiels wird in FLOPs oder MACs gemessen. Diese beschreiben die benötigten Rechenoperationen bis das Modell zu einem Ergebnis kommt. Ein FLOP beschreibt eine einzige Berechnung, ein MAC beschreibt eine zusammengefasste Multiplikation und Addition, also  $ax + b$ . Daher können MACs grob in FLOPs umgerechnet werden:  $MAC = 0,5 \cdot FLOP$ . Durch eine Parallelisierung der Rechenschritte, zum Beispiel in einer GPU, kann die Trainingszeit drastisch reduziert werden.

**Die Aufgabe Bild-Segmentierung beschreibt im Kontext der Hochwasser-Erkennung einen Weg automatisch Pixel der Satellitenbilder in Klassen zuzuordnen um dadurch Wasser auf diesen zu markieren.** Dabei kann der Erfolg einer solchen Erkennung über verschiedene Metriken wie zum Beispiel dem IoU ermittelt werden.

## 2.4 Statistische Modelle zur Erkennung von Überschwemmungen auf Satelliten

Wasser ist auf Satellitenbildern über diverse Kombinationen an Bändern, wie zum Beispiel dem NDWI, leicht von Landmassen zu unterscheiden.<sup>58</sup> Der NDWI nutzt den Unterschied in der Absorption von Licht durch Gewässer zwischen dem grünen und dem nahen infraroten Teil des

---

<sup>57</sup>Vgl. Otter/Medina/Kalita 2021, S. 604

<sup>58</sup>Vgl. Mateo-Garcia u. a. 2021, S. 4

elektromagnetischen Spektrums, im Falle der Sentinel 2 Daten also  $\frac{B8-B4}{B8+B4}$ . Bei Überschwemmungen vermischt sich das Wasser jedoch mit diversen anderen Materialien wie zum Beispiel Erde, Gestein, Biomasse, Trümmer und weitere, wodurch die Unterscheidung von Wasser und Land erschwert wird.<sup>59</sup> Weiterhin beeinflussen Wolken und ihre Schatten die Satellitenaufnahmen beträchtlich.

Durch verschiedene Methoden können diese Probleme zu einer gewissen Effektivität gelöst werden. Im Folgenden werden verschiedene statistische Ansätze, welche dieses Problem lösen, erläutert und diskutiert.

#### 2.4.1 Ansätze basierend auf Grenzwerten

Ein schon angesprochener Ansatz *Wasser* von *Nicht-Wasser* zu unterscheiden ist die Nutzung des NDWI.<sup>60</sup> Die Einordnung in die Klassen *Wasser* und *Nicht-Wasser* geschieht über einen Grenzwert, Bonafilia u. a. 2020 nutzte zum Beispiel den Wert 0,3.<sup>61</sup> Wenn der NDWI des Pixels diesen Grenzwert überschreitet zählt der Pixel zur Klasse *Wasser*, sonst zur Klasse *Nicht-Wasser*. Durch einen Abgleich mit einer Karte von permanentem Wasser wie zum Beispiel JRC-Permanent-Water von Pekel u. a. 2016 können so überschwemmte Gebiete erkannt werden.

Da sowohl elektromagnetische Wellen im NIR Bereich als auch im Grünen Bereich von Wolken absorbiert werden, verhindern Wolken die Einordnung in die richtige Klasse. Da der optische Sensor eines Satelliten bei einer Wolke nur wenig Strahlung misst, fällt der NDWI an der entsprechenden Stelle gering aus. Somit kann es vorkommen, dass Wasserflächen irrtümlich als *Nicht-Wasser* markiert werden. Die Autor\*innen McFeeters 1996 welche den NDWI eingeführt haben sahen den Nutzen primär in der Abschätzung und qualitativen Analyse von Wasserflächen.<sup>62</sup> Sie führten entsprechend keine quantitative Analyse durch, weshalb es keine konkreten Metriken zu dieser Methode gibt.

Die Anwendung eines Grenzwertes kann auch über andere Daten und Indexe zur Trennung von *Wasser* und *Nicht-Wasser* genutzt werden. Neben dem NDWI werden so zum Beispiel auch SAR-Daten genutzt, da Wasser Radarwellen auf besondere Weise reflektiert.<sup>63</sup> Da Radarwellen nur sehr schwach von Wolken absorbiert werden und gleichzeitig von Land anders beeinflusst werden als von Wasser, können Land und Wasser sehr gut auf RadARBildern unterschieden werden.<sup>64</sup> Jedoch entstehen hierbei Ungenauigkeiten und falsche Resultate durch Rauschen und Störsignale.

Die Wahl des Grenzwerts kann entweder über den Stochastischer Gradienten Abstieg (SGD), über die Verteilung des Datensatzes, über eine Precision-Recall-Curve oder über die Fachkenntnis von Expert\*innen gewählt werden. Der SGD basiert auf dem Algorithmus von Robbins/Monro 1951,

<sup>59</sup>Vgl. auch im Folgenden Mateo-Garcia u. a. 2021, S. 4

<sup>60</sup>Vgl. auch im Folgenden McFeeters 1996

<sup>61</sup>Vgl. Bonafilia u. a. 2020, S. 839

<sup>62</sup>Vgl. McFeeters 1996, S. 1433

<sup>63</sup>Vgl. Moreira u. a. 2013, S. 15

<sup>64</sup>Vgl. Bonafilia u. a. 2020, S.839

<sup>65</sup>Entnommen aus: Donchyts u. a. 2016, S. 395

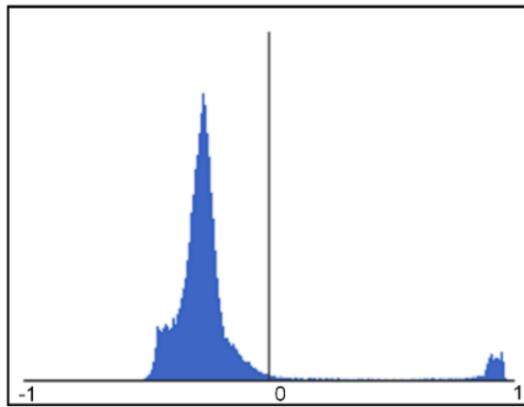


Fig. 6: Histogramm der Verteilung der NDWI Werte einer Satellitenaufnahme über Australien.  
(Bestimmung eines Grenzwert über Verteilung eines Datensatzes)<sup>65</sup>

bei welchem ein zufälliger Grenzwert initialisiert und statistisch schrittweise geändert wird. Nach jeder Änderung wird eine Metrik über den verwendeten Datensatz berechnet und der Grenzwert basierend auf dieser angepasst. Dies wird solange wiederholt bis ein Abbruchkriterium erreicht wird. Diese Methode ist sehr rechenaufwändig und kann bei unausgeglichenen Datensätzen zu einer Überanpassung an die überwiegende Klasse und somit falschen Ergebnissen führen.<sup>66</sup> Der optimale Grenzwert kann auch über die Werte-Verteilung des Training-Datensatzes gefunden werden, von welcher ein exemplarisches Beispiel in Abbildung 6 gezeigt ist. Dabei können Expert\*innen basierend auf der Verteilung der Werte über den Wertebereich für beide Klassen sowie ihren Erfahrungen einen Grenzwert bestimmen. Dies ist deutlich weniger rechenaufwändig, dafür jedoch subjektiv. Eine objektive Bestimmung des Grenzwertes kann auch über die *Otsu* Methode gefunden werden. Dabei sucht der verwendete Algorithmus nach dem Grenzwert, welcher die Intraklassenvarianz minimiert.<sup>67</sup> Bei zwei Klassen 0 und 1 ist diese definiert als gewichtete Summe der beiden Varianzen  $\sigma_{c \in 0,1}$ :

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t)$$

Die Precision-Recall-Curve vergleicht verschiedene Grenzwerte (Parameter) indem für jeden Grenzwert die Sensitivität der Genauigkeit gegenübergestellt wird, wie in Abbildung 7 gezeigt. Die Kurve lässt mehrere Möglichkeiten der Auswertung mit unterschiedlichen Zielen und Ergebnissen zu. Eine häufig verwendete Auswertungs-Methode ist die Betrachtung der Fläche unter der Kurve.<sup>68</sup> Mit dieser kann zum Beispiel die Qualität der Voraussagen eines Grenzwert-basierten, binären Klassifizierung-Algorithmus gemessen werden. Angenommen solch ein Algorithmus oder Modell gibt Werte zwischen 0 und 1 zurück und der Grenzwert bestimmt ab welchem Wert etwas einer Klasse zugeordnet wird. Wenn ein Algorithmus sich sehr sicher bezüglich seiner

<sup>66</sup>Vgl. Denevi u. a. 2019, S. 15667

<sup>67</sup>Vgl. Otsu 1979, S. 64 f.

<sup>68</sup>Vgl. Boyd/Eng/Page 2013, S. 452

Klassifizierung ist er lediglich Werte sehr nahe an 0 und 1 zurück, wodurch die Änderung des Grenzwertes wenig an der Sensitivität und der Genauigkeit ändert. Wenn weiterhin Sensitivität und Genauigkeit sehr gut sind, ist auch die Fläche unterhalb der Kurve sehr groß, was für einen sehr guten Algorithmus spricht. Eine weitere Möglichkeit ist die Betrachtung von Koinzidenz-Intervallen, welche aus der Kurve hergeleitet werden.<sup>69</sup> Dazu wird basierend auf der Aufgabenstellung eine Bedingung gestellt, zum Beispiel, dass Sensitivität des Algorithmus mindestens 95% betragen soll. Basierend darauf wird der Grenzwert gewählt, welcher die höchste Genauigkeit aufweist und gleichzeitig diese Bedingung erfüllt.

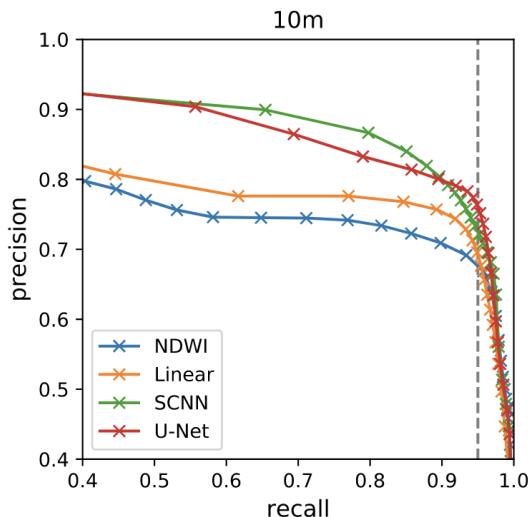


Fig. 7: Precision-Recall-Curve als Ergebnis mehrerer Modell-Architekturen mit verschiedenen Trainings-Parametern. Darunter auch ein Grenzwert-Modell mit dem NDWI (Blau). Die gestrichelte, graue Linie zeigt die von den Forschenden festgelegte Mindest-Sensitivität (Recall) auf, welche ein Modell aufweisen muss. (Beispiel einer Precision-Recall-Curve)<sup>70</sup>

#### 2.4.2 Entscheidungsbäume und Random-Forests

Eine erweiterte Form des Grenzwert-Ansatzes sind Entscheidungsbäume. Diese klassifizieren anhand mehrerer, sequentieller Entscheidungen basierend auf separaten Grenzwerten. Random-Forests sind eine erweiterte Form der Entscheidungsbäume, dabei handelt es sich um zufällig erzeugte, dafür aber gewichtete Entscheidungsbäume. Zur Bestimmung der Gewichte werden verschiedene Methoden des maschinellen Lernens eingesetzt.

Zhou, Z.-H./Feng 2017 entwickelten basierend auf Künstliche Nueronale Netze (KNN) und Random-Forests einen neuen Ansatz: Deep Forests. Deep Forests sind vielschichtige Modelle mit ähnlicher Komplexität wie Deep-Neural-Networks (DNN), jedoch im Gegensatz zu DNN nicht differenzierbar. Als Beispiel eines Deep Forests entwickelten die Forschenden die neue Modell-Architektur multi-Grained Cascade Forest (gcForest), mit welcher Bilder klassifiziert werden können. Wie in Abbildung 8 zu erkennen ist diese Architektur in ein „Multi-Grained Scanning“

<sup>69</sup>Vgl. auch im Folgenden Boyd/Eng/Page 2013, S. 456 f.

<sup>70</sup>Entnommen aus: Mateo-Garcia u. a. 2021, S. 8

und einen „Cascade Forest“ aufgeteilt. Das Multi-Grained Scanning verarbeitet alle Features eines Bildes und formt sie in eine lineare Sequenz um, inspiriert von CNN.<sup>71</sup> Dadurch wird auch der Kontext eines Pixels, also zum Beispiel seine Nachbarn, berücksichtigt.

Der Cascade Forest ist das Kernstück der vorgeschlagenen Architektur: hier werden über verschiedene Baum- und Wald-Arten die verschiedenen Features verarbeitet und ein Ergebnis gefunden.<sup>72</sup> Laut Zhou, Z.-H./Feng 2017 soll das Training der gcForest Architektur zu herkömmlichen DNN vergleichbar aufwendig bezüglich der Laufzeit und Rechenaufwands sein.<sup>73</sup> Weitere Optimierungen seien durch Parallelisierung und somit Ausführbarkeit auf einer GPU möglich.<sup>74</sup> Aufgrund der fehlenden Differenzierbarkeit des Modells können Optimierungen durch Parallelisierung von DNN nicht genutzt werden. Ebenso sind die *Wenn-Dann* Abfragen der Entscheidungsbäume sehr rechenaufwendig, da Computer besser Berechnungen als Vergleiche durchführen können.

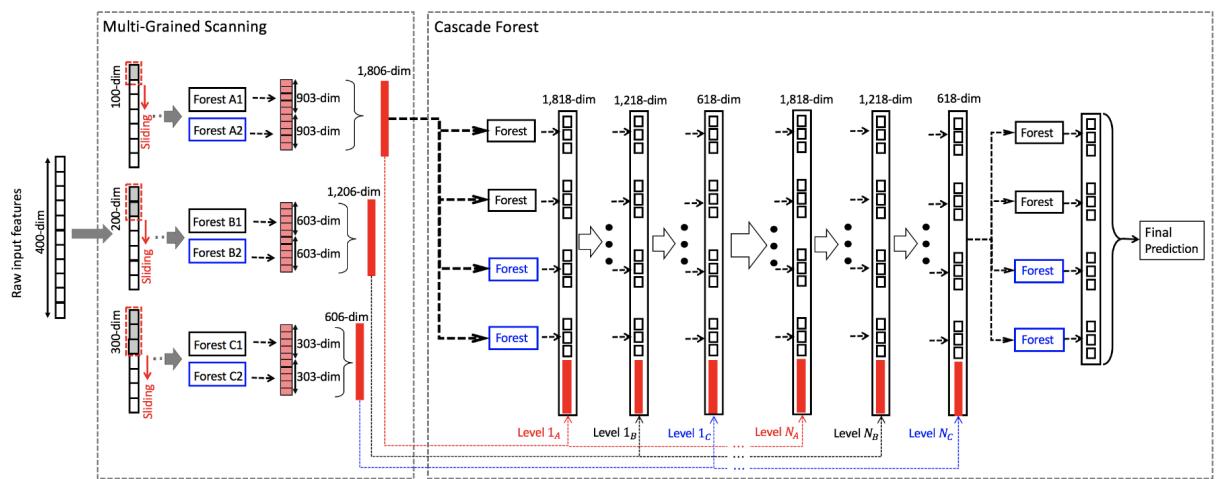


Fig. 8: Die Architektur besteht aus zwei Teilen: Multi-Grained Scanning und Cascade Forest. Im Multi-Grained Scanning werden Features extrahiert und im Cascade Forest diese verarbeitet und klassifiziert. (Deep Forest: gcForest Architektur)<sup>75</sup>

Deep Forests wurden im Laufe der Forschungen von Jamali u. a. 2021 genutzt um Wasser auf Satellitendaten zu klassifizieren. Dabei verwendeten die Forschenden die gcForest Architektur für ein Modell, welches auf einen Datensatz zur Gewässer-Erkennung in Kanada trainiert wurde und verglichen diesen mit einem herkömmlichen CNN sowie zwei weiteren auf Entscheidungsbäumen basierenden Modellen.<sup>76</sup> Dabei stellten sie eine signifikante Verbesserung der Deep Forest Methode gegenüber anderen Modellen fest.

<sup>71</sup>Vgl. Zhou, Z.-H./Feng 2017, S. 3564 f.

<sup>72</sup>Vgl. Zhou, Z.-H./Feng 2017, S. 3562

<sup>73</sup>Vgl. Zhou, Z.-H./Feng 2017, S. 3574 f.

<sup>74</sup>Vgl. Zhou, Z.-H./Feng 2017, S. 3575, 3582 f.

<sup>75</sup>Entnommen aus: Zhou, Z.-H./Feng 2017, S. 3565

<sup>76</sup>Vgl. auch im Folgenden Jamali u. a. 2021, S. 1088

### 2.4.3 Einsatz von Algorithmen der Cluster Erkennung

Neben der Klassifizierung von einzelnen Pixel in die Klassen Land und Wasser, teilt der Ansatz des Clustering ganze Pixelgruppen einer Klasse zu. Eine der häufig verwendete Methoden des Clustering sind Support Vektor Machines (SVM). SVM nutzen einen Stützvektor um basierend auf den Trainingsdaten einen Datensatz in zwei oder (mit Anpassung) mehrere Gruppen zu klassifizieren. Wie in Abbildung 9 zu sehen ist wird der Stützvektor  $W$  so gewählt, dass der Abstand (Margin) zwischen der Trennlinie und den beiden Gruppen maximiert wird.<sup>77</sup> Über bestimmte Methoden ist es möglich, trotz fehlerhafter Klassifizierungen innerhalb des Trainingsdatensatzes erfolgreich SVM zu trainieren, ebenso können über mathematische Transformationen auch mehrdimensionale Probleme mit SVM gelöst werden.<sup>78</sup>

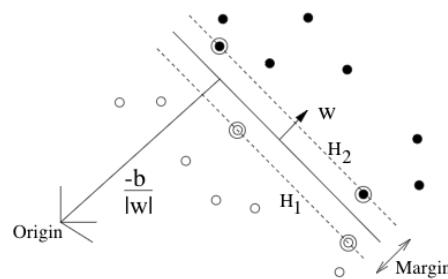


Fig. 9: Beispielhafter Aufbau einer SVM. Über den Stützvektor  $W$  entsteht eine Trennlinie (schwarz) welche zwei Gruppen voneinander trennt. (Support-Vector-Machine)<sup>79</sup>

Auch bei der Auswertung von Satellitendaten wurden in der Vergangenheit SVM genutzt um Pixel zwischen unterschiedlichen Klassen zu unterscheiden. Die Forschenden um Mountrakis/Im/Ogole 2011 haben hierzu über 120 verschiedene Arbeiten, welche Satelliten- und Luft-aufnahmen mit SVM analysierten, ausgewertet. Dabei stellten sie fest, dass viele Arbeiten mit SVM keine großen Trainings-Datensätze benötigten, um im Vergleich zu DNN bereits gute Ergebnisse zu erzielen.<sup>80</sup> Jedoch seien SVM sehr anfällig für verrauschte Daten, da Ausreißer sehr starken Einfluss auf das Training von SVM haben. Ebenso seien SVM wenig geeignet für mehr-dimensionale Aufgaben, da diese sehr viel Rechenaufwand benötigen. Beides ist bei Satellitenaufnahmen häufig der Fall, da atmosphärisches Rauschen nicht zu vermeiden und durch die hohe Anzahl an verfügbaren Features durch zum Beispiel verschiedene Wellenlängen Bänder eine hohe Anzahl verfügbaren Dimensionen einhergeht.

Die vorgestellten statistischen Modelle sind grundsätzlich dazu in der Lage zwischen verschiedenen Klassen zu unterscheiden. **So könnten statistischen Modelle wie Grenzwert-Modelle,**

---

<sup>77</sup>Vgl. Burges 1998, S. 128 f.

<sup>78</sup>Vgl. Burges 1998, S. 135 f., 138

<sup>79</sup>Entnommen aus: Burges 1998, S. 129

<sup>80</sup>Vgl. auch im Folgenden Mountrakis/Im/Ogole 2011, S. 249, 256

**Random Forest oder SVM dazu eingesetzt werden um Wasser Pixel auf Satelliten-Bildern zu erkennen.**

## **2.5 Vorstellung und historische Entwicklung neuronaler Modelle der Bild-Segmentierung**

Durch die immer größer zur Verfügung stehenden Rechenkapazitäten können auch kompliziertere und aufwendigere Deep-Learning Algorithmen genutzt werden. Daher nutzen neuere Ansätze DNN um die Pixel der Satellitenbilder in die Klassen *Wasser* und *Land* zu kategorisieren, wie zum Beispiel die Arbeiten von Bonafilia u. a. 2020, Mateo-Garcia u. a. 2021 oder Lopez u. a. 2020.

### **2.5.1 Funktionsweise von Encoder-Decoder Netzwerke**

Klassische neuronale Netze in Form von Multi-Layer Feed-Forward Neural Network (MLF) sind sehr aufwändig in der Berechnung wenn viele Features über mehrere Schichten im Netzwerk verarbeitet werden müssen. Auf Bildern der Größe  $8 \times 8$  Pixel können MLF mit heutigen Prozessoren problemlos z. B. Ziffern erkennen.<sup>81</sup> Auf größeren Bildern mit mehr Details des zu erkennenden Objektes kommen hingegen selbst die modernsten Grafikkarten an ihre Grenzen. Daher entwickelten LeCun u. a. 1999 CNN. Grundlage von CNN sind Filter, welche basierend auf den Bildkanälen neue Kanäle erzeugen, auf welchen Bild-Merkmale hervorgehoben werden. Der grundlegendste Filter ist ein Parameterfeld aus  $3 \times 3$  Parametern, hierbei ist auch von  $3 \times 3$  Faltung die Rede. Für alle Pixel, mit Ausnahme der Pixel am Bildrand, wird ein neuer Wert basierend auf der  $3 \times 3$  Parameter und den Pixelwerten des ursprünglichen Bildkanals berechnet. Da Pixel am Bildrand in eine, bzw. Pixel an den Bildecken zwei Richtungen keine Nachbarn haben, kann dort der Filter nicht angewendet werden, entsprechend verkleinert sich das resultierende Bild bei jeder Anwendung des Filters. Je nach Anpassung der Parameter des Filters können dabei unterschiedliche Eigenschaften und Merkmale der auf dem Bild abgebildeten Objekte extrahiert werden. Weiterhin ist auch die genaue Position dieser Objekt-Merkmale irrelevant, da die  $3 \times 3$  Faltung sich auf alle Pixel positionsunabhängig angewandt wird. Bei dieser Art der Faltung werden lediglich die Werte eines einzelnen Bildkanals benutzt, anders hingegen bei der sogenannten  $1 \times 1$  Faltung. Der  $1 \times 1$  Filter besitzt zusätzlich eine Tiefe basierend auf der Anzahl an Bildkanälen. So werden mehrere Bildkanäle zu einem Kanal zusammengefasst, dieser Prozess ähnelt sehr stark einem MLF bestehend aus zwei Schichten mit einem einzigen Knoten in der Ausgangs-Schicht.

In einem CNN werden verschiedene Filter-Schichten mit mehreren Pooling-Schichten aneinander gereiht. Ein Pooling summiert vier benachbarte Pixel zusammen, wodurch die resultierende Anzahl an Pixeln geringer ist. Somit wird Platz geschaffen um mehrere Filter auf das Eingangs-Bild anzuwenden. Dadurch wird mit jeder zunehmender Schicht die Anzahl an Kanälen mit

---

<sup>81</sup>Siehe Cun u. a. 1989

<sup>82</sup>Entnommen aus: Lecun/Bengio/Hinton, G. 2015, S. 438

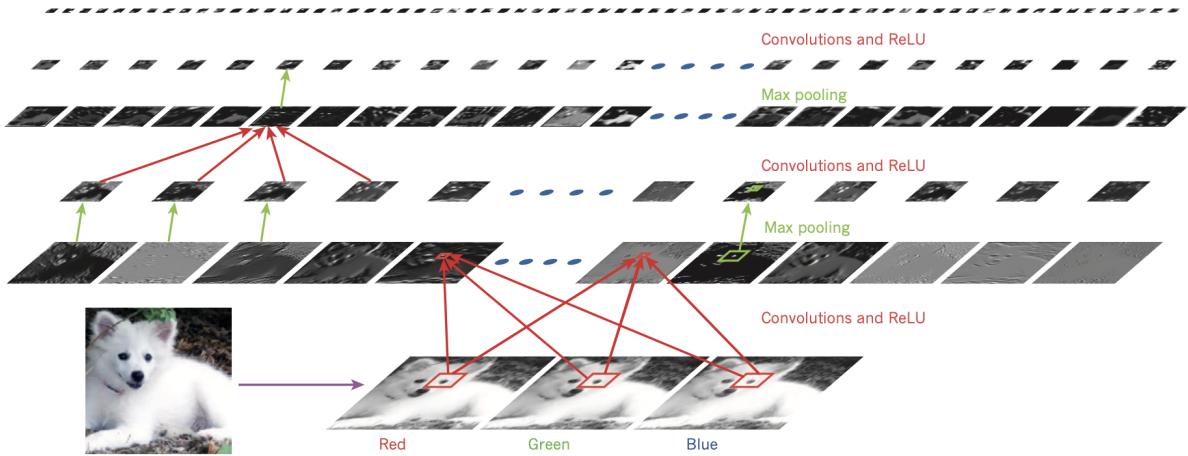


Fig. 10: Aus der ursprünglichen Abbildung eines Hundes werden über Faltungen einzelne Features extrahiert. Diese werden über Pooling-Schichten reduziert und darauf diese beiden Schritte so lange wiederholt, bis jedes Feature lediglich ein Skalar ist. Diese Skalare können dann über ein einschichtiges MLF zu einem Ergebnis reduziert werden. (Aufbau eines CNN)<sup>82</sup>

Objekt-Merkmalen größer, die Größe der Bilder kleiner. Dieser Abschnitt der CNN Architektur wird auch Encoder genannt. Das encodierte Bild, also die vielen Kanäle geringer Größer werden daraufhin mit einem kleinen MLF miteinander verrechnet, wodurch eine Klasse für das gesamte Bild bestimmt werden kann. Das MLF wird in dieser Architektur auch Decoder genannt. Das beschriebene Encoder-Decoder-Konzept ist Grundlage für viele weitere Architekturen, welche die Ideen des CNN aufnehmen, erweitern und optimieren.

### 2.5.2 Historische Entwicklung von Encoder Modellen

Die gewonnene Effizienz der Encoder-Decoder Architekturen wurde schnell dazu genutzt, größere und komplexere Modelle zu entwickeln. 2012 entwickelte Krizhevsky/Sutskever/Hinton, G. E. 2012 eine Architektur, welche mit einer GPU trainiert werden kann: AlexNet. AlexNet basiert auf der parallelen Berechnung der einzelnen Filter und der Knoten des MLF. Dadurch ist es möglich AlexNet sehr viel schneller als bisherige Modelle zu trainieren, wodurch wiederum die Architektur umso komplexer und das Training umso ausführlicher gestaltet werden kann.<sup>83</sup> Dies war ein wichtiger Schritt, da seitdem das Training von DNN ohne GPU nicht mehr vorstellbar ist. Ein weiterer Schritt bei der Optimierung von CNN Architekturen kam durch die Arbeit von Simonyan/Zisserman 2014. Diese entwickelten die Architektur Visual Geometry Group Network (VGG)-Net, welche auf dem AlexNet basiert und anstelle von  $5 \times 5$  und  $11 \times 11$  Filter ausschließlich  $3 \times 3$  Filter nutzt.<sup>84</sup> Da zwei  $3 \times 3$  Filter hintereinander das Bild auf die gleiche Größe wie  $5 \times 5$  Filter reduzieren und gleichzeitig die gleiche Anzahl an Merkmalen extrahieren können, jedoch anstelle von 25 erlernbaren Parametern lediglich 18 benötigen, ist eine Reihe an  $3 \times 3$  Fil-

<sup>83</sup>Vgl. Krizhevsky/Sutskever/Hinton, G. E. 2012

<sup>84</sup>Vgl. Simonyan/Zisserman 2014, S. 1

ter effizienter als größere Filter. Dies reduziert die Anzahl an trainierbaren Parametern für eine Filter-Operation, wodurch abermals tiefere, also Netze mit mehr Schichten bei gleichbleibendem Rechenaufwand möglich sind. Liu, Z. u. a. 2015 entwickelten das Deep Parsing Network (DPN), eine Encoder Architektur, welche das VGG erweitert und auf Markov Zufallsfelder anwendet. Dabei sollen diese Zufallsfelder die Anzahl an benötigten Epochen im Training reduzieren, indem Resultate der Backpropagation stochastisch angenähert werden.<sup>85</sup>

Durch Versuche mit immer tieferen Netzen entstand das Problem des verschwindenden Gradienten, bei welchem (zu) einfache Probleme nicht mehr vom Netzwerk erlernt werden können, da es selbst zu komplex ist.<sup>86</sup> Nehme man ein einfaches Problem, wie zum Beispiel die Multiplikation mit 2 und den Eingangssignalen. Ein sehr einfaches, „flaches“ neuronales Netzwerk bestehend aus einem Perzepron kann dies relativ simpel erlernen, da für die Lösung des Problems lediglich ein Gewicht nötig ist. Ein DNN hingegen benötigt deutlich länger um auf dieses Problem trainiert zu werden und ist zusätzlich sehr anfällig für Störungen und ungesehene Eingaben, da nun mehr Gewichte miteinander verrechnet werden. Um das Problem des verschwindenden Gradienten zu lösen entwickelten die Forschenden He u. a. 2015 eine Architektur basierend auf Residuen, daher wird diese auch ResNet genannt. ResNet führt sogenannte „Skip-Connections“ ein, wodurch manche Schichten, Residuen, optional übersprungen werden können.<sup>87</sup> Eine Weiterentwicklung zu ResNet stellt das von Xie u. a. 2017 entwickelte *ResNext* dar, welche ResNet um die Dimension der Kardinalität erweitert. Dazu werden die Filter-Blöcke in den Residuen des Netzwerks in viele kleinere Teile geteilt, wodurch jeder Block nun aus vielen kleinen Filtern anstelle eines großen besteht.<sup>88</sup> Huang u. a. 2016 entwickelten ResNets ebenfalls weiter, wodurch die Netzwerkarchitektur DenseNet entstand. DenseNet nutzt nicht wie ResNet mehrere MLF am Ende jeder Schicht, sondern ein großes, Schichten-übergreifendes MLF. Dadurch sollen in höheren Schichten erkannte Merkmale auch in tieferen genutzt werden können, wodurch sich die Erkennung von Merkmalen insgesamt verbessert. Da die verschiedenen Architekturen unterschiedlich viele Schichten (Tiefe des Netzwerks), welche wiederum unterschiedlich groß sind (Breite des Netzwerks), aufweisen, suchten Tan/Le 2019 nach optimalen Kombinationen aus Tiefe, Breite und Bildgröße für solche Netzwerke. Dadurch entstanden die EfficientNet Encoder Netzwerke, dessen Größe mittels drei Koeffizienten festgelegt werden kann.<sup>89</sup> Dies ermöglicht eine optimale Skalierung des Encoder-Netzwerks.

Vaswani u. a. 2017 führten Transformer Modelle im Bereich des Natural Language Processing (NLP) ein. Diese Modelle basieren auf einem Mechanismus namens „Self-Attention“, bei welchem ein Modell den Wörtern in einem Satz eine Gewichtung basierend ihrer Wichtigkeit, bzw. Aussagefähigkeit zuweist. Weiterhin sind die Modelle an die eindimensionale Struktur von Sätzen angepasst. Forschende fanden heraus, dass Menschen lediglich einen geringen Teil der eintretenden visuellen Informationen verarbeiten, sie besitzen also vermutlich einen ähnlichen

---

<sup>85</sup>Vgl. Liu, Z. u. a. 2015, S. 1

<sup>86</sup>Vgl. He u. a. 2015, S. 770 f.

<sup>87</sup>Vgl. auch im Folgenden He u. a. 2015, S. 773

<sup>88</sup>Vgl. Xie u. a. 2017, S. 5990

<sup>89</sup>Vgl. Tan/Le 2019, S. 10695 f.

Attention-Mechanismus wie der von Transformern.<sup>90</sup> Naheliegend ist daher eine Implementierung des Mechanismus für Bilder. Aufgrund der zweidimensionalen Natur von Bildern, bei Bildern mit verschiedenen Kanälen sogar dreidimensionalen Natur, ist dieser Mechanismus bei Bildern umständlicher. Erste Versuche, den Attention-Mechanismus auf Bilder zu übertragen wurden in Form von Transformern von Kolesnikov u. a. 2021 durchgeführt. Sogenannte Visual Transformer (ViT) reduzieren die zweidimensionalen Bilder in eindimensionale Sequenzen welche und speisen diese in einen Transformer.<sup>91</sup> Eine alternative Methode um Self-Attention auch auf Bilder anzuwenden entwickelten Guo u. a. 2022: Large Kernel Attention (LKA). LKA ist eine Kombination aus drei verschiedenen Faltungen, welche in Abbildung 11 beschrieben ist. Diese Kombination besteht aus: Der tiefenweisen Faltung, z. B. einer  $3 \times 3$ , der ausgedehnten, tiefenweisen Faltung, bei welcher Abstände zwischen den faltenden Pixeln existieren und der 1-Pixel Faltung, welche verschiedene Kanäle miteinander verbindet.<sup>92</sup> Dadurch wird bei jeder Anwendung eines Filters nicht nur ein kleiner Teil des Bildes berücksichtigt, sondern direkt das gesamte Bild. Die Filter trainieren somit eine Erkennung, welche Teile des Bildes wichtig sind und welche nicht.

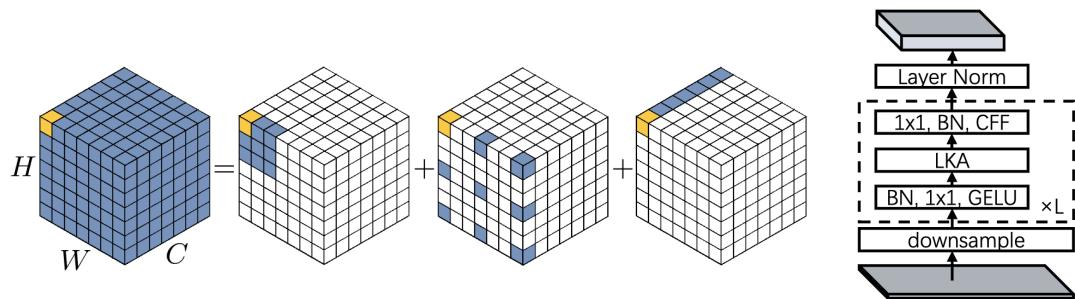


Fig. 11: Links: Aufbau eines LKA. Rechts: Architektur einer Stufe des VAN. (VAN Architektur)<sup>93</sup>

Basierend auf dem LKA entwickelten die Forschenden eine neue Basis-Architektur namens VAN, welche ebenfalls in Abbildung 11 beschrieben ist. Diese besitzt einen stufenweisen Aufbau, bei welcher jede Stufe eine geringere Auflösung, jedoch mehr Feature-Kanäle aufweist, ähnlich zu bestehenden Architekturen wie dem ResNet.<sup>94</sup> Für jede Stufe wird zunächst das Eingabebild herunterskaliert, für alle weiteren Schritte einer Stufe bleibt darauf die Größe des Bildes gleich. Nach der Anwendung einer Normalisierung, einem  $1 \times 1$  Filter und der GELU-Aktivierungsfunktion wird der LKA angewendet. Darauf werden die Merkmale mit einem gewöhnlichem CNN mit anschließendem MLF extrahiert. Schließlich wird am Ende jeder Stufe eine Normalisierung durchgeführt. Laut Angaben der Autoren schnitt das VAN deutlich besser als bisherige State-of-the-Art Modelle wie zum Beispiel das ResNet ab.<sup>95</sup>

<sup>90</sup>Vgl. Guo u. a. 2022, S. 1 f.

<sup>91</sup>Vgl. Kolesnikov u. a. 2021, S. 3 ff.

<sup>92</sup>Vgl. Guo u. a. 2022, S. 3

<sup>93</sup>Entnommen aus: Guo u. a. 2022, S. 4 f.

<sup>94</sup>Vgl. auch im Folgenden Guo u. a. 2022, S. 6 f.

<sup>95</sup>Vgl. Guo u. a. 2022, S. 12 ff.

### 2.5.3 Decoder-Architekturen in der Bild-Segmentierung

Das DNN-Decoder-Netzwerk des CNN kann lediglich eine Klasse für ein ganzes Bild bestimmen, jedoch nicht für jeden Pixel eine Klasse bestimmen. Dadurch ist es z. B. nicht möglich, mehrere Objekte gleichzeitig in einem Bild zu erkennen und den entsprechenden Pixeln zuzuordnen. Daher eignen sich CNN aufgrund ihrer architektonischen Struktur nicht um auf Satellitendaten Wasser zu erkennen. Die Forschenden Long/Shelhamer/Darrell 2015 entwickelten die Decoder-Architektur FCN basierend auf CNN welche auch Klassifikation auf Pixel-Ebene erlaubt, wie in Abbildung 12 zu erkennen ist.<sup>96</sup> Die kleinen, encodierten Kanäle des Bildes werden im Decoder des FCN über Upsampling-Methoden wieder auf die Größe des ursprünglichen Bildes zusammengefasst und vergrößert.

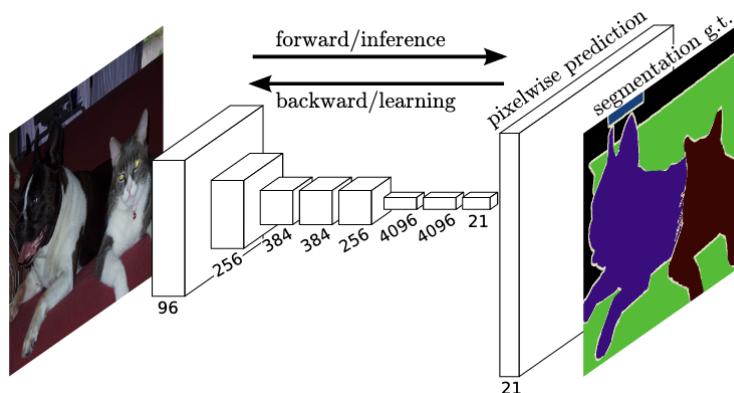


Fig. 12: Durch eine Hochskalierung der Features aus dem Encoder kann ein FCN jedem Pixel eines Bildes eine Klasse zuweisen. (FCN)<sup>97</sup>

2015 entwickelte das Forschungsteam Ronneberger/Fischer/Brox 2015 ein sehr erfolgreiches Modell basierend auf FCN: U-Net. Dieses Modell gewann mehrere Wettbewerbe und kommt seither in verschiedenen Bereichen zum Einsatz, zum Beispiel bei der Erkennung von Straßen auf Satellitenbildern.<sup>98</sup> Die vorangegangenen Architekturen basierend auf CNN und FCN waren aufgrund ihrer linearen Architektur vergleichsweise ineffektiv.<sup>99</sup> Es waren viele Trainingsbeispiele notwendig, um diese Architekturen zu trainieren, weiterhin war aufgrund der stetig steigenden Feature-Anzahl beider Architekturen viel Rechenkapazität nötig. Die Forschenden konnten mit dieser neuen Architektur sehr gute Ergebnisse mit sehr wenig Trainingsbeispielen und wenig Rechenkapazität erlangen. Dies konnte durch die Übertragung von Zwischenresultaten des Encoders auf Operationen im Decoder erreicht werden, welche in Abbildung 13 schematisch dargestellt ist. So können im Upsampling-Prozess, dem Decoder, höher-auflösende Informationen genutzt werden, um die kleinen, vielschichtigen Kanäle höher aufzulösen. Durch Augmentierung konnten die Forschenden die Anzahl an Trainingsbeispielen künstlich erhöhen, wodurch bessere Ergebnisse erzielt werden konnten. Zhou, Z. u. a. 2018 entwickelten eine verbesserte Methode des U-Net:

<sup>96</sup>Vgl. auch im Folgenden Long/Shelhamer/Darrell 2015, S. 2

<sup>97</sup>Entnommen aus: Long/Shelhamer/Darrell 2015, S. 1

<sup>98</sup>Vgl. Ronneberger/Fischer/Brox 2015, S. 237 f. Vgl. Zhang, Z./Liu, Q./Wang 2018

<sup>99</sup>Vgl. auch im Folgenden: Ronneberger/Fischer/Brox 2015, S. 236 ff.

U-Net++. Dieses nutzt Techniken des DenseNet von Huang u. a. 2016 um die Übertragung der Zwischenresultate des Encoders auf den Decoder noch effektiver zu gestalten, indem auch hier Filter und MLF die Signale abermals verarbeiten.<sup>100</sup>

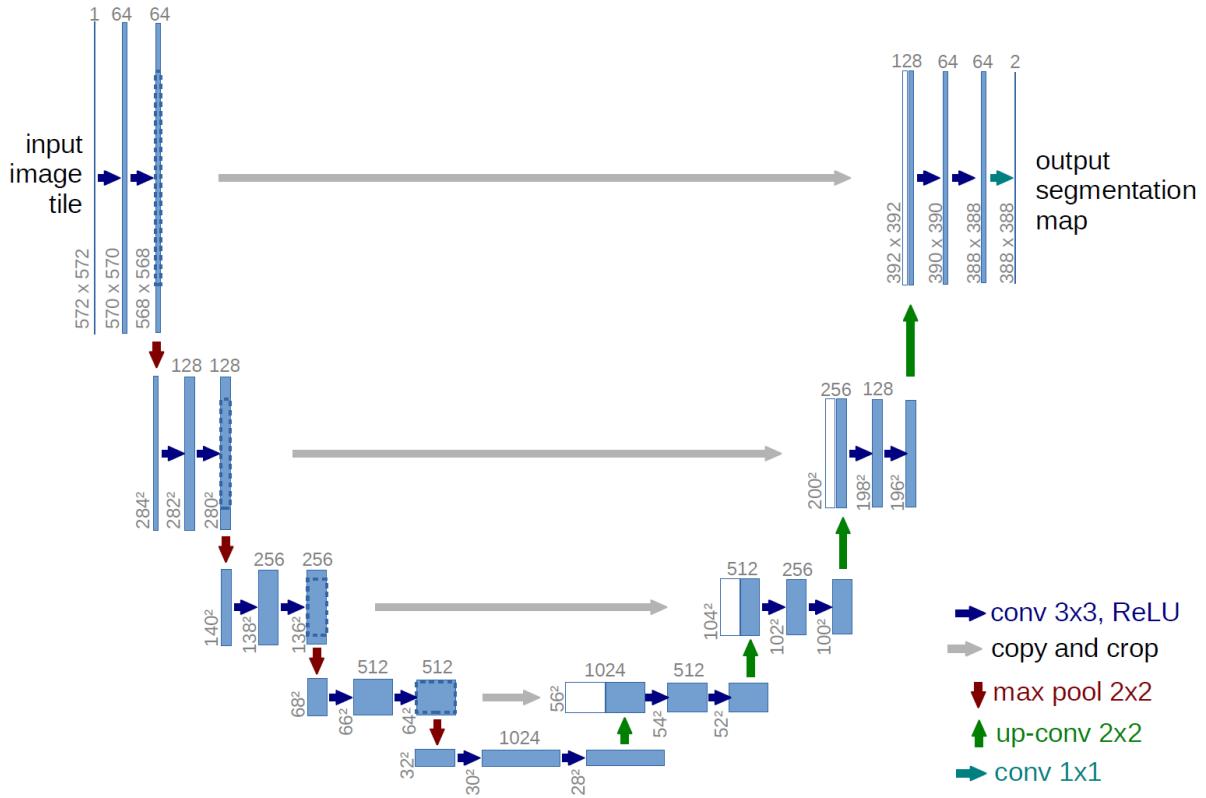


Fig. 13: Anstelle einer linearen Weiterverarbeitung der gefalteten Features werden Feature-Bereiche aus dem Encoder-Netzwerk in das Decoder-Netzwerk kopiert, um eine genauere Auflösung bei der Hochskalierung kleinerer Features zu erhalten. Informationen gehen somit bei der Encodierung nicht verloren, sondern können bei der Decodierung gezielt verwendet werden. (U-Net Architektur)<sup>101</sup>

Auch in der Entwicklung von Decoder-Architekturen wurde die Idee der Attention-Methode aufgenommen. Xu/Lam/Jia 2021 nutzen für ihre vorgeschlagene Architektur MaNet Attention-Mechanismen, welche in die Encodern ResNet, VGG und Inceptionv3 injiziert werden. MaNet folgt ebenfalls dem architektonischen Aufbau des U-Net.<sup>102</sup> Feature Pyramid Network (FPN) hingegen nutzen einen neuartigen Aufbau, welcher ähnlich wie das U-Net Verbindungen zwischen Encoder und Decoder Schichten zulässt.<sup>103</sup> Dabei generiert jede Schicht des Decoders eigene Vorhersagen, welche im weiteren Verlauf in MLF genutzt werden.

<sup>100</sup>Vgl. Zhou, Z. u. a. 2018, S. 6 f.

<sup>101</sup>Entnommen aus: Ronneberger/Fischer/Brox 2015, S. 235

<sup>102</sup>Vgl. Xu/Lam/Jia 2021, S. 99 f.

<sup>103</sup>Vgl. auch im Folgenden Xu/Lam/Jia 2021, S. 3 ff.

#### 2.5.4 Methoden beim Training von neuronalen Modellen

Da neuronale Netze sehr komplex sind, wurden im Laufe der letzte Jahre eine Reihe an Methoden entwickelt, welche das Training solcher Netze gezielt beeinflussen, um eine Steigerung der Effektivität und Effizienz des Trainings zu vergrößern.

Viele neuronale Netze besitzen einstellbare Parameter, sogenannte Hyperparameter. Dazu zählt zum Beispiel die Lernrate oder die Batch-Größe, also die Anzahl an Trainingsbeispielen auf welche gleichzeitig die Backpropagation angewandt wird. Diese Parameter werden auch *Hyper-Parameter* genannt.<sup>104</sup> Da Modelle beim maschinellen Lernen ihre Parameter stark an die Daten des Datensatzes anpassen, ist eine Evaluierung mit demselben Datensatz nicht aussagekräftig, weshalb zwischen einem Trainings- und Test-Datensatz unterschieden wird. Dabei bekommt das Modell den Test-Datensatz erst bei der Evaluierung des Modells zu sehen, so kann auch die Fähigkeit zur Generalisierung des Modells auf andere Eingangsdaten evaluiert werden. Die Teilung eines Datensatzes in Training und Test kann starke Auswirkungen auf die Ergebnisse haben, da vor allem die Größe des Training-Datensatzes eine große Rolle spielt.<sup>105</sup> Bei Modellen mit Hyper-Parametern wird neben dem Trainings- und Test-Datensatz auch zwischen einem Validierungs-Datensatz unterschieden. Bei dem Training der Parameter des Modells kommt der Trainingsdatensatz zum Einsatz, bei der Optimierung der Hyper-Parameter der Validierungs-Datensatz. Bei einem neuronalen Netzwerk wird dazu beispielsweise eine Rastersuche durchgeführt.<sup>106</sup> Dabei wird das Modell wiederholt mit unterschiedlichen Hyper-Parametern trainiert und mit dem Validierungs-Datensatz evaluiert. Das Modell mit der besten Kombination an Hyper-Parameter wird darauf gegen den Test-Datensatz getestet und diese Ergebnisse mit anderen Modellen verglichen. Um die Suche nach geeigneten Hyper-Parametern zu verkürzen, kann die Methode des *Pruning* angewandt werden.<sup>107</sup> Dabei wird wiederholt nach einer kleinen Anzahl an Epochen der Erfolg des Trainings frühzeitig evaluiert, sollte dieser Erfolg geringer sein als bisherige Trainings wird das Training abgebrochen.

Eine Annäherung der optimalen Lernrate, auch ein Hyper-Parameter, kann auch über die sogenannte *Cyclical Learning Rates* Methode gefunden werden.<sup>108</sup> Dazu wird über mehrere Epochen die Lernrate langsam erhöht, bis eine Lernrate gefunden wurde, mit welcher das Modell schnell zum Optimum gelangt. Da diese Methode nicht das reale Training, also mit einem Training mit einer konstanten Lernrate, widerspiegeln kann, ist der ermittelte Wert lediglich eine Annäherung an die optimale Lernrate. Gotmare u. a. 2018 untersuchten weitere heuristische Ansätze zur Anpassung der Lernrate während des Trainings. Bei einer *Aufwärm-Phase* wird die Lernrate über die ersten Epochen langsam erhöht. Dadurch wird der Fokus auf im Training-Datensatz früh aufkommende Beispiele unterdrückt, indem diesen bei der Backpropagation weniger Beachtung geschenkt werden.<sup>109</sup> Auch weitere Strategien zur Anpassung der Lernrate wie zum Beispiel die

---

<sup>104</sup>Vgl. Akiba u. a. 2019, S. 2623

<sup>105</sup>Vgl. Rácz/Bajusz/Héberger 2021, S 1115

<sup>106</sup>Vgl. auch im Folgenden Akiba u. a. 2019, S. 2624 f.

<sup>107</sup>Vgl. auch im Folgenden Akiba u. a. 2019, S. 2626 f.

<sup>108</sup>Vgl. auch im Folgenden Smith 1981, S. 465 f.

<sup>109</sup>Vgl. auch im Folgenden Gotmare u. a. 2018, S. 468 f., 470 f.

*stochastic gradient descent with restarts* Methode können zur Verbesserung des Trainings genutzt werden. Diese Strategie lässt die Lernraten entlang einer Kosinuskurve abklingen und setzt sie dann am Ende des Abklingens wieder auf den Ausgangswert zurück. Dadurch soll die Optimierung eines lokalen Optimums verhindert und weiterhin sichergestellt werden, dass ein gefundenes Optimum trotzdem voll ausgenutzt wird.

Anpassungen der Trainingsdaten vor dem Training können neuronalen Netzen helfen, schneller auf ein Problem trainiert zu werden. Normalisierung ist eine Technik, um alle Werte aneinander anzupassen, sodass sie sich in einem ähnlichen Spektrum befinden. Es konnte gezeigt werden, dass die Anwendung einer Normalisierung vor dem Training einen positiven Effekt auf den Lernprozess eines Modells haben kann.<sup>110</sup> Durch eine weitere Anpassung der Daten vor dem Training kann die sogenannte *Augmentierung* bieten. Bei der Augmentierung werden die Bilder zufallsbasiert gespiegelt, verdreht, ausgeschnitten oder auf andere Weisen transformiert.<sup>111</sup> Auch Rauschen oder Bewegungs-Artefakte kann bei der Augmentierung dem Bild künstlich hinzugefügt werden. Dadurch entstehen aus einem Trainings-Bild mehrere Beispiele für das Training. Dies ist vor allem bei kleinen Datensätzen sehr nützlich, da künstlich die Anzahl an Trainings-Beispielen erhöht wird. Weiterhin verringert sich so die Gefahr eines Overfittings des Modells auf die Trainings-Bilder, da es Eigenschaften wie Ausrichtung und ähnliche nicht erlernt. Ein Beispiel hierfür ist die Erkennung eines Hundes, welcher umgedreht auf dem Rücken liegt. Wenn das Modell mit einem Datensatz trainiert wurde, auf dem alle Hunde auf dem Bauch liegen, entsprechend ihr Gesicht „richtig herum“ ist, die Bilder aber durch Augmentierung gedreht wurden, kann das Modell auch Hunde mit gedrehten Gesichtern erkennen.

Neuronale Modelle eignen sich besonders gut zur Erkennung von Objekten auf Bildern, da sie einerseits die nötige Komplexität für diese Aufgabe aufweisen, als auch dazu in der Lage sind über bestimmte Methoden sich an den Kontext der Bilder anzupassen. **Daher können neuronale Modelle dazu genutzt werden, die Aufgabe der Bild-Segmentierung im Kontext der Hochwasser-Erkennung automatisch zu lösen.**

---

<sup>110</sup>Vgl. Shanker/Hu/Hung 1996, S. 389 f.

<sup>111</sup>Vgl. auch im Folgenden Cubuk u. a. 2019, S. 114 f.

### 3 Erläuterung und Durchführung der Experimente

Im Folgenden wird das theoretische Wissen aus dem Kapitel 2 in Experimenten zur Fluterkennung angewandt, mit dem Ziel die Fragestellung *Ist es mit heutigem Wissensstand möglich eine ähnliche, wenn nicht sogar bessere Genauigkeit und Zuverlässigkeit bei der Erkennung von Hochwasser auf Satellitendaten als durch manuelle, menschliche Annotierung zu erzielen?* zu beantworten. Dazu wird im ersten Schritt der Sen1Floods11 Datensatz beschrieben und verschiedene Normalisierungs-Methoden erläutert. Dieser Datensatz wird in allen folgenden Experimenten verwendet. Jedes Experiment wird mit einer einheitlichen Evaluierungs-Methodik ausgewertet, welche in Kapitel 3.2 beschrieben wird und folgen der DSR Struktur, welche in Kapitel 1.1 beschrieben ist. Die Vielzahl der Experimente resultiert aus der großen Komplexität des Themas: Angenommen man zählt die Kombinationen aller einstellbaren Parameter eines Modells zusammen, so ergibt sich eine Kombinations-Vielfalt von mehreren Milliarden möglicher Kombinationen. Um diese greifbarer zu gestalten werden die Kombinationsmöglichkeiten aus Normalisierung, Verlustfunktion, Band-Kombination und der Modell-Architektur sequentiell erschlossen.

#### 3.1 Nähere Untersuchung des Datensatzes

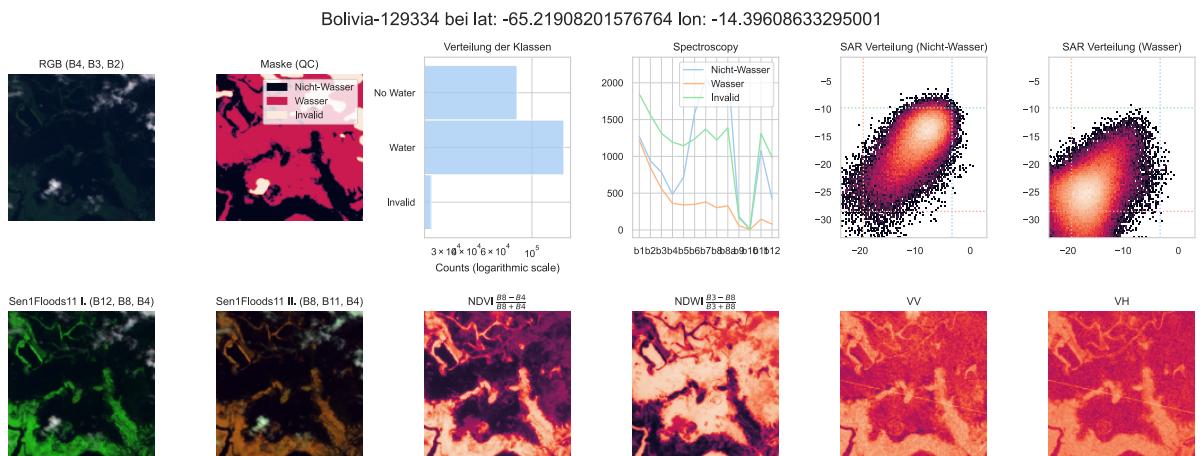


Fig. 14: Obere Reihe von links nach rechts: RGB Bild, Maske, Verteilung der drei Klassen, Spektroskopie der optischen Bänder, Verteilung der SAR Werte aller als Wasser gekennzeichneten Pixel, selbiges für Nicht-Wasser Pixel. Untere Reihe von links nach rechts: Falschfarbendarstellung der Bänder Kombination [B12, B8, B4], Falschfarbendarstellung der Bänder Kombination [B8, B11, B4], NDVI, NDWI, VV SAR-Band, VH SAR-Band. (Visualisierungen zu einer Satellitenaufnahme eines Gebiets in Bolivien)

Diese Arbeit benutzt ausschließlich den händisch-annotierten Datensatz Sen1Floods11 von Bonafilia u. a. 2020 für die Durchführung der Experimente. Dieser Datensatz enthält 446 Bilder der Größe  $512 \times 512$  Pixel von 11 verschiedenen Regionen, wovon alle Bilder Aufnahmen von

Überschwemmungen sind. Die Pixel des Datensatzes sind entweder als *Nicht-Wasser*, *Wasser* oder *Invalid* gekennzeichnet, *Wasser* bedeutet daher in diesem Zusammenhang meist auch überchwemmtes Land. *Invalid* Pixel können entweder Wolken oder aufgrund der Flugbahn von den Satelliten nicht erfasste Pixel sein. Im Folgenden wird der Datensatz von Bonafilia u. a. 2020 genauer untersucht, um ein grundlegendes Verständnis der Daten aufzubauen. Dazu zeigt Abbildung 14 verschiedene Visualisierungen einer Satellitenaufnahme eines Gebiets in Bolivien. In Kombination mit den statistischen Eigenschaften aus Tabelle 2 sollen indikative Einblicke in den Aufbau und die Eigenschaften des Datensatzes geben werden. Auf den Visualisierungen der unteren Reihe kann visuell grob zwischen *Wasser* und *Nicht-Wasser* unterschieden werden. Dies zeigt, dass die Daten der optischen Bänder mittels Verrechnung mit z. B. dem NDWI oder Falschfarbendarstellung eine Klassifizierung von *Wasser* auf Satellitendaten zulassen. Jedoch ist ein deutliches Rauschen, vor allem an den Grenzen zwischen der *Wasser* und *Nicht-Wasser* Klasse zu erkennen. Die Visualisierungen der beiden SAR-Bänder lassen auch eine visuelle Erkennung der beiden Klassen zu, jedoch ist ein starkes Rauschen zu erkennen. Über die Verteilung der SAR-Bänder kann ein grober Unterschied in der Verteilung der Werte von *Wasser* und *Nicht-Wasser* Pixeln erkannt werden.

	Durchschnitt	Std	Min	Max	1. Percentil	99. Percentil
vv	-10.2	4.4	-89.8	36.8	-19.3	-3.3
vh	-16.9	5.4	-103.9	17.4	-28.5	-9.8
b1	1612.3	694.6	0.0	9994.0	1252.3	2872.9
b2	1379.9	734.6	0.0	19182.0	962.9	2822.1
b3	1344.4	731.6	0.0	15100.0	840.4	2811.1
b4	1195.2	860.6	0.0	17418.0	588.2	2942.2
b5	1439.2	771.4	0.0	10975.0	768.6	3001.8
b6	2344.2	921.6	0.0	11847.0	1083.3	4004.9
b7	2796.5	1088.0	0.0	12611.0	1247.6	4680.2
b8	2578.4	1029.2	0.0	19199.0	1080.9	4375.1
b8a	3023.8	1205.1	0.0	13253.0	1259.1	5015.5
b9	476.7	331.7	0.0	3947.0	226.5	888.4
b10	59.2	130.4	0.0	1735.0	35.9	113.4
b11	1989.2	993.7	0.0	15031.0	681.3	3829.7
b12	1152.5	768.9	0.0	13472.0	357.9	2707.6

Tab. 2: Durchschnitt, Standardabweichung, Minima, Maxima und die 1% Quantile der verschiedenen Bänder des Sen1Floods11 Datensatzes. Alle Werte wurden auf eine Nachkommastelle gerundet. (Sen1Floods11 Datensatz: Statistische Eigenschaften)

Die Tabelle 2 zeigt die statistischen Eigenschaften des gesamten Datensatzes auf. Die Werte des Sentinel-1 SARs bewegen sich zumeist im negativen Bereich, die Werte der optischen Bänder stets im positiven. Weiterhin weisen die SAR-Werte im Gegensatz zu den optischen Werten eine relativ geringe Standardabweichung auf. Unter den optischen Bändern befinden sich zwei Bänder, B9 und B10, welche ebenfalls eine vergleichsweise geringe Standardabweichung aufweisen. Diese scheinen aufgrund der geringen Standardabweichung und am Durchschnitt nah anliegenden 1% Quantilen eine sehr geringe Menge an starken Ausreißern zu besitzen. Die hohen Maximalwerte der anderen

optischen Bänder lassen in Verbindung mit den vom Durchschnitt weit entfernten 1% Quantilen auf eine allgemein bessere Verteilung der Werte und eine geringe Menge aus schwachen Ausreißern schließen. Ähnlich verhält es sich mit den niedrigen Minimalwerten der SAR-Bänder.

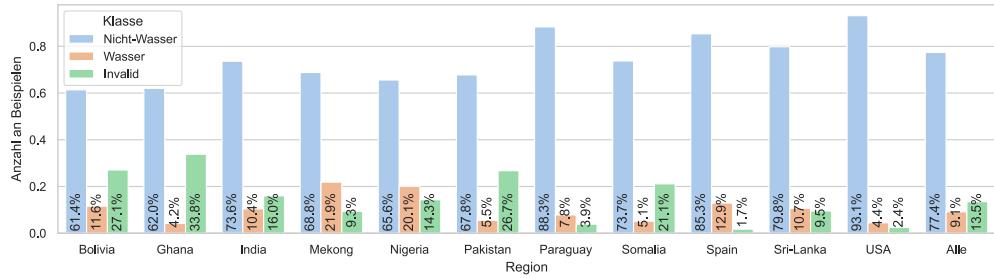


Fig. 15: Verteilung der Klassen *Wasser*, *Nicht-Wasser* und *Invalid* über die verschiedenen Regionen. Es ist zu deutlich zu erkennen, dass die Pixel überwiegend zur Klasse *Nicht-Wasser* gehören. (Sen1Floods11 Datensatz: Verteilung der Klassen)

Grafik 15 zeigt die Verteilung der Klassen über die verschiedenen Regionen. Es fällt auf, dass die Klasse *Nicht-Wasser* deutlich häufiger vertreten ist als die anderen beiden Klassen. Der Datensatz weist also ein Bias bezüglich der *Nicht-Wasser* Klasse auf. Ebenso ist zu erkennen, dass die Regionen *Bolivien*, *Ghana*, *Pakistan* und *Somalia* zusätzlich einen deutlich höheren Anteil an *Invaliden* Pixeln aufweisen. Dies hat jedoch keine nähere Bedeutung, da diese Pixel neben Wolken auch schlicht von den Satelliten nicht erfasst wurden.

### 3.1.1 Normalisierungs-Methoden

Im Folgenden werden die drei unterschiedlichen Normalisierungsmethoden *Standard-Normalisierung*, *MinMax-Normalisierung* und *3-Sigma-Normalisierung* auf den Datensatz angewandt und exemplarische Grafiken aufgezeigt. Es werden alle Normalisierungen auf jedes Band einzeln angewandt, sodass jedes Band als eigenständiger Datensatz verstanden werden kann.

Die erste Normalisierungsmethode *Standard-Normalisierung* wurde von der Python-Bibliothek Albumentations implementiert und nutzt den Mittelwert und die Standardabweichung zur Berechnung der normalisierten Werte:

$$n_i = \frac{x_i \cdot \text{Mean}(x)}{\text{STD}(X)} \quad (3.1)$$

Diese Normalisierungs-Methode ist im Bereich des maschinellen Lernens sehr beliebt, da sie die Werte so normalisiert, dass der Datensatz nach der Normalisierung einen Durchschnitt von 0 und eine Standardabweichung von 1 aufweist. Dies ist auch in der statistischen Auswertung in Tabelle 12, welche dem Anhang 1 hinzugefügt wurde, zu sehen: Alle Bänder weisen einen Durchschnitt von 0 und eine Standardabweichung von 1 auf. Ein Problem bei dieser Normalisierung fällt beim Betrachten des NDVI und NDWI auf: Alle Werte liegen in einem sehr kleinen Bereich, bis auf ein

paar wenige Ausreißer, deren Beträge sehr groß sind. Abbildung 30, welche ebenfalls dem Anhang 1 hinzugefügt wurde, zeigt Beispiel ähnlich der Abbildung 14 aller folgenden Normalisierungen. In den obersten NDVI und NDWI Darstellungen der Abbildung machen sich die Ausreißer dadurch erkenntlich, dass alles in einer Farbe dargestellt wird: die Werte sind alle sehr ähnlich bis auf sehr wenige Ausnahmen. Dies geschieht, da die Werte der Komponenten zur Berechnung der beiden Indizes sowohl positive, als auch negative Werte annehmen können. Dadurch können sie sich bei der Berechnung ausgleichen und schaffen somit ein starkes Ungleichgewicht zwischen Numerator und Denominator. Daher ist diese Normalisierungs-Methode nicht geeignet zur Analyse von berechneten Bändern.

Die *MinMax-Normalisierung* normiert die Werte auf die Skala (0|1). Somit befinden sich alle Werte im positiven Bereich und können entsprechend besser miteinander verrechnet werden. Dazu nutzt sie den Minimal- und den Maximalwert eines Datensatzes:

$$n_i = \frac{x_i - \text{Min}(X)}{\text{Max}(X) - \text{Min}(X)} \quad (3.2)$$

In dem zweiten Teil der Abbildung 30 sind die berechneten NDVI und NDWI Werte wieder visuell sichtbar. Diese Methode berücksichtigt jedoch keine Ausreißer in den Daten. Auch dies ist in Abbildung 30 zu erkennen, die Falschfarben-Darstellungen sind sehr dunkel, was auf einen Bias der normalisierten Werte gegen 0 schließen lässt. Durch Ausreißer in den Daten wird das Min-Max-Verhältnis stark verändert, so entsteht wie hier ein Bias. Auch in der angehängten Tabelle 13 ist dieser Bias anhand der Durchschnitts-Werte der Bänder zu erkennen. Insbesondere die optischen Bänder weisen einen sehr geringen Durchschnitt auf, vermutlich da der Maximal-Wert jedes Bandes sich weit von den durchschnittlichen Daten befindet.

Um dies zu lösen normiert die dritte Normalisierungs-Methode *3-Sigma-Normalisierung* die Werte mit demselben Prinzip wie die *MinMax-Normalisierung* Methode normalisiert. Jedoch anstelle des Minimums, respektive Maximums, wird das 1% bzw. das 99% Percentil genutzt:

$$n_i = \frac{x_i - \text{Perc1}(X)}{\text{Perc99}(X) - \text{Perc1}(X)} \quad (3.3)$$

So werden entsprechende Ausreißer ignoriert. Da diese nun jedoch nach der Normalisierung wieder negative Werte, bzw. Werte über 1 annehmen können, werden diese auf  $0 + \text{eps}$  und  $1 - \text{eps}$  gesetzt. Das IEEE hat  $\text{eps} = 2^{**} - 52$  definiert, was die kleinste Zahl über 0 repräsentiert.<sup>112</sup> Im dritten und untersten Teil der Abbildung 30 sind im Vergleich zu den bisherigen Abbildungen die Unterschiede zwischen Wasser und Nicht-Wasser Flächen deutlich besser sichtbar. Insbesondere die berechneten Bänder NDVI und NDWI profitieren stark von der Normalisierung. Die statistischen Eigenschaften des Datensatzes nach der Anwendung der 3-Sigma-Normalisierung zeigen eine große Stabilität aller Werte. Der Durchschnitt aller Bänder beträgt 0.5, alle Werte befinden sich zwischen 0 und 1. Ebenso ähnelt sich die Standardabweichung aller Bänder sehr und ist

---

<sup>112</sup>Vgl. Committee 2019

vergleichsweise gering. Bis auf die Werte des optischen Bandes B10 befinden sich 98% zwischen 0.3 und 0.7 um 0.5 herum. Durch die Vereinheitlichung der Werte durch die 3-Sigma-Normalisierung soll es den folgenden Modellen leichter fallen, Zusammenhänge in den Daten zu erkennen und zu nutzen.

#### 3.1.2 Einstellung des Datensatzes zur Verwendung in Experimenten

In dieser Arbeit wurden alle Beispiele aus Ghana dem Test-Datensatz zugeordnet, alle anderen dem Training-Datensatz. So sollen die Modelle nicht nur keine Beispiele aus dem Test-Datensatz bei dem Training nutzen, sondern auch keine Beispiele welche geografische Ähnlichkeiten mit diesem aufweisen. So wird automatisch die Robustheit gegenüber ungewöhnlichen Regionen der Modelle getestet. Bei Experimenten, welche zusätzlich einen Validierung-Datensatz erfordern, wurde der Training-Datensatz zufällig in einen 85% Anteil für das Training und einen 15% Anteil für die Validierung geteilt. Weiterhin wird die 3-Sigma-Normalisierung für die folgenden Experimente genutzt.

## 3.2 Methodik zur Evaluation

Bei der Auswertung der Experimente wurde die Evaluierung der Modelle in drei Teile geteilt: Eine quantitative Analyse um die objektive Effektivität eines Modells zu bestimmen, eine qualitative Analyse um die subjektive Effektivität eines Modells zu beurteilen sowie eine Aufwand-Analyse um die Effizienz eines Modells zu bestimmen. Nicht alle Experimente nutzen alle drei Teile der Evaluierung, da sie entweder zu aufwendig sind oder zu wenig Aussagekraft besitzen. Weiterhin wurde diese Form der Evaluierung lediglich auf Experimente angewandt bei welchen Modelle mit maschinellem Lernen trainiert wurden. Ausgenommen sind daher die Experimente *Verhalten von Verlustfunktionen in unausgeglichenen Datensätzen* und *Spektroskopische Betrachtung des Datensatzes*.

In der quantitativen Analyse wurden zu jedem Modell folgende Metriken aufgenommen: Mikro-Genauigkeit, mIoU sowie für die Klasse *Wasser* Präzision, Sensitivität und IoU. Diese große Anzahl an erhobener Metriken ist notwendig, um einerseits präzisere Aussagen zum Verhalten der Modelle treffen zu können, andererseits aufgrund des Bias im Datensatz. Der stark überwiegende Anteil an Landfläche gegenüber Wasser beziehungsweise überschwemmter Fläche verschiebt die Aussagefähigkeit bestimmter Metriken. So würde ein sehr schlechtes Modell, welches für jeden Pixel jedes Bildes ausschließlich die Klasse *Nicht-Wasser* vorhersagt, eine sehr hohe Genauigkeit von rund 90% aufweisen, da dies der Anteil an *Nicht-Wasser* Pixeln im Datensatz ist. Mit der Mikro-Genauigkeit in Verbindung mit dem mIoU soll der allgemeine Erfolg bezüglich der Segmentierung der verschiedenen Klassen erfasst werden. Der IoU der *Wasser* Klasse soll genaue Hinweise geben, wie gut das Modell Wasser auf den Satellitendaten erkennen kann. Die

Präzision des Modells bezüglich der Klasse *Wasser* soll aufzeigen, wie sicher ein als *Wasser* markierter Pixel tatsächlich *Wasser* abbildet. Hingegen die Sensitivität bezüglich der Klasse *Wasser* soll aufzeigen, mit welcher Wahrscheinlichkeit ein Modell einen *Wasser* Pixel auch als *Wasser* markiert. Die Präzision und die Sensitivität geben zusammen die Verlässlichkeit des Modells wieder. Alle Metriken wurden von der Python Bibliothek *Torchmetrics* implementiert.<sup>113</sup> Um die Aussagekraft der erfassten Metriken zu steigern, wurden die Experimente mindestens fünfmal wiederholt.

Bei der qualitativen Analyse werden exemplarisch zwei Beispiele visuell genauer betrachtet. Ein Beispiel zeigt eine große Landfläche mit zwei kleinen Seen, das andere einen Fluss, welcher eine sehr große Fläche umliegendes Land überschwemmt sowie einen kleinen Teil an Wolken aufweist. So soll evaluiert werden können ob ein Modell (1) auch kleine Wasserstellen auf den Satellitendaten erkennt, (2) Überschwemmungen erkennt, (4) die Grenze zwischen Landfläche und Wasser korrekt zeichnet und (3) von Wolken stark beeinflusst wird. Ebenso kann so der Erfolg bezüglich fluvialen und pluvialen Fluten untersucht werden. Da es sich hierbei jedoch um subjektive Betrachtungen handelt wird die qualitative Analyse weniger gewichtet.

Als letzter Teil der Evaluation von Modellen wurde eine Aufwand-Analyse durchgeführt. Dabei wird zu einem Modell die Anzahl an trainierbaren Parametern sowie die benötigten MACs um ein Bild der Größe  $512 \times 512$  zu segmentieren erfassst. Die Anzahl trainierbarer Parameter spiegelt neben dem Verbrauch bei der Speicherung der Gewichte des Modells auch den benötigten Platz im Arbeitsspeicher eines Modells wider. Die benötigten MACs spiegeln wiederum den benötigten Energieverbrauch sowie Laufzeit-Aufwand wider.

## 3.3 Training zweier Vergleichsmodelle

Um die genutzten Modelle in ihrer Effektivität zu beurteilen wurden zwei grundlegende Modelle (Baselines) implementiert und evaluiert. Das erste Modell nutzt einen linearen Klassifizierungsalgorithmus, das zweite ein vor-trainiertes U-Net der Python-Bibliothek Segmentation Models Pytorch (SMP).<sup>114</sup>

### 3.3.1 Statistisches SAR-Grenzwert Modell

Der lineare Klassifizierungsalgorithmus klassifiziert jeden Pixel basierend auf den SAR-Werten von Sentinel 1. Dazu werden die VV und VH Bänder mit der 3-Sigma Normalisierung normiert, auf einer zweidimensionalen Ebene aufgespannt und die Verteilung beider Bänder abgebildet. Alle Werte, die unterhalb einer Trennlinie  $y = mx + n$  liegen werden der Klasse *Wasser* und alle anderen der Klasse *Nicht-Wasser* zugeordnet. Diese Trennlinie wurde händisch festgelegt. Dazu wurde zuerst anhand mehrerer Beispiele des Datensatzes erkannt, dass die Steigung der

---

<sup>113</sup>Detlefsen u. a. 2022

<sup>114</sup>Yakubovskiy 2020

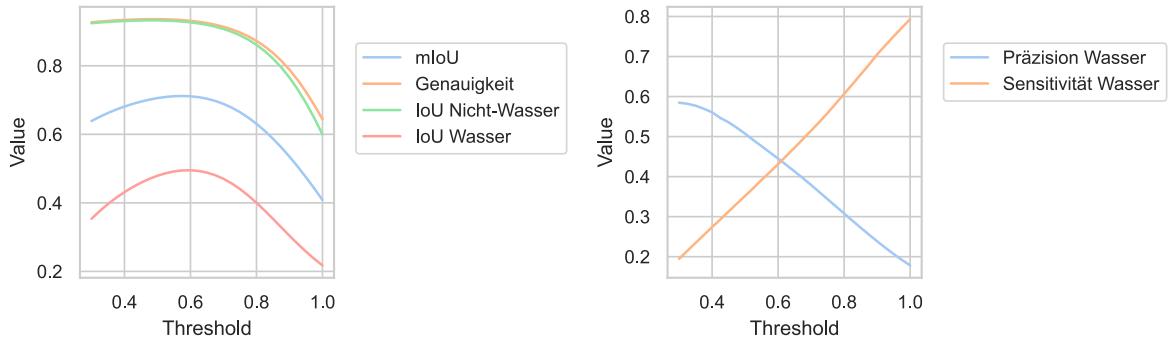


Fig. 16: Errechnete Metriken der Rastersuche für Grenzwert (Threshold)  $n \in [0.3; 1]$ . Links: OA (Micro Average), IoU für beide Klassen und mIoU. Rechts: Präzision (Precision) und Sensitivität (Recall) der Klasse Wasser. (Lineares Modell: Resultate der Rastersuche)

Trennlinie grob  $m = -1$  beträgt. Darauf wurden verschiedene Ordinatenabschnitte  $n$  gegen den Datensatz getestet und verschiedene Metriken miteinander verglichen. In Abbildung 16 werden die Ergebnisse dieser Rastersuche gezeigt. Der mIoU und der IoU der Wasser Klasse besitzen beide bei  $n = 0.625$  mit  $mIoU = 70,7\%$  und  $IoU_{Water} = 48,7$  ihren Scheitelpunkt. Die Präzision und Sensitivität der Wasser Klasse treffen sich ebenfalls an dieser Stelle. Da der Fokus der Modelle eher auf der *Wasser* Klasse als auf der *Nicht-Wasser* Klasse beruhen soll, kann beides ein Indiz dafür sein, dass 0.625 der optimale Wert für  $n$  ist.

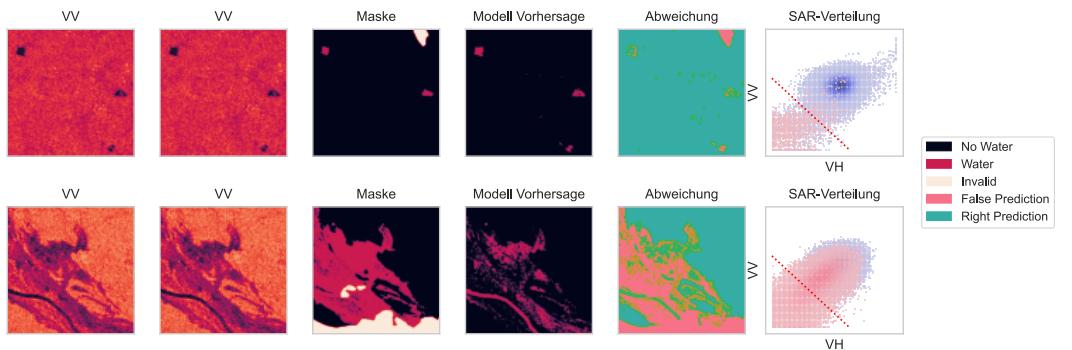


Fig. 17: Zwei Beispiel Gebiete in Ghana. Von links nach rechts: VV-Band, VH-Band, annotierte Maske, vom linearem Modell vorhergesagte Maske, Unterschiede beider Masken, Verteilung beider SAR-Bänder mit Trennlinie (rot). (Lineare Klassifikation anhand SAR-Verteilung)

Die Abbildung 17 zeigt zwei Beispiele dieser Klassifizierung mit  $n = 0.625$ , Tabelle 3 quantitative Metriken. Die Grafiken ganz rechts zeigen die Verteilung der Klassen *Wasser* (Blau) und *Nicht-Wasser* (Violett). Im oberen Beispiel ist gut zu erkennen, dass sich zwei Cluster gebildet haben, welche über eine Linie bis auf wenige Ausnahmen gut getrennt werden können. Im oberen fünften Bild von links ist der Erfolg dieses Ansatzes zu erkennen: Das Modell klassifiziert die angegebenen Gewässer bis auf wenige Ausnahmen am Rand der Gewässer korrekt. Das untere Beispiel jedoch konnte nicht so gut erkannt werden, es sind viele Falsch-Klassifikationen sichtbar. Anhand der

Verteilung beider Bänder in der unteren rechten Grafik der Abbildung 17 lässt sich erkennen, dass viele annotierte *Wasser* Pixel sehr ähnliche SAR-Werte wie *Nicht-Wasser* Pixel haben. Daher ist es dem linearen Modell nicht möglich diese richtig zu erkennen. Dieses Modell kann also zwar kleine Wasserstellen erkennen, erkennt jedoch überschwemmte Gebiete nicht sonderlich gut, wird dafür wiederum nicht von Wolken beeinflusst.

Metrik	2D-Grenzwert	U-Net
<b>Genauigkeit</b>	91,87%	75,69% $\pm$ 7,1
<b>mIoU</b>	70,69%	47% $\pm$ 10,11
<b>IoU Nicht-Wasser</b>	92,63%	71,64% $\pm$ 6,88
<b>IoU Wasser</b>	48,75%	24,4% $\pm$ 14,42
<b>Präzision Wasser</b>	41,4%	11,52% $\pm$ 5,37
<b>Sensitivität Wasser</b>	41,88%	8,54% $\pm$ 6,29

Tab. 3: Erfasste Metriken des linearen Klassifizierungs-Algorithmus für  $n = 0.625$  und des unoptimierten U-Net. (Vergleichsmodelle: Erfasste Metriken)

Das lineare Modell sagt *Nicht-Wasser* Pixel um 43,88% IoU besser voraus als *Wasser* Pixel. Dies zeigt einen starken Bias im Modell gegenüber der *Nicht-Wasser* Klasse auf. Dieser Bias basiert vermutlich auf der ungleichen Verteilung beider Klassen im Trainingsdatensatz in Verbindung mit einem großen Anteil an Überschneidungen in der Verteilung der SAR-Werte beider Klassen. Aufgrund dieser ungleichen Verteilung beider Klassen im Datensatz kann die Mikro-Genauigkeit auch nicht als aussagekräftige Metrik herangezogen werden. Stattdessen ist der mIoU mit 70,69% hier deutlich aussagekräftiger. Weiterhin ist die Verlässlichkeit des Modells mit einer Präzision von 41,4% und einer Sensitivität von 41,88% der Klasse *Wasser* gering. Dafür ist das Modell jedoch äußerst performant, um ein Bild der Größe  $512 \times 512$  zu segmentieren benötigt es  $512 \cdot 512 \cdot 4 = 1\text{Mega-MAC}$ , weiterhin besitzt es lediglich einen trainierbaren Parameter und ist somit sehr speicher freundlich.

### 3.3.2 Naives Training eines U-Net

Das zweite Vergleichsmodell nutzt ein vor-trainiertes U-Net welches Objekte auf RGB-Bildern klassifiziert. Die Implementierung und vor-trainierten Gewichte stammen aus der Python Bibliothek *SMP*.<sup>115</sup> Die Implementierung der Modelle der Bibliothek folgen dem Schema der Encoder-Decoder Modelle und erlaubt weiterhin verschiedene Encoder, Decoder und Aktivierungsfunktionen über Parameter einzustellen. In diesem und allen folgenden Experimenten wird der gleiche Versuchsaufbau bezüglich des Trainings genutzt, dabei wurde die Python-Bibliothek *Pytorch-Lightning* genutzt.<sup>116</sup> Die genutzte Modell-Architektur inklusive Encoder, Decoder und Aktivierungsfunktion wird im Vorhinein über einstellbare Parameter definiert. Alle vor-trainierten Gewichte stammen aus Trainings mit Daten aus dem Imagenet Datensatz.<sup>117</sup> Jedes Experiment

---

<sup>115</sup>Vgl. Yakubovskiy 2020

<sup>116</sup>Falcon u. a. 2022

<sup>117</sup>Deng u. a. 2009

wiederholt das Training des eingestellten Modells mehrere Male um aussagekräftige Ergebnisse, unabhängig von zufallsbasierten Faktoren, wie zum Beispiel die Reihenfolge der Trainingsdaten oder die Augmentierungen dieser, treffen zu können. Bei dem Training wurde Pruning angewandt, also nachdem sich der Verlust der Validierungsphase über drei Epochen nicht mehr verbessert hat abgebrochen. Ein frühzeitiger Abbruch führt zu einem unvollständigen Training und somit zu einem unfertig trainierten Modell. Dies sorgt für ein verfälschtes Ergebnis der Testmetriken, was jedoch vernachlässigt werden kann, da diese Verfälschung nur sehr geringfügig ist und ausreicht um indikative Schlüsse ziehen zu können. Durch den vorzeitigen Abbruch des Trainings wird dafür Zeit und Rechenkapazität gespart, wodurch eine Wiederholung des Experiments möglich ist, um aussagekräftigere Ergebnisse zu liefern. Trainiert wurde in allen Experimenten mit einem Adam Optimierungsalgorithmus, welcher einen über Parameter einstellbaren Verlust der Vorhersage gegenüber der Ziel-Maske minimiert. Ebenfalls einstellbar ist die Kombination der genutzten Modalitäten, anhand derer das Modell eine Vorhersage produziert. Die Hyperparameter Batch-Größe und Lernrate kann ebenfalls über Parameter des Versuchsaufbaus eingestellt werden. Weiterhin einstellbar ist die Form der Datenaufbereitung wie Normalisierung und Augmentierung, alle Experimente wenden horizontale und vertikale Spiegelungen sowie zufällige 90° Drehungen an, um die Menge an Trainingsbeispielen künstlich zu erhöhen.

<b>Modell Architektur</b>	UNet
<b>Encoder</b>	Resnet50
<b>Aktivierungs-Funktion</b>	Sigmoid
<b>Normierungs-Methode</b>	SMP
<b>Modalitäten</b>	RGB [B4, B3, B2]
<b>Verlustfunktion</b>	Kreuz-Entropie
<b>Lernrate</b>	5e-4
<b>Batch Größe</b>	16
<b>Wiederholungen</b>	10

Tab. 4: Parameter-Einstellungen vom Versuchsaufbau des U-Net Vergleichsmodell Trainings.  
(Training-Parameter: U-Net Vergleichsmodell)

Tabelle 4 zeigt die Versuchseinstellungen des zweiten Vergleichsmodells. Die Daten wurden vorher nicht mit der 3-Sigma Normalisierung normiert, da das vortrainierte U-Net für RGB-Bilder eine eigene Normierung, bzw. Datenaufbereitung implementiert. Das U-Net wurde mit den Trainingsdaten der Sentinel 2 Bänder 4, 3 und 2, also den RGB Bändern, trainiert um das Modell an die Flut-Klassifikation anzupassen. Trainiert wurde mit einem Adam Optimierungsalgorithmus welcher den Kreuz-Entropie-Verlust der Vorhersage gegenüber der Ziel-Maske minimieren soll. Ein Training dauerte zwischen 7 und 20 Minuten über 6 bis 18 Epochen.

Tabelle 3 zeigt die quantitativen Ergebnisse des Trainings. Die Genauigkeit und der mIoU sind mit  $75,69\% \pm 7,1$  und  $47\% \pm 10,11$  deutlich schlechter als die des linearen Vergleichsmodells. Auch die unterrepräsentierte Klasse *Wasser* weist lediglich einen IoU von  $24,4\% \pm 14,42$  mit einer sehr hohen Standardabweichung. Dies zeigt eine große Varianz sowie Zufalls-Abhängigkeit der Trainingsdaten auf, das Modell verhält sich nicht robust. Besonders gering ist die Verlässlichkeit des Modells gegenüber der Erkennung von *Wasser* Pixeln mit einer Präzision von gerade mal

$11,52\% \pm 5,37$  und einer Sensitivität von  $8,54\% \pm 6,29$ . Das U-Net besitzt 32,5 Millionen trainierbare Parameter und ist somit deutlich komplexer und aufwendiger als das lineare Modell mit einem einzigen trainierbarem Parameter. Ebenso benötigt das U-Net für eine einzige Vorhersage 42.83 Giga-MACs und ist somit auch deutlich rechen-aufwendiger als das lineare Modell welches 1 Mega-MAC für eine Vorhersage benötigt.

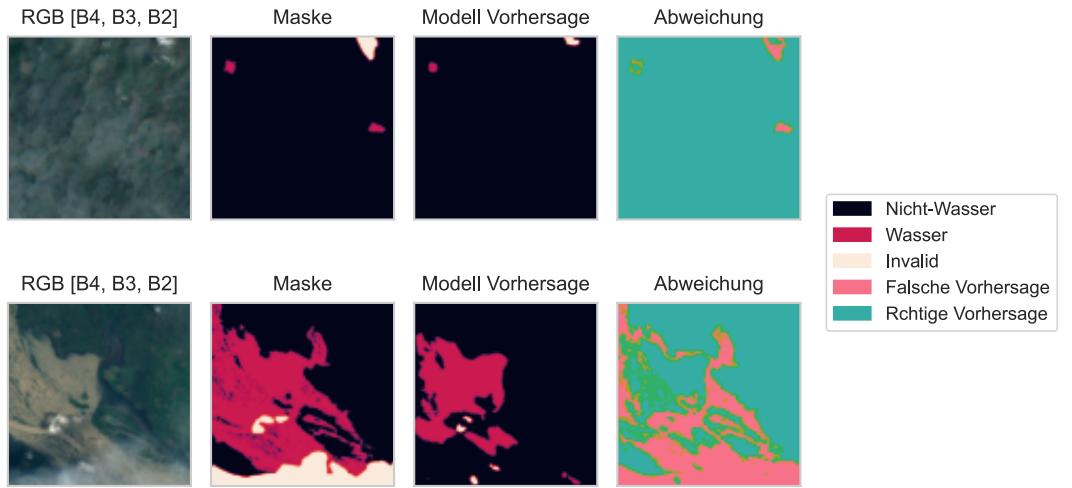


Fig. 18: Zwei Beispiel Gebiete in Ghana. Von links nach rechts: RGB Darstellung, annotierte Maske, vom U-Net Vergleichsmodell vorhergesagte Maske, Unterschiede beider Masken. (Qualitative Visualisierung: U-Net Vergleichsmodell)

In Abbildung 18 sind zwei Beispiele einer Bild-Segmentierung des Modells zu sehen. Im ersten Beispiel konnte das Modell keine der beiden Wasserstellen erkennen, hat dafür jede invalide Pixel gekennzeichnet, wo keine sind. Im zweiten Beispiel konnte das Modell jedoch den übergelaufenen Fluss grob erkennen, auch vereinzelt invalide Pixel wurden richtig klassifiziert. Jedoch weichen die vorhergesagten Formen, also die Begrenzungen der *Wasser* und *Nicht-Wasser* Flächen stark von der annotierten Maske ab. Dieses Modell kann weder kleine Wasserstellen erkennen, noch die Grenze zwischen *Wasser* und *Nicht-Wasser* richtig kennzeichnen. Vermutlich führen Ungenauigkeiten der annotierten Masken zu diesem Ergebnis, da das Modell empfindlich auf solche Störungen reagiert. Eine weitere Ursache kann in der Wahl der Modalitäten liegen: Die gewählten Bänder B5, B4 und B3 (die sichtbaren RGB Bänder) lassen eine Unterscheidung zwischen *Wasser* und *Nicht-Wasser* nur an wenigen Stellen zu. Dies lässt sich auch auf den Beispielbildern erkennen: Abbildung 18 zeigt links zwei Beispiele visualisiert mit den RGB Bändern. Besonders bei dem oberen Beispiel ist es schwer, wenn nicht sogar unmöglich, mit bloßem Auge die zwei auf der Maske markierten Wasserstellen ausfindig zu machen. Das Modell stößt auf dieselben Schwierigkeiten, da künstliche, neuronale Netze wie dieses von den neuronalen Netzen der Menschen abstammen.

### 3.4 Verschiedene Verlustfunktionen im Vergleich

Um neuronale Netze wie das U-Net trainieren zu können, benötigt es einer geeigneten Verlustfunktion, mit welcher der Optimierer die Parameter des Modells anpassen kann. Das Baseline U-Net nutzt die Kreuz-Entropie Verlustfunktion zum Training des Modells, welche unausgeglichenen Daten nicht berücksichtigt. Im Falle der Hochwasser-Erkennung auf Satellitendaten mit einer Unter-Präsentation der Klasse *Wasser* soll eine optimale Verlustfunktion alle Abweichungen von mehreren Klassen der vorhergesagten Maske gegenüber der annotierten Maske und weiterhin die ungleiche Verteilung der Klassen berücksichtigen.

#### 3.4.1 Verhalten von Verlustfunktionen in unausgeglichenen Datensätzen

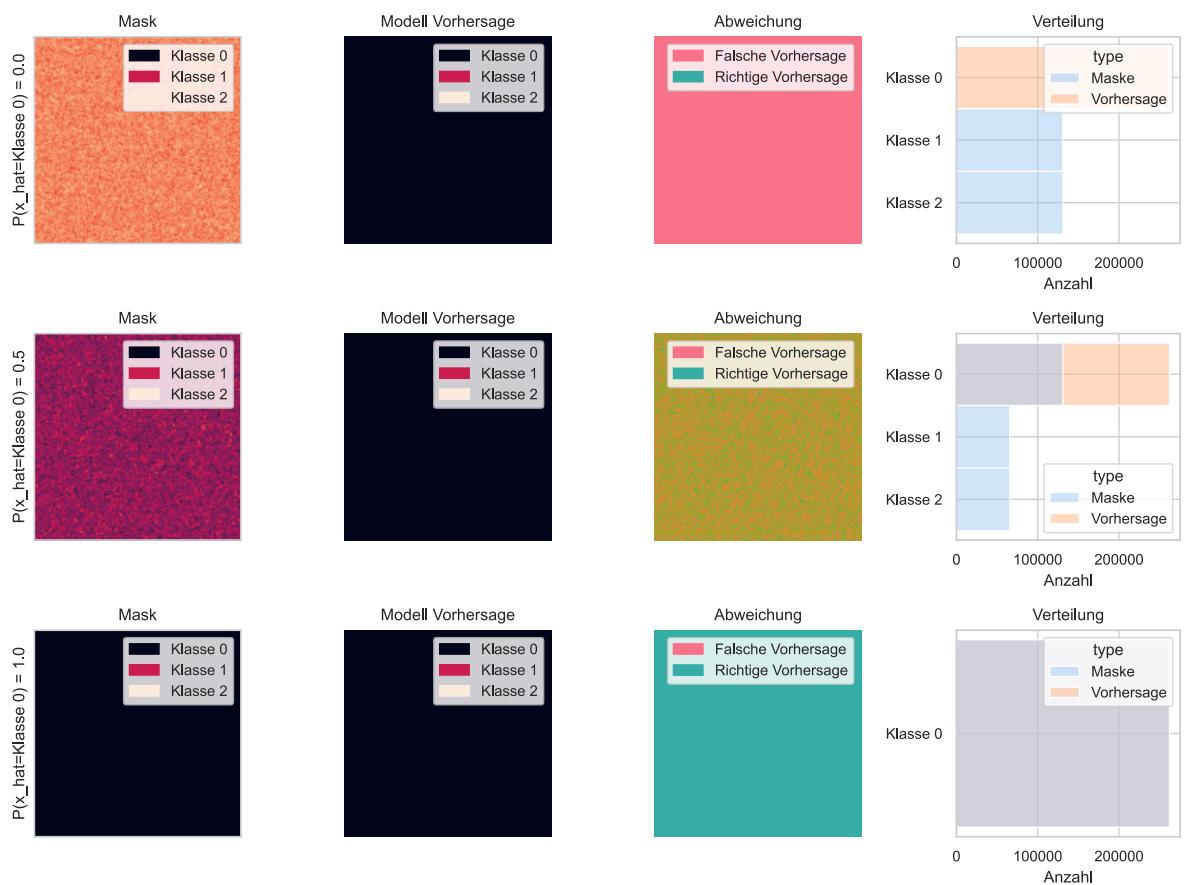


Fig. 19: Exemplarischer Aufbau des Experiments: drei zufällig generierte Masken mit  $P(y = \text{Class}0) = 0$ ,  $P(y = \text{Class}0) = 0.5$  und  $P(y = \text{Class}0) = 1$  abgeglichen mit dem Immer-Klasse-0 Modell. (Verlustfunktionen: Experimentaufbau)

Dazu wurden verschiedene Verlustfunktionen in einem Experiment miteinander verglichen. Dieses Experiment ist angelehnt an die Versuche von Lin u. a. 2017. Lin u. a. 2017 untersuchten die Auswirkungen des Faktors  $\gamma$  auf die Verlustfunktion, welcher Fokus auf schwer erlernbaren Klassen

legen soll. Ziel des Experiments ist es diesen Fokus zu quantifizieren und somit indikativ das Verhalten der Verlustfunktion bei schwer erlernbaren Klassen gegenüber leicht erlernbaren Klassen zu untersuchen. Es spiegelt daher nicht die Realität wider und kann daher nur bedingt Aussagen über diese treffen. Ausgangslage des Experiments ist ein Klassifizierungsmodell, welches immer Klasse 0 vorhersagt, daher gilt  $p_0 = 1$ . Daraufhin wurden 28 verschiedene Zielmasken  $y$  zufällig basierend auf einem Wahrscheinlichkeitswert  $P(y = \text{Class0}) \in [0; 1]$  generiert. So wurden unterschiedliche Verteilungen der Klassen in den Pseudo-Trainingsdaten dargestellt. Die Abbildung 19 stellt diesen Versuchsaufbau grafisch dar. Zu jeder Zielmaske wurden die Verluste mit einer Reihe an Verlustfunktionen berechnet und in der Abbildung 20 grafisch gegenübergestellt.

Es wurden die Verlustfunktionen Kreuz-Entropie, Dice-Kreuz-Entropie, Tversky, Dice und IoU verwendet und miteinander verglichen. Dazu wurden die Verlustfunktionen Tversky, Dice und IoU eigens implementiert und die Kreuz-Entropie Verlustfunktion Implementierung der Python Bibliothek PyTorch genutzt. Für die Dice-Kreuz-Entropie Verlustfunktion wurden die Dice und die Kreuz-Entropie Verlustfunktion addiert. Für die eigens implementierten Verlustfunktionen wurden zusätzlich  $\alpha$  und  $\gamma$  Parameter implementiert. Der  $\alpha$  Parameter ermöglicht eine Gewichtung einzelner Klassen. Panchapagesan u. a. 2016 nutzen für das Training ihres Modells ein Gewicht für jede Klasse um so das Ungleichgewicht des Datensatzes auszugleichen. Dabei wird der berechnete Verlust mit dem Parameter  $\alpha$  der jeweiligen Klasse multipliziert.<sup>118</sup> Der Parameter  $\alpha$  wird entweder manuell auf die gewünschte Gewichtung gewählt oder anhand der Verteilung der Klassen im Datensatz berechnet. Die Kreuz-Entropie Implementierung der PyTorch Bibliothek lässt auch die Nutzung der Gewichte  $\alpha$  zu. Der zweite Parameter  $\gamma$  ändert die Fokussierung der Verlustfunktion auf schwer erlernbare Beispiele.<sup>119</sup> Ziel des Faktors ist es, niedrigen Verlust noch geringer zu machen und hohen Verlust weiter zu vergrößern, wodurch der Bereich, in dem ein Beispiel einen geringen Verlust erhält, erweitert wird. Die Berechnung einer beliebigen eigen-implementierten Metrik  $m$  des Verlustes einer Vorhersage  $p$  gegenüber einer Maske  $t$  gemittelt über die Klassen  $C$  wird im Folgenden mathematisch dargestellt:

$$\begin{aligned} loss_c &= 1 - m_c \\ loss_\alpha &= \frac{1}{|C|} \sum_{c \in C} \alpha loss_c \\ loss_\gamma &= loss_\alpha^\gamma \end{aligned}$$

Da die IoU und der Dice-Koeffizient nicht differenzierbar sind, wird für beide in der Literatur eine differenzierbare Annäherung genutzt:<sup>120</sup>

---

<sup>118</sup>Vgl. Panchapagesan u. a. 2016, S. 761

<sup>119</sup>Vgl. Lin u. a. 2017, S. 321

<sup>120</sup>Vgl. Eelbode u. a. 2020, S. 3681

$$\begin{aligned}
 intersection &= \sum_{h \in H, w \in W} p_{h,w} \cdot t_{h,w} \\
 union &= \sum_{h \in H, w \in W} p_{h,w} t_{h,w} - p_{h,w} \cdot t_{h,w} \\
 IoU_{loss} &= \frac{intersection + 1}{union + 1} \\
 sum &= \sum_{h \in H, w \in W} p_{h,w} + t_{h,w} \\
 Dice_{loss} &= \frac{2 \cdot intersection + 1}{sum + 1}
 \end{aligned}$$

Die dritte implementierte Verlustfunktion ist der Tversky Verlust. Dieser wurde erstmals von Salehi/Erdogmus/Gholipour 2017 eingesetzt und basiert auf dem Tversky-Ähnlichkeitsindex von Tversky 1977. Dabei wurde neben  $\alpha$  und  $\gamma$  ein dritter Parameter  $\beta$  zur Anpassung des Verlustes in Bezug auf die positive Klasse hinzugefügt.<sup>121</sup> Anzumerken sei, dass im Fall  $\beta = 0.5$  der Tversky-Index mit dem Dice-Koeffizienten und somit auch der F1-Genauigkeit übereinstimmt. Daher wurden indikativ zwei verschiedene  $\beta$  in den Experimenten genutzt, welche nah an 0, bzw. 1 liegen:  $\beta = 0.1$  und  $\beta = 0.9$ . Der Tversky Verlust wurde wie folgt berechnet:

$$\begin{aligned}
 TP &= \sum_{h \in H, w \in W} p_{h,w} \cdot t_{h,w} \\
 FP &= \sum_{h \in H, w \in W} p_{h,w} \cdot (1 - t_{h,w}) \\
 FN &= \sum_{h \in H, w \in W} (1 - p_{h,w}) \cdot t_{h,w} \\
 Tverskyloss &= \frac{TP + 1}{TP + (1 - \beta) \cdot FP + \beta \cdot FN + 1}
 \end{aligned}$$

Die Abbildung 20 zeigt die Ergebnisse des Experiments. Dabei wird die Größe des Verlusts gegenüber dem Anteil an positiven Beispielen, also Pixeln der Klasse 0, in der Maske verglichen. Die Verluste sind nach genutzter Methode in Graphen sortiert, also ob die Klassen gemittelt wurde, ob ein Gewicht  $\alpha$  angewandt wurde oder welcher Fokusparameter  $\gamma$  gewählt wurde. Alle Graphen zeigen die vier Verlustfunktionen IoU, Dice, Tversky  $\beta = 0.1$  und Tversky  $\beta = 0.9$ . Für die beiden Graphen der gemittelten Klassen links wurde zum Vergleich auch der Kreuz-Entropie-Verlust abgebildet, bei den ungewichteten davon zusätzlich auch der Dice-Kreuz-Entropie-Verlust. Allgemein sagt der Verlauf einer Funktion aus, wie stark sich der Verlust über eine größer werdende Anzahl an richtig klassifizierten Pixel verhält. In diesem Kontext wird nach einer Verlustfunktion gesucht, welche für eine geringe Anzahl einen sehr hohen Verlust aufbringt und für eine mittlere

---

<sup>121</sup>Vgl. auch im Folgenden Salehi/Erdogmus/Gholipour 2017, S. 382 f.

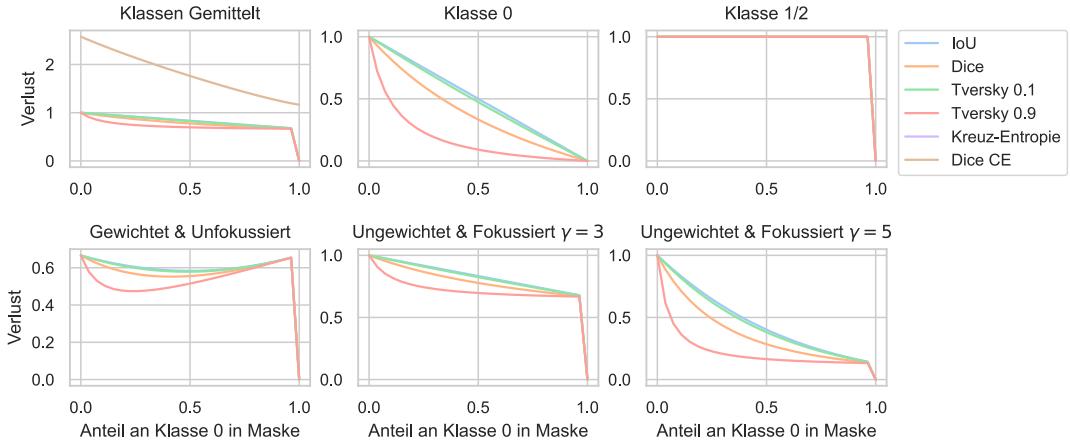


Fig. 20: Ergebnisse geordnet nach genutzter Methode. Alle Verlustfunktionen der ersten Reihe rechnen ungewichtet und unfokussiert, also mit  $\alpha = 1$  und  $\gamma = 1$ . Von links nach rechts: gemittelt über alle Klassen, binäre Berechnung der über-repräsentierten Klasse 0, binäre Berechnung der über-repräsentierten Klasse 0. Zweite Reihe mittelt über alle Klassen. Von links nach rechts: gewichtet nach Anteilen in der Grund-Wahrheit, fokussiert mit  $\gamma = 3$  und fokussiert mit  $\gamma = 5$ . (Verlustfunktionen: Ergebnisse des Experiments)

sowie hohe Anzahl einen kleinen Verlust aufbringt. Wichtig ist jedoch, dass es auch einen deutlichen Unterschied des Verlusts zwischen mittleren und hohen Anzahlen an richtig klassifizierten Pixel gibt, da sonst das Modell auf ein mittel wertiges Ziel trainiert wird. Es soll also einen Fokus auf falsch-klassifizierte Pixel geben, ohne dabei die Unterscheidung zwischen mittleren und hohen Anzahlen richtig klassifizierter Pixel zu verlieren.

Der ungewichtete Kreuz-Entropie-Verlust verhält sich wie der IoU Verlust gegenüber den anderen Verlusten linear, ist jedoch größer für starke Abweichungen der Vorhersage gegenüber der Maske und geringer für schwache Abweichungen als der IoU und ähnliche Verluste. Der gewichtete Kreuz-Entropie-Verlust ist der einzige gewichtete Verlust, welcher für steigende Genauigkeit des Modells kontinuierlich sinkt, alle anderen untersuchten gewichteten Verlustfunktionen weisen lokale Minima auf. Lokale Minima würden ein Training verhindern, da der Fokus des Verlusts sowohl auf falsch, als auch auf richtig klassifizierte Bilder legen würde und somit auf ein Gleichgewicht beider trainiert werden würde. Die zwei oberen rechten Graphen zeigen den Anteil der jeweiligen Klassen auf den gemittelten Verlust: Da Klassen 1 und 2 niemals vom Modell vorhergesagt werden ist entsprechend jeder Verlust 1, also 100%, die Verluste nur bezogen auf Klasse 0 zeigen deutlicher das Verhalten der Verlustfunktionen. Der Tversky Verlust mit  $\beta = 0,9$  zeigt die stärkste Krümmung, ohne dabei den Verlust bei mittleren Anzahlen richtig klassifizierter Pixel zu niedrig anzusetzen. Alle anderen Verlustfunktionen verhalten sich sehr nah an dem linearem IoU Verlust, also ein Verlust ohne Fokussierung oder Ähnlichem. Die untere Reihe zeigt den Einfluss des Faktors  $\gamma$  auf den Verlust. Bei der Betrachtung der Graphen der unteren Reihe fällt dabei auf, dass die Krümmung der Kurven mit zunehmendem  $\gamma$  stärker wird, somit der Fokus wie gewünscht auf falsch-klassifizierte Pixel gelegt wird. Bei sehr hohem  $\gamma$  kann der Verlust jedoch nicht mehr gut zwischen mittelmäßig und guten Klassifizierungen unterscheiden, wodurch

das Modell möglicherweise gehindert werden kann, weiter als mittelmäßige Klassifizierungen zu trainieren.

**Resultierend können drei Verlustfunktionen als individuell interessant herausgefiltert werden:** (1) der gewichtete Kreuz-Entropie-Verlust, (2) der Tversky Verlust mit  $\beta = 0.9$  und der IoU Verlust mit  $gamma = 5$ , also einem sehr großem Gamma. Der gewichtete Kreuz-Entropie-Verlust ist der einzige gewichtete Verlust ohne Minimum, somit der einzige funktionierende Verlust welcher die Gewichtungen der Klassen nutzen kann. Weiterhin basiert er auf dem normalen Kreuz-Entropie-Verlust, welcher sehr häufig in der Literatur zu finden ist und entsprechend gute Ergebnisse liefert. Der Tversky Verlust mit  $\beta = 0.9$  hingegen ist aufgrund seiner starken Krümmung interessant und vielversprechend. Der IoU Verlust mit  $gamma = 5$  verhält sich weiterhin quasi linear, reduziert aber den Verlust, wenn eine einzige Klasse einen großen Verlust hat. Somit kann er falsch-klassifizierte Pixel einer Klasse ignorieren, bzw. vernachlässigen.

### 3.4.2 Anwendung ausgewählter Verlustfunktionen

Mit diesen drei Verlustfunktionen *gewichteter Kreuz-Entropie*, *Tversky*  $\beta = 0,9$  und *fokussierter IoU* wurde das U-Net auf die gleiche Art trainiert wie das Vergleichsmodell aus Kapitel 3.3.2. Entsprechend können die Resultate des neuen Trainings direkt mit dem Vergleichsmodell verglichen werden. Tabelle 5 zeigt die Einstellungen des Versuchsaufbaus, die einzige Änderung zu den Einstellungen des Vergleichs-Experiments ist die Anzahl an Wiederholungen und natürlich die genutzte Verlustfunktion. Die Anzahl an Wiederholungen wurde reduziert um Zeit- und Rechenressourcen zu sparen.

<b>Modell Architektur</b>	U-Net
<b>Encoder</b>	ResNet50
<b>Aktivierungs-Funktion</b>	Sigmoid
<b>Normierungs-Methode</b>	SMP
<b>Modalitäten</b>	RGB [B4, B3, B2]
<b>Verlustfunktion</b>	- Gewichtete Kreuz-Entropie - Tversky ( $\beta = 0,9$ ) - IoU ( $\gamma = 5$ )
<b>Lernrate</b>	5e-4
<b>Batch Größe</b>	16
<b>Wiederholungen</b>	5

Tab. 5: Parameter-Einstellungen des Versuchsaufbaus zur Ermittlung der geeigneten Verlustfunktion. (Training-Parameter: Verlustfunktionen)

Abbildung 21 zeigt den Vergleich der erfassten Metriken für die Trainings mit verschiedenen Verlustfunktionen. Es zeigt sich über fast alle erfassten Metriken eine Verbesserung der drei neuen Verlustfunktionen gegenüber dem Vergleichsmodell. Auch ist sichtbar, dass der gewichtete Kreuz-Entropie-Verlust und der Tversky Verlust im Vergleich zu den anderen beiden Verlustfunktionen

besser abschneiden. Der fokussierte IoU Verlust hingegen weist eine sehr große Standardabweichung bei vielen Metriken auf woraus sich schließen lässt, dass mit diesem Verlust trainierte Modelle nicht sehr robust sind. Die Modelle des gewichteten Kreuz-Entropie-Verlusts weisen um 2,14% mIoU, 1,27% Genauigkeit, 3,14% IoU der Klasse *Nicht-Wasser* und 2,61% IoU der Klasse *Wasser* höhere Metriken als die Modelle des Tversky Verlusts auf. Jedoch ist die Streuung der Metriken der Modelle des gewichteten Kreuz-Entropie-Verlusts wesentlich höher, was auf größere Zufalls-Abhängigkeit der Trainingsdaten und somit eine vermutlich schlechtere Generalisierbarkeit der Modelle schließen lässt. Weiterhin zeigen die Modelle des Tversky Verlusts mit einer 2,04% höheren Präzision und einer 3,89% höheren Sensitivität verlässlichere Ergebnisse liefern als die Modelle des gewichteten Kreuz-Entropie-Verlusts.

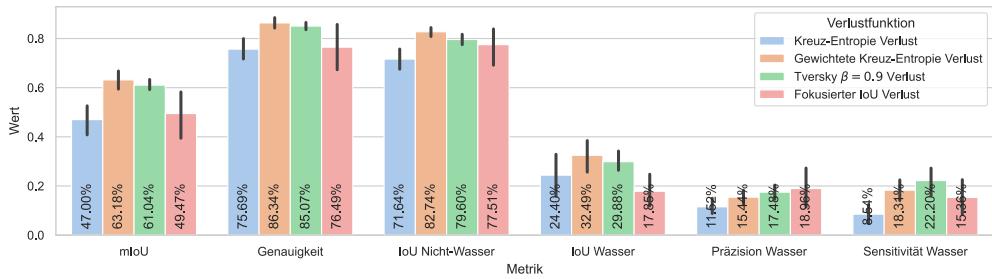


Fig. 21: Erfasste Metriken des Trainings mit den Verlustfunktionen Kreuz-Entropie, gewichtete Kreuz-Entropie, Tversky und fokussierter IoU. (Verlustfunktionen: Ergebnisse des Trainings)

Dieser Zustand gibt sich vor allem in der qualitativen Analyse der Modelle erkennen: Abbildung 32 des Anhangs 2 zeigt ähnlich wie Abbildung 18 vorhergesagte Masken zu ausgewählten Test-Beispielen. Das Modell, welches mit dem Tversky Verlust trainiert wurde, ist deutlich besser dazu in der Lage, die Grenze zwischen der *Wasser* und *Nicht-Wasser* Klasse zu zeichnen. Das Modell des Tversky Verlusts scheint jedoch einen Bias gegenüber der Klasse *Invalid* aufzuweisen, da deutlich mehr Pixel als solche klassifiziert werden. Trotzdem scheint das Modell des Tversky Verlusts auch dünne Flüsse auf den Bildern zu erkennen.

**Aufgrund der höheren Verlässlichkeit bei der Erkennung der Klasse *Wasser* wird in dieser Arbeit der Tversky Verlust den anderen Verlustfunktionen vorgezogen.** Es ist jedoch anzumerken, dass auch der gewichtete Kreuz-Entropie-Verlust vergleichbar gute Ergebnisse liefert.

## 3.5 Einfluss der Verwendung verschiedener Modalitäten

Elektromagnetische Strahlen verschiedener Wellenlängen werden von unterschiedlichen Materialien unterschiedlich stark reflektiert. Da Satelliten diese elektromagnetische Strahlung messen, um daraus Satellitenbilder zu erzeugen, kann man sich dies zunutzen machen, um *Wasser* von *Nicht-Wasser* Pixeln zu unterscheiden. Daher wird im Folgenden eine Spektroskopie mit

dem Sen1Floods11 Datensatz durchgeführt, um Wellenlängen herauszufinden, welche von *Wasser* von *Nicht-Wasser* besonders unterschiedlich stark reflektiert werden. Mit den Daten dieser Wellenlängen werden darauf Modelle trainiert um die Auswirkungen der Wahl von verschiedenen Modalitäten auf neuronale Netze zu untersuchen.

### 3.5.1 Spektroskopische Betrachtung des Datensatzes

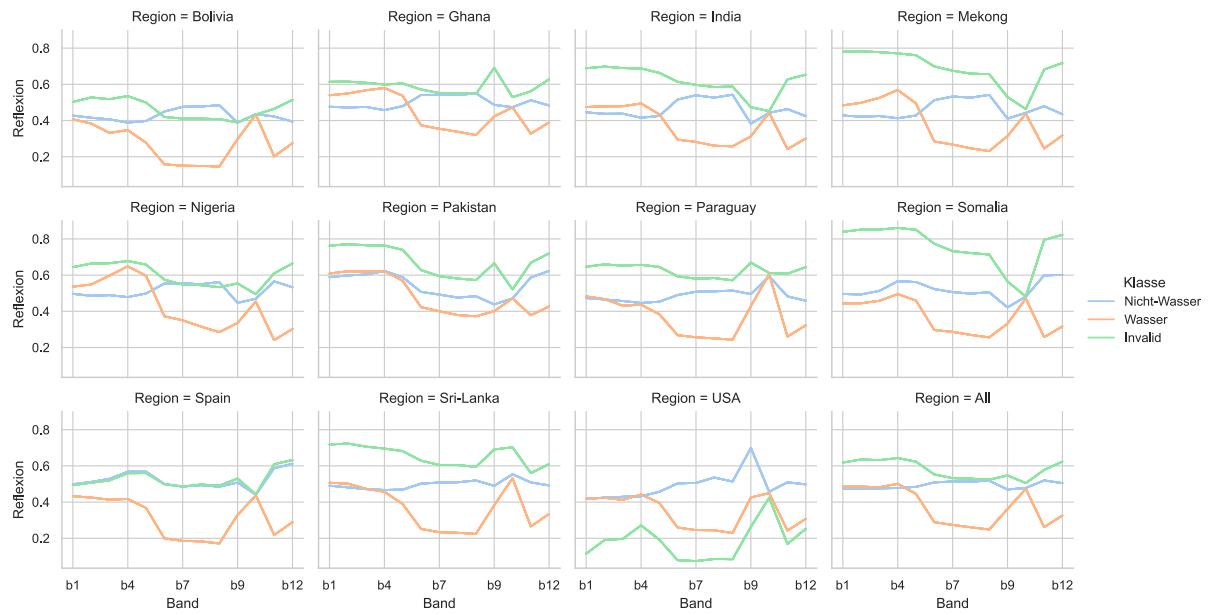


Fig. 22: Spektroskopie der drei Klassen des Sen1Flood11 Datensatzes. Aufgeteilt über alle 11 Regionen sowie Zusammenfassung aller Regionen. (Spektroskopie des Sen1Flood11 Datensatzes)

Zur Durchführung der Spektroskopie wurden alle optischen Daten des Datensatzes in ihre angehörige Region unterteilt und mit der 3-Sigma Normalisierung normiert. Darauf wurde für jede Region der Durchschnitt jedes Bandes für alle drei Klassen *Wasser*, *Nicht-Wasser* und *Invalid* berechnet. Alle Durchschnitte der Regionen wurden ebenfalls nochmals gemittelt um eine globale, von den Regionen unabhängige Spektroskopie zu erhalten. Die Ergebnisse sind in Abbildung 22 abgebildet. Es fällt auf, dass vor allem bei den Bändern im NIR-Bereich (B6-B9) ein großer Unterschied zwischen den Klassen *Wasser* und *Nicht-Wasser* liegt. Ein ähnlich großer Unterschied ist im SWIR-Bereich (B11, B12) zu erkennen. *Invalid* Pixel scheinen sich vor allem auf Bändern des sichtbaren Bereichs (B2-B4) von den anderen Klassen abzuheben. **Zusammenfassend scheinen die Bänder im NIR und SWIR-Bereichs in Verbindung mit Bändern des sichtbaren Bereichs besonders vielversprechend um die drei Klassen *Nicht-Wasser*, *Wasser* und *Invalid* voneinander zu unterscheiden.**

### 3.5.2 Training des U-Net mit ausgewählten Kombinationen von Modalitäten

Aufgrund des steigenden Rechenaufwands des Modells mit steigender Modalitäten-Anzahl ist für das finale Modell nicht sinnvoll alle optischen und SAR-Bänder in das Modell einzuspeisen. Basierend auf den Erkenntnissen der Spektroskopie sowie gängiger Praxis der Literatur wurden folgende 5 Kombination an Bändern zur weiteren Analyse durch Anwendung im Training ausgewählt. (1) Die erste Kombination *Infrared* besteht aus den Bändern B6, B7, B8 und B8a, welche alle Bänder im NIR-Bereich sind und laut Spektroskopie besonders gut zwischen *Wasser* und *Nicht-Wasser* unterscheiden können. (2) Die *Land/Wasser* Kombination der Bänder B8, B11 und B4 wurde schon bei der Entwicklung der Annotationen des Datensatzes durch Bonafilia u. a. 2020 angewandt und schließt mit B8 und B11 zwei Bänder welche *Wasser* und *Nicht-Wasser* gut unterscheiden sowie mit B4 eines welches die *Invalid* von den anderen unterscheidet mit ein. (3) Die Kombination *Indexe* aus den berechneten Indexen NDVI, NDWI und AWEI soll durch die bereits sehr kontrastreiche Hervorhebung der Klasse *Wasser* dem Modell helfen, diese Klasse direkt besser zu erkennen. (4) Die ausgewählten Bänder B8, B11 und B4 werden zusammen mit den SAR-Daten und dem AWEI zusammen in das Modell gegeben und ergeben zusammen die *gemischte* Kombination, um gegenseitige Schwächen auszugleichen, ohne dabei zu viele Modalitäten zu nutzen. (5) Theoretisch ist die Nutzung aller Bänder die beste Kombination, da hier dem Modell die meisten Informationen zur Verfügung stehen. Daher dient diese Kombination *Alle* als Vergleichswert zu den anderen Kombinationen, auch wenn sie zur schlussendlichen Auswahl nicht infrage kommt. Der Versuchsaufbau folgt dem gleichen Muster wie die Versuche zur Verlustfunktion und dem Vergleichsmodell. Tabelle 6 zeigt die genutzten Parameter des Versuchs. Es wurde bereits die in der vorherigen Sektion gefundene Verlustfunktion eingesetzt, damit das Training direkt auf den richtigen Verlust optimiert.

<b>Modell Architektur</b>	U-Net
<b>Encoder</b>	ResNet50
<b>Aktivierungs-Funktion</b>	Sigmoid
<b>Normierungs-Methode</b>	3-Sigma Normalisierung
<b>Modalitäten</b>	<ul style="list-style-type: none"> <li>- NIR [B6, B7, B8, B8a]</li> <li>- Land/Wasser [B8, B11, B4]</li> <li>- Indexe [NDVI, NDWI, AWEI]</li> <li>- Gemischt [AWEI, VV, VH, B8, B11, B4]</li> <li>- Alle</li> </ul>
<b>Verlustfunktion</b>	Tversky ( $\beta = 0,9$ )
<b>Lernrate</b>	5e-4
<b>Batch Größe</b>	16
<b>Wiederholungen</b>	5

Tab. 6: Parameter-Einstellungen des Trainings/Versuchsaufbaus zur Ermittlung einer geeigneten Kombination an Modalitäten. (Training-Parameter: Modalitäten)

Die quantitativen Ergebnisse sind in Abbildung 23 zusammengefasst. Wie zu erwarten ist die Kombination *Alle* bestehend aus allen Bändern in den Metriken zur Bestimmung der Effektivität des Modells mit 69,93% mIoU, 60,73% *Wasser*-IoU, 82,63% *Nicht-Wasser*-IoU und 86,64%

Genauigkeit den anderen Kombinationen überlegen. Ebenso ist die Standardabweichung über alle Metriken hinweg vergleichsweise gering, was auf eine gute Stabilität und Generalisierbarkeit der Modelle hinweist. Anders hingegen weisen die drei fokussierten Kombinationen *NIR*, *Land/Wasser* und *Indexe* relativ hohe Standardabweichungen auf und konnten auch schlechter zwischen den Klassen unterscheiden. Von diesen drei Kombinationen konnte die *Land/Wasser* Kombination sehr gut zwischen mit einem IoU von 56,81 % am besten zwischen *Wasser* und anderen Klassen unterscheiden. Die *gemischte* Kombination konnte um 3,39% IoU der Klasse *Wasser* besser abschneiden als die Kombination *Land/Wasser*. Ebenso sorgt das Training mit der *gemischten* Kombination für ähnlich stabile und generalisierte Modelle wie das Training mit allen Bändern. Lediglich bei der Vorhersage der *Nicht-Wasser* Klasse konnten andere Kombinationen besser abschneiden. Weiterhin zeigt sich, dass die Verwendung weiterer Modalitäten als die sichtbaren Bänder eine deutliche Verbesserung der Modelle bewirken. Modelle der Kombinationen *NIR*, *Land/Wasser* und *Indexe* benötigen ca. 42,5 Giga-MACs für die Segmentierung eines 512x512 Bilder, die *gemischte* Kombination 43 Giga-MACs und die Nutzung aller Bänder 45 Giga-MACs. Die Anzahl an trainierbaren Parametern ändert sich nicht viel.

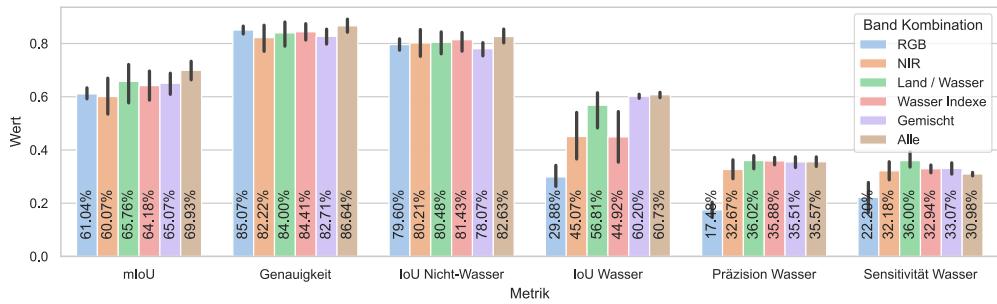


Fig. 23: Erfasste Metriken der Modelle auf den Test-Datensatz nach Training mit unterschiedlichen Modalitäten. (Modalitäten: Ergebnisse des Trainings)

Abbildung 31 des Anhang 2 zeigt Vorhersagen der Modelle, um diese qualitativ zu bewerten. Das Modell, welches die NIR Bänder nutzt, kennzeichnet Wasserstellen eher grob als die genauen Ränder zu erfassen. So sind auf der vorhergesagten Maske des zweiten Beispiels vom NIR Modell nur wenige Mittelinseln sichtbar. Betrachtet man lediglich die Klasse *Wasser*, so unterscheiden sich die restlichen Modelle nicht sonderlich. Die Modelle der Kombinationen *gemischte* und *Alle* erkennen bei dem Beispiel mit dem großen überschwemmtem Gebiet mehr kleinere Mittelinseln als der Rest. Jedoch ist auffällig, dass alle Modelle entweder dazu tendieren die Klasse *Wasser* oder die Klasse *Invalid* vorherzusagen. Ähnlich wie bei der qualitativen Analyse der Modelle der Verlustfunktion kann hier ein Bias der Modelle aufgezeigt werden, welcher entweder für die Klasse *Wasser* oder *Invalid* ist. Vermutlich entstammt dieser Bias aus der unzureichenden Qualität der annotierten Masken. Das Ziel von Bonaflia u. a. 2020 bei der Annotation der Daten war es nicht Wolken richtig zu klassifizieren, sondern den Fokus auf die Klasse *Wasser* zu legen. Die Klasse *Invalid* existiert also, um die Qualität der *Wasser*-Annotationen zu gewährleisten und wird daher aus deren Perspektive in diesem Kontext falsch verwendet. Eine Alternative wäre es, die *Invaliden* Pixel während des Trainings und bei der Evaluierung der Modelle zu ignorieren. So

würde ein Modell – ähnlich wie das lineare Vergleichsmodell – lediglich zwischen zwei Klassen unterscheiden können.

**Aus der quantitativen Analyse geht hervor, dass die *gemischte* Kombination abgesehen von der *All* Kombination die vielversprechendste bezüglich dem Erkennen von *Wasser*, jedoch auch die rechen-aufwendigste der untersuchten Kombinationen ist.** Der Unterschied des Rechenaufwands ist jedoch so gering, dass er vernachlässigt werden kann. Weiterhin können aus der qualitativen Analyse aufgrund des Bias der Modelle keine Schlüsse bezüglich der am besten geeigneten Kombination gezogen werden. Es konnte dafür festgestellt werden, dass die Berücksichtigung der Klasse *Invalid* zu schlechteren Ergebnissen führen kann.

## 3.6 Optimierung eines neuronalen Modells

Die Python Bibliothek SMP bietet neben dem U-Net noch weitere vor-implementierte Modell-Architekturen, darunter U-Net++, FPN und MaNet sowie eine große Anzahl an vortrainierten Encodern und 5 Aktivierungs-Funktionen. Im Zuge dieser Arbeit wurde die vermutlich beste Kombination dieser drei Hyperparameter in drei aufeinander-folgenden Versuchen durch Rastersuchen gefunden. Da dies nicht der übliche Prozess zur Findung solcher Hyperparameter-Konfigurationen ist, wurde zum Vergleich auch eine große Hyperparameter-Rastersuche durchgeführt. Bei dieser Hyperparameter-Rastersuche wurde aufgrund der Erkenntnisse aus 3.5.2 die Klasse *Invalid* beim Training und bei der Evaluation ignoriert.

### 3.6.1 Sequentielle Ermittlung der Architektur

Der Versuchsaufbau der drei sequentiellen Versuche zur Bestimmung der Architektur gleicht dem der Versuche zum Vergleichsmodell, Verlustfunktion und Band-Kombination, jedoch mit 3 Trainings-Wiederholungen pro Kombination anstelle von 5 oder 10. Durch die geringe Anzahl an Wiederholungen schwindet auch die Aussagefähigkeit der Versuche, war jedoch aufgrund des hohen Trainingsaufwands nötig und reicht dennoch aus, um Tendenzen zu erkennen. Weiterhin musste aufgrund von Begrenzungen des Arbeitsspeichers die Batch-Größe auf 8 Beispiele heruntergesetzt werden. Im ersten Versuch wurden die vier Modell-Architekturen U-Net, U-Net++, FPN und MaNet untersucht und miteinander verglichen. Die Auswahl dieser Architekturen geschah willkürlich, da nur eine geringe Anzahl aufgrund des hohen Trainingsaufwands möglich war. Die beste Architektur wurde im zweiten Versuch dazu verwendet, die 6 Encoder *ResNet 50*, *ResNext 50*, *DenseNet 161*, *EfficientNet B5*, *DPN 68* und *VGG 19*. Auch diese Auswahl geschah willkürlich, jedoch wurde darauf geachtet, dass die Anzahl an Parameter der Encoder zwischen 20 und 30 Millionen Parameter liegen, wodurch eine bessere Vergleichbarkeit der Encoder hergestellt wird. Der dritte Versuch wiederum nutzt sowohl die beste Architektur als auch den besten Encoder um eine optimale Aktivierungs-Funktion zu finden. SMP implementierte die 5 Aktivierungs-Funktionen *Sigmoid*, *Softmax*, *LogSoftmax*, *TanH* und *Identity*, also eine lineare

Aktivierungs-Funktion. Da die genutzte Tversky Verlustfunktion nicht an die Rückgabe-Intervalle der LogSoftmax ( $[-inf, 0)$ ) und der Identity ( $[-inf, inf]$ ) Aktivierungs-Funktionen angepasst ist wurden lediglich die Sigmoid, Softmax und TanH Aktivierungs-Funktion untersucht und miteinander verglichen. Tabelle 7 zeigt die genutzten Parameter der drei Versuche.

<b>Modell Architektur</b>	<ul style="list-style-type: none"> <li>- U-Net</li> <li>- U-Net++</li> <li>- MaNet</li> <li>- FPN</li> </ul>
<b>Encoder</b>	<ul style="list-style-type: none"> <li>- ResNet 50 (23M Parameter)</li> <li>- ResNext 50 (22M)</li> <li>- DenseNet 161 (26M)</li> <li>- EfficientNet B5 (28M)</li> <li>- DPN 68 (11M)</li> <li>- VGG 19 (20M)</li> </ul>
<b>Aktivierungs-Funktion</b>	<ul style="list-style-type: none"> <li>- Sigmoid</li> <li>- Softmax</li> <li>- TanH</li> </ul>
<b>Normierungs-Methode</b>	3-Sigma Normalisierung
<b>Modalitäten</b>	Gemischt [AWEI, VV, VH, B8, B11, B4]
<b>Verlustfunktion</b>	Tversky ( $\beta = 0, 9$ )
<b>Lernrate</b>	5e-4
<b>Batch Größe</b>	8
<b>Wiederholungen</b>	3

Tab. 7: Parameter-Einstellungen des Versuchsaufbaus zur Ermittlung der geeigneten Architektur, Encoders und Aktivierungs-Funktion in einem sequentiellen Verfahren. (Training-Parameter: Architektur)

Abbildung 24 zeigt die Ergebnisse der drei Versuche. Die Ergebnisse des ersten Versuchs zeigen, dass die U-Net++ und die MaNet Architekturen mit einem mIoU von durchschnittlich 3 % Verbesserung gegenüber dem normalem U-Net deutlich besser abschneiden, Modelle beider Architekturen performen ähnlich gut. Die Standardabweichung des IoU der Klassen *Wasser* und *Nicht-Wasser* sind bei den Modellen der U-Net++ Architektur um 1% geringer als die der MaNet Architektur. Dies ist jedoch wenig aussagekräftig, da lediglich drei Wiederholungen des Trainings durchgeführt wurden und somit kleine Abweichungen wenig aussagen. Die Modelle der FPN Architektur wiesen durchschnittlich einen halb so großen mIoU auf und performten vor allem bei der Vorhersage der unterrepräsentierten Klasse *Wasser* vergleichsweise schlecht. Auch zeigt die hohe Standardabweichung bei nahezu allen aufgenommenen Metriken die geringe Robustheit der FPN Architektur auf. Jedoch könnte auch dies dem Zufall geschuldet sein. Eine qualitative Analyse wurde aufgrund des Bias im Modell zwischen den beiden unterrepräsentierten Klassen *Wasser* und *Invalid* ausgelassen. Aufgrund der geringeren Standardabweichung der U-Net++ Architektur wurde für die weiteren Versuche diese Architektur benutzt. Allgemein ist der Versuch aufgrund des Bias in den Modellen gegen einer der unterrepräsentierten Klassen, auf-

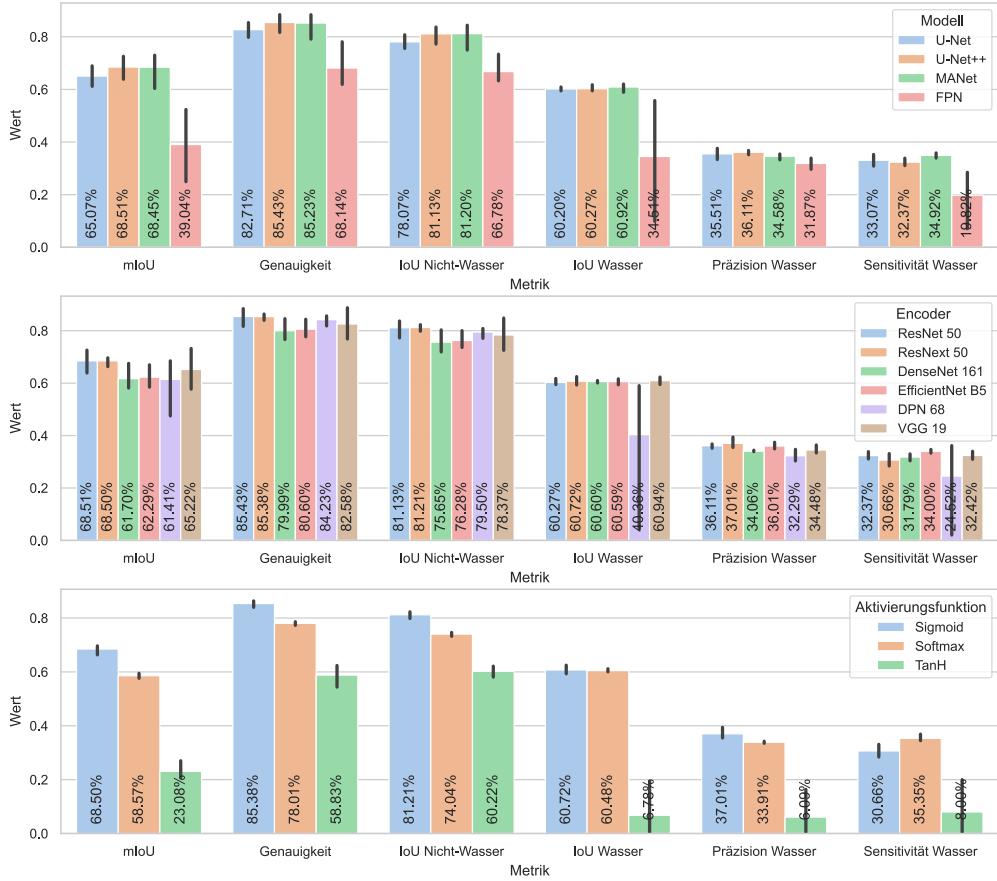


Fig. 24: Erfasste Metriken verschiedener Modelle im Vergleich. Vergleiche von oben nach unten: Decoder-Architekturen, Encoder-Architekturen, Aktivierung-Funktionen. (Architektur: Ergebnisse des Trainings)

grund der geringen Anzahl an genutzten Architekturen sowie aufgrund der geringen Anzahl an Wiederholungen des Trainings nicht aussagekräftig genug, um daraus eine optimale Architektur auswählen zu können.

Aus denselben Gründen sind die Ergebnisse des zweiten Versuchs ebenfalls nicht aussagekräftig genug. Die *ResNet* und *ResNext* Encoder trennen mit einem mIoU von 68,51% und 68,5% wesentlich besser als andere Encoder. Weiterhin weist der *ResNext* Encoder eine deutlich geringere Standardabweichung als der *ResNet* Encoder auf, weshalb dieser für den letzten Versuch genutzt wurde. Im dritten Versuch zeigt sich, dass die *Sigmoid* Aktivierungs-Funktion besser funktioniert als andere, bis auf die Sensitivität der *Wasser* Klasse schneidet die *Sigmoid* Funktion in allen Metriken besser ab. Dies könnte daran liegen, dass die vor-trainierten Gewichte des Encoders ebenfalls mit der *Sigmoid* Aktivierungs-Funktion trainiert wurden und sich somit auf diese

optimiert haben. Bei einem potenziell längerem Training könnten daher andere Aktivierungs-Funktionen besser abschneiden. Da jedoch auch die Konvergenz, also das schnelle Training des Modells relevant ist, scheint die Sigmoid Aktivierungs-Funktion eine geeignete Wahl zu sein. **Die Kombination aus dem U-Net++ Decoder, dem ResNext Encoder und der Sigmoid Aktivierungsfunktion scheint besser Wasser auf Satellitendaten zu erkennen als andere Kombinationen.**

### 3.6.2 Statistische Hyperparameter-Suche

Mit allen bisher gewonnenen Erkenntnissen wurde zusätzlich eine große Rastersuche nach Hyperparametern mit der Python Bibliothek *Optuna* durchgeführt.<sup>122</sup> Der Aufbau dieses Versuchs ist leicht anders als die bisherigen. Anders als bei den anderen Versuchen wurden hier *Invalide* Pixel ignoriert. So soll es einem Modell nicht mehr möglich sein, eine der beiden unterrepräsentierten Klassen zu bevorzugen. Weiterhin wurden auch die Lernrate und die Batch-Größe in die Hyperparametersuche mit einbezogen, um auch diese zu optimieren. Dafür wurde die Aktivierungs-Funktion auf Sigmoid festgesetzt, da in dem vergangenen Experiment Sigmoid deutlich bessere Ergebnisse lieferte als andere Funktionen. Außerdem hilft eine geringere Anzahl an variablen Hyperparametern dem Optuna-Algorithmus schneller eine optimale Konfiguration zu finden. Der Optuna-Algorithmus wurde so eingestellt, dass er den mIoU der Modelle auf den Validierungs-Datensatz maximiert. Insgesamt wurden über 40 Modelle mit verschiedenen Kombinationen an Hyperparametern trainiert. Alle möglichen Parameter sind in der Tabelle 8 abgebildet.

Abbildung 25 zeigt die Ergebnisse der Rastersuche. Wie zu sehen ist, hat die Batch-Größe keine besondere Auswirkung auf den Erfolg des Trainings, da die Werte über den gesamten Wertebereich der Batch-Größe verteilt sind. Weiterhin scheint die optimale Kombination aus den Hyperparametern U-Net, ResNet sowie einer Lernrate von  $2e - 4$  zu bestehen, das Modell mit diesen Hyperparametern konnte mIoU von 88,19% auf den Validierungs-Datensatz aufweisen. Jedoch wurde im Verlauf dieser Rastersuche aufgrund des Optuna-Optimierungsalgorithmus vermehrt mit dieser Kombination anstelle anderer Kombinationen trainiert. Somit wurden zu wenige andere Kombinationen ausprobiert, um eine sichere Aussage treffen zu können. Weiterhin ist auch fraglich, ob die Encoder-Decoder Architektur auf diese Weise ermittelt werden kann, da der Optuna-Algorithmus dazu optimiert wurde die Hyperparameter einer einzigen Architektur zu bestimmen. Die Kombination aus U-Net und ResNet findet in der Forschung jedoch sehr große Verwendung was somit mit diesen Ergebnissen übereinstimmt. **Folgernd geht aus der Rastersuche hervor, dass die Kombination bestehend aus dem U-Net Decoder, dem ResNet Encoder trainiert mit einer Lernrate von  $2e - 4$  Wasser besser als andere ausprobierte Kombinationen auf Satellitendaten erkennt.**

---

<sup>122</sup>Akiba u. a. 2019

<b>Modell Architektur</b>	<ul style="list-style-type: none"> <li>- U-Net</li> <li>- U-Net++</li> <li>- MaNet</li> <li>- FPN</li> </ul>
<b>Encoder</b>	<ul style="list-style-type: none"> <li>- ResNet 50 (23M Parameter)</li> <li>- ResNext 50 (22M)</li> <li>- DenseNet 161 (26M)</li> <li>- EfficientNet B5 (28M)</li> <li>- DPN 68 (11M)</li> <li>- VGG 19 (20M)</li> </ul>
<b>Aktivierungs-Funktion</b>	Sigmoid
<b>Normierungs-Methode</b>	3-Sigma Normalisierung
<b>Modalitäten</b>	Gemischt [AWEI, VV, VH, B8, B11, B4]
<b>Verlustfunktion</b>	Tversky ( $\beta = 0, 9$ )
<b>Lernrate</b>	(1e-5; 1e-2)
<b>Batch Größe</b>	(2; 10)
<b>Versuche</b>	100

Tab. 8: Parameter-Einstellungen des Versuchsaufbaus zur Ermittlung der geeigneten Architektur, Encoders und Aktivierungs-Funktion über eine Rastersuche. (Training-Parameter: Rastersuche)

### 3.7 Training eines Modells mit optimierten Hyperparametern

Basierend auf den Ergebnissen der vorangegangenen Versuche wurden schlussendlich zwei Modelle mit den vermuteten optimalen Konfigurationen trainiert, diese ist in Tabelle 9 dokumentiert. Für das Training wurde die Region Ghana aus dem Sen1Floods11 Datensatz entfernt und daraufhin eine zufällige Teilung der übrigen Beispiele durchgeführt, wodurch ein Trainings-Datensatz bestehend aus 86% der übrigen Beispiele und ein Validierungs-Datensatz aus den restlichen 15% entstanden ist. Die Beispiele wurden zufällig horizontal und vertikal gespiegelt sowie um 90° gedreht, um den Datensatz künstlich zu vergrößern. Weiterhin wurde die 3-Sigma Normalisierung auf alle Beispiele angewandt, um dem Modell zu helfen, die Beispiele besser zu segmentieren. Beide Modelle nutzen die Kombination der Bänder AWEI, VV, VH, B8, B11 und B4 um eine Maske zu erzeugen. Während des Trainings verarbeiteten die Modelle 8 Beispiele in einem Batch gleichzeitig, um ein Überlaufen des Arbeitsspeichers zu vermeiden. Ziel des Trainings war es, den Tversky  $\beta = 0, 9$  Verlust mit dem Adam-Optimierungsalgorithmus zu reduzieren. Die U-Net++ Architektur ergibt zusammen mit dem ResNext 101 Encoder, als einer größeren Version des ResNext 50 Encoders, und der Sigmoid Aktivierungsfunktion das genutzte Modell des ersten Trainings. Das zweite Modell besteht aus der Kombination aus U-Net, ResNet 50 und die Sigmoid Aktivierungsfunktion. Beide Encoder wurden auf den Imagenet Datensatz vor-trainiert.

Das erste Training wurde nach 6 Stunden abgebrochen, dadurch sollte dem Modell genug Zeit geben werden die Aufgabe vollständig zu erlernen. Über eine Lernrate-Suche der Python Bibliothek Pytorch-Lightning wurde  $3 - e3$  als bester geeigneter Startwert für die Lernrate des ersten

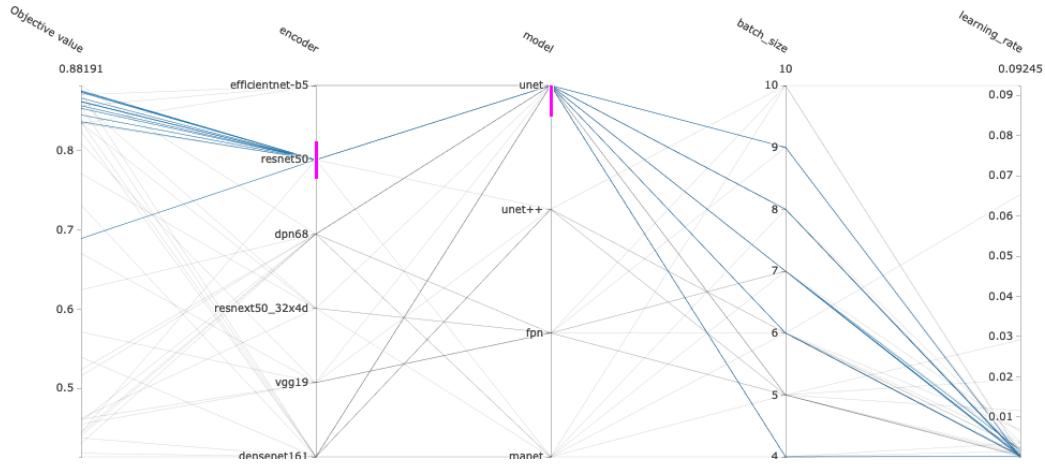


Fig. 25:  $\|$ -Koordinaten Diagramm der Rastersuche. U-Net und ResNet wurden markiert um den Erfolg dieser Kombination zu zeigen. (Architektur: Ergebnisse der Rastersuche)

Trainings gefunden.<sup>123</sup> Daher wurde die initiale Lernrate auf  $3e - 3$  gesetzt, nach 3 Epochen gemäß der Warmup-Methode um den Faktor 10 auf  $3e - 4$  verringert, weitere Verringerungen der Lernrate um den Faktor 10 wurden nach der 30., 100. und 200. Epoche durchgeführt. Da durch die Hyperparametersuche aus Kapitel 3.6.2 die geeignete Lernrate ebenfalls gefunden, wurde die Lernrate auf diesen Wert  $2e - 4$  gesetzt. Das Modell des zweiten Trainings erkennt im Gegensatz zum ersten die Klasse *Invalid* nicht. Dies markiert den wesentlichen Unterschied zwischen beiden Modellen

<b>Modell Architektur</b>	U-Net++	U-Net
<b>Encoder</b>	ResNext 101	ResNet 50
<b>Aktivierungs-Funktion</b>	Sigmoid	
<b>Normierungs-Methode</b>	3-Sigma Normalisierung	
<b>Modalitäten</b>	Gemischt [AWEI, VV, VH, B8, B11, B4]	
<b>Verlustfunktion</b>	Tversky ( $\beta = 0, 9$ )	
<b>Lernrate</b>	5e-3	2e-4
<b>Batch Größe</b>	8	
<b>Wiederholungen</b>	4	10

Tab. 9: Parameter-Einstellungen des finalen Trainings der beiden ermittelten optimierten Architekturen. Die Modelle beider Architekturen wurden in einem weiteren Punkt ungleich eingestellt: Das optimierte U-Net++ kann im Gegensatz zum optimierten U-Net neben *Wasser* und *Nicht-Wasser* auch *invalid* Pixel erkennen. (Training-Parameter: Optimierte Modelle)

Das erste Training wurde 4-mal, das zweite 10-mal durchgeführt um Ausreißer durch die zufällige Abfolge der Trainingsdaten sowie der zufälligen Initialisierung der Modell-Gewichte entgegenzutreten. Die Ergebnisse dieses Trainings werden in Kapitel 4 evaluiert und mit anderen Modellen aus der Forschung verglichen.

<sup>123</sup>Falcon u. a. 2022

## 4 Evaluierung und Vergleich des Modells mit vergleichbaren Arbeiten

In diesem Kapitel werden die Resultate des finalen Trainings genauer analysiert und bezüglich Performance, Rechenaufwand und Robustheit der Modelle bewertet. Basierend darauf wird auf den Erfolg der genutzten Methodik zur Optimierung der Modelle genauer eingegangen. Die Ergebnisse der Analyse werden daraufhin den Ergebnissen der in dieser Arbeit vorgestellten Vergleichsmodellen gegenübergestellt und darauf weiterhin mit vergleichbaren Arbeiten verglichen. Schlussendlich wird ein Ausblick auf aufkommende Forschungsfragen gegeben.

### 4.1 Qualitative Evaluation der optimierten Modelle

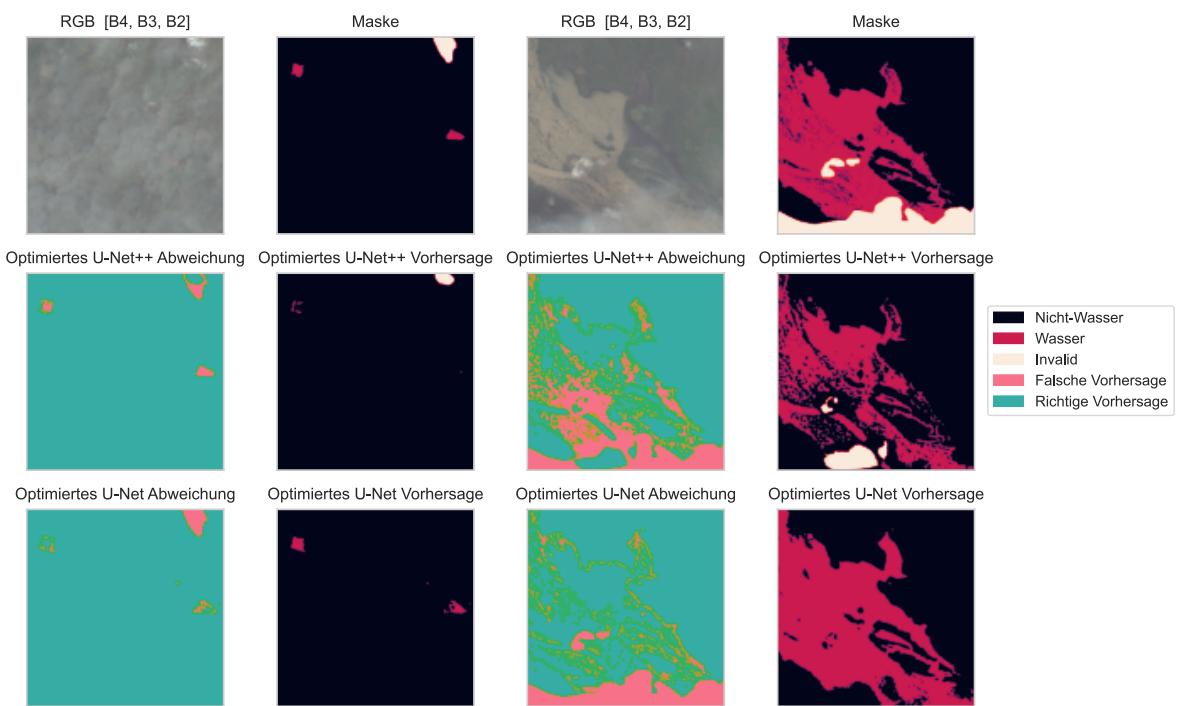


Fig. 26: Zwei Beispiele aus Ghana. Jeweils 2 Spalten gehören zu einem Beispiel. Zu jedem Beispiel ist die RGB Aufnahme sowie die annotierte Maske abgebildet. Darunter sind die Vorhersagen und Abweichungen dieser von den beiden optimierten Modellen zu sehen. Das optimierte U-Net++ in Reihe zwei und das optimierte U-Net in Reihe drei. (Qualitative Ergebnisse der optimierten Modelle)

Abbildung 26 zeigt ähnlich wie die Abbildungen vorheriger qualitativer Analysen zwei Beispiele aus dem Test-Datensatz, die dazugehörigen Annotationen sowie die berechneten Masken der beiden optimierten Modelle. Es ist zu erkennen, dass das U-Net++, welches ebenfalls die Klasse *Invalid* vorhersagt, kleine Wasserstellen so gut wie gar nicht erkennt und auch die Grenzen zwischen Land und Wasser bei einer Überschwemmung nicht deutlich kennzeichnen kann. Das

optimierte U-Net erkennt hingegen auch die kleinen Wasserstellen des ersten Beispiels sehr gut und kennzeichnet die Land-Wasser-Grenzen des zweiten Beispiels besser. Trotzdem kennzeichnete das U-Net auch kleinere Mittelinseln des zweiten Beispiels fälschlicherweise als *Wasser*.

## 4.2 Quantitative Evaluation der optimierten Modelle

Tabelle 10 zeigt die Genauigkeit, den gemittelten sowie den klassenabhängigen IoU für die Klassen *Nicht-Wasser* und *Wasser* sowie die Präzision und Sensitivität der Klasse *Wasser* auf den Test-Datensatz Ghana und den Validierungs-Datensatz aller anderen Regionen. Wie zu erwarten performen die trainierten Modelle auf den Test-Datensatz in allen erfassten Metriken schlechter als auf den Validierungs-Datensatz. So besteht zum Beispiel eine Abweichung von fast 30% zwischen der durchschnittlichen Präzision aller Modelle auf des Validierungs-Datensatzes und der des Test-Datensatzes. Da alle Beispiele aus dem Validierungs-Datensatz aus den gleichen Regionen stammen wie die für das Modell während des Trainings sichtbaren Beispiele des Trainings-Datensatzes sind die Modelle in der Lage, das erlernte Wissen über die regions-spezifischen Eigenschaften in die Verarbeitung der Validierungs-Beispiele einfließen zu lassen. Da die Beispiele der Region Ghana von dem Training ausgeschlossen wurden, konnten die Modelle diese Eigenschaften der Region Ghana nicht erlernen und können somit auch kein Vor-Wissen über diese Region in die Verarbeitung einfließen lassen. Betrachtet man die relativen Abweichungen von  $\frac{IoU_{val,water} - IoU_{test,water}}{IoU_{val,water}} = \frac{61,89\% - 51,75\%}{61,89\%} = 16,3\%$  des IoU der Klasse *Wasser* und vergleicht diese mit der relativen Abweichungen von 11,93% des IoU der Klasse *Nicht-Wasser*, so fällt auf, dass bei fehlendem Vor-Wissen über regionsspezifischen Eigenschaften besonders die Genauigkeit der Vorhersagen der unterrepräsentierten Klasse *Wasser* abnimmt. **Es ist also ersichtlich, dass die regionsspezifische Eigenschaften des Datensatzes relevant für die Vorhersage einer Wasser-Maske durch die trainierten Modelle sind.**

Metrik	Test	Validierung
<b>Genauigkeit</b>	$81,79\% \pm 2,14$	$89,28\% \pm 1,21$
<b>mIoU</b>	$61,39\% \pm 4,48$	$66,86\% \pm 2,65$
<b>IoU Nicht-Wasser</b>	$77,25\% \pm 2,13$	$87,72\% \pm 1,38$
<b>IoU Wasser</b>	$51,75\% \pm 7,24$	$61,89\% \pm 13,76$
<b>Präzision Wasser</b>	$35,42\% \pm 1,16$	$65,36\% \pm 4,57$
<b>Sensitivität Wasser</b>	$26,07\% \pm 3,5$	$54,39\% \pm 6,76$

Tab. 10: Erfasste Metriken der Vorhersagen von den optimierten U-Net++ Modellen, welche auf den Test-Datensatz der Region Ghana und respektive dem Validierungs-Datensatz der restlichen Regionen trainiert wurden. (Ergebnisse des finalen Trainings der ersten Architektur)

Die Genauigkeit der Modelle auf den Test-Datensatz ist mit  $81,79\% \pm 2,14$  deutlich besser als die eines Zufallsmodells mit 33,33%, was für eine gute Segmentierung durch die Modelle spricht. Betrachtet man nun den mIoU 61,39%±4,48 fällt auf, dass die Modelle zwar immer noch bessere

Masken berechnen als Zufallsmodelle, jedoch wesentlich schlechter als die Genauigkeit vermuten lässt. Der mIoU wird aus den IoU der drei verschiedenen Klassen berechnet, da jedoch die Klasse *Invalid* aufgrund der Aufgabenstellung der Trennung von *Wasser* von *Nicht-Wasser* nicht von Relevanz für die Beurteilung der Modelle ist, werden im Folgenden lediglich die anderen beiden Klassen betrachtet. Der IoU der Klasse *Nicht-Wasser* beträgt  $77,25\% \pm 2,13$  ist deutlich höher als der mIoU und der IoU der *Wasser* Klasse mit  $51,75\% \pm 7,24$ . Es zeigt sich daher deutlich, dass die überrepräsentierte Klasse *Nicht-Wasser* auch besser erkannt werden kann als die Klasse *Wasser*. Da die Klasse *Nicht-Wasser* aufgrund der hohen Anzahl an Beispielen deutlich mehr Einfluss auf die Genauigkeit hat, ist auch die Genauigkeit der Modelle relativ hoch. Weiterhin zeigt sich durch die Standardabweichung der IoU Metriken für beide Klassen, dass die *Nicht-Wasser* Klasse von den Modellen deutlich robuster erkannt werden kann, als die *Wasser* Klasse bei welcher die Modelle mit einer Standardabweichung von  $7,24\%$  diese eher zufällig gut erkennen können. Die Präzision der *Wasser* Klasse gibt an, wie verlässlich die erkannte Wasser-Maske der Modelle ist. In diesem Fall beträgt die Präzision der *Wasser* Klasse  $35,42\% \pm 1,16$ . Auch die Sensitivität der Modelle ist mit  $26,07\% \pm 3,5$  bezüglich der Klasse *Wasser* relativ niedrig.

Die Kombination aus der U-Net++ Architektur gekoppelt an den ResNext-101 Encoder besitzt mehr als 112 Millionen trainierbare Parameter, um diese Parameter abzuspeichern benötigt es 1,35 GB an Speicherplatz. Um mit einem solchen Modell eine Vorhersage zu tätigen sind 295 GMACs Rechenoperationen notwendig, was ungefähr 590 GFLOPs entspricht.

Metrik	Test
<b>Genauigkeit</b>	$88,25\% \pm 0,14$
<b>mIoU</b>	$84,28\% \pm 1,01$
<b>IoU Nicht-Wasser</b>	$97,77\% \pm 0,14$
<b>IoU Wasser</b>	$70,78\% \pm 1,19$
<b>Präzision Wasser</b>	$35,00\% \pm 1,16$
<b>Sensitivität Wasser</b>	$34,61\% \pm 1,82$

Tab. 11: Erfasste Metriken der Vorhersagen von den optimierten U-Net Modellen, welche auf den Test-Datensatz der Region Ghana trainiert wurden. (Ergebnisse des finalen Trainings der zweiten Architektur)

Analog zur quantitativen Analyse des optimierten U-Net++ Modells zeigt die Tabelle 11 die Ergebnisse des optimierten Trainings des U-Net Modells. Es zeigt sich, dass dieses Modell *Wasser* mit einem IoU von 70,78% und einer Standardabweichung von gerade mal  $\pm 1,19\%$  relativ gut und vom Zufall unabhängig erkennt. Das Modell trennt *Wasser* und *Nicht-Wasser* ebenfalls sehr gut wie der relative hohe mIoU mit 84,28% zeigt. Trotzdem zeigt sich mit einer Präzision von 35,00% und einer Sensitivität von 34,61% eine geringe Zuverlässigkeit der Vorhersagen, einerseits wurden wenig *Wasser* Pixel erkannt als auch viele *Nicht-Wasser* Pixel als *Wasser* gekennzeichnet. Die Kombination aus U-Net und ResNet 50 benötigt 43 GMAC Rechenoperationen, um ein Bild der Größe  $512 \times 512$  Pixel zu segmentieren. Dies entspricht ungefähr 86 GFLOPs. Da das Modell 32 Millionen trainierbare Parameter besitzt, benötigt es in etwa 400 MB an Speicherplatz.

Im direkten Vergleich der beiden Modelle zeigt sich, dass die Reduzierung zu einem binären Klassifizierungsproblem bei dem optimiertem U-Net einen größeren Erfolg des Trainings der Modelle verspricht.

### 4.3 Erfolgs der durchgeführten Experimente

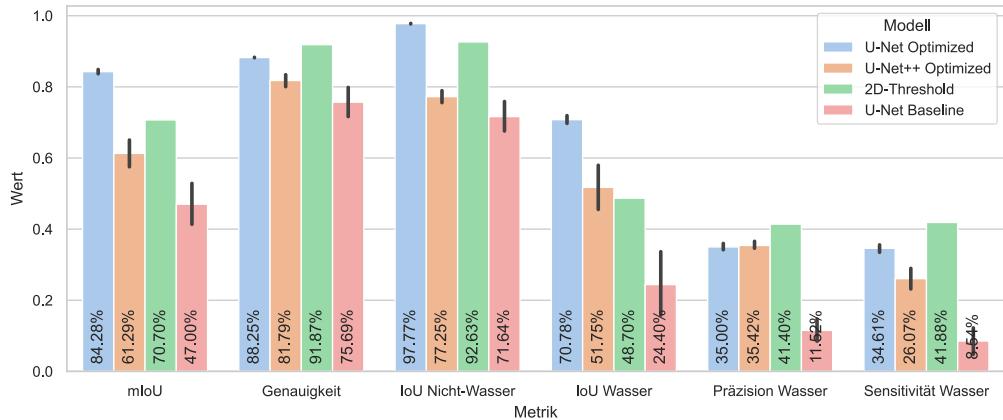


Fig. 27: Erfasste Metriken der unoptimierten Vergleichsmodelle und den optimierten Modellen auf den Test-Datensatz der Region Ghana. (Vergleich zwischen optimierten und unoptimierten Modellen)

Abbildung 27 zeigt den Vergleich zwischen den beiden Vergleichsmodellen und den optimierten Modellen. Es ist zu erkennen, dass die beiden optimierten Modelle in allen Metriken besser abschneiden als das neuronale Vergleichsmodell. Dies spricht für den Erfolg der genutzten Optimierungs-Methoden und zeigt gleichzeitig die hohe Relevanz der verschiedenen Einstellungen beim Training eines Modells. Das lineare Modell schneidet sowohl in allen Metriken besser als das neuronale Vergleichsmodell ab, als auch mit einer Genauigkeit von 88,25%, 41,4% Präzision und 41,88% Sensitivität besser als die optimierten neuronalen Modelle. Daher kann gesagt werden, dass das lineare Modell Wasser Pixel verlässlicher erkennt als die neuronalen Modelle. Vorsicht muss jedoch bei dem Vergleich des mIoU gewahrt werden, da das lineare Modell und das optimierte U-Net lediglich die *Wasser* und *Nicht-Wasser* klassifizieren kann und daher alle *Invaliden* Pixel bei der Berechnung der Metrik ignoriert werden. Es zeigt sich jedoch deutlich, dass die neuronalen Modelle besser bei der Erkennung von Wasser sind, sowohl das optimierte U-Net, als auch das optimierte U-Net++ sind mit einem IoU der Klasse *Wasser* von 70,78%, bzw. 51,75% besser als das lineare Vergleichsmodell mit 48,7%. Weiterhin ist auch das optimierte neuronale U-Net Modell mit einem IoU von 97,77% besser als das lineare Modell die Klasse *Nicht-Wasser* zu erkennen. Diese beiden Punkte spiegeln sich auch in den Ergebnissen des mIoU wider: Das optimierte U-Net kann mit einem mIoU von 84,28% gefolgt vom linearen Modell mit 70,7% mIoU am besten die beiden Klassen segmentieren. Dafür ist das lineare Modell wesentlich schneller und Ressourcen-sparender bei der Segmentierung eines Bildes aufgrund der sehr gerin-

gen Anzahl an trainierbaren Parameter. Daher kann abschließend gesagt werden, dass die angewandten Optimierungs-Methoden erfolgreich waren. Bei richtiger Einstellung der Trainingsparameter können neuronale Netze besser Wasser von Nicht-Wasser auf Satellitenbildern trennen als einfache lineare Modelle.

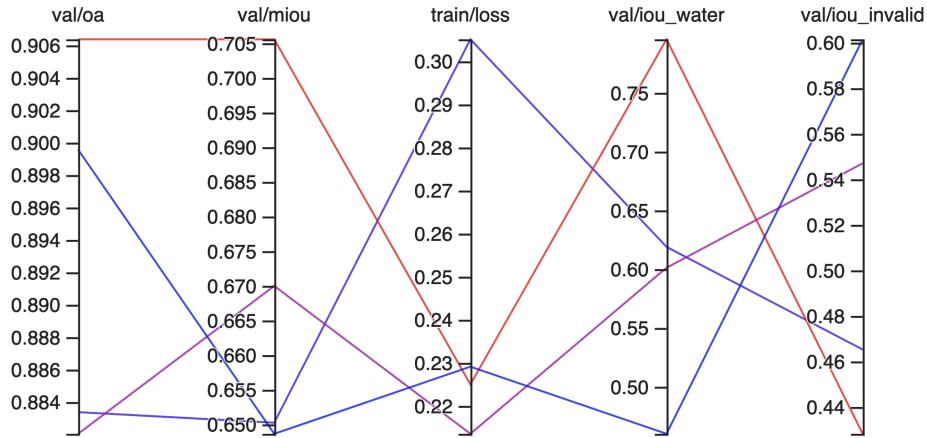


Fig. 28: Die erfassten Metriken mIoU, IoU aller Klassen und die Genauigkeit im Vergleich mit dem Tversky-Verlust, dargestellt in einem ||-Koordinaten Diagramm für mehrere Trainings. Die Metriken wurden über die Vorhersagen des optimierten U-Net++ Modells, welches auch *invalid* Pixel erkennen kann, berechnet. (Metriken mehrerer Trainings im Vergleich mit dem Verlust)

Abbildung 28 zeigt eine sogenannte parallele Koordinaten-Grafik welche erfasste Metriken der Validierungsphase des Trainings vom optimierten U-Net++ mit dem Tversky-Verlust vergleicht. Die Trainings, bei denen der Verlust relativ gering war, wie zum Beispiel das Training repräsentiert durch die rote Linie, sind die Ergebnisse der erfassten Metriken besser als bei Trainings mit hohem Validierungs-Verlust. Dies spricht für das Funktionieren der genutzten Verlustfunktion, welche in Kapitel 3.4 ermittelt wurde. Weiterhin wird der Bias eines Trainings bezüglich einer der unterrepräsentierten Klassen sichtbar: Modelle, welche gut die Klasse *Wasser* erkennen und somit einen hohen IoU aufweisen, erkennen weniger gut die Klasse *Invalid* und vice versa. **Der Vergleich zwischen dem unoptimierten U-Net und dem optimierten U-Net zeigt weiterhin, dass die Nutzung verschiedener Modalitäten einen größeren Erfolg bei der Erkennung von *Wasser* verspricht.**

#### 4.4 Vergleich mit ähnlichen Arbeiten aus der Forschung

Auf Abbildung 29 wurden die Ergebnisse der Arbeiten von Bonafilia u. a. 2020, Mateo-Garcia u. a. 2021, Rahnemoonfar u. a. 2020 mit denen dieser Arbeit verglichen. Nicht alle Arbeiten erfassten die gleichen Metriken, weshalb manche fehlen. Weiterhin muss angemerkt werden, dass alle Arbeiten unterschiedliche Ziele verfolgen und entsprechend die Rahmenbedingungen anders sind. So nutzen zum Beispiel Rahnemoonfar u. a. 2020 und Mateo-Garcia u. a. 2021 nicht den

Sen1Floods11 Datensatz zum Training und zur Evaluierung ihrer Ergebnisse. Anders als diese Arbeit führten Bonafilia u. a. 2020 keine Optimierung ihres Modells durch.

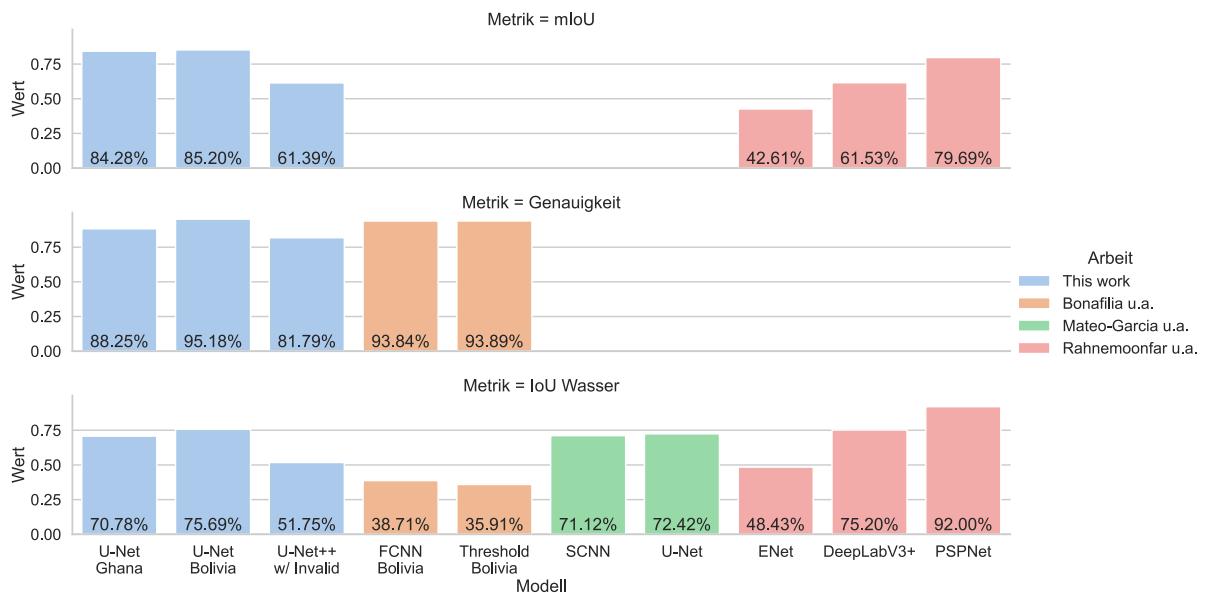


Fig. 29: Die erfassten Metriken mIoU, IoU der Klasse *Wasser* und die Genauigkeit für Modelle der Arbeiten von Bonafilia u. a. 2020, Mateo-Garcia u. a. 2021, Rahnemoonfar u. a. 2020 und dieser Arbeit. Da nicht alle Arbeiten die selben Metriken erfassen fehlen Einträge. Weiterhin können Präzision und Sensitivität aufgrund fehlender Daten nicht verglichen werden. (Quantitativer Vergleich mit anderen Arbeiten)<sup>124</sup>

Es wird neben dem optimierten U-Net, bei welchem Beispiele der Region Ghana während von den Trainingsdaten ausgeschlossen und als Testdaten genutzt wurde, auch die Metriken des gleichen Modells, bei welchen hingegen die Region Bolivien ausgeschlossen wurde, erfasst, um so direkte Vergleichbarkeit mit Bonafilia u. a. 2020 herzustellen. Bei dem Vergleich der blau gekennzeichneten Modelle, also Modelle dieser Arbeit, lässt sich beobachten, dass der naive Grenzwert-basierte Ansatz eine höhere Genauigkeit sowie einen höheren mIoU aufweist als die neuronalen Modelle. Die hohe Genauigkeit ist jedoch wahrscheinlich auf eine zu geringe Vorhersage der Klasse *Wasser* und die Tatsache zurückzuführen, dass ein Großteil der Fläche in jedem Beispiel trockenes Land ist, was die Genauigkeit stark beeinflusst.<sup>125</sup> Betrachtet man daher lediglich den IoU der Klasse *Wasser*, zeigt sich, dass das optimal trainierte neuronale Modell besser abschneidet als das Grenzwert-Modell. Auch schneidet das Modell deutlich besser als das U-Net Vergleichsmodell in allen erfassten Metriken (+14,39% mIoU, +27,35% IoU Wasser, +6,19% Genauigkeit) ab. **Diese Verbesserung des optimierten U-Net gegenüber dem Vergleichsmodell zeigt den Erfolg der genaueren Untersuchung und Optimierung der neuronalen Modelle.**

Da die Modelle von Bonafilia u. a. 2020 mit demselben Datensatz trainiert und bewertet wurden, können diese direkt mit denen dieser Arbeit verglichen werden. Bonafilia u. a. 2020 erfassten lediglich die Genauigkeit ihrer Modelle sowie die IoU der Klasse *Wasser*, daher kann nicht mit

<sup>124</sup> Werte entnommen aus: Bonafilia u. a. 2020, S. 841 f. Mateo-Garcia u. a. 2021, S. 7, Rahnemoonfar u. a. 2020, S. 89651

<sup>125</sup> Vgl. Bonafilia u. a. 2020, S. 841

dem mIoU verglichen werden. Sowohl das FCN als auch der Grenzwert-Ansatz von Bonafilia u. a. 2020 weisen eine sehr hohe Genauigkeit mit 93,84% sowie 93,89% auf. Dies ist deutlich höher als die Genauigkeit des in dieser Arbeit trainierten U-Net++ Modells, welches lediglich eine Genauigkeit von 81,79% auf den Testdatensatz aufweisen kann. Wie schon angesprochen merkte Bonafilia u. a. 2020 ihrer Arbeit an, dass eine hohe Genauigkeit wenig aussagekräftig bezüglich der erfolgreichen Segmentierung der Klasse *Wasser* ist. Vergleicht man daher die IoU der Klasse *Wasser*, zeigt sich, dass das U-Net++ mit 51,75% deutlich besser abschneidet als die Modelle von Bonafilia u. a. 2020 mit 36,71%, beziehungsweise 35,91%. Daher kann gesagt werden, dass das in dieser Arbeit trainierte U-Net++ besser *Wasser* auf Satelliten-Daten erkennt. Bei dem Vergleich mit den Modellen der Arbeiten von Rahnemoonfar u. a. 2020 und Mateo-Garcia u. a. 2021 schneidet das U-Net++ schlechter ab. Grund dafür kann eine unterschiedliche Verteilung der Klassen im Datensatz, eine andere Definition der gestellten Aufgabe, z. B. Erkennung von Wasser anstelle von Überschwemmung oder andere unterschiedliche Eigenschaften der genutzten Methoden sein. Trotzdem kann festgehalten werden, dass es noch Potenzial bei der Verbesserung der Erkennung von *Wasser* auf Satellitendaten gibt.

## 4.5 Möglichkeiten der weiteren Optimierung, Ausblick und Implikationen für die Wissenschaft

Basierend auf den Ergebnissen dieser Forschungsarbeit können weitere Aspekte der Erkennung von *Wasser* auf Satellitenbildern untersucht werden. Mit der in dieser Arbeit genutzten Kombination an Modalitäten (AWEI, VV, VH, B8, B11, B4) können weitere Modelle, welche nicht auf den sehr aufwendigen neuronalen Netzen basieren, entwickelt werden. So könnte zum Beispiel Random Forests oder auch Deep Forests mit den entsprechenden Daten des Sen1Floods11 Datensatzes trainiert werden, um den Trainingsaufwand und die verbrauchten Rechen-Ressourcen zu reduzieren. Es könnten auch die Nutzung anderer neuronaler Modelle untersucht werden, wie zum Beispiel die neuartigen LKA Netzwerke, welche zum Beispiel als Encoder in einem U-Net dienen könnten. Auch müssen vor der Anwendung dieser Modelle noch Untersuchungen zur Robustheit und Vertrauenswürdigkeit getätigter werden, zum Beispiel durch Anomalien-Detektion oder Mixup-Attacken.

Auch der Einfluss der Trainingsdaten könnte genauer untersucht werden. So könnte zum Beispiel ein neuer Datensatz entwickelt werden, welcher genauere Annotationen bietet. Die Qualität der bestehenden oder zukünftigen Annotationen könnte ebenfalls durch eine gezielte Gewichtung einzelner Teile des Bildes stattfinden. So könnten zum Beispiel sehr gut annotierte Flüsse oder kleinere Seen mehr Gewichtung beim Training geschenkt werden oder bei schlecht annotierten Grenzen zwischen überschwemmten und nicht überschwemmten Gebieten diesen eine geringere Gewichtung geben. Andernfalls könnten Methoden zur Behebung von Annotationsfehlern untersucht werden, wie zum Beispiel die Anwendung einer künstlichen Bild-Unschärfe auf die annotierte Maske. Dadurch könnte das Modell mögliche Unsicherheiten in der Annotation, meist

Grenzen zwischen *Wasser* und *Nicht-Wasser*, erlernen und sich beim Training auf die sicheren Beispiele konzentrieren.

Bonafilia u. a. 2020 zeigten, dass die Anwendung eines einfachen Modells zur automatischen Annotierung der Daten bei der Erzeugung eines großen Datensatzes ausreichend ist, um mit diesem Datensatz weitere Modelle mit signifikant besserer Genauigkeit trainieren zu können. Durch die starke Verbesserung des Modells dieser Arbeit im Vergleich zu dem sehr einfachen Modell von Bonafilia u. a. 2020 könnte sich dieser Effekt deutlich verstärken. Demnach könnte mit dem deutlich besserem Modell eine größere Datenmenge generiert werden, welche wiederum für weitere Trainings von Modellen genutzt werden kann, um noch höhere Genauigkeiten zu erreichen. Dabei könnte auch die Skalierbarkeit dieses Effekts genauer untersucht werden, also ob die Genauigkeit des finalen Modells linear oder exponentiell mit zunehmender Genauigkeit des Modells zur Daten-Generierung wächst.

## 5 Fazit

Am Anfang dieser Arbeit stellte sich die Frage: *Ist es mit heutigem Wissenstand möglich, eine ähnliche, wenn nicht sogar bessere Genauigkeit und Zuverlässigkeit bei der Erkennung von Hochwasser auf Satellitendaten als durch manuelle, menschliche Annotierung zu erzielen?* Um diese Frage zu beantworten wurde eine Reihe an Experimenten zur Verbesserung bestehender Modelle durchgeführt.

Es konnte gezeigt werden, dass mit der 3-Sigma-Normalisierung die Werte eines Datensatzes zur Erkennung von Überschwemmungen auf Satellitendaten optisch sichtlich verbessert werden können. In einem einheitlichen Evaluierungs-Verfahren zur Bewertung von Modellen zur Erkennung von Hochwasser auf Satellitendaten konnten in anwendungsbezogenen Experimenten eine Auswahl an geeigneten Verlustfunktionen gefunden werden. Diese Arbeit nutzte im weiteren Verlauf den Tversky-Verlust, welcher sowohl an die Aufgabe der Bild-Segmentierung angepasst ist als auch den Bias des Datensatzes reflektiert. Bei der Anwendung des Tversky-Verlustes konnte eine Verbesserung gegenüber dem Vergleichsmodell festgestellt werden. Durch die Anwendung der Kombinationen bestehend aus den optischen Bändern B11, B8, B4, den SAR-Bändern VV und VH sowie der verrechneten Kombination zum Index AWEI konnte ebenfalls eine signifikante Verbesserung festgestellt werden. In einer Hyperparameter-Rastersuche konnten die optimalen Trainingseinstellungen für das Modell gefunden werden, was abermals eine Verbesserung gegenüber dem Vergleichsmodell darstellt.

Im Vergleich mit anderen Arbeiten der Forschungsliteratur zeigte sich, dass das schlussendlich optimierte Modell eine wesentliche Verbesserung gegenüber anderen auf den gleichen Datensatz trainierten Modellen darstellt. Die Nutzung verschiedener Regionen hat, wie gezeigt werden konnte, sehr großen Einfluss auf den Erfolg des Trainings der Modelle, ebenso wie die Reduzierung der Problematik auf eine binäre Klassifizierung in die Klasse *Wasser* bzw. *Nicht-Wasser*. Weiterhin konnte auch gezeigt werden, dass dieses optimierte Modell ähnlich gute, wenn nicht sogar bessere Ergebnisse bei der Erkennung von Hochwasser auf Satellitenbildern liefert als Modelle von im Forschungsfeld führenden Arbeiten. Somit konnte sich diese Arbeit dem State of the Art anschließen.

Das in dieser Arbeit entwickelte Modell weist jedoch weder eine bessere noch eine ähnliche Genauigkeit und Zuverlässigkeit bei der Erkennung von Hochwasser auf Satellitendaten als menschliche Annotationen auf. Die Möglichkeit, diesen Grad an Genauigkeit und Zuverlässigkeit zu erreichen, steht jedoch weiterhin infrage, da auch zu dem in dieser Arbeit entwickelten Modell noch Optimierungspotential steckt.

## Literaturverzeichnis

- Akiba, T./Sano, S./Yanase, T./Ohta, T./Koyama, M. (2019):** Optuna: A Next-generation Hyperparameter Optimization Framework. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, S. 2623–2631. DOI: 10.1145/3292500.3330701. URL: <https://doi.org/10.1145/3292500.3330701>.
- Augusteijn, M. F./Warrender, C. E. (1998):** Wetland classification using optical and radar data and neural network classification. In: <http://dx.doi.org/10.1080/014311698215342> 19 (8), S. 1545–1560. ISSN: 13665901. DOI: 10.1080/014311698215342. URL: <https://www.tandfonline.com/doi/abs/10.1080/014311698215342>.
- Bates, P. D./Quinn, N./Sampson, C./Smith, A./Wing, O./Sosa, J./Savage, J./Olcese, G./Neal, J./Schumann, G./Giustarini, L./Coxon, G./Porter, J. R./Amodeo, M. F./Chu, Z./Lewis-Gruss, S./Freeman, N. B./Houser, T./Delgado, M./Hamidi, A./Bolliger, I./McCusker, K./Emanuel, K./Ferreira, C. M./Khalid, A./Haigh, I. D./Couasnon, A./Kopp, R./Hsiang, S./Krajewski, W. F. (2021):** Combined Modeling of US Fluvial, Pluvial, and Coastal Flood Hazard Under Current and Future Climates. In: *Water Resources Research* 57.2. DOI: 10.1029/2020wr028673.
- Bell, R./Lawrence, J./Allan, S./Blackett, P./Stephens, T. (2017):** Coastal Hazards and Climate Change - Guidance for local government. Forschungsber. New Zealand Ministry for the Environment.
- Bonafilia, D./Tellman, B./Anderson, T./Issenberg, E. (2020):** Sen1Floods11: a geo-referenced dataset to train and test deep learning flood algorithms for Sentinel-1. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Seattle, WA, USA: IEEE, S. 835–845. ISBN: 978-1-72819-360-1. DOI: 10.1109/CVPRW50498.2020.90113. URL: <https://ieeexplore.ieee.org/document/9150760/> (besucht am 04.01.2022).
- Boyd, K./Eng, K. H./Page, C. D. (2013):** Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8190 LNAI (PART 3), S. 451–466. ISSN: 03029743. DOI: 10.1007/978-3-642-40994-3\_29. URL: [https://link.springer.com/chapter/10.1007/978-3-642-40994-3\\_29](https://link.springer.com/chapter/10.1007/978-3-642-40994-3_29).
- BROWN, W. M. (1967):** Synthetic Aperture Radar. In: *IEEE Transactions on Aerospace and Electronic Systems* AES-3 (2), S. 217–229. ISSN: 00189251. DOI: 10.1109/TAES.1967.5408745.
- Brunner, M. I./Swain, D. L./Wood, R. R./Willkofer, F./Done, J. M./Gilleland, E./Ludwig, R. (2021):** An extremeness threshold determines the regional response of floods to changes in rainfall extremes. In: *Communications Earth & Environment* 2 (1), S. 173. ISSN: 2662-4435. DOI: 10.1038/s43247-021-00248-x. URL: <https://www.nature.com/articles/s43247-021-00248-x>.
- Burges, C. J. (1998):** A Tutorial on Support Vector Machines for Pattern Recognition. In: *Data Mining and Knowledge Discovery 1998 2:2* 2 (2), S. 121–167. ISSN: 1573-756X. DOI:

- 10 . 1023 / A : 1009715923555. URL: <https://link.springer.com/article/10.1023/A:1009715923555>.
- Chen, A. S./Djordjević, S./Leandro, J./Savić, D. A. (2010):** An analysis of the combined consequences of pluvial and fluvial flooding. In: *Water Science and Technology* 62.7, S. 1491–1498. DOI: 10.2166/wst.2010.486.
- Committee, M. S. (2019):** 754-2019 - IEEE Standard for Floating-Point Arithmetic. In: *IEEE Std 754-2019 (Revision of IEEE 754-2008)*, S. 1–84.
- Cubuk, E. D./Zoph, B./Mane, D./Vasudevan, V./Le, Q. V. (2019):** AutoAugment: Learning Augmentation Strategies From Data. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Cun, Y. L./Guyon, I./Jackel, L. D./Henderson, D./Boser, B./Howard, R. E./Denker, J. S./Hubbard, W./Graf, H. P. (1989):** Handwritten Digit Recognition: Applications of Neural Network Chips and Automatic Learning. In: *IEEE Communications Magazine* 27 (11), S. 41–46. ISSN: 01636804. DOI: 10.1109/35.41400.
- Czymzik, M./Brauer, A./Dulski, P./Plessen, B./Naumann, R./von Grafenstein, U./Scheffler, R. (2013):** Orbital and solar forcing of shifts in Mid- to Late Holocene flood intensity from varved sediments of pre-alpine Lake Ammersee (southern Germany). In: *Quaternary Science Reviews* 61, S. 96–110. ISSN: 0277-3791. DOI: <https://doi.org/10.1016/j.quascirev.2012.11.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0277379112004817>.
- Denevi, G./Ciliberto, C./Grazzi, R./Pontil, M. (2019):** Learning-to-Learn Stochastic Gradient Descent with Biased Regularization. In: *Proceedings of the 36th International Conference on Machine Learning*. Hrsg. von Kamalika Chaudhuri/Ruslan Salakhutdinov. Bd. 97. Proceedings of Machine Learning Research. PMLR, S. 1566–1575. URL: <https://proceedings.mlr.press/v97/denevi19a.html>.
- Deng, J./Dong, W./Socher, R./Li, L.-J./Li, K./Fei-Fei, L. (2009):** ImageNet: A Large-Scale Hierarchical Image Database. In: *CVPR09*.
- Detlefsen, N. S./Borovec, J./Schock, J./Harsh, A./Koker, T./Liello, L. D./Stanci, D./Quan, C./Grechkin, M./Falcon, W. (2022):** TorchMetrics - Measuring Reproducibility in PyTorch. DOI: 10.5281/ZENODO.6459048. URL: <https://zenodo.org/record/6459048>.
- Donchyts, G./Schellekens, J./Winsemius, H./Eisemann, E./van de Giesen, N. (2016):** A 30 m Resolution Surface Water Mask Including Estimation of Positional and Thematic Differences Using Landsat 8, SRTM and OpenStreetMap: A Case Study in the Murray-Darling Basin, Australia. In: *Remote Sensing 2016, Vol. 8, Page 386* 8 (5), S. 386. ISSN: 2072-4292. DOI: 10.3390/RS8050386. URL: <https://www.mdpi.com/2072-4292/8/5/386>.
- Doocy, S./Daniels, A./Murray, S./Kirsch, T. D. (2013):** The Human Impact of Floods: a Historical Review of Events 1980-2009 and Systematic Literature Review. In: *PLoS Currents* 5, *ecurrents.dis.f4deb457904936b07c09daa98ee8171a*. ISSN: 2157-3999. DOI: 10.1371/currents.dis.f4deb457904936b07c09daa98ee8171a. URL: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3644291/> (besucht am 01.03.2022).

- Eelbode, T./Bertels, J./Berman, M./Vandermeulen, D./Maes, F./Bisschops, R./Blaschko, M. B. (2020)**: Optimization for Medical Image Segmentation: Theory and Practice When Evaluating With Dice Score or Jaccard Index. In: *IEEE transactions on medical imaging* 39 (11), S. 3679–3690. ISSN: 1558254X. DOI: 10.1109/TMI.2020.3002417.
- Falcon, W./Borovec, J./Wälchli, A./Eggert, N./Schock, J./Jordan, J./Skafte, N./Ir1dXD/Bereznyuk, V./Harris, E./Murrell, T./Yu, P./Præsius, S./Addair, T./Zhong, J./Lipin, D./Uchida, S./Bapat, S./Schröter, H./Dayma, B./Karnachev, A./Kulkarni, A./Komatsu, S./Martin.B/SCHIRATTI, J.-B./Mary, H./Byrne, D./Eyzaguirre, C./cinjon/Bakhtin, A. (2022)**: PyTorch Lightning. DOI: 10.5281/zenodo.3828935. URL: <https://www.pytorchlightning.ai>.
- Feyisa, G. L./Meilby, H./Fensholt, R./Proud, S. R. (2014)**: Automated Water Extraction Index: A new technique for surface water mapping using Landsat imagery. In: *Remote Sensing of Environment* 140, S. 23–35. ISSN: 0034-4257. DOI: 10.1016/J.RSE.2013.08.029.
- Fretwell, P. T./Staniland, I. J./Forcada, J. (2014)**: Whales from Space: Counting Southern Right Whales by Satellite. In: *PLOS ONE* 9.2, e88655. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0088655. URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0088655> (besucht am 02.03.2022).
- Gotmare, A./Keskar, N. S./Xiong, C./Socher, R. (2018)**: A Closer Look at Deep Learning Heuristics: Learning rate restarts, Warmup and Distillation. In: *7th International Conference on Learning Representations, ICLR 2019*. DOI: 10.48550/arxiv.1810.13243. URL: <https://arxiv.org/abs/1810.13243v1>.
- Gregor, S./Hevner, A. R. (2013)**: Positioning and Presenting Design Science Research for Maximum Impact. In: *MIS Q.* 37.2, S. 337–355. URL: <http://misq.org/positioning-and-presenting-design-science-research-for-maximum-impact.html>.
- Guo, M.-H./Lu, C.-Z./Liu, Z.-N./Cheng, M.-M./Hu, S.-M. (2022)**: Visual Attention Network. In: *arXiv preprint arXiv:2202.09741*. DOI: 10.48550/arxiv.2202.09741. URL: <https://arxiv.org/abs/2202.09741v3>.
- Harouni, M./Baghmaleki, H. (2020)**: Color Image Segmentation Metrics. In: *Encyclopedia of Image Processing*. Hrsg. von Phillip A. Laplante. CRC Press, S. 95–113. ISBN: 9781482244908. DOI: 10.1201/9781351110273-140000183. URL: <https://arxiv.org/abs/2010.09907>.
- Hartmann, T./Juepner, R. (2017)**: The flood risk management plan between spatial planning and water engineering. In: *Journal of Flood Risk Management* 10 (2), S. 143–144. ISSN: 1753-318X. DOI: 10.1111/jfr3.12101. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1111/jfr3.12101>.
- He, K./Zhang, X./Ren, S./Sun, J. (2015)**: Deep Residual Learning for Image Recognition. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-December, S. 770–778. ISSN: 10636919. DOI: 10.48550/arxiv.1512.03385. URL: <https://arxiv.org/abs/1512.03385v1>.
- Huang, G./Liu, Z./Maaten, L. V. D./Weinberger, K. Q. (2016)**: Densely Connected Convolutional Networks. In: *Proceedings - 30th IEEE Conference on Computer Vision and*

- Pattern Recognition, CVPR 2017* 2017-January, S. 2261–2269. DOI: 10.48550/arxiv.1608.06993. URL: <https://arxiv.org/abs/1608.06993v5>.
- Jamali, A./Mahdianpari, M./Brisco, B./Granger, J./Mohammadianesh, F./Salehi, B. (2021)**: Deep Forest classifier for wetland mapping using the combination of Sentinel-1 and Sentinel-2 data. In: <https://doi.org/10.1080/15481603.2021.1965399> 58 (7), S. 1072–1089. ISSN: 15481603. DOI: 10.1080/15481603.2021.1965399. URL: <https://www.tandfonline.com/doi/abs/10.1080/15481603.2021.1965399>.
- Jaramillo, C./Romero, I./D'Apolito, C./Bayona, G./Duarte, E./Louwye, S./Escoobar, J./Luque, J./Carrillo-Briceño, J. D./Zapata, V./Mora, A./Schouten, S./Zavada, M./Harrington, G./Ortiz, J./Wesselingh, F. P. (2017)**: Miocene flooding events of western Amazonia. In: *Science Advances* 3 (5). ISSN: 23752548. DOI: 10.1126/SCIAADV.1601693/SUPPL\_FILE/1601693\_TABLESS1\_TO\_S12.ZIP. URL: <https://www.science.org>.
- Jegou, S./Drozdzal, M./Vazquez, D./Romero, A./Bengio, Y. (2016)**: The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 2017-July, S. 1175–1183. ISSN: 21607516. DOI: 10.48550/arxiv.1611.09326. URL: <https://arxiv.org/abs/1611.09326v3>.
- Jordan, J. (2018)**: Evaluating image segmentation models. URL: <https://www.jeremyjordan.me/evaluating-image-segmentation-models/>.
- Khatami, R./Mountrakis, G./Stehman, S. V. (2016)**: A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: General guidelines for practitioners and future research. In: *Remote Sensing of Environment* 177, S. 89–100. ISSN: 0034-4257. DOI: 10.1016/J.RSE.2016.02.028. URL: <https://www.sciencedirect.com/science/article/abs/pii/S0034425716300578>.
- Kolesnikov, A./Dosovitskiy, A./Weissenborn, D./Heigold, G./Uszkoreit, J./Beyer, L./Minderer, M./Dehghani, M./Houlsby, N./Gelly, S./Unterthiner, T./Zhai, X. (2021)**: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. DOI: 10.48550/arxiv.2010.11929. URL: <https://openreview.net/forum?id=YicbFdNTTy>.
- Krizhevsky, A./Sutskever, I./Hinton, G. E. (2012)**: ImageNet Classification with Deep Convolutional Neural Networks. In: *Advances in Neural Information Processing Systems*. Hrsg. von F. Pereira/C.J. Burges/L. Bottou/K.Q. Weinberger. Bd. 25. Curran Associates, Inc. URL: <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
- Lecun, Y./Bengio, Y./Hinton, G. (2015)**: Deep learning. In: *Nature* 2015 521:7553 521 (7553), S. 436–444. ISSN: 1476-4687. DOI: 10.1038/nature14539. URL: <https://www.nature.com/articles/nature14539>.
- LeCun, Y./Haffner, P./Bottou, L./Bengio, Y. (1999)**: Object Recognition with Gradient-Based Learning. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 1681, S. 319–345. ISSN: 16113349.

- DOI: 10.1007/3-540-46805-6\_19. URL: [https://link.springer.com/chapter/10.1007/3-540-46805-6\\_19](https://link.springer.com/chapter/10.1007/3-540-46805-6_19).
- Lin, T. Y./Goyal, P./Girshick, R./He, K./Dollar, P. (2017):** Focal Loss for Dense Object Detection. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2), S. 318–327. ISSN: 19393539. DOI: 10.48550/arxiv.1708.02002. URL: <https://arxiv.org/abs/1708.02002v2>.
- Liu, Z./Li, X./Luo, P./Loy, C. C./Tang, X. (2015):** Semantic Image Segmentation via Deep Parsing Network. In: *CoRR*. DOI: 10.1109/ICCV.2015.162. URL: <http://arxiv.org/abs/1509.02634>.
- Long, J./Shelhamer, E./Darrell, T. (2015):** Fully Convolutional Networks for Semantic Segmentation. In: *arXiv:1411.4038 [cs]*. URL: <http://arxiv.org/abs/1411.4038>.
- Lopez, T./Al Bitar, A./Biancamaria, S./Güntner, A./Jäggi, A. (2020):** On the Use of Satellite Remote Sensing to Detect Floods and Droughts at Large Scales. In: *Surveys in Geophysics* 41.6, S. 1461–1487. ISSN: 1573-0956. DOI: 10.1007/s10712-020-09618-0. URL: <https://doi.org/10.1007/s10712-020-09618-0> (besucht am 28.02.2022).
- Madry, S. (2013):** Introduction and History of Space Remote Sensing. In: *Handbook of Satellite Applications*. Hrsg. von Joseph N. Pelton/Scott Madry/Sergio Camacho-Lara. New York, NY: Springer, S. 657–666. ISBN: 9781441976710. DOI: 10.1007/978-1-4419-7671-0\_37. URL: [https://doi.org/10.1007/978-1-4419-7671-0\\_37](https://doi.org/10.1007/978-1-4419-7671-0_37) (besucht am 02.03.2022).
- Masson-Delmotte, V./P. Zhai, A. P./Connors, S. L./Péan, C./Berger, S./Caud, N./Chen, Y./Goldfarb, L./Gomis, M. I./Huang, M./Leitzell, K./Lonnoy, E./Matthews, J. B. R./Maycock, T. K./Waterfield, T./Yelekçi, O./Yu, R./Zhou, B. (2021):** IPCC, 2021: Summary for Policymakers. In: *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*.
- Mateo-Garcia, G./Veitch-Michaelis, J./Smith, L./Oprea, S. V./Schumann, G./Gal, Y./Baydin, A. G./Backes, D. (2021):** Towards global flood mapping onboard low cost satellites with machine learning. In: *Scientific Reports* 11.1. DOI: 10.1038/s41598-021-86650-z.
- McFeeters, S. K. (1996):** The use of the Normalized Difference Water Index (NDWI) in the delineation of open water features. In: <https://doi.org/10.1080/01431169608948714> 17 (7), S. 1425–1432. ISSN: 13665901. DOI: 10.1080/01431169608948714. URL: <https://www.tandfonline.com/doi/abs/10.1080/01431169608948714>.
- Minaee, S./Boykov, Y. Y./Porikli, F./Plaza, A. J./Kehtarnavaz, N./Terzopoulos, D. (2021):** Image Segmentation Using Deep Learning: A Survey. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. ISSN: 19393539. DOI: 10.1109/TPAMI.2021.3059968.
- Mitchell, T. M. (1997):** Machine Learning. WBC/McGraw-Hill. ISBN: 0070428077.
- Moreira, A./Prats-Iraola, P./Younis, M./Krieger, G./Hajnsek, I./Papathanassiou, K. P. (2013):** A tutorial on synthetic aperture radar. In: *IEEE Geoscience and Remote Sensing Magazine* 1 (1), S. 6–43. ISSN: 21686831. DOI: 10.1109/MGRS.2013.2248301.

- Mountrakis, G./Im, J./Ogole, C. (2011):** Support vector machines in remote sensing: A review. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (3), S. 247–259. ISSN: 0924-2716. DOI: 10.1016/J.ISPRSJPRS.2010.11.001.
- Online, S. (2015):** MultiSpectral Instrument (MSI) Overview. ESA. URL: <https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument> (besucht am 21.04.2022).
- Otsu, N. (1979):** THRESHOLD SELECTION METHOD FROM GRAY-LEVEL HISTOGRAMS. In: *IEEE Trans Syst Man Cybern SMC-9* (1), S. 62–66. ISSN: 00189472. DOI: 10.1109/TSMC.1979.4310076.
- Otter, D. W./Medina, J. R./Kalita, J. K. (2021):** A Survey of the Usages of Deep Learning for Natural Language Processing. In: *IEEE Transactions on Neural Networks and Learning Systems* 32 (2), S. 604–624. ISSN: 21622388. DOI: 10.1109/TNNLS.2020.2979670.
- Panchapagesan, S./Sun, M./Khare, A./Matsoukas, S./Mandal, A./Hoffmeister, B./Vitaladevuni, S. (2016):** Multi-Task Learning and Weighted Cross-Entropy for DNN-Based Keyword Spotting. In: *Interspeech 2016*. ISCA, S. 760–764. DOI: 10.21437/Interspeech.2016-1485.
- Pekel, J. F./Cottam, A./Gorelick, N./Belward, A. S. (2016):** High-resolution mapping of global surface water and its long-term changes. In: *Nature* 2016 540:7633 540 (7633), S. 418–422. ISSN: 1476-4687. DOI: 10.1038/nature20584. URL: <https://www.nature.com/articles/nature20584>.
- Pettorelli, N. (2013):** The Normalized Difference Vegetation Index. Oxford University Press. ISBN: 978-0-19-151272-8.
- Rácz, A./Bajusz, D./Héberger, K. (2021):** Effect of Dataset Size and Train/Test Split Ratios in QSAR/QSPR Multiclass Classification. In: *Molecules* 2021, Vol. 26, Page 1111 26 (4), S. 1111. ISSN: 1420-3049. DOI: 10.3390/MOLECULES26041111. URL: <https://www.mdpi.com/1420-3049/26/4/1111>.
- Rahnemoonfar, M./Chowdhury, T./Sarkar, A./Varshney, D./Yari, M./Murphy, R. R. (2020):** FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding. In: *IEEE Access* 9, S. 89644–89654. ISSN: 21693536. DOI: 10.48550/arxiv.2012.02951. URL: <https://arxiv.org/abs/2012.02951v1>.
- Robbins, H./Monro, S. (1951):** A Stochastic Approximation Method. In: <https://doi.org/10.1214/aoms/1177729586> 22 (3), S. 400–407. ISSN: 0003-4851. DOI: 10.1214/AOMS/1177729586. URL: <https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-3/A-Stochastic-Approximation-Method/10.1214/aoms/1177729586.full>.
- Ronneberger, O./Fischer, P./Brox, T. (2015):** U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*. Hrsg. von Nassir Navab/Joachim Hornegger/William M. Wells/Alejandro F. Frangi. Lecture Notes in Computer Science. Cham: Springer International Publishing, S. 234–241. ISBN: 9783319245744. DOI: 10.1007/978-3-319-24574-4\_28.
- Salehi, S. S. M./Erdogmus, D./Gholipour, A. (2017):** Tversky loss function for image segmentation using 3D fully convolutional deep networks. In: *Lecture Notes in Computer Science*

- (including subseries *Lecture Notes in Artificial Intelligence* and *Lecture Notes in Bioinformatics*) 10541 LNCS, S. 379–387. ISSN: 16113349. DOI: 10.48550/arxiv.1706.05721. URL: <https://arxiv.org/abs/1706.05721v1>.
- Schmidt, W. (2014)**: Optische Spektroskopie: Eine Einführung. Google-Books-ID: 49dLBAAQ-BAJ. John Wiley & Sons. 448 S. ISBN: 9783527663347.
- Shanker, M./Hu, M./Hung, M. (1996)**: Effect of data standardization on neural network training. In: *Omega* 24.4, S. 385–397. ISSN: 0305-0483. DOI: [https://doi.org/10.1016/0305-0483\(96\)00010-2](https://doi.org/10.1016/0305-0483(96)00010-2). URL: <https://www.sciencedirect.com/science/article/pii/0305048396000102>.
- Short, N. M. (1982)**: The Landsat Tutorial Workbook: Basics of Satellite Remote Sensing. Google-Books-ID: 9RsrAAAAIAAJ. National Aeronautics/Space Administration, Scientific/Technical Information Branch. 576 S.
- Simonyan, K./Zisserman, A. (2014)**: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. DOI: 10.48550/arxiv.1409.1556. URL: <https://arxiv.org/abs/1409.1556v6>.
- Smith, D. I. (1981)**: Actual and potential flood damage: a case study for urban Lismore, NSW, Australia. In: *Applied Geography* 1 (1), S. 31–39. ISSN: 0143-6228. DOI: 10.1016/0143-6228(81)90004-7.
- Tan, M./Le, Q. V. (2019)**: EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In: *36th International Conference on Machine Learning, ICML 2019* 2019-June, S. 10691–10700. DOI: 10.48550/arxiv.1905.11946. URL: <https://arxiv.org/abs/1905.11946v5>.
- Thépaut, J.-N./Dee, D./Engelen, R./Pinty, B. (2018)**: The Copernicus Programme and its Climate Change Service. In: *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*. IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium. ISSN: 2153-7003, S. 1591–1593. DOI: 10.1109/IGARSS.2018.8518067.
- Tsakiris, G. (2014)**: Flood risk assessment: concepts, modelling, applications. In: *Natural Hazards and Earth System Sciences* 14 (5), S. 1361–1369. ISSN: 1684-9981. DOI: 10.5194/nhess-14-1361-2014. URL: <https://nhess.copernicus.org/articles/14/1361/2014/>.
- Tversky, A. (1977)**: Features of similarity. In: *Psychological Review* 84 (4), S. 327–352. ISSN: 0033295X. DOI: 10.1037/0033-295X.84.4.327. URL: /record/1978-09287-001.
- Vaswani, A./Shazeer, N./Parmar, N./Uszkoreit, J./Jones, L./Gomez, A. N./Kaiser, L./Polosukhin, I. (2017)**: Attention Is All You Need. In: *Advances in Neural Information Processing Systems* 2017-December, S. 5999–6009. ISSN: 10495258. DOI: 10.48550/arxiv.1706.03762. URL: <https://arxiv.org/abs/1706.03762v5>.
- Xie, S./Girshick, R./Dollár, P./Tu, Z./He, K. (2017)**: Aggregated residual transformations for deep neural networks. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-January, S. 5987–5995. DOI: 10.1109/CVPR.2017.634.

- Xu, Y./Lam, H. K./Jia, G. (2021):** MANet: A two-stage deep learning method for classification of COVID-19 from Chest X-ray images. In: *Neurocomputing* 443, S. 96–105. ISSN: 0925-2312. DOI: 10.1016/J.NEUROCOM.2021.03.034.
- Yakubovskiy, P. (2020):** Segmentation Models Pytorch. [https://github.com/qubvel/segmentation\\_models.pytorch](https://github.com/qubvel/segmentation_models.pytorch).
- Yeo, S. W. (2002):** Flooding in Australia: A Review of Events in 1998. In: *Natural Hazards 2002* 25:2 25 (2), S. 177–191. ISSN: 1573-0840. DOI: 10.1023/A:1013765303567. URL: <https://link.springer.com/article/10.1023/A:1013765303567>.
- Zhang, H./Dana, K./Shi, J./Zhang, Z./Wang, X./Tyagi, A./Agrawal, A. (2018):** Context Encoding for Semantic Segmentation. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, S. 7151–7160. ISSN: 10636919. DOI: 10.48550/arxiv.1803.08904. URL: <https://arxiv.org/abs/1803.08904v1>.
- Zhang, Z./Liu, Q./Wang, Y. (2018):** Road Extraction by Deep Residual U-Net. In: *IEEE Geoscience and Remote Sensing Letters* 15.5, S. 749–753. ISSN: 1558-0571. DOI: 10.1109/LGRS.2018.2802944.
- Zhou, Z.-H./Feng, J. (2017):** Deep Forest: Towards An Alternative to Deep Neural Networks. In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, S. 3553–3559. DOI: 10.24963/ijcai.2017/497. URL: <https://doi.org/10.24963/ijcai.2017/497>.
- Zhou, Z./Siddiquee, M. M. R./Tajbakhsh, N./Liang, J. (2018):** UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11045 LNCS, S. 3–11. ISSN: 16113349. DOI: 10.48550/arxiv.1807.10165. URL: <https://arxiv.org/abs/1807.10165v1>.
- Zurich (o.J.):** Three common types of flood explained. URL: <https://www.zurich.com/knowledge/topics/flood-and-water-damage/three-common-types-of-flood> (besucht am 17.02.2022).

# **Appendix**

## **Appendix Directory**

Anhang 1	Tabellen und Visualisierungen der Statistischen Eigenschaften des Datensatzes	74
Anhang 2	Beispiele zur qualitativen Analyse	77

## Anhang 1: Tabellen und Visualisierungen der Statistischen Eigenschaften des Datensatzes

	Durchschnitt	Std	Min	Max	1. Percentil	99. Percentil
vv	0.0	1.0	-18.1	10.7	-2.1	1.6
vh	0.0	1.0	-16.0	6.3	-2.1	1.3
b1	0.0	1.0	-2.3	12.1	-0.5	1.8
b2	0.0	1.0	-1.9	24.2	-0.6	2.0
b3	0.0	1.0	-1.8	18.8	-0.7	2.0
b4	0.0	1.0	-1.4	18.9	-0.7	2.0
b5	0.0	1.0	-1.9	12.4	-0.9	2.0
b6	0.0	1.0	-2.5	10.3	-1.4	1.8
b7	0.0	1.0	-2.6	9.0	-1.4	1.7
b8	0.0	1.0	-2.5	16.1	-1.5	1.7
b8a	0.0	1.0	-2.5	8.5	-1.5	1.7
b9	0.0	1.0	-1.4	10.5	-0.8	1.2
b10	0.0	1.0	-0.5	12.9	-0.2	0.4
b11	0.0	1.0	-2.0	13.1	-1.3	1.9
b12	0.0	1.0	-1.5	16.0	-1.0	2.0

Tab. 12: Durchschnitt, Standardabweichung, Minima, Maxima und die 1% Quantile der verschiedenen Bänder des Sen1Floods11 Datensatzes nach Anwendung der Albumentation-Normalisierung. Alle Werte wurden auf eine Nachkommastelle gerundet. (Sen1Floods11 Datensatz: Statistische Eigenschaften nach Normalisierung über die Standardabweichung)

	Durchschnitt	Std	Min	Max	1. Percentil	99. Percentil
vv	0.6	0.0	0.0	1.0	0.6	0.7
vh	0.7	0.0	0.0	1.0	0.6	0.8
b1	0.2	0.1	0.0	1.0	0.1	0.3
b2	0.1	0.0	0.0	1.0	0.1	0.1
b3	0.1	0.0	0.0	1.0	0.1	0.2
b4	0.1	0.0	0.0	1.0	0.0	0.2
b5	0.1	0.1	0.0	1.0	0.1	0.3
b6	0.2	0.1	0.0	1.0	0.1	0.3
b7	0.2	0.1	0.0	1.0	0.1	0.4
b8	0.1	0.1	0.0	1.0	0.1	0.2
b8a	0.2	0.1	0.0	1.0	0.1	0.4
b9	0.1	0.1	0.0	1.0	0.1	0.2
b10	0.0	0.1	0.0	1.0	0.0	0.1
b11	0.1	0.1	0.0	1.0	0.0	0.3
b12	0.1	0.1	0.0	1.0	0.0	0.2

Tab. 13: Durchschnitt, Standardabweichung, Minima, Maxima und die 1% Quantile der verschiedenen Bänder des Sen1Floods11 Datensatzes nach Anwendung der MinMax-Normalisierung. Alle Werte wurden auf eine Nachkommastelle gerundet. (Sen1Floods11 Datensatz: Statistische Eigenschaften nach MinMax-Normalisierung)

	Durchschnitt	Std	Min	Max	1. Percentil	99. Percentil
vv	0.5	0.2	0.0	1.0	0.2	0.8
vh	0.5	0.2	0.0	1.0	0.2	0.7
b1	0.5	0.1	0.0	1.0	0.4	0.7
b2	0.5	0.1	0.0	1.0	0.4	0.7
b3	0.5	0.1	0.0	1.0	0.4	0.7
b4	0.5	0.1	0.0	1.0	0.4	0.7
b5	0.5	0.1	0.0	1.0	0.4	0.7
b6	0.5	0.2	0.0	1.0	0.3	0.8
b7	0.5	0.2	0.0	1.0	0.3	0.8
b8	0.5	0.2	0.0	1.0	0.3	0.8
b8a	0.5	0.2	0.0	1.0	0.3	0.8
b9	0.5	0.2	0.0	1.0	0.4	0.7
b10	0.5	0.1	0.0	1.0	0.5	0.5
b11	0.5	0.2	0.0	1.0	0.3	0.8
b12	0.5	0.2	0.0	1.0	0.3	0.8

Tab. 14: Durchschnitt, Standardabweichung, Minima, Maxima und die 1% Quantile der verschiedenen Bänder des Sen1Floods11 Datensatzes nach Anwendung der 3-Sigma-Normalisierung. Alle Werte wurden auf eine Nachkommastelle gerundet. (Sen1Floods11 Datensatz: Statistische Eigenschaften nach 3-Sigma-Normalisierung)

## Appendix

---

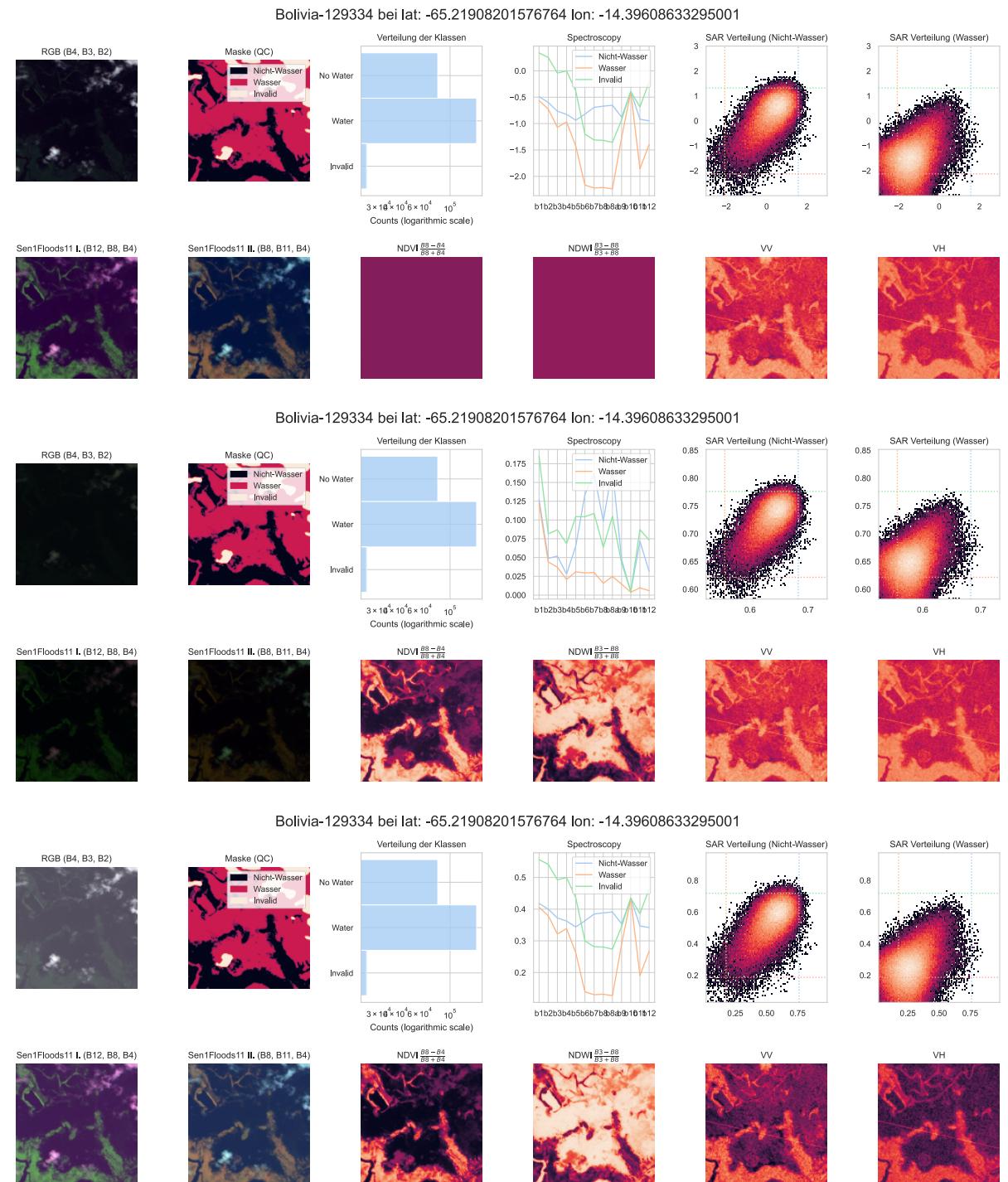


Fig. 30: Diese Abbildungen besitzen den gleichen Aufbau wie Abbildung 14. Normalisierungen von oben nach unten: Albumentation-Normalisierung, MinMax-Normalisierung, 3-Sigma-Normalisierung. (Visualisierungen zu einer normalisierten Satellitenaufnahme eines Gebiets in Bolivien)

## Anhang 2: Beispiele zur qualitativen Analyse

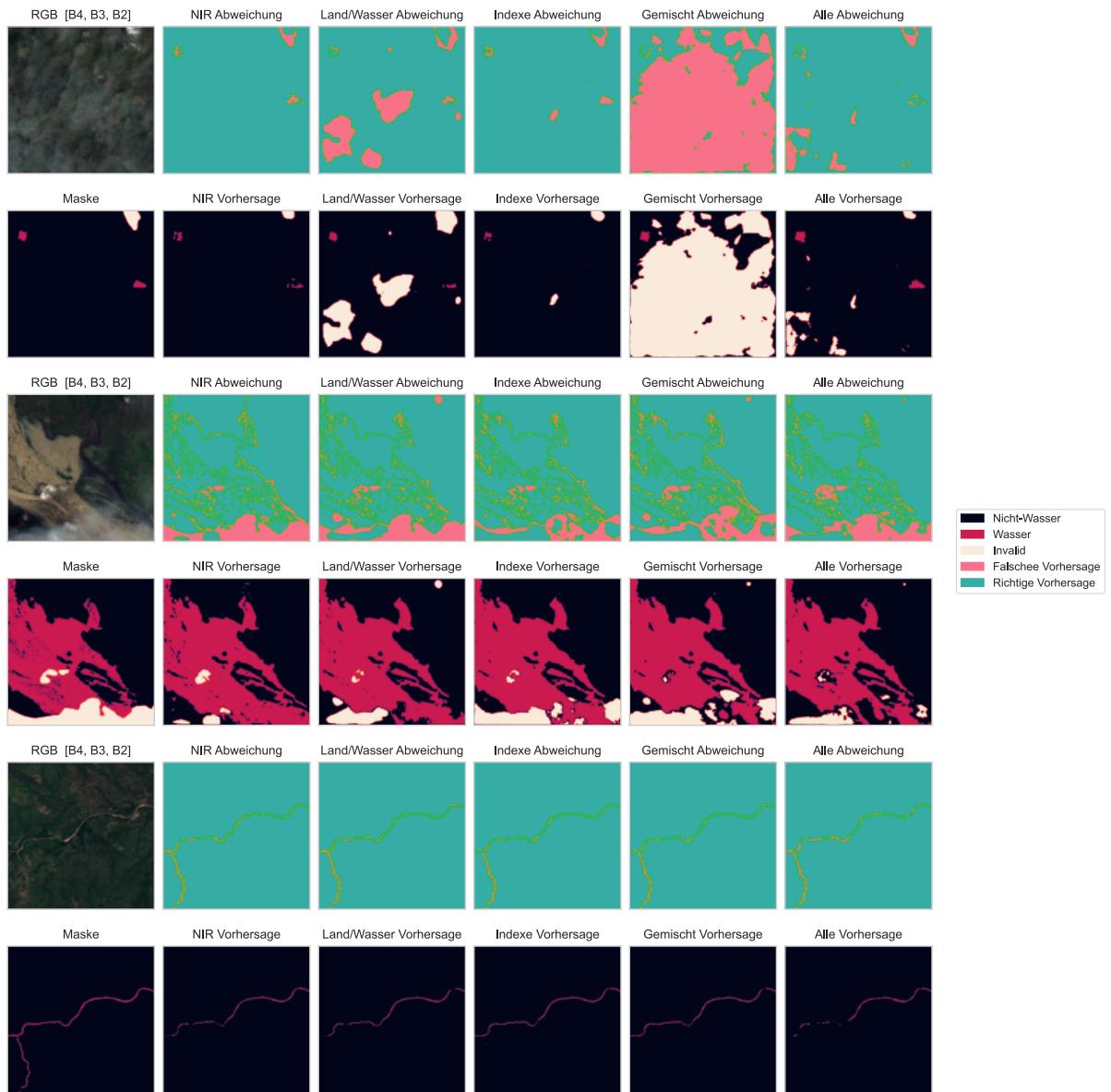


Fig. 31: Vorhersagen von Modellen von drei Beispielen, nach dem Training mit verschiedenen Kombinationen von Modalitäten (Qualitative Ergebnisse des Trainings zur Bewertung von Modalität-Kombinationen)

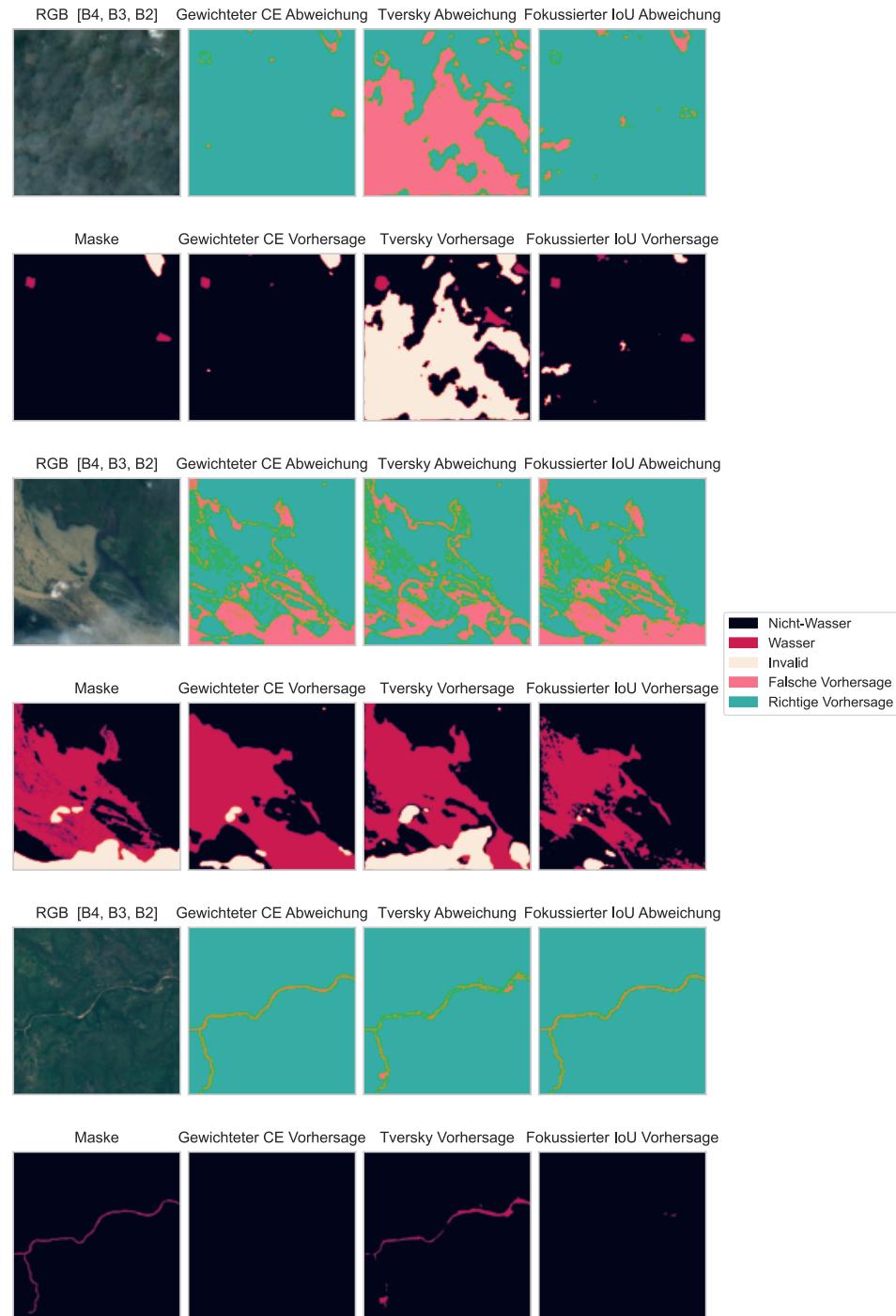


Fig. 32: Vorhersagen von Modellen von drei Beispielen, nach dem Training mit verschiedenen Verlustfunktionen. (Qualitative Ergebnisse des Trainings zur Bewertung von Verlustfunktionen)

## **Erklärung**

Ich versichere hiermit, dass ich meine Bachelorarbeit mit dem Thema: *Vergleich verschiedener Modell-Architekturen und Modalitäten zur Erkennung von Hochwasser mit Satellitendaten* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

(Ort, Datum)

(Unterschrift)