Ludwig-Maximilians-Universität
München
Fakultät für Geowissenschaften

Masterarbeit

für die Prüfung zum Master of Science

im Studiengang Umweltsysteme und Nachhaltigkeit: Monitoring, Modellierung, Management

Systematische Evaluierung geeigneter Verfahren der künstlichen Intelligenz für die semantische Segmentierung von Landbedeckung in fernerkundungsgestützten Bilddaten

Julia Maria Holzner

Deutsches Zentrum
DLR für Luft- und Raumfahrt

Julia Maria Holzner

Systematische Evaluierung geeigneter Verfahren der künstlichen Intelligenz für die semantische Segmentierung von Landbedeckung in fernerkundungsgestützten Bilddaten

1. Betreuer: **Prof. Dr. Ralf Ludwig**,
Ludwig-Maximilians-Universität München,
Fakultät für Geowissenschaften,
Lehr- und Forschungseinheit für physische Geographie
und komplexe Umweltsysteme

2. Betreuer: **Dr. Michael Wurm**,
Deutsches Zentrum für Luft- und Raumfahrt (DLR),
Deutsches Fernerkundungsdatenzentrum (DFD),
Abteilung Georisiken und zivile Sicherheit,
Team Stadt und Gesellschaft

Matrikelnummer: 11555607

Abgabedatum: 18. November 2022

Zusammenfassung

Zusammenfassung

Im Zuge des globalen Wandels gewinnt die Analyse des Zustands der Erdoberfläche und ihrer Veränderungen immer mehr an Bedeutung. Die Bereitstellung geeigneter Informationen hinsichtlich der Landbedeckung und Landnutzung erfolgt dabei basierend auf der Auswertung fernerkundungsgestützter Bilddaten. Für die Auswertung werden gegenwärtig überwiegend hybride semi-automatische Klassifizierungsmethoden genutzt. Die aktuellen Entwicklungen im Hinblick auf eine steigende Datenqualität und stetig zunehmende verfügbare Datenmengen lassen jedoch den Bedarf nach geeigneteren Verfahren entstehen, um den Ansprüchen und dem Potenzial dieser Daten gerecht werden zu können. Durch entscheidende Fortschritte im Hardwarebereich und der Prozessparallelisierung rücken automatisierte, auf künstlicher Intelligenz basierende Klassifizierungsmethoden zunehmend in den Fokus der Wissenschaft. Diese Arbeit hat das Ziel, das Potenzial von *Deep Learning* Anwendungen für die Klassifizierung von Landbedeckung und Landnutzung zu evaluieren. Dazu wird ein neuronales Netzerk implementiert und mit einem fernerkundungsgestützten Bilddatensatz für Deutschland trainiert und getestet. Die gewählte Methode zeigt großes Potenzial hinsichtlich einer automatisierten Klassifizierung von Landbedeckung mit klassenspezifischen f1-Werten bis zu 0,93.

Abstract

In the course of global change, the analysis of the state of the earth's surface and its changes is increasingly important. The availability of suitable information regarding land cover and land use is based on the processing of remotely sensed image data. Currently, hybrid semi-automatic classification methods are predominantly used for this purpose. However, current developments in terms of increasing data quality and steadily growing amounts of available data raise the need for more suitable methods to meet the demands and potential of the data. Due to crucial advances in hardware and process parallelization, automated classification methods based on artificial intelligence gain more and more attention in the scientific community. This work aims to evaluate the potential of Deep Learning applications for land cover and land use classification. For this purpose, a neural network is implemented which is trained and tested with a remote sensing based image dataset for Germany. The chosen method shows great potential regarding an automated classification of land cover with class specific f1-scores up to 0.93.

Inhaltsverzeichnis

Abbildungsverzeichnis **7**

Tabellenverzeichnis **9**

1. Einleitung **10**
1.1. Motivation . . . 10
1.2. Problemstellung und Ziel der Arbeit . . . 11
1.3. Aufbau der Arbeit . . . 12

2. Daten und Methodik **13**
2.1. Bilddaten aus der Fernerkundung . . . 13
2.2. Digitales Landbedeckungsmodell . . . 15
2.2.1. Landbedeckung (lbakt) . . . 16
2.2.2. Landnutzung (lnakt) . . . 19
2.2.3. CORINE Land Cover (CLC) . . . 20
2.3. Deep Learning für Landbedeckungs- und Landnutzungsklassifizierung . 23
2.3.1. Deep Learning mit neuronalen Netzwerken . . . 23
2.3.2. Semantische Segmentierung mit Deep Learning . . . 25

3. Experimentelles Setup **34**
3.1. Vorbereitung des Datensatzes . . . 34
3.1.1. Szenenvorauswahl . . . 34
3.1.2. Datenvorbearbeitung . . . 35
3.1.3. Datensampling . . . 36
3.2. Hyperparameter . . . 40
3.2.1. Datenaugmentierung . . . 41
3.2.2. Batch Size und Batch Normalisierung . . . 41
3.2.3. Aktivierungsfunktion . . . 41
3.2.4. Verlustfunktion . . . 42
3.2.5. Lernrate und Epochenzahl . . . 44
3.2.6. Optimierer . . . 44

4. Ergebnisse **45**
4.1. Experimente . . . 45
4.2. Metriken zur Bewertung der Modellgüte . . . 46
4.3. Einfluss der Samplingmethode auf Datensatz- und Modellgüte . . . 48
4.4. Landbedeckungsmodell (lbakt) . . . 50

4.5. Aggregiertes Landbedeckungsmodell (lbakt) 53
4.6. Landnutzungsmodell (lnakt) 56
4.7. CORINE Land Cover (CLC) 59

5. Diskussion **63**
5.1. Einfluss der Sampling Methode 63
5.2. Klassifizierungsergebnisse 63

6. Zusammenfassung **67**

A. Anhang **68**
A.1. Einfluss der Samplingmethode auf die Qualität der Teildatensätze . . . 68
A.2. Landbedeckungsmodell (lbakt) - Stratifiziertes Sampling 69
A.3. Landnutzungsmodell (lnakt) - Stratifiziertes Sampling 71
A.4. CORINE Land Cover (CLC) - Stratifiziertes Sampling 73

Literatur **75**

Abbildungsverzeichnis

2.1. RapidEye2015 Mosaik 14
2.2. RapidEye2015 Datensatz - Szenenauswahl 15
2.3. Klassenverteilung im lbakt-Datensatz 17
2.4. Szenen pro Klasse für lbakt 18
2.5. Klassenverteilung im lnakt-Datensatz 19
2.6. Szenen pro Klasse für lnakt 20
2.7. Klassenverteilung im CLC-Datensatz 22
2.8. Aufbau eines Neuronalen Netzwerks 24
2.9. Vergleich eines normalen Neuronalen Netzwerks mit einem Konvolutionsnetzwerk 25
2.10. Aufbau eines Konvolutionsnetzwerkes 26
2.11. Vanishing/Exploding Gradient in tiefen neuronalen Netzwerken 27
2.12. Ein Residualblock 27
2.13. Entwicklung eines Residualen Netzwerks aus einem einfachen Netzwerk 29
2.14. Training von einfachen Netzen und Residualnetzen auf dem ImageNet-Datensatz 30
2.15. Bottleneck-Blöcke für tiefere Residualnetzwerke 31
2.16. Training von tieferen Residualnetzwerken 31
2.17. Aufbau eines Feature Pyramid Networks 32
2.18. Ein Upsampling-Block mit lateraler und Top-Down-Verbindung 33

3.1. RapidEye2015 Datensatz für Deutschland 35
3.2. Einfluss der Random Sampling Methode auf die Qualität der Teildatensätze 37

4.1. Konfusionsmatrix zur Berechnung der klassenspezifischen Gütemaße . . 47
4.2. Ergebnisse des Samplings für lbakt 48
4.3. Konfusionsmatrix für Experiment 11 50
4.4. Beispiele der Klassifizierungsergebnisse des Modells aus Experiment 11 51
4.5. Klassenspezifische Metriken für das Modell aus Experiment 11 52
4.6. Aggregierungsschema für Experiment 13 53
4.7. Konfusionsmatrix für Experiment 13 54
4.8. Beispiele der Klassifizierungsergebnisse des Modells aus Experiment 13 55
4.9. Klassenspezifische Metriken für das Modell aus Experiment 13 56
4.10. Konfusionsmatrix für Experiment 14 57
4.11. Beispiele der Klassifizierungsergebnisse des Modells aus Experiment 14 58
4.12. Klassenspezifische Metriken für das Modell aus Experiment 14 59

4.13. Konfusionsmatrix für Experiment 16 60
4.14. Beispiele der Klassifizierungsergebnisse des Modells aus Experiment 16 61
4.15. Klassenspezifische Metriken für das Modell aus Experiment 16 62

5.1. Landbedeckungs- und Landnutzungsinformation für eine exemplarische Rapideye-Szene 65

A.1. Ergebnisse des Samplings für lnakt 68
A.2. Ergebnisse des Samplings für CLC 68
A.3. Konfusionsmatrix für Experiment 12 69
A.4. Klassenspezifische Metriken für das Modell aus Experiment 12 70
A.5. Konfusionsmatrix für Experiment 15 71
A.6. Klassenspezifische Metriken für das Modell aus Experiment 15 72
A.7. Konfusionsmatrix für Experiment 17 73
A.8. Klassenspezifische Metriken für das Modell aus Experiment 17 74

Tabellenverzeichnis

2.1. Instrumentenparameter der RapidEye Mission 13
2.2. lbakt-Nomenklatur 16
2.3. lnakt-Nomenklatur 19
2.4. CLC-Nomenklatur 21
2.5. Deep Learning Methoden der Bildanalyse mit Neuronalen Netzwerken . 24

3.1. Klassenspezifische Ungleichgewichte der Pixelverteilung auf Szenenebene 38
3.2. Szenenanzahl der Teildatensätze für Training, Validierung und Test . . 39
3.3. Experimente zur Anpassung der Hyperparameter 40

4.1. Durchgeführte Experimente 45
4.2. Gütemaße für die Experimente mit LBM-DE 49

1. Einleitung

1.1. Motivation

Die Klassifizierung von Landbedeckung und Landnutzung ist ein zentraler Bestandteil der Fernerkundung. Besonders hinsichtlich der zunehmenden Auswirkungen der anthropogenen Einflussnahme auf die Erdoberfläche ist eine detaillierte Zustands- und Änderungsanalyse der Landbedeckung und Landnutzung entscheidend. Geodatenprodukte mit Informationen über die Beschaffenheit der Erdoberfläche sind essenziell für zahlreiche wissenschaftliche und operationelle Anwendungen, sowie für Entscheidungsträger der Politik, Industrie, etc. Die Anwendungsfelder sind dabei vielfältig, wie zum Beispiel urbane und ländliche Landnutzungsplanungen, Ressourcenmanagement, sowie Katastrophenmanagement und -vorsorge (Hu u. a., 2015). Für diese Anwendungen werden hochauflösende und aktuelle Informationen benötigt. Aktuell werden die erforderlichen Daten überwiegend in hybriden Ansätzen mit semi-automatischen Methoden gewonnen. Entsprechende Datenbanken wie zum Beispiel die *CORINE Land Cover* der europäischen Union (Copernicus, 2022b) oder das Landbedeckungsmodell für Deutschland (Bundesamt für Kartographie und Geodäsie (BKG), 2021) stehen frei zu Verfügung, allerdings können sie die steigenden Ansprüche an Auflösung und Aktualität durch die ihnen zugrundeliegende Methodik nur bedingt erfüllen. Die Bereitstellung höher aufgelöster Informationen ist in der traditionellen Klassifizierungsmethodik mit hohen Kosten und langen Produktionszeiten verbunden. Außerdem können Genauigkeit und Konsistenz der Informationen zeitlich und abhängig vom Bearbeiter variieren.

Die Fernerkundung wir zudem zunehmend mit der Herausforderung der Verarbeitung von *Big Data* konfrontiert. Mit dem Beginn des europäischen Erdbeobachtungsprogramms Copernicus stehen seit 2014 große und weiterhin anwachsende Mengen von Erdbeobachtungsdaten zur Verfügung (Copernicus, 2022a; Zhu u. a., 2017; Camps-Valls u. a., 2021). Dadurch entsteht der Bedarf nach geeigneteren Klassifizierungsmethoden, um diese aktuellen Herausforderungen zu bewältigen und den Anforderungen gerecht zu werden. In den letzten zehn Jahren entwickelte sich *Deep Learning* als neue Methode zum stark wachsenden Forschungszweig der fernerkungsgestützten Landbedeckungs- und Landnutzungsklassifizierung (Camps-Valls u. a., 2021). Neben der zunehmenden Verfügbarkeit geeigneter Datenmengen waren dafür Entwicklungen im Hardwarebereich, in Form von leistungfähigeren Grafikprozessoren, und Fortschritte in der Prozessparallelisierung von entscheidender Bedeutung. Gegenwärtig wird *Deep Learning* verstärkt für neue Ansätze der Datenverarbeitung in der Fernerkundung

genutzt (Reichstein u. a., 2019). Traditionelle, *hand-crafted* Methoden (DeFries u. a., 1999; Hansen u. a., 2012) werden abgelöst. Das Forschungsfeld wächst seit 2014 stetig an (Camps-Valls u. a., 2021). Daten von verschiedenen Sensoren werden in Kombination mit unterschiedlichen *Deep Learning* Anwendungen für ein breites Spektrum spezifischer Forschungsfragen genutzt: Im urbanen Raum können beispielsweise strukturelle Änderungen analysiert (Mansour u. a., 2021; McGlinchy u. a., 2019), informelle Siedlungsräume identifiziert (Stark u. a., 2020; Wurm, Stark u. a., 2019), oder die Bebauung hinsichtlich verschiedener Faktoren, wie zum Beispiel ihrer Energiebilanz (Wurm, Droin u. a., 2021), bewertet werden. Im Agrarbereich werden neuronale Netzwerke zur Feststellung spezifischer Anbausorten genutzt (Kussul u. a., 2017). Für die flächendeckende Klassifizierung von Landbedeckung und Landnutzung auf Grundlage fernerkundungsgestützter Bilddaten existieren international bereits mehrere verschiedene Ansätze im *Deep Learning* Bereich (Voelsen u. a., 2022; Stoian u. a., 2019; Karra u. a., 2021; Alp u. a., 2022; Storie u. a., 2018; Castelluccio u. a., 2015).

1.2. Problemstellung und Ziel der Arbeit

Das Landbedeckungsmodell für Deutschland (LBM-DE), basierend auf dem amtlichen topographischen-kartograpischen Informationssystem (ATKIS) stellt den nationalen Beitrag Deutschlands zum europäischen Copernicus-Programm (Copernicus, 2022a) dar. Das Hauptanwendungsziel ist die Ableitung der CORINE *Land Cover* (CLC) für Deutschland durch die Bestimmung und Analyse landschaftlicher Zustände und Änderungen. Das LBM-DE wird aktuell durch hybride semi-automatische Klassifizierungsmethoden erzeugt. Die vorliegende Arbeit untersucht, inwiefern diese Prozesse durch die Nutzung von *Deep Learning* Methoden automatisiert werden können, um den gegenwärtigen Herausforderungen steigender Datenmengen mit zunehmender Komplexität und Auflösung gerecht zu werden. Dazu wird eine *Deep Learning* Methode ausgearbeitet und hinsichtlich ihrer Eignung zur Klassifizierung von Landbedeckung und Landnutzung im Rahmen mehrerer Experimente getestet. Als Datengrundlage dienen Fernerkundungsdaten der RapidEye Mission sowie die Landbedeckungs- und Landnutzungsinformationen des LBM-DE. Das Ziel der Arbeit ist eine Einschätzung, inwiefern die Landbedeckungs- und Landnutzungsinformation für Deutschland durch Methoden der künstlichen Intelligenz automatisiert erzeugt werden kann.

1.3. Aufbau der Arbeit

Im ersten Abschnitt der Arbeit (Kapitel 2) werden die genutzten Daten vorgestellt und anschließend die Grundlagen der gewählten *Deep Learning* Architektur erläutert. Das darauf folgende Kapitel (Kapitel 3) beschreibt das experimentelle Setup. Dazu werden zuerst die nötigen Schritte der Datenvorbereitung erläutert und anschließend die für die Anwendung gewählten Hyperparametereinstellungen vorgestellt. Kapitel 4 stellt die Ergebnisse der durchgeführten Experimente vor: diese werden in Kapitel 5 diskutiert. Kapitel 6 schließt die Arbeit mit einer Zusammenfassung der gewonnenen Erkenntnisse ab.

2. Daten und Methodik

Nachfolgend werden die für die Arbeit genutzten Daten und die gewählte Methodik erläutert. Als Datengrundlage werden einerseits Bilddaten aus der Fernerkundung (Kapitel 2.1) und andererseits die entsprechenden Daten der zugehörigen Landbedeckung beziehungsweise Landnutzung (Kapitel 2.2) verwendet. Die Daten sowie deren Eigenschaften werden im Anschluss vorgestellt. Die danach folgende Methodik der Arbeit basiert auf der Wahl einer geeigneten Anwendung der künstlichen Intelligenz für den Trainings- und Klassifizierungsprozess. Charakteristika und Funktionsweise der gewählten Architekturen werden in Kapitel 2.3 aufgezeigt.

2.1. Bilddaten aus der Fernerkundung

Bei den genutzten Fernerkundungdaten handelt es sich um einen für Deutschland flächendeckenden Datensatz von RapidEye (The European Space Agency (ESA), 2022a) mit Referenzjahr 2015. Aufgrund der Ausstattung mit hochauflösenden optischen Kameras, sowie der hohen Aktualität der Aufnahmen durch eine kurze globale Wiederkehrzeit eignen sich die Bilddaten von RapidEye für vielfältige Anwendungen mit hohem Detailanspruch. Tabelle 2.1 fasst die Instrumentenparameter von RapidEye zusammen.

Tabelle 2.1.: Instrumentenparameter der RapidEye Mission (Bundesamt für Kartographie und Geodäsie (BKG), 2021; The European Space Agency (ESA), 2022a; Deutsche Raumfahrtagentur, 2008).

Orbit	Typ: sonnensynchron Höhe: 630 km Inklination: 97,9° Umlaufdauer: 96.7 min Globale Wiederkehrzeit: 5 Tage
Auflösung	6,5 m im Nadir, resampled auf 5 m Pixelgröße bei orthorektifizierten Produkten
Schwadweite	± 6,75° über dem Nadir (entspricht 77 km)
Bänder	Blau (440-510 nm) Grün (520-590 nm) Rot (630-685 nm) Red Edge (690-730 nm) NIR (760-850 nm)

Der für die Arbeit vorliegende RapidEye Datensatz besteht aus 842 Szenen in 16 bit in den fünf zur Verfügung stehenden Kanälen (Tabelle 2.1). Die Bodenpixelauflösung beträgt fünf Meter. Abbildung 2.1 zeigt den Datensatz als Mosaik.

Abbildung 2.1.: RapidEye2015 Mosaik. Die im Datensatz enthaltene Meeresfläche ist nicht mit abgebildet. Die im Datensatz enthaltenen Szenen reichen über die Bundesgrenze hinaus.

Die hohe Auflösung der RapidEye Szenen ermöglicht eine detaillierte Untersuchung der Erdoberfläche, beispielsweise für Fragestellungen hinsichtlich der Vegetationsdecke, der Gletscher- und Schneebedeckung, der Meereisausprägung oder auch der Landbedeckung und Landnutzung (Zhang u. a., 2021; The European Space Agency (ESA), 2022b). Abbildung 2.2 zeigt eine Auswahl von Aufnahmen aus dem Datensatz. Verschiedene Bedeckungsarten wie Bebauung, Ackerland, Baumbestand und Gewässer sind deutlich erkennbar. Ebenso visuell unterscheidbar sind Nutzungsarten wie Verkehrsinfrastruktur (Straßen, Schienennetze, Flughäfen), Wohnsiedlungen, oder Anbaugebiete.

Abbildung 2.2.: RapidEye2015 Datensatz - Szenenauswahl. Verschiedene Bebauungsstrukturen (oben) wie Wohnsiedlung, Industrie und Infrastruktur, sowie Anbaugebiete, naturnahe Flächen und Gewässer (unten) sind visuell deutlich unterscheidbar.

2.2. Digitales Landbedeckungsmodell

Die Datengrundlage für die Landbedeckung und Landnutzung liefert das Digitale Landbedeckungsmodell für Deutschland (LBM-DE) (Bundesamt für Kartographie und Geodäsie (BKG), 2021). Es beschreibt die geometrischen Objekte der Landbedeckung und Landnutzung im Vektorformat. Der Datensatz beinhaltet lückenlos und überlappungsfrei aneinandergrenzende Flächenobjekte. Im Datensatz enthalten sind die Modelle der Landbedeckung (lbakt), der Landnutzung (lnakt), sowie die CORINE

Land Cover. Seit 2012 erfolgt die räumlich vollständige Erhebung beziehungsweise Aktualisierung im Abstand von drei Jahren. Der vorliegende Datensatz LBM-DE2015 hat das Referenzjahr 2015. Als Datenquelle werden flächenhafte Objekte des ATKIS Basis-DLM, die Vorgängermodelle von LBM-DE, Aufnahmen der Sensoren RapidEye sowie DMC, Image2015 Daten und digitale Orthophotos genutzt (Hovenbitzer u. a., 2014). Die Mindestkartierfläche für Objekte im LBM-DE Datensatz beträgt 1 ha, die Mindestkartierbreite 15 m. Unterschreiten Objekte diese Vorgaben, werden sie nicht in den Datensatz aufgenommen sondern generalisiert: Die Ähnlichkeit zu benachbarten Flächen entscheidet, welcher Klasse sie zugeordnet werden. Diese Methode kann für automatisierte Prozesse Probleme verursachen, da es sich um die subjektiv visuelle Wahrnehmung des Bearbeiters handelt. Die Klassifizierung von Mischflächen beruht auf der Struktur und Verteilung der Einzelbestandteile. Enthält eine potentielle Mischfläche Einzelbestandteile, die größer sind als 1 ha, müssen diese separat entsprechend ihrer Landbedeckungsklasse erfasst werden. Die minimale Kartierfläche des europäischen CORINE-Datensatzes liegt bei 25 ha, die minimale Kartierbreite bei 100 m. Letztere greift beispielsweise bei Flüssen oder Straßen, die mindestens 100 m lang sein müssen, um in den Datensatz aufgenommen werden zu können. Seit 2012 liegt parallel dazu auf nationaler Ebene ein Datensatz mit einer Mindeskartierfläche von 5 ha als öffentlich zugänglicher Datensatz vor. Dieser wird für diese Arbeit genutzt.

Die folgenden Abschnitte stellen die im LBM-DE enthaltenen Datensätze der Landbedeckung und Landnutzung, sowie der CORINE *Land Cover* vor.

2.2.1. Landbedeckung (lbakt)

Die Landbedeckung beschreibt die physische Beschaffenheit der Erdoberfläche. Die Einteilung in Klassen erfolgt nach verschiedenen naturnahen oder anthropogen geprägten Kategorien. Das vorliegende Landbedeckungsmodell besteht aus 31 Klassen (Tabelle 2.2).

Tabelle 2.2.: lbakt-Nomenklatur (Bundesamt für Kartographie und Geodäsie (BKG), 2021).

Klasse 0	nodata	Klasse 16	Nadelbäume
Klasse 1	Bebauung	Klasse 17	Laub-/ Nadelbäume
Klasse 2	Anlagen	Klasse 18	Sand, Steine, Erde
Klasse 3	versiegelte gebäudelose Flächen	Klasse 19	Fels
Klasse 4	Mischflächen	Klasse 20	Brandfläche
Klasse 5	Ackerland	Klasse 21	Schnee und Eis
Klasse 6	Weinbau	Klasse 22	Sumpf
Klasse 7	Obst- und Beerenobst	Klasse 23	Moor
Klasse 8	Hopfen	Klasse 24	Sumpf mit Bäumen
Klasse 9	Homogenes Grünland	Klasse 25	Moor mit Bäumen
Klasse 10	Inhomogenes Grünland	Klasse 26	Watt
Klasse 11	Grasland mit Bäumen	Klasse 27	Wasserlauf
Klasse 12	Zwergsträucher (Heide)	Klasse 28	Wasserfläche
Klasse 13	Büsche und Sträucher	Klasse 29	Lagune
Klasse 14	Aufforstung	Klasse 30	Mündungstrichter
Klasse 15	Laubbäume	Klasse 31	Offenes Meer

Die Klassen lassen sich in sieben Überkategorien gliedern: Bebauung und versiegelte Flächen (Klasse 1-4), Anbauflächen (Klasse 5-8), Grünland (Klasse 9-11), Flächen mit Strauch- und Baumbewuchs (Klasse 12-17), natürliche Flächen ohne Bewuchs (Klasse 18-21), Sumpf- und Moorflächen (Klasse 22-25) und Wasserflächen (Klasse 26-31). Neben in sich homogenen Bedeckungsarten wie beispielsweise Ackerland, Homogenes Grünland, Laubbäume, oder Wasserflächen sind Mischklassen wie Mischflächen, Grasland mit Bäumen oder Sumpf mit Bäumen definiert, deren Zusammensetzung heterogener ist.

Auf Szenen- und Pixelebene zeigt der Datensatz einzelne Klassen betreffend starke Klassenungleichgewichte (Abbildung 2.3). Die Ungleichgewichte sind unter anderem auf die Regionalität kleinerer Klassen zurückzuführen (Abbildung 2.4).

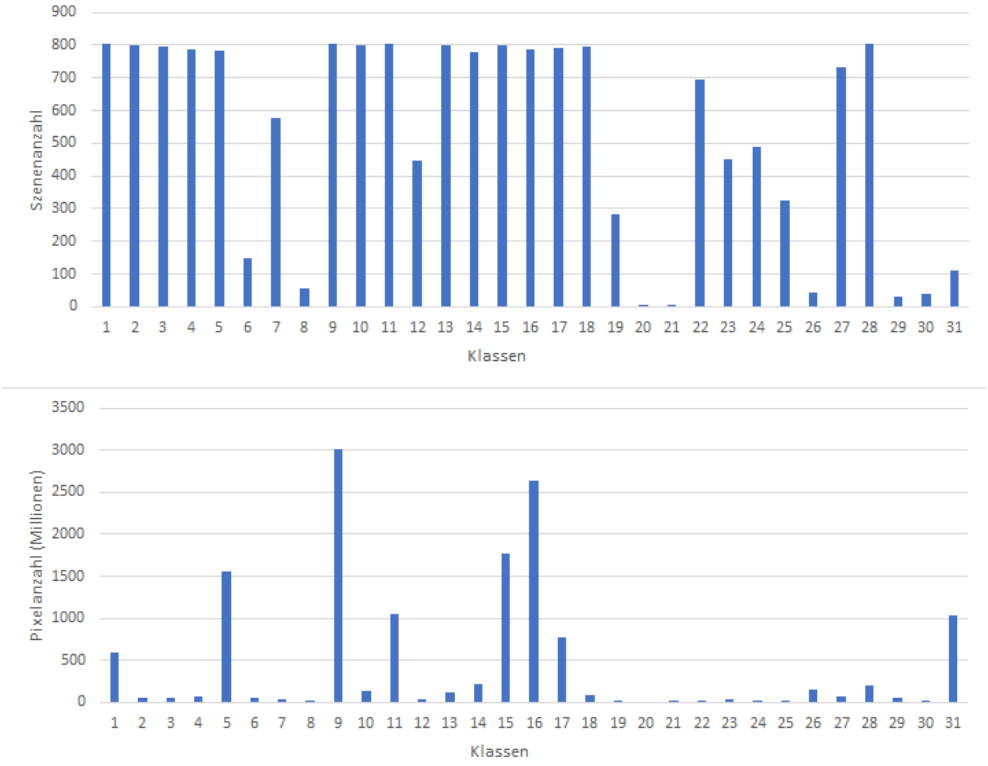

Abbildung 2.3.: Klassenverteilung im lbakt-Datensatz. Oben: Szenenanzahl pro Klasse. Unten: Pixelanzahl pro Klasse in Millionen.

Bei den Klassen Weinbau und Hopfen handelt es sich um Sonderkulturen, die in Deutschland nur auf kleiner Fläche angebaut werden. Die Klassen Watt, Lagune, Mündungstrichter und Offenes Meer sind regional stark begrenzte Wasserklassen, die nur auf wenigen Szenen an der Norddeutschen Küste auftreten. Gleiches gilt für Schnee und Eis: Die Klasse tritt nur in der süddeutschen Alpenregion auf. Die Klasse Brandfläche kommt nur auf einer Szene vor. Auf Pixelebene dominieren die Klassen Ackerbau, homogenes Grünland, sowie Laub- beziehungsweise Nadelbäume.

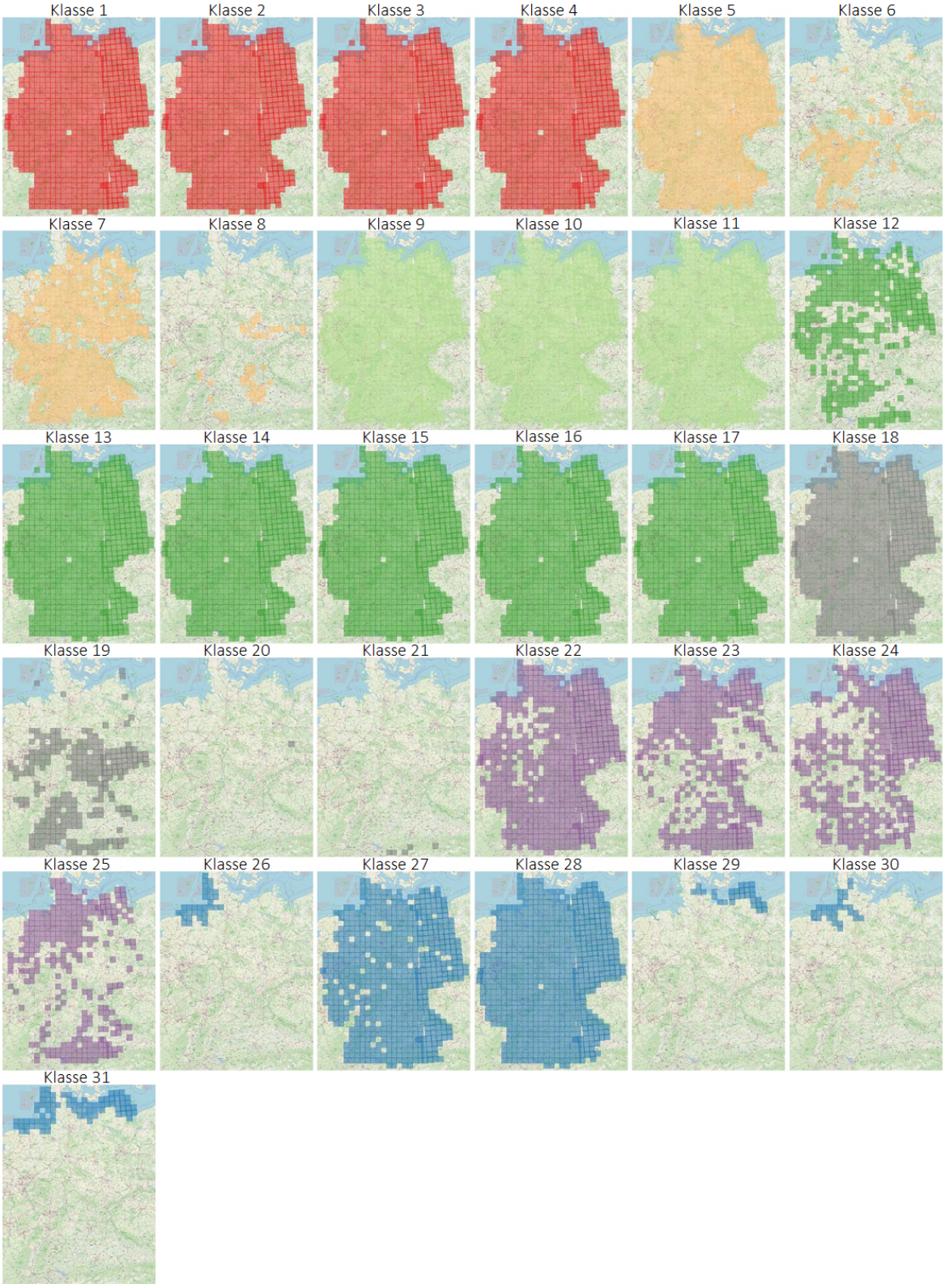

Abbildung 2.4.: Szenen pro Klasse für lbakt, farblich sortiert nach den fünf Hauptgruppen Bebauung und versiegelte Flächen (rot), Anbauflächen (gelb), Grünland (hellgrün), Flächen mit Strauch- und Baumbewuchs (dunkelgrün), natürliche Flächen ohne Bewuchs (grau), Sumpf- und Moorflächen (violett) und Wasserflächen (blau).

2.2.2. Landnutzung (lnakt)

Die Landnutzung beschreibt die Art der anthropogenen Inanspruchnahme der Erdoberfläche. Sie unterscheidet sich von der Landbedeckung dahingehend, dass hier der Fokus auf dem Nutzungsaspekt liegt. Die Flächennutzung legt den Schwerpunkt stärker auf die bauliche Prägung der Erdoberfläche, während die Landnutzung überwiegend unbebautes Gelände spezifiziert. Das vorliegende Landnutzungsmodell beinhaltet 15 Klassen (Tabelle 2.3). Sie lassen sich in die vier Kategorien Bebauung, Verkehrsinfrastruktur, unbebaute (naturnahe) Flächen und Wasser untergliedern.

Tabelle 2.3.: lnakt-Nomenklatur (Bundesamt für Kartographie und Geodäsie (BKG), 2021).

Klasse 0	keine Nutzung erkennbar	Klasse 8	Flugverkehr
Klasse 1	Wohnen	Klasse 9	Sport und Freizeit
Klasse 2	Produktion	Klasse 10	Städtische Grünfläche
Klasse 3	Öffentlichkeit	Klasse 11	Wasser
Klasse 4	Hafen	Klasse 12	Landwirtschaft (intensiv)
Klasse 5	Abbaufläche	Klasse 13	Extensive Nutzung
Klasse 6	Deponie	Klasse 14	Forstwirtschaft
Klasse 7	Straßen und Bahnverkehr	Klasse 15	Im Bau

Wie auch bei lbakt treten Klassenungleichgewichte auf. Auf Szenenebene sind diese wesentlich geringer (Abbildung 2.5).

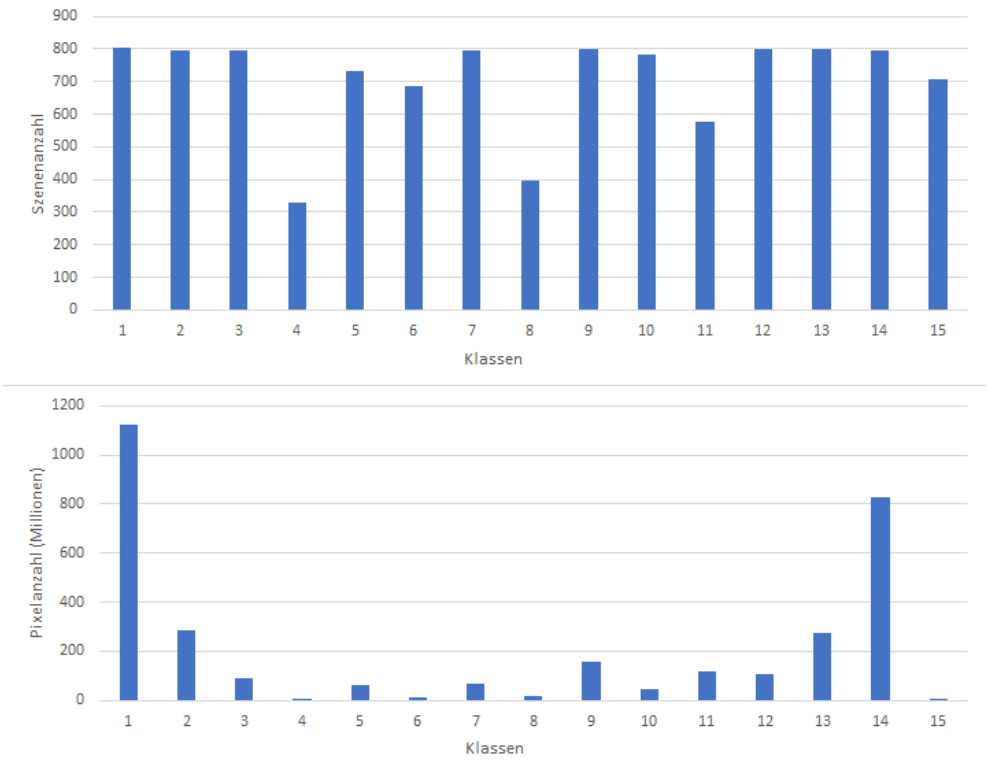

Abbildung 2.5.: Klassenverteilung im lnakt-Datensatz. Oben: Szenenanzahl pro Klasse. Unten: Pixelanzahl pro Klasse in Millionen.

Die kleinsten Klassen Hafen und Flugverkehr sind jeweils auf über 300 der 842 Szenen vertreten und weisen keine starke Regionalität auf (Abbildung 2.6). Deutlicher werden die Ungleichgewichte auf Pixelebene (Abbildung 2.5). Die beiden größten Klassen Wohnen und Forstwirtschaft dominieren. Die auf Szenenebene häufig vorkommenden Klassen weisen auf Pixelebene eine starke Kleinräumigkeit auf.

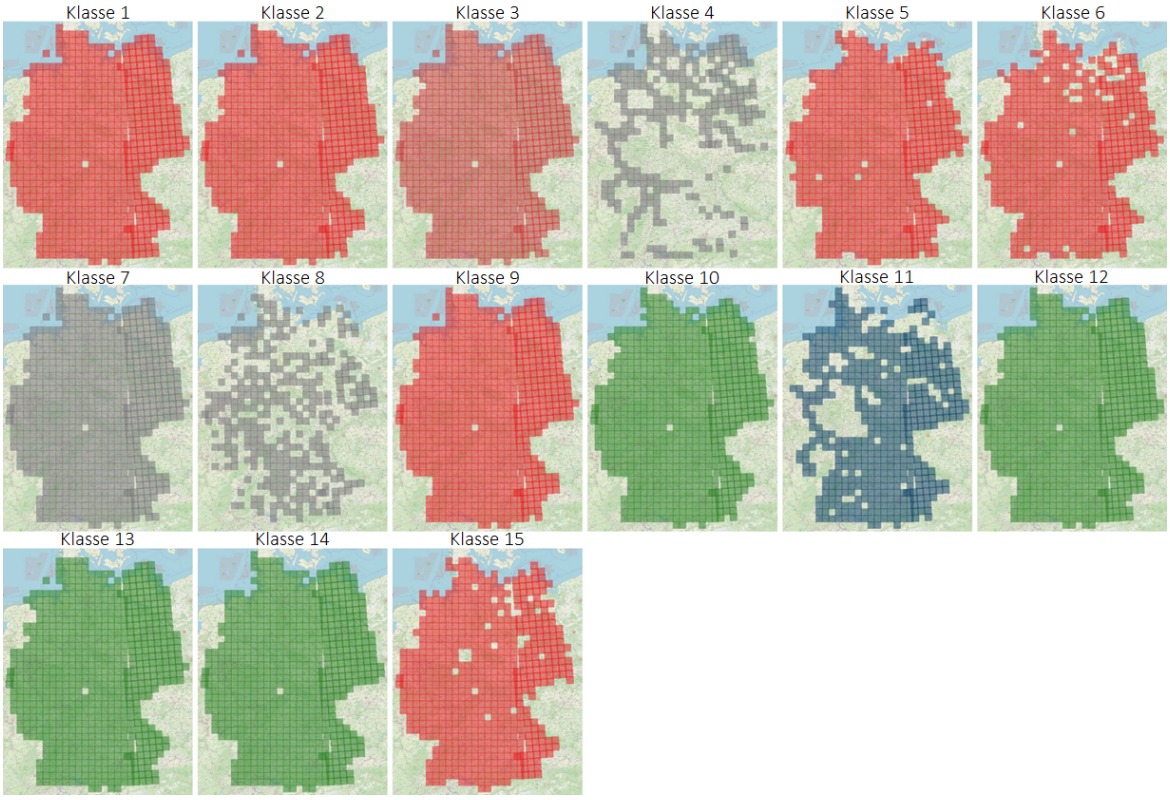

Abbildung 2.6.: Szenen pro Klasse für lnakt, farblich sortiert nach den vier Hauptgruppen Bebauung (rot), Verkehrsinfrastruktur (grau), unbebaute (naturnahe) Flächen (grün) und Wasserflächen (blau).

2.2.3. CORINE Land Cover (CLC)

Das CORINE-Programm (*Coordination of Information on the Environment*) wurde von der Europäischen Kommission eingeführt und dient der Standardisierung der Datenerhebung für Landbedeckung und Landnutzung in Europa (Copernicus, 2022b; Büttner, 2014). Die Teildatensätze werden auf nationaler Ebene erstellt und dann auf europäischer Ebene zur *CORINE Land Cover* zusammengeführt. Seit 2000 wird alle sechs Jahre ein nationaler CLC-Datensatz für Deutschland an die europäische Umweltagentur übergeben. Die Methodik zur Ableitung des Datensatzes beruht auf einer Kreuztabelle, welche die Informationen lbakt und lnakt kombiniert. Zusätzlich werden der Vegetationsanteil und der Versiegelungsgrad der untersuchten Fläche berücksichtigt (Hovenbitzer u. a., 2014). Auf europäischer Ebene besteht die Nomenklatur der

CORINE *Land Cover* aus 44 Klassen, davon sind 35 in dem vorliegenden Datensatz für Deutschland enthalten (Tabelle 2.4).

Tabelle 2.4.: CLC-Nomenklatur. Die blau gedruckten Klassen kommen im Datensatz für Deutschland nicht vor (Bundesamt für Kartographie und Geodäsie (BKG), 2021).

Klasse 0	nodata
Klasse 1	durchgängig städtische Prägung
Klasse 2	nicht durchgängig städtische Prägung
Klasse 3	Industrie- und Gewerbeflächen, öffentliche Einrichtungen
Klasse 4	Straßen-, Eisenbahnnetze und funktionell zugeordnete Flächen
Klasse 5	Hafengebiete
Klasse 6	Flughäfen
Klasse 7	Abbauflächen
Klasse 8	Deponien und Abraumhalden
Klasse 9	Baustellen
Klasse 10	Städtische Grünflächen
Klasse 11	Sport- und Freizeitanlagen
Klasse 12	nicht bewässertes Ackerland
Klasse 13	Regelmäßig bewässertes Ackerland
Klasse 14	Reisfelder
Klasse 15	Weinbauflächen
Klasse 16	Obst- und Beerenobstbestände
Klasse 17	Olivenhaine
Klasse 18	Wiesen und Weiden
Klasse 19	Einjährige Kulturen in Verbindung mit Dauerkulturen
Klasse 20	Komplexe Parzellenstrukturen
Klasse 21	Landwirtschaftlich genutztes Land mit Flächen natürlicher Bodenbedeckung von signifikanter Größe
Klasse 22	Land- und Forstwirtschaftliche Flächen
Klasse 23	Laubwälder
Klasse 24	Nadelwälder
Klasse 25	Mischwälder
Klasse 26	Natürliches Grünland
Klasse 27	Heiden und Moorheiden
Klasse 28	Hartlaubbewuchs
Klasse 29	Wald-Strauch-Übergangsstadien
Klasse 30	Strände, Dünen und Sandflächen
Klasse 31	Felsflächen ohne Vegetation
Klasse 32	Flächen mit spärlicher Vegetation
Klasse 33	Brandflächen
Klasse 34	Gletscher und Dauerschneegebiete
Klasse 35	Sümpfe
Klasse 36	Torfmoore
Klasse 37	Salzwiesen
Klasse 38	Salinen
Klasse 39	In der Gezeitenzone liegende Flächen
Klasse 40	Gewässerläufe
Klasse 41	Wasserflächen
Klasse 42	Lagunen
Klasse 43	Mündungsgebiete
Klasse 44	Meere und Ozeane

Es handelt sich sowohl um Landnutzungs-, Landbedeckungs-, als auch Mischklassen. Die Nomenklatur ist hierarchisch aufgebaut, es ergeben sich fünf Hauptgruppen ähnlich denen der Landbedeckung:

1. Bebaute Flächen (Klasse 1-11)
2. Landwirtschaftliche Flächen (Klasse 12-22)
3. Wälder und naturnahe Flächen (Klasse 23-34)
4. Feuchtflächen (Klasse 35-39)
5. Wasserflächen (Klasse 40-44)

Auch bei CORINE treten, sowohl auf Szenen- als auch auf Pixelebene, starke, teils durch Regionalität bedingte Klassenungleichgewichte auf (Abbildung 2.7), die sich aus den kleinen Klassen von lbakt und lnakt ableiten lassen. Die kleinsten, in Deutschland vertretenen Klassen des Datensatzes sind, wie bei lbakt, die Klassen Brandfläche sowie Gletscher und Dauerschneegebiete.

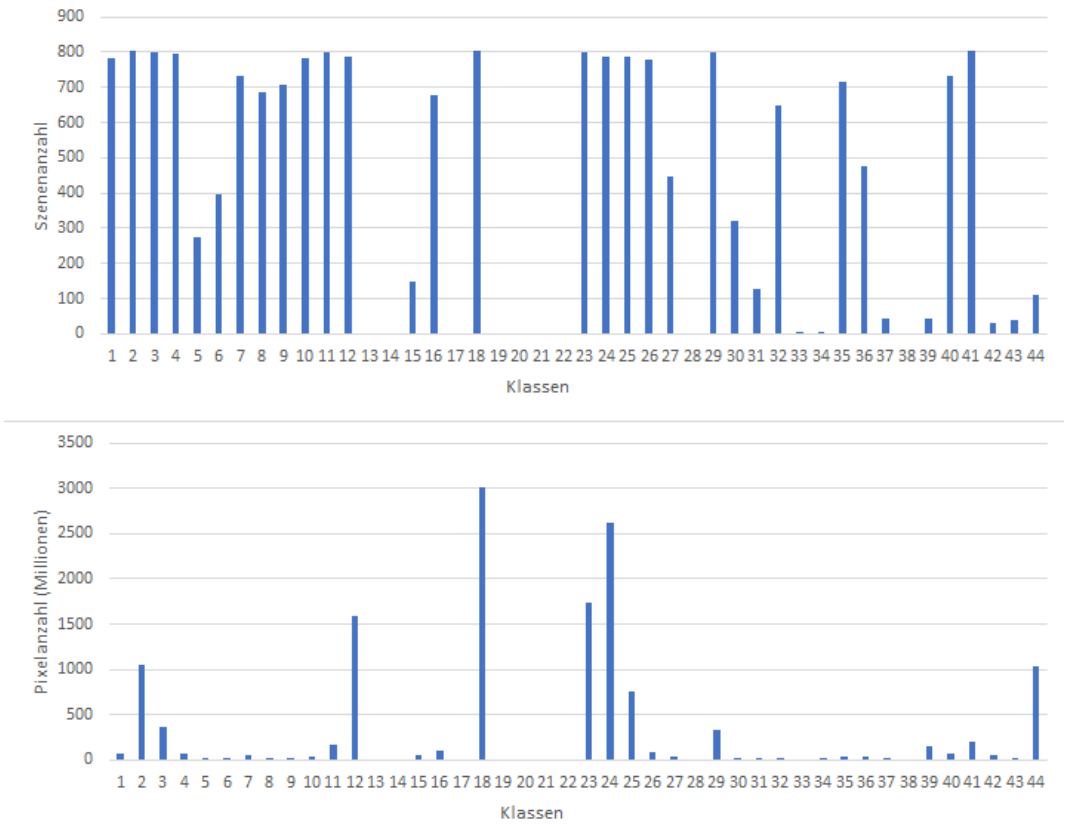

Abbildung 2.7.: Klassenverteilung im CLC-Datensatz. Oben: Szenenanzahl pro Klasse. Unten: Pixelanzahl pro Klasse in Millionen.

Auf Pixelebene dominieren wenige große Klassen. Die größte Klasse ist die Klasse Wiesen und Weiden, gefolgt von Nadelwäldern, Laubwäldern und nicht bewässertem Ackerland.

2.3. Deep Learning für Landbedeckungs- und Landnutzungsklassifizierung

Mit der zunehmenden räumlichen Auflösung von Fernerkundungsbildern können herkömmliche Klassifizierungsansätze die spektralen und räumlichen Informationen der Daten nicht mehr vollständig abbilden, was zu Qualitätseinbußen bei den Klassifizierungsergebnissen führt. Aktuell zeigen Methoden der semantischen Segmentierung mit tiefen neuronalen Netzwerken ein großes Potenzial zur Lösung dieser Herausforderung (Pedrayes u. a., 2021).

Nachfolgend werden die Grundprinzipien und die Funktionsweise neuronaler Netzwerke erläutert. Anschließend wird die für diese Arbeit gewählte Analysemethode, die semantische Segmentierung, vorgestellt und die ihr zugrundeliegende Netzwerkarchitektur dargestellt und diskutiert.

2.3.1. Deep Learning mit neuronalen Netzwerken

Deep Learning ist der am schnellsten wachsende Teilbereich in der Analyse von *Big Data* und zählte 2013 zu den zehn bahnbrechendsten Technologien des Jahres (MIT Technology Review, 2022). *Deep Learning* ist eine Methode der Informationsverarbeitung und ein Teilbereich des maschinellen Lernens (Goodfellow u. a., 2016). In den letzten zehn Jahren entwickelte sich *Deep Learning* durch entscheidende technologische Innovationen und der stetig steigenden Menge verfügbarer Daten hier zum führenden Paradigma (Camps-Valls u. a., 2021). *Deep Learning* nutzt neuronale Netzwerke mit hierarchischen Architekturen von zunehmender Komplexität, um große Datensätze zu analysieren (Reichstein u. a., 2019). Bei Neuronalen Netzwerken handelt es sich um eine Methode des *Supervised Learning*: dem Netzwerk werden Eingaben mit den gewünschten Ausgaben übergeben, um daraus die jeweiligen Beziehungen zwischen Ein- und Ausgabe zu erlernen. Bei *Unsupervised Learning* würden dem Netzwerk nur die Eingaben übergeben werden.

Ein neuronales Netzwerk besteht aus mindestens drei Schichten: Der Eingabeschicht, einer oder mehr verborgenen Schichten und der Ausgabeschicht (Abbildung 2.8). Die Anzahl der Schichten bezeichnet die Netzwerktiefe (Ketkar u. a., 2021). Von einem tiefen Netzwerk spricht man bei mehr als zwei verborgenen Schichten. Eine Schicht ist jeweils aus Neuronen aufgebaut und mit der Nächsten verknüpft. Jede dieser Verbindungen besitzt ein spezifisches Gewicht, das die Stärke des Signals bestimmt, das durch die Verbindung geleitet wird. Wird eine Eingabe in der *Forward Propagation* durch diese Verbindung geleitet, wird sie mit der Gewichtung der Verbindung multipliziert. Das Ergebnis der Matrixmultiplikation geht dann in eine Aktivierungsfunktion ein (Pointer, 2021). Aktivierungsfunktionen fügen dem neuronalen Netz Nichlinearität hinzu. Dadurch kann das Netz komplexere Daten modellieren. Das Ergebnis der Aktivierungsfunktion gelangt in die nächste Schicht. Diese Prozesse entsprechen dem vorwärtsgerichteten Teil eines Netzwerks. Der zweite zentrale Bestandteil eines neu-

ronalen Netzes ist die *Backpropagation* mit Gradientenabstieg. Dabei werden Informationen rückwärtsgerichtet durch das Netzwerk geleitet, um dessen Performance zu verbessern. Dazu werden zwei Teilprozesse ausgeführt: Mit der Berechnung einer Verlustfunktion (Pointer, 2021) wird die Differenz zwischen der Vorhersage des Netzwerkes und dem tatsächlichen Ergebnis als Gradient quantifiziert. Dieses Ergebnis nutzt dann ein Optimierer im Gradientenabstiegsverfahren (Pointer, 2021), um die Gewichte des Netzwerks anzupassen. Ziel ist es, die Verlustfunktion so weit wie möglich zu minimieren.

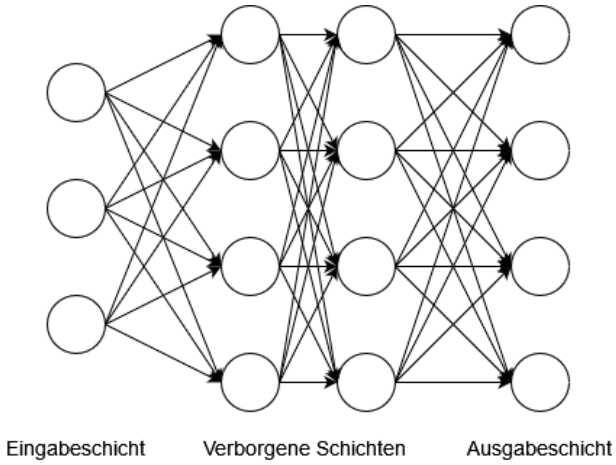

Abbildung 2.8.: Aufbau eines Neuronalen Netzwerks.

Es existieren verschiedene Genauigkeitsebenen, auf denen Bilder auf Grundlage von neuronalen Netzwerken analysiert beziehungsweise klassifiziert werden können (Tabelle 2.5). Während Methoden wie die *Image Classification* oder die *Object Detection* die Klassifizierung und Lokalisierung von einem oder mehreren spezifischen Objekten durchführen, funktioniert die semantische Segmentierung auf Pixelebene. Jeder Pixel des Eingabebildes wird zu einer semantischen Klasse attributiert (Sugimoto u. a., 2021; Tuia u. a., 2021), dadurch ergibt sich als Ausgabe ein hochauflösendes Bild, das der Größe der Eingabe entspricht.

Tabelle 2.5.: *Deep Learning* Methoden der Bildanalyse mit Neuronalen Netzwerken.

Methode	**Annahme**	**Ausgabe**
Image Classification	nur ein Objekt im Bild	diskretes Label
Classification and Localization	nur ein Objekt im Bild	diskretes Label + Bounding Box
Object Detection	mehrere Objekte im Bild	diskretes Label + Bounding Box
Semantic Segmentation	Pixellevel	hochaufgelöstes Bild, jedem Pixel wird eine Klasse zugeordnet
Instance Segmentation	Pixellevel	hochaufgelöstes Bild, jedem Pixel wird eine Klasse zugeordnet, Instanzen einer Klasse werden separat klassifiziert

2.3.2. Semantische Segmentierung mit Deep Learning

Deep Learning Anwendungen zeigen im Vergleich zu traditionellen Klasifizierungsmethoden großes Potenzial hinsichtlich der Qualitätssteigerung der Ergebnisse. Insbesondere mit semantischer Segmentierung kann die Modellgenauigkeit für die Identifizierung von Landbedeckung und Landnutzung gesteigert werden (Pedrayes u. a., 2021; Li u. a., 2020; Stoian u. a., 2019; Marmanis u. a., 2018). Aus diesem Grund wird für die vorliegende Arbeit als grundlegende Methode die semantische Segmentierung gewählt.

Ein Netzwerk zur Semantischen Segmentierung besteht aus *Backbone* (*Encoder*) und *Head* (*Decoder*), die in einer *Bottom-Up* und *Top-Down* Struktur miteinander gekoppelt sind (Sugimoto u. a., 2021; Tuia u. a., 2021). Der *Backbone* extrahiert in einem *Downsampling* Netzwerk die Merkmale (*Features*) der Eingabe; der *Head* führt die Segmentierung der Merkmale in einem *Upsampling*-Prozess durch. Für das Modell der vorliegenden Arbeit wird die Kombination aus ResNet als *Encoder* und FPN als *Decoder* genutzt. Die beiden Komponenten werden nachfolgend dargestellt.

Encoder ResNet

Der *Encoder* ResNet wurde von Microsoft Research entwickelt. Er gewann 2015 nach AlexNet (2012) und VGG und GoogLeNet/Inception (2014) den ILSVRC-Wettbewerb (*ImageNet Large Scale Visual Recognition Challenge*) (Sugimoto u. a., 2021). Im Rahmen des Wettbewerbs werden Algorithmen zur Objekterkennung und Bildklassifizierung in großem Maßstab bewertet (Stanford Vision Lab, 2020).

Sowohl bei ResNet als auch seinen Vorgängern AlexNet (Krizhevsy u. a., 2012), VGG (Simonyan u. a., 2014) und GoogLeNet/Inception (Szegedy u. a., 2014) handelt es sich um Neuronale Konvolutionsnetze (CNNs) (Pointer, 2021), deren zentrale Einheit die Faltungsschichten bilden (Abbildung 2.9).

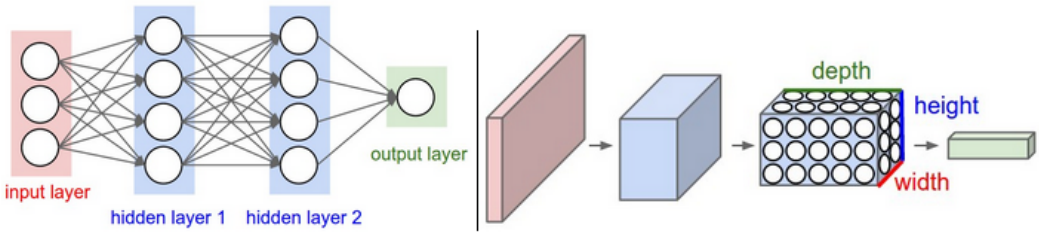

Abbildung 2.9.: Vergleich eines normalen Neuronalen Netzwerks mit einem Konvolutionsnetzwerk. Links: Normales Neuronales Netzwerk mit drei Schichten. Alle Neuronen werden nebeneinander aufgebaut. Rechts: Ein CNN organisiert die Neuronen in drei Dimensionen (Breite, Höhe, Tiefe). Die Tiefe entspricht der Anzahl der Eingabekanäle (zum Beispiel RGB) (Stanford University, 2022).

Verglichen mit normalen Neuronalen Netzen besitzen Konvolutionsnetzwerke sehr viel weniger Einzelverbindungen und damit weniger Gewichte und Parameter. Dadurch

werden die nötigen Interaktionen des Netzwerks reduziert (Ketkar u. a., 2021). Konvolutionsnetzwerke ermöglichen folglich eine schnellere Berechnung und beugen gleichzeitig einer Überanpassung (*Overfitting*) vor. Für CNNs besteht die Eingabe explizit aus einem Bild, die Architektur ist speziell auf dessen Dimensionen angepasst.

Die Eingabe einer Faltungsoperation ist ein Bild der Größe $n_{in} \cdot n_{in} \cdot C$ mit n als Bildhöhe und -breite und C als Anzahl der Kanäle sowie ein Filtersatz der Größe $f \cdot f \cdot C$ mit f als Filterhöhe und -Breite mit k Filtern . Als Ausgabe erhält man eine *Feature Map* der Größe $n_{out} \cdot n_{out} \cdot k$. In der Konvolutionsschicht (Abbildung 2.10) wird jeweils ein Filter von links oben nach rechts unten über das Eingabebild bewegt (Ketkar u. a., 2021). Für jeden Schritt wird eine Matrixmultiplikation durchgeführt und das Ergebnis aufsummiert. Daraus ergibt sich die *Feature Map*, die die Summen der Matrixmultiplikationen enthält. Zusätzlich beeinflussbar sind bei einer Faltungsoperation die Schrittweite (*stride*) des Filters und dessen Verhalten an den Randbereichen des Eingabebildes. Durch Auffüllen (*padding*) können unvollständige Filtereingaben ergänzt werden.

Im Anschluss an die Faltung folgt eine Aktivierungs- sowie eine *Pooling*- Schicht. In letzterer wird die Auflösung der Eingabe verringert, wodurch gleichzeitig die Parameterzahl in den folgenden Schichten reduziert wird (Hu u. a., 2015). Die Ausgabe wird in einer vollständig verknüpften Schicht in Form von Klassenwerten berechnet.

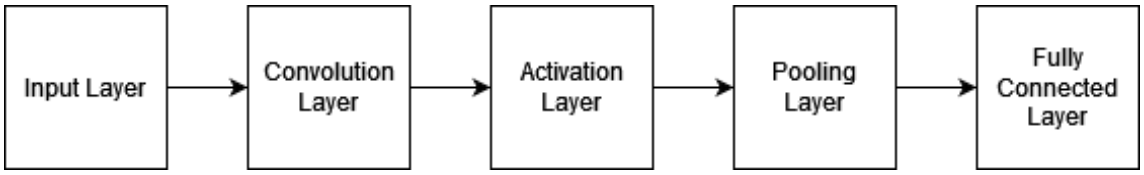

Abbildung 2.10.: Aufbau eines Konvolutionsnetzwerkes.

Seit AlexNet (8 Schichten) besteht der Trend, tiefere Netzwerke zu entwerfen, die mehr Schichten enthalten, um dadurch die Modellleistung zu verbessern (Simonyan u. a., 2014; Szegedy u. a., 2014). Beispiele sind VGG mit 16 beziehungsweise 19 und GoogLeNet/Inception mit 22 Schichten. Tiefere Modelle sind jedoch schwieriger zu trainieren. Ein simples Anfügen weiterer Schichten an bereits bestehende Architekturen ist nicht möglich. Durch das Hinzufügen weiterer Schichten kann die Modellperformance eine Sättigung erreichen oder sich verschlechtern.

Abbildung 2.11 zeigt den Trainingsfehler (links) und den Testfehler (rechts) für zwei unterschiedlich tiefe einfache Netzwerke, trainiert auf dem CIFAR-10 Datensatz. Beide Fehlerraten sind für das tiefere Netzwerk mit 56 Schichten höher als für das flachere Netzwerk mit 20 Schichten. Würde es sich hierbei um ein *Overfitting*-Problem handeln, läge nur der Testfehler des tieferen Netzwerks höher. Da der Fehler aber sowohl im Training als auch im Test des tieferen Netzwerks ansteigt, handelt es sich um das Problem des *Vanishing/Exploding Gradient*. Die Problematik ergibt sich aus dem Aufbau einfacher Netzwerke wie AlexNet und VGG. Diese Netzwerke arbeiten mit einem *Downscaling* der Auflösung des Bildes, einhergehend mit einem *Upscaling*

der Filteranzahl. Fügt man einem solchen Netzwerk weitere Schichten hinzu, sollten diese die Identitätsfunktion der ursprünglichen Schichten lernen. Damit sollte mindestens die selbe Modellgüte wie in dem flacheren Netzwerk zu erreichen sein. Dies ist aber nicht der Fall, da die Gewichte der Schichten gegen null initialisert werden; wenn der Gradient in der Backpropagation an frühere Schichten zurückgegeben wird, wird er unendlich klein. Bereits vor ResNet bestanden Ansätze zum Umgang mit der Problematik, wie zum Beispiel das Hinzufügen einer zusätzlichen Verlustfunktion in einer mittleren Schicht als zusätzliche Überwachung (Szegedy u. a., 2014). Eine finale Lösung existierte nicht.

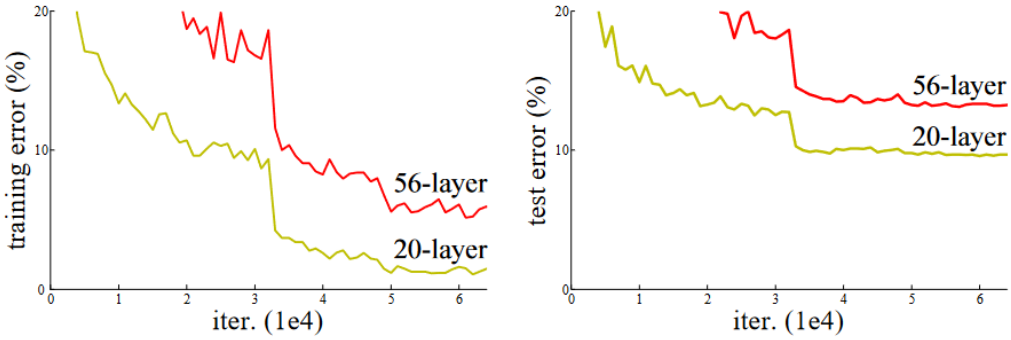

Abbildung 2.11.: *Vanishing/Exploding Gradient* in tiefen neuronalen Netzwerken. Trainingsfehler (links) und Testfehler (rechts) trainiert auf CIFAR-10 mit einfachen 20-Schichten und 56-Schichten Netzwerken (He u. a., 2016).

ResNet liefert einen neuen Ansatz: Es werden Residualblöcke mit *Shortcut Connections* eingeführt (Abbildung 2.12). Diese Verbindungen überspringen eine oder mehrere Schichten, um die Aktivierungen der vorderen mit denen der nachfolgenden Schicht zu verbinden. Dadurch entsteht ein Residualblock. In Kombination ergeben diese ein Residuales Netzwerk (ResNet). *Shortcut Connections* fügen dabei weder zusätzliche Parameter noch Rechenkomplexität hinzu. Die ursprüngliche Eingabe in jedem Block wird in die nächste Schicht weitergeleitet, sodass das Signal der Trainingsdaten durch ein tieferes Netzwerk gehen kann. Mit dieser Methode können Netzwerke ausgebildet werden, die wesentlich mehr Schichten beinhalten als die bisherigen wie VGG und Inception, dabei aber keine höhere Fehlerrate aufweisen (Pointer, 2021).

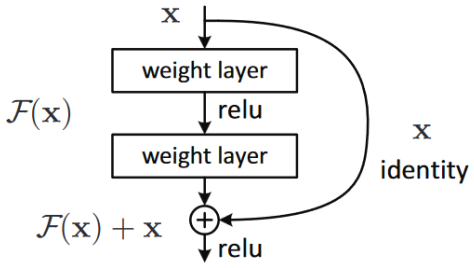

Abbildung 2.12.: Ein Residualblock (He u. a., 2016).

Residuales Lernen entspringt der folgenden Hypothese: $H(x)$ entspricht einem zugrundeliegenden Bild, das durch einige gestapelte Layer angepasst wird. x bezeichnet dabei die Eingabe in der ersten Schicht. Wenn es möglich ist, dass mehrere nichtlineare Schichten komplizierte Funktionen asymptotisch approximieren, müssen sie ebenso die Residualfunktion approximieren können. Die Residualfunktion enstpricht dabei $H(x)-x$. Es gilt die Annahme dass Ein- und Ausgabe dieselben Dimensionen besitzen. Anstatt also eine Annäherung der Schichten an $H(x)$ anzustreben, sollen durch Residuales Lernen die Schichten explizit eine Residualfunktion $F(x) := H(x) - x$ approximieren. Die ursprüngliche Funktion wird zu $F(x) + x$. Diese Methode wird auf die Residualblöcke übertragen (Abbildung 2.12). Ein Baustein ist definiert als

$$y = F(x, W_i) + x \tag{2.1}$$

x und y sind die Ein- und Ausgabevektoren. *Shortcut Connections* führen mit ihrer Funktionsweise weder zusätzliche Parameter ein, noch nimmt die Rechenkomplexität zu. Die Funktion $F(x, W_i) + x$ entspricht dem *Residual Mapping*, das gelernt werden soll. Die *Shortcut*-Verbindungen führen eine Identitätszuordnung durch; ihre Ausgabe wird zur Ausgabe der gestapelten Schichten addiert. Eine Optimierung des *Residual Mapping* ist einfacher als die des ursprünglichen, nicht referenzierten *Mappings*. Statt einer nicht referenzierten Funktion wird die Residualfunktion mit Bezug auf die Eingabe erlernt. Die Änderung, die innerhalb eines solchen Blocks gelernt wird, ist wesentlich kleiner als das Lernergebnis einer Schicht in einem einfachen Netzwerk. Dadurch wird der Lernvorgang für das Netzwerk erleichtert. Abbildung 2.13 stellt die Entwicklung von VGG 19 (links) über ein tieferes einfaches Netzwerk (mitte) hin zu ResNet (rechts) dar. Bei VGG 19 erfolgt das *Downscaling* der Auflösung, sowie das *Upscaling* der Filter mit zunehmender Schichtentiefe. Dadurch ergibt sich eine hohe Anzahl von Parametern und gleichzeitig ein hoher Rechenaufwand: VGG 19 führt 19,6 Milliarden Gleitkommaoperationen (FLOPs) aus.

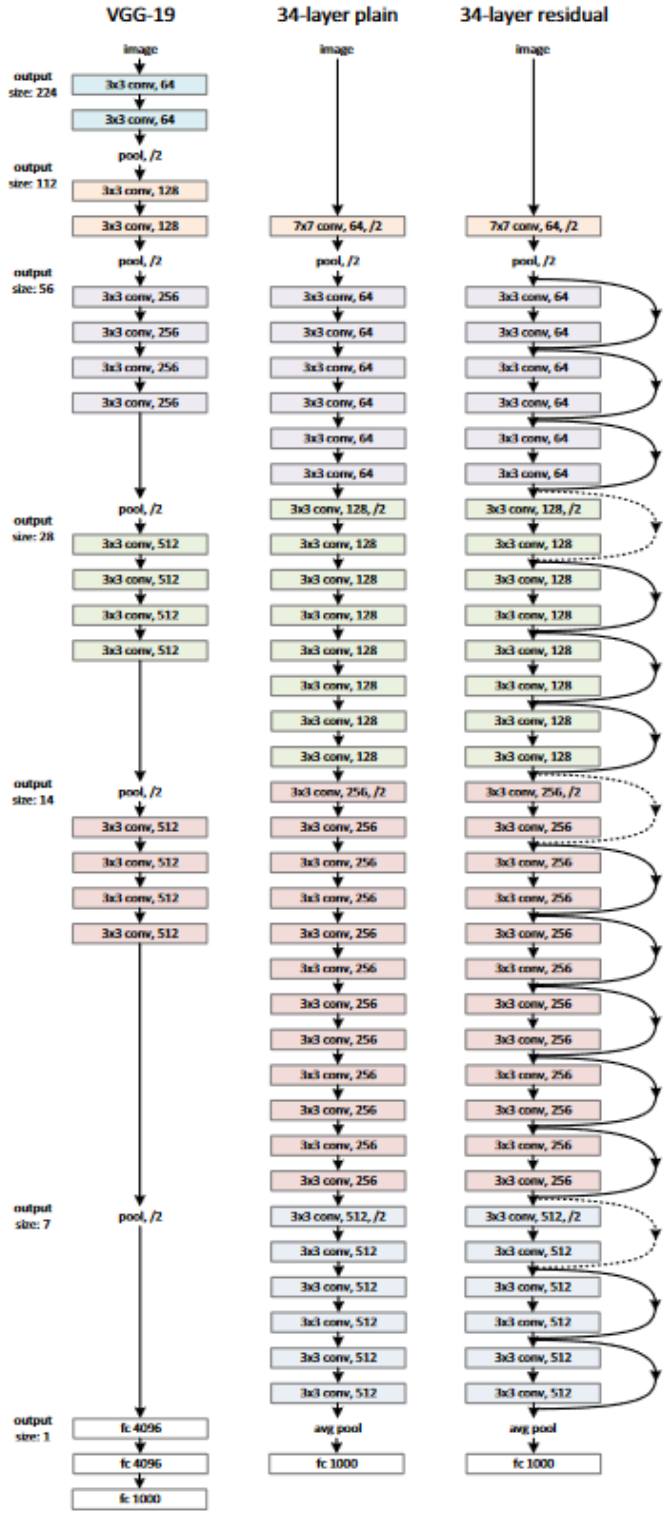

Abbildung 2.13.: Entwicklung einer Residualen Netzwerkarchitektur. Links: VGG-19-Modell (19,6 Milliarden FLOPs) als Referenz. Mitte: ein einfaches Netz mit 34 Schichten (3,6 Milliarden FLOPs). Rechts: ResNet mit 34 Schichten (3,6 Milliarden FLOPs) (He u. a., 2016).

Für einen ersten Vergleich werden dem tieferen einfachen Netzwerk weitere Schichten hinzugefügt (Gesamtzahl 34). In einem Trainingslauf mit ImageNet Daten erhöht sich der Trainingsfehler des tieferen Netzwerks (Plain 34) gegenüber dem flacheren (Plain 18) (Abbildung 2.14 links). Fügt man den Netzwerken noch die *Shortcut Connections* hinzu (Abbildung 2.13 rechts), und vergleicht dann das flachere Residualnetzwerk mit dem tieferen, fällt der Fehler durch die höhere Anzahl an Schichten deutlich ab (Abbildung 2.14 rechts). Das Problem des *Vanishing Gradient* kann also durch die Einführung von Residualblöcken mit *Shortcut Connections* gelöst werden.

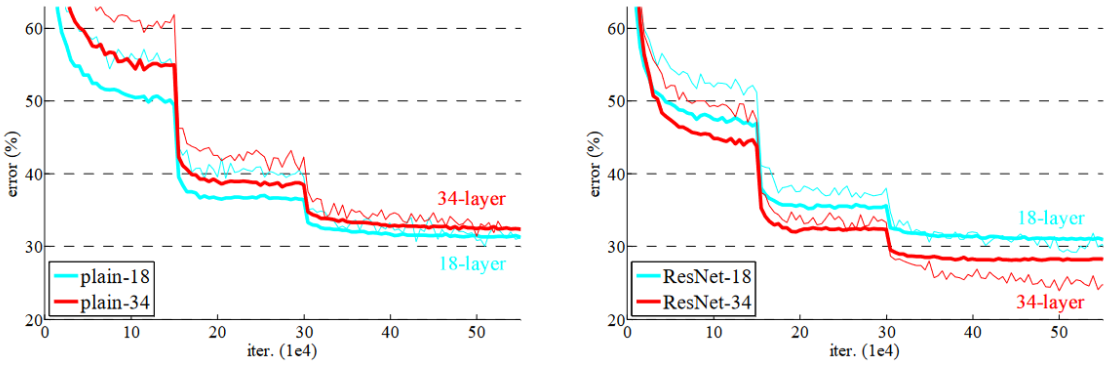

Abbildung 2.14.: Training auf dem ImageNet-Datensatz. Schmale Kurven entsprechen dem Trainingsfehler, normale Kurven dem Validierungsfehler. Links: Einfache Netze mit 18 und 34 Schichten. Rechts: ResNets mit 18 und 34 Schichten (He u. a., 2016).

ResNet bietet zudem durch seinen Aufbau den Vorteil einer massiv verringerten Komplexität. Ein Residuales Netzwerk mit 34 Schichten führt nur 3,6 Milliarden Gleitkommaoperationen aus. Gleichzeitig können mit ResNet sehr viel tiefere Netzwerke trainiert werden, ohne dass der Berechnungsaufwand stark ansteigt, da weniger Filter als in einfachen Netzwerken eingesetzt werden. Weitere Experimente mit tieferen ResNets (50, 101, 152) zeigen, dass der Fehler mit zunehmender Anzahl an Schichten weiter abnimmt. Um Berechnungsaufwand und -zeit in diesen tieferen Netzwerken niedrig zu halten, werden gegenüber der Grundlage von ResNet 34 Anpassungen an den Residualblöcken vorgenommen. Jeder Residualblock erhält statt den ursprünglichen zwei Schichten einen *Stack* aus drei Schichten (Abbildung 2.15). Ein solcher *Bottleneck Block* (Abbildung 2.15 rechts) besitzt die gleiche Komplexität wie die Residualblöcke in ResNet 34 (Abbildung 2.15 links). Die 1x1 Konvolutionsschichten dienen der Reduzierung und Wiederherstellung der Dimensionen. Zentral liegt die *Bottleneck*-Schicht mit reduzierten Ein- und Ausgabedimensionen. Mit dieser Methode lassen sich tiefe Netzwerke mit einer hohen Schichtenanzahl trainieren (Abbildung 2.16 links).

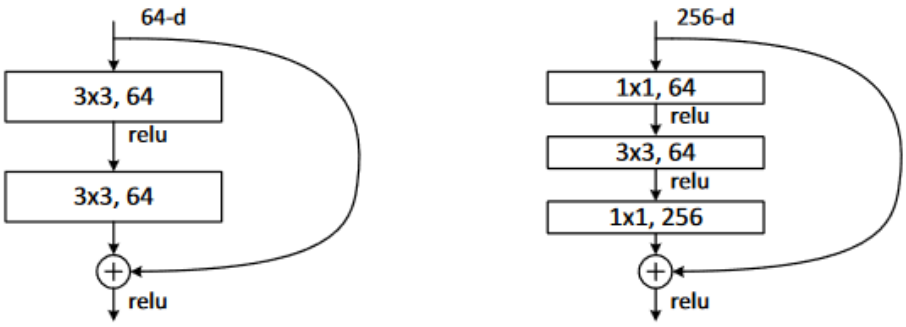

Abbildung 2.15.: Links: Residualblock für ResNet34. Rechts: Ein *Bottleneck* Block für ResNet50/101/152 (He u. a., 2016).

Je mehr Schichten dem Netzwerk hinzugefügt werden, desto geringer werden sowohl Trainings- als auch Testfehler. Bei einem Vergleich von ResNet 110 mit ResNet 1202 ändert sich dieser Trend (Abbildung 2.16 rechts): während der Trainingsfehler beider Netzwerke gleich gering bleibt, liegt der Testfehler des tieferen Netzwerks mit 1202 Layern deutlich höher: Hier handelt es sich also um ein *Overfitting* des Modells, der Anstieg ist nicht auf den *Vanishing Gradient* zurückzuführen.

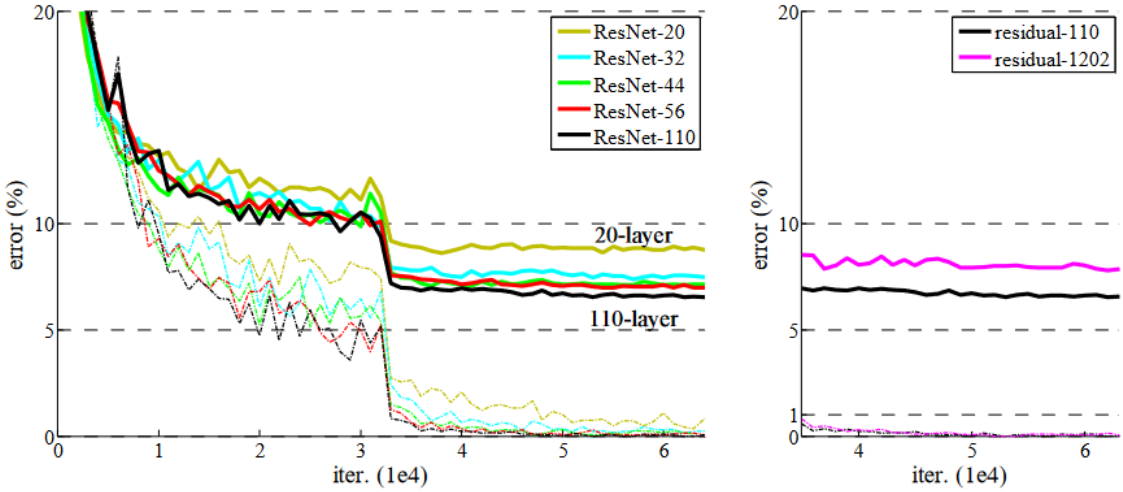

Abbildung 2.16.: Training von tieferen Residualnetzwerken auf dem CIFAR-10 Datensatz. Die gestrichelten Linien entsprechen dem Trainingsfehler, die normalen Linien dem Testfehler. Links: ResNets mit 20 bis 110 Schichten. Rechts: ResNets mit 110 und 1202 Schichten (He u. a., 2016).

Mit ResNet liefert Microsoft Research eine Architektur, die das *Vanishing Gradient* Problem löst und die, ohne den Rechenaufwand zu erhöhen, bis in eine Tiefe trainiert werden kann, die schließlich durch *Overfitting* des Modells limitiert wird. ResNet ist damit die ideale Architektur für verschiedenste Bildverarbeitungsaufgaben. In der vorliegenden Arbeit wird ResNet50 (50 Schichten) als *Encoder* genutzt.

Decoder FPN

Das *Feature Pyramid Network* (FPN) (Lin u. a., 2016) wurde 2017 von *Facebook AI Research* (FAIR) vorgestellt. Die Methode kombiniert niedrig aufgelöste, semantisch starke Merkmale mit hochaufgelösten, semantisch schwachen Merkmalen durch einen *Bottom-Up* und *Top-Down* Ansatz mit lateralen Verbindungen.

Vorgänger von FPN waren FCN (*Fully Convolutional Network*) (Long u. a., 2014) und U-Net (Ronneberger u. a., 2015). Die Idee von FCN beruht darauf, alle vollständig verbundenen Schichten durch Konvolutionsschichten zu ersetzen. Ein FCN hat keine vollständig verbundene Schicht am Ende, wie sie üblicherweise für die *Image Classification* genutzt werden. Dadurch besitzt die finale Ausgabeschicht zwar die selbe Bildgröße wie die Eingabe, die Anzahl der Kanäle entspricht dabei aber der Anzahl der Klassen. Die Ausgabeschicht erhält also die Dimension $Höhe \cdot Breite \cdot Klassenanzahl$. Mit einer Aktivierungsfunktion in der letzten Schicht erhält man für jeden Pixel die wahrscheinlichste Klasse . Bereits FCN nutzt das Grundkonzept des *Down-* und *Upsampling* für eine pixelgenaue Vorhersage, welches auch U-Net und FPN integrieren werden. Ebenso werden bereits hier laterale Verbindungen eingeführt, die die *Downsampling Feature Map* und die *Upsampling Feature Map* derselben Ebene durch Addition kombinieren. Für die Vorhersage wird nur die finale *Upsampling Feature Map* verwendet. Der Nachfolger U-Net wurde erstmals 2015 für die Bildsegmentierung im Biomedizinischen Bereich vorgestellt. Das Gesamtkonzept ist dem von FCN sehr ähnlich. Der *Down-* und *Upsampling* Ansatz bleibt gleich; ebenso wird nur die letzte *Feature Map* aus dem *Upsampling* für die Vorhersage verwendet (Abbildung 2.17 oben).

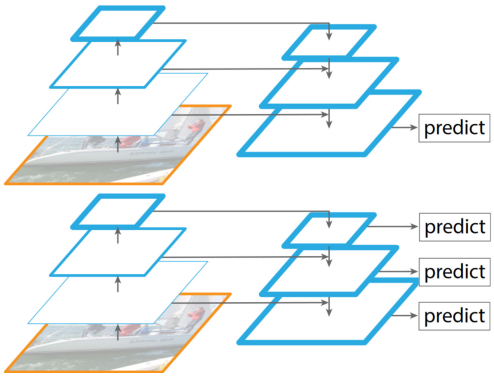

Abbildung 2.17.: Oben: *Top-Down*-Architektur (zum Beispiel U-Net) mit lateralen Verbindungen. Eine Vorhersage findet nur auf letzten Ebene statt. Unten: FPN mit ähnlicher Struktur; Vorhersagen werden aber unabhängig auf allen Ebenen getroffen. *Feature Maps* sind durch blaue Umrisse gekennzeichnet; dickere Umrisse bezeichnen semantisch stärkere *Features* (Lin u. a., 2016).

Eine Neuerung in U-Net gegenüber FCN ist eine größere Anzahl von Merkmalskanälen im *Upsampling*, die die Architektur symmetrischer macht. Ein weiterer Unterschied besteht in der Methodik: Die Addition der lateralen Verbindung mit sofort anschließendem *Upsampling* bei FCN wird bei U-Net ersetzt: Die *Feature Maps* werden verkettet

und durchlaufen dann Konvolutionsschichten zur weiteren Vearbeitung. Die Zentrale Idee, die dann FPN von U-Net unterscheidet, ist, dass die Vorhersage nicht nur auf der letzten Upsampling Ebene getroffen wird, sondern für jede der Ebenen in verschiedenen Auflösungen (Abbildung 2.17 unten). FPN fügt dem *Bottom-Up* Netzwerk ResNet einen *Top-Down*-Pfad mit seitlichen Verbindungen hinzu. Der *Top-Down*-Pfad startet in der tiefsten Schicht des Netzwerks und führt schrittweise ein *Upsampling* durch; die räumliche Auflösung wird dabei um den Faktor 2 erhöht. Parallel werden über die lateralen Verbindungen transformierte Versionen derselben räumlichen Größe aus dem *Bottom-Up*-Pfad elementweise addiert (Abbildung 2.18).

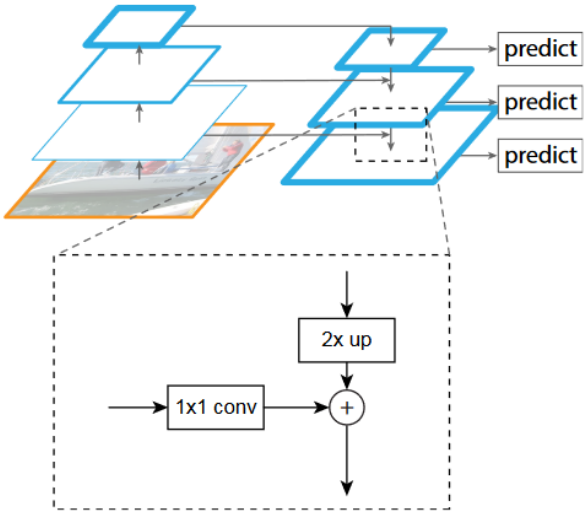

Abbildung 2.18.: Ein *Upsampling*-Block mit lateraler und *Top-Down*-Verbindung, durch Addition zusammengefügt (Lin u. a., 2016).

Dabei wird an den *Feature Maps* aus dem *Bottom-Up*-Pfad eine Faltung durchgeführt, um die Kanaldimensionen zu reduzieren. FPN erzeugt damit eine Pyramide mit standardmäßigen Auflösungen von 1/32 bis 1/4, in der jede Ebene die gleiche Kanaldimension (standardmäßig 256) hat. Die Leistungsfähigkeit der Architektur wurde in einem Vergleich mit zwei weiteren gängigen *Backbone* Architekturen evaluiert (Kirillov u. a., 2019). Dabei zeigt sich unter anderem, dass FPN sehr viel weniger Aktivierungen hat und etwa zwei Mal effizienter ist als eine symmetrische *Encoder-Decoder* Architektur (zum Beispiel U-Net). Mit der Verwendung von FPN kann die Güte des Modells gesteigert werden (Sugimoto u. a., 2021). Es handelt sich um einen asymmetrischen, unkomplizierten *Decoder*, dessen *Top-Down*-Pfad nur einen Block pro Ebene hat und eine gemeinsame Kanaldimension verwendet.

3. Experimentelles Setup

Die Einstellungen des Experimentellen Setups setzen sich aus der Vorbereitung der Eingangsdaten (Kapitel 3.1) und der Wahl geeigneter Hyperparameter für das Netzwerk (Kapitel 3.2) zusammen. Auf Basis der in diesem Kapitel aufgezeigten Erkenntnisse werden die anschließenden Experimente zur Landbedeckungs- und Landnutzungsklassifizierung durchgeführt.

3.1. Vorbereitung des Datensatzes

Um den Datensatz für die Nutzung vorzubereiten und die Qualität der Eingabedaten zu gewährleisten, sind mehrere Schritte nötig. Die nachfolgend dargestellten Abläufe basieren auf empirischen Erhebungen sowie Empfehlungen aus der aktuellen Literatur.

Im Rahmen der Szenenvorauswahl (Kapitel 3.1.1) werden auftretende Szenendoppelungen und -überschneidungen entfernt. Anschließend folgt die Datenvorbearbeitung (Kapitel 3.1.2) mit Anpassungen bezüglich Datengröße, Struktur und Zusammensetzung. Im *Datensampling* (Kapitel 3.1.3) werden die Szenen dann auf für den Trainingsprozess und die Bewertung benötigten Teildatensätze aufgeteilt.

3.1.1. Szenenvorauswahl

Der RapidEye Datensatz kann nicht vollständig genutzt werden. Grund dafür sind Szenenüberschneidungen sowie Szenendoppelungen (Abbildung 3.1). Um diese Überschneidungen weitgehend aus dem Datensatz zu entfernen, werden alle Szenen miteinander verglichen und deren Überlappungsbereich in Prozent berechnet. Wird eine Überlappung festgestellt, gilt bei der Überschreitung eines festgelegten Schwellenwertes: es wird diejenige der verglichenen Szenen ausgeschlossen, die weniger valide Pixel aufweist. Ist die Anzahl invalider Pixel beider Szenen gleich, wird die zweite Szene ausgeschlossen. Mit dieser Methode werden Schwellenwerte von 70% und 60% Überlappungsbereich getestet. Dadurch werden 78 bzw. 85 Szenen ausgeschlossen.

Aufgrund der besseren Ergebnisse wird ein Schwellenwert von 60% für Szenenüberlappungen genutzt.

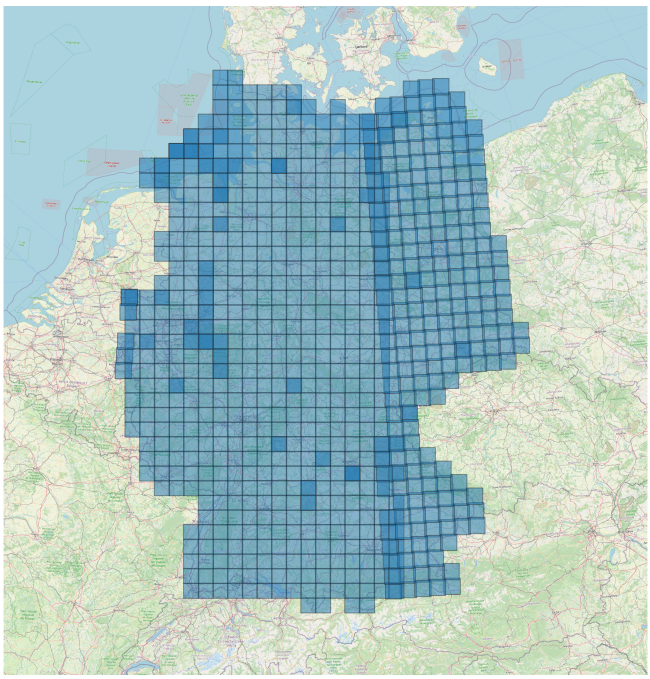

Abbildung 3.1.: RapidEye2015 Datensatz für Deutschland. Überschneidungsbereiche und Szenendoppelungen werden durch die Überlappung dunkler dargestellt.

3.1.2. Datenvorbearbeitung

Die Datenvorbearbeitung besteht aus mehreren Schritten, die den Datensatz an die Anforderungen des Netzwerks anpassen. Zuerst wird ein *Data Whitening* verwendet (Pal u. a., 2016), um Korrelationen und Abhängigkeiten zwischen Pixeln im Datensatz zu entfernen. Für alle Szenen werden der Mittelwert und die Standardabweichung der fünf Kanäle berechnet und daraus jeweils der Mittelwert bestimmt. Die 16 bit Bilder werden neu skaliert, um Pixelwerte zwischen 0 und 1 zu erhalten. Dann wird der Mittelwert auf Pixelebene subtrahiert und das Ergebnis durch die Standardabweichung geteilt, um die Pixelwerte um 0 zu zentrieren. Anschließend werden die RapidEye Szenen in 320 x 320 Pixel große Kacheln aufgeteilt; das entspricht der größtmöglichen Kachelgröße die vom Netzwerk verarbeitet werden kann. Von den Kacheln werden nur diejenigen in den Trainingsdatensatz übernommen, deren Label mindestens 95% valide Pixel enthalten, also nicht der Klasse 0 angehören. Die Landbedeckungsmodelle liegen nur bis zu den deutschen Landesgrenzen vor, Bereiche außerhalb sind flächendeckend als Klasse 0 bezeichnet. Die RapidEye Daten dagegen erstrecken sich in voller Szenengröße darüber hinaus. Kacheln, die außerhalb der Landesgrenzen liegen, könnten so das Training negativ beeinflussen, da hier keine Landbedeckungs- beziehungsweise Nutzungsinformation gegeben ist.

3.1.3. Datensampling

Die RapidEye-Daten mit dem jeweils zugehörigen Landbedeckungs- bzw. Landnutzungsmodell werden auf drei Teildatensätze aufgeteilt (Pointer, 2021). Jeder Teildatensatz soll dabei die Klassenverteilung des gesamten Datensatzes korrekt repräsentieren. Für die Güte des Modells spielen die Menge und Qualität, sowie die Zusammensetzung der verfügbaren Daten eine wichtige Rolle (Bruzzone u. a., 2014). Der Trainingsdatensatz wird während des Trainingsvorgangs zur Aktualisierung des Modells verwendet. Der Validierungsdatensatz wird zur Bewertung des Modells nach jeder Epoche verwendet. Die Validierung des Modells dient dazu, eine Überanpassung (*Overfitting*) zu verhindern. Das Modell soll in der Lage sein, trainierte Zusammenhänge zu verallgemeinern und nicht nur die gelernten Trainingsdaten abzubilden. Mit dem Testdatensatz wird die Qualität des Modells nach Abschluss des Trainings bewertet.

Die verfügbaren Daten können nicht zu mehreren Teilsätzen gleichzeitig hinzugefügt werden. Aus diesem Grund werden bestehende Überschneidungen im Datensatz vor dem *Sampling* entfernt (Kapitel 3.1.1).

Um die Szenen in die drei Teildatensätze für Training, Validierung und Test aufzuteilen, stehen zwei Methoden zur Verfügung. Die initiale Methode war ein zufälliges *Sampling*. Aufgrund von auftretenden Klassenungleichgewichten wird zudem ein stratifiziertes *Sampling* eingeführt.

Random Sampling

Mit der Methode des *Random Sampling* werden die Szenen zufällig auf Training (70%), Validierung (15%) und Test (15%) aufgeteilt. Um den Einfluss dieser zufälligen Zuordnung auf die Qualität der Klassenverteilung zu untersuchen, werden mehrere *Random Sampling* Tests durchgeführt. Als Landbedeckungsmodell wird aufgrund seiner starken Klassenungleichgewichte (Kapitel 2.2.1) lbakt genutzt.

Abbildung 3.2.: Einfluss der Random Sampling Methode auf die Qualität der Teildatensätze. Zur Evaluierung werden drei Experimente mit lbakt mit variirenden *Shuffle*-Einstellungen durchgeführt.

Abbildung 3.2 stellt die Ergebnisse der Tests dar. Die kleinen Klassen (Klasse 6-8, verschiedene Anbaufrüchte; Klasse 26, 29 und 30, verschiedene Wasserklassen) zeigen eine überwiegend hohe Heterogenität in der Verteilung. Die kleinste Klasse 20

(Brandfläche) kann mit nur einer Szene nicht gesampelt werden. Bei der nächstgrößeren Klasse 21 (Schnee (permanent) und Eis), die auf fünf Szenen vertreten ist, findet nur in einem Experiment eine Aufteilung in alle drei Teildatensätze statt. Das *Random Sampling* ist also nicht dazu geeignet, Ungleichgewichten vorzubeugen und insbesondere die kleineren Klassen gezielt den Teildatensätzen zuzuweisen. Aufgrund dessen wird eine zweite Methode eingeführt. Für Trainingsläufe mit *Random Sampling* wird weiterhin der initiale Shuffle-Typ 1 genutzt, um die Vergleichbarkeit der Experimente zu gewährleisten.

Stratifiziertes Sampling

Mit dem stratifizierten *Sampling* werden die Szenen nach pixelbasierten Schwellenwerten auf Training, Validierung und Test aufgeteilt. Das Ziel der Methode besteht darin, eine ausgewogenere Pixelverteilung kleiner Klassen zu erreichen.

Die für das *Random Sampling* genutzten Schwellenwerte (70% Training, 15% Validierung, 15% Test) werden hier auf 80% Training, 10% Validierung und 10% Test angepasst. Der Grund dafür liegt darin, dass die Schwellenwerte im Anschluss zwar auf Pixelebene berechnet werden, das Sampling aber auf Szenenebene durchgeführt wird. Gerade bei kleinen Klassen variieren die klassenspezifischen Pixelzahlen der einzelnen Szenen, was zu einer starken Überschreitung der Schwellenwerte führen kann (Tabelle 3.1).

Tabelle 3.1.: Klassenspezifische Ungleichgewichte der Pixelverteilung auf Szenenebene am Beispiel der Klasse 21 (Schnee (permanent) und Eis) des Landbedeckungsmodells lbakt. Die Gesamtpixelzahl im Datensatz beträgt 43364 auf fünf Szenen.

Szenen-ID	Pixelanzahl	Gesamtanteil
3260918_2015_08_30_re1_3a_574323	532	1%
3260919_2015_05_18_re1_3a_577254	6556	15%
3260921_2015_08_27_re2_3a_574322	26463	61%
3261022_2015_07_31_re4_3a_574320	341	<1%
3361008_2015_08_26_re2_3a_574322	9472	21%

Auf Basis der festgelegten prozentualen Aufteilung werden alle Pixel des genutzten Landbedeckungsmodells pro Klasse addiert, um daraus die entsprechenden Schwellenwerte auf Pixelebene zu berechnen. Die Schwellenwerte werden im anschließenden Datensampling dazu genutzt, von der kleinsten Klasse aufsteigend alle noch nicht gesampelten Szenen zu durchsuchen und die entsprechende Klasse enthaltende Szenen so lange dem Teildatensatz hinzuzufügen, bis der Pixelschwellenwert überschritten wird. Die Methode wird für die Validierungs- und Testssätze durchgeführt, alle übrigen Szenen werden dem Trainingsdatensatz hinzugefügt.

Je nach genutztem LBM-DE Datensatz unterscheidet sich die konkrete Umsetzung des stratifizierten *Samplings*. Im Fall von lbakt können die Klassen 20 und 21 aufgrund ihrer geringen Szenen- und Pixelanzahl nicht berücksichtigt werden. Bei Klasse 20, die

nur auf einer Szene vertreten ist, ist diese Methode grundsätzlich nicht möglich. Klasse 21 ist auf fünf Szenen vertreten, die Pixelverteilung dabei aber so heterogen (Tabelle 3.1), dass ein gezieltes *Sampling* nach einer 80-10-10 Aufteilung nicht möglich ist.

Für das Landnutzungsmodell (lnakt) können nach demselbem Prinzip alle Klassen gesampelt werden. Die auf Szenenebene gleichmäßigere Klassenverteilung erübrigt einen Klassenausschluss, wie er bei lbakt angewendet werden muss. Für die CORINE *Land Cover* wird die Methode unter Ausschluss der neun Klassen angewendet, die in Deutschland nicht vertreten sind. Außerdem können wie auch bei lbakt die beiden kleinsten Klassen (CLC Klassen 33 und 34) aufgrund ihrer zu geringen Szenenanzahl nicht in das *Sampling* integriert werden.

Tabelle 3.2.: Szenenanzahl der Teildatensätze für Training, Validierung und Test.

Landnutzungsmodell	**Random Sampling**	**Stratifiziertes Sampling**
Landbedeckung (lbakt)	527, 113, 113	497, 111, 145
Landnutzung (lnakt)	512, 109, 110	546, 89, 96
CORINE Land Cover	521, 111, 112	522, 114, 108

Aus den beiden zu Verfügung stehenden *Sampling* Methoden in Kombination mit dem Schwellenwert-Mechanismus in der Datenvorprozessierung (Kapitel 3.1.2) ergeben sich je nach ausgewähltem Landbedeckungsmodell und *Sampling* Methode variierende Größen der Teildatensätze (Tabelle 3.2).

3.2. Hyperparameter

Zur Optimierung des Modells können verschiedene Hyperparameter angepasst werden. Die Auswahl der im Rahmen dieser Arbeit genutzten Hyperparameter beruht einerseits auf einer empirischen Evaluierung, andererseits auf den Empfehlungen der aktuellen Forschung.

Tabelle 3.3.: Experimente zur Anpassung der Hyperparameter. Experiment 1-5: Anpassung der Lernrate, des Lernraten *Drops* und der Verlustfunktion. Experiment 6: Szenenüberlappungen (Schwellenwert = 60%) entfernt. Experiment 7: Szenenüberlappungen (Schwellenwert = 70%) enfernt; angepasste Datenaugmentierung und Klassengewichtung für die Verlustfunktion. Experiment 8-10: *Warm Up* Lernrate, Anpassen des Lernraten *Drops*, Anpassen der Epochenzahl.

Exp.	**Lernrate**	**Drop**	**Epochen**	**Verlustfunktion**	**Val IoU**
1	0.0001	50	40	Dice	62%
2	0.0001	50	23	$CE_{labelsmoothing}$	48%
3	0.005	20	76	$CE_{labelsmoothing}$	51%
4	0.005	30	20	Dice	18%
5	0.0001	30	16	Dice	63%
6	0.005	20	28	$CE_{labelsmoothing}$	37%
7	0.001	2	13	$CE_{labelsmoothing}$	12%
8	0.001 warm up $1e^{-8}$	10	24	$CE_{labelsmoothing}$	13%
9	0.001 warm up $1e^{-8}$	15	12	$CE_{labelsmoothing}$	44%
10	0.001 warm up $1e^{-8}$	20	67	$CE_{labelsmoothing}$	51%

Die empirische Evaluierung (Tabelle 3.3) erfolgt auf Grundlage des Landbedeckungsmodells (lbakt) mit 31 Klassen. Genutzt wird für alle Tests ein FPN-ResNet Modell. Zur initialen Bewertung der Modellgüte nach Abschluss eines Trainingslaufs wird der Mittelwert der Validierungs-IoU (*Intersection over Union*) aller Validierungsszenen genutzt. Die IoU bezeichnet den Grad der Überlappung des Eingabebildes und der entsprechenden Vorhersage. Da es sich hierbei nur um einen Mittelwert handelt, werden zur zusätzlichen Überprüfung die Ergebnisse der Klassifizierung herangezogen, um beispielsweise klassenspezifische Probleme präzise identifizieren zu können.

Die Einstellungen aus Experiment 10 (Tabelle 3.3) erzielen die besten Ergebnisse. Aus diesem Grund werden die hier genutzten Parameter mit einer Epochenzahl von 60 für zukünftige Experimente übernommen. Die Hyperparameter werden nachfolgend genauer erläutert.

3.2.1. Datenaugmentierung

Für den Trainings- und den Validierungsdatensatz werden Datenaugmentierungsmethoden angewandt. Die Datenaugmentierung oder Datenanreicherung dient der künstlichen Erweiterung des vorliegenden Datensatzes (Shane, 2021). Die Daten werden durch verschiedene Methoden leicht variiert. Dadurch kann einem *Overfitting* des Modells vorgebeugt werden. *Overfitting* tritt bei einem Modell auf, das dazu tendiert, die Trainingsdaten widerzuspiegeln, anstatt eine verallgemeinernde Lösung zu erlernen. Auf den Trainingsdatensatz wird die *Resize*-, die *HorizontalFlip*- und die *ShiftScaleRotate*-Transformation angewendet (Buslaev u. a., 2020). Die *Resize*-Transformation skaliert alle Daten auf die selbe Größe. Die *HorizontalFlip*- und *ShiftScaleRotate*-Transformationen spiegeln das Bild an der Horizontalen beziehungsweise verschieben, skalieren und drehen die Eingabe. Alle Transformationen können mit einer Auftrittswahrscheinlichkeit angegeben werden, um eine zufällige Datenanreicherung zu erzielen. Auf den Validierungsdatensatz wird ebenfalls die *Resize*-Transformation angewendet. Zusätzlich ist die *PadIfNeeded*-Transformation integriert. Diese füllt die Ränder eines Bildes auf eine angegegebene Größe (320 x 320 Pixel) auf.

3.2.2. Batch Size und Batch Normalisierung

Die *Batch Size* legt die Menge an Bildern fest, die das Netzwerk durchlaufen, bevor es während der *Backpropagation* aktualisiert wird. Je kleiner der *Batch*, desto weniger Speicherplatz wird benötigt (Pointer, 2021). Die *Batch Size* für das genutzte Modell ist 16. In ResNet ist nach jeder Konvolution beziehungsweise vor jeder Aktivierung eine *Batch* Normalisierungsschicht integriert (He u. a., 2016). Diese Schicht normalisiert jeden *Batch*, der das Netzwerk durchläuft, auf einen Mittelwert von 0 mit einer Varianz von 1 (Ioffe u. a., 2015). Bei tiefen Netzwerken beugt dieser Mechanismus dem *Vanishing* und *Exploding Gradient* vor und macht Netzwerke dadurch schneller und stabiler.

3.2.3. Aktivierungsfunktion

Als Aktivierungsfunktion der verborgenen Schichten wird bei ResNet und weiteren CNNs standardmäßig die ReLu (*Rectified Linear Unit*) genutzt (He u. a., 2016; Richards, 2022; Ketkar u. a., 2021). Sie ist definiert als

$$f(x) = max(0, x) \tag{3.1}$$

Für negative Eingaben ist die ReLu-Funktion null, für positive Eingaben verläuft sie linear. Ihre Vorteile liegen zum einen in der unkomplizierten Berechnung, die die Laufzeit des Trainings positiv beeinflusst. Zum anderen beugt die Linearität dem *Vanishing/Exploding Gradient* vor, indem die Funktionsausgabe keine Sättigung erreichen

kann, wenn die Eingabe größer wird. Da negative Eingaben zudem zur Ausgabe null führen, wird die Funktion seltener aktiviert; dadurch können präzisere Modelle trainiert werden, die oft eine bessere Vorhersagekraft und weniger *Overfitting* aufweisen. Auch ist ein Netzwerk mit weniger Aktivierungen schneller, da weniger Berechnungen durchgeführt werden. Der Nachteil der Aktivierungsfunktion liegt in einem Phänomen, das als *dying* ReLu bezeichnet wird und auftreten kann, wenn die Lernrate (Kapitel 3.2.5) zu groß gewählt ist. Ein ReLU-Neuron ist "tot", wenn es im negativen Bereich feststeckt und immer 0 ausgibt. Es ist möglich, dass während eines Trainingslaufs Teile des Netzwerks durch die *dying* ReLu inaktiv werden.

Als Aktivierungsfunktion für die Ausgabeschicht wird die Sigmoid-Funktion (Ketkar u. a., 2021) genutzt. Sie ist definiert als

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3.2}$$

Die Sigmoid-Funktion gibt Werte im Bereich von 0 bis 1 aus. Je größer/positiver die Eingabe, desto näher liegt der Ausgabewert bei 1; je kleiner/negativer die Eingabe, desto näher liegt der Ausgabewert bei 0. Sie wird aufgrund ihres Wertebereichs verwendet, um Wahrscheinlichkeiten als Ausgabe zu erhalten.

3.2.4. Verlustfunktion

Die Verlustfunktion berechnet den Unterschied zwischen der Vorhersage des Modells und dem tatsächlichen Wert. Sie liefert Informationen über die Güte des Modells und damit die Grundlage für die Aktualisierung des Netzwerks.

Im Trainingssetup wird der Kreuzentropieverlust verwendet. Er wird für mehrklassige Klassifizierungsaufgaben empfohlen, um auftretende Klassenungleichgewichte im Datensatz zu kompensieren (Ketkar u. a., 2021; Tuia u. a., 2021). Mit der Funktion wird jede vorhergesagte Klassenwahrscheinlichkeit mit dem tatsächlichen Wert verglichen. Daraus wird der Verlust berechnet und auf die ausgegebene Wahrscheinlichkeit angewandt. Die Anwendung erfolgt dabei logarithmisch.

Der Kreuzentropieverlust ist definiert als

$$L_{CE} = -\sum_{i=1}^{n} t_i \cdot log(p_i) \tag{3.3}$$

n ist die Klassenanzahl, t_i das tatsächliche Label und p_i die vom Modell ausgegebene Wahrscheinlichkeit. Für kleinere Vorhersagewahrscheinlichkeiten wächst der Verlust exponentiell an und wird für $p_i = 0$ unendlich. Für $p_i = 1$ geht er gegen Null.

Zusätzlich ist in die Verlustfunktion *Label Smoothing* (Müller u. a., 2019) integriert um einer *Overconfidence* des Modells vorzubeugen. *Overconfidence* führt bei schlecht kalibrierten Modellen dazu, dass die vorhergesagten Wahrscheinlichkeiten höher liegen als die Gesamtgenauigkeit (*Overall Accuracy*). Während des Gradientenabstiegs

$$\nabla CE = p - y = Sigmoid - y \tag{3.4}$$

mit p als Ausgabewahrscheinlichkeit und y als Klassenverteilung sollen p und y so weit wie möglich angenähert werden. Der Gradient liegt dabei zwischen -1 und 1. *One-hot-encoded* Eingaben, deren Werte nur 0 oder 1 sein können, führen dazu, dass die größtmöglichen *Logit*-Lücken in die Verlustfunktion gehen. Diese ergeben in Kombination mit dem begrenzten Wertebereich des Gradienten weniger anpassungsfähige Modelle mit dem Problem der *Overconfidence*. *Label Smoothing* kann die *Logit*-Lücken verkleinern und so für eine bessere Kalibrierung des Modells sorgen. Es verändert die *one-hot-encoded* Eingabe y_{hot} mithilfe einer Gleichverteilungsfunktion:

$$y_{ls} = (1 - \alpha) \cdot y_{hot} + \alpha \div K \tag{3.5}$$

K enspricht der Klassenanzahl; α ist der Hyperparameter, der den *Smoothing*-Grad bestimmt. Wenn $\alpha = 0$ bleibt y_{hot} unverändert, mit $\alpha = 1$ erhält man die Gleichverteilungsfunktion. Im vorliegenden Modell ist $\alpha = 0,1$, wie empfohlen (Müller u. a., 2019).

Der Funktion kann außerdem bei auftretenden Klassenungleichgewichten eine Gewichtung zugeschaltet werden. Die Gewichtung ergibt sich aus folgenden Schritten: Zuerst wird die Gesamtpixelanzahl des Datensatzes und deren 30. und 70. Perzentil bestimmt. Je nach genutztem Landbedeckungsmodell werden alle Pixel pro Klasse aufsummiert. Bei der Einzelbetrachtung der Pixelsummen pro Klasse gilt dann: Wenn die Pixelsumme kleiner als das 30. Perzentil ist, wird sie durch das 30. Perzentil ersetzt, wenn sie größer als das 70. Perzentil ist, wird sie durch das 70. Perzentil ersetzt, wenn sie zwischen den Perzentilen liegt oder gleich diesen ist, bleibt die Pixelsumme bestehen.

Aus den angepassten Pixelanzahlen y_n werden dann die Gewichte pro Klasse w_n berechnet:

$$w_n = (\frac{y_n}{\sum(y_n)})^{-1} \tag{3.6}$$

Die Gewichte werden abschließend noch skaliert

$$ws_n = \frac{w_n}{\sum w_n} \tag{3.7}$$

und in dieser Form in die Verlustfunktion übergeben.

3.2.5. Lernrate und Epochenzahl

Die Lernrate ist eines der wichtigsten Hyperparameter für die Optimierung neuronaler Netzwerke (Pointer, 2021). Sie bestimmt die Schrittweite des Optimierers mit dem Ziel der Minimierung der Verlustfunktion. Mit einer niedrigeren Lernrate ist das Training zuverlässiger; die Optimierung des Netzwerks dauert jedoch länger, da die Schritteweite in Richtung des Minimums der Verlustfunktion gering ist. Mit einer zu hohen Lernrate kann es passieren, dass die Gewichtsänderungen so groß sind, dass der Optimierer das Minimum verfehlt und den Verlust verschlechtert.

Die für das Netzwerk verwendete initiale Lernrate wird auf 0,0001 festgelegt. Alle 20 Epochen wird die Lernrate halbiert; ein Trainingslauf besteht aus 60 Epochen. Alle Parameter sind empirisch getestet (Tabelle 4.1) und die Modellgüte anhand der Validierungs-IoU bewertet.

Zusätzlich wird für die ersten fünf Epochen eines Trainingslaufs eine *Warm Up* Lernrate (He u. a., 2016; Nakamura u. a., 2021; Goyal u. a., 2017) von $1e^{-8}$ festgelegt. Das *Warm Up* der Lernrate verwendet in den ersten Trainingsepochen eine sehr kleine Schrittgröße. Da die Modellparameter mit einer Zufallsverteilung initialisiert werden, unterscheidet sich das initiale Modell stark von einem optimierten Modell; ein präzises Training des initialen Modells in den ersten Epochen, gefolgt von einer größeren Lernrate, führt zu einer höheren Genauigkeit.

3.2.6. Optimierer

Während der *Backpropagation* werden die Gewichte des Netzwerks durch das Gradientenabstiegsverfahren angepasst. Der negative Gradient der Verlustfunktion wird hierbei als Abstiegsrichtung gewählt. Dazu aktualisiert der Optimierer das neuronale Netz auf Grundlage der Ergebnisse der Verlustfunktion (Ketkar u. a., 2021) unter Berücksichtigung der Lernrate. Diese gibt vor, wie weit der Optimierer die Gewichte für einen *Batch* in die dem Gradienten entgegengesetzte Richtung verschieben soll. Das Ziel besteht darin, die Verlustfunktion zu minimieren; je geringer der Verlust, desto besser ist das Modell. Die klassische Optimierungsmethode ist der stochastische Gradientenabstieg. Dabei wird eine einzelne globale Lernrate für alle Aktualisierungen der Gewichte genutzt. Im vorliegenden Netzwerk wird der Adam-Optimierer (*Adaptive Moment Estimation*) (Kingma u. a., 2014) genutzt. Adam kombiniert Funktionen aus zwei anderen Methoden: rmsprop (*Root mean Square*), der quadratische Gradienten zur Skalierung der Lernrate verwendet und AdaGrad (*Adaptive Gradient Algorithm*), welcher gleitende Mittelwerte des Gradienten anstelle der direkten Verwendung des Gradienten nutzt (Ketkar u. a., 2021). Adam berechnet den exponentiellen gleitenden Mittelwert des Gradienten und den quadrierten Gradienten pro Parameter und passt auf Grundlage dessen die globale Lernrate individuell pro Parameter an. Die Methode ist rechnerisch effizient und benötigt wenig Speicherplatz. Außerdem eignet sie sich gut für Anwendungen mit großen Datenmengen oder hoher Parameterzahl.

4. Ergebnisse

Nachfolgend werden die Ergebnisse der Arbeit sowie die zugrundeliegende Methodik zur Auswertung vorgestellt. Kapitel 4.1 beschreibt die durchgeführten Experimente sowie die zentralen Fragestellungen, die untersucht werden sollen. Im folgenden Kapitel 4.2 werden die zur Bewertung der Ergebnisse genutzten Metriken erläutert. Die anschließenden Kapitel 4.3 bis 4.7 enthalten die Resultate der Experimente.

4.1. Experimente

Auf Grundlage der Evaluierungsergebnisse aus Kapitel 3 werden sieben Experimente (Tabelle 4.1) durchgeführt. Untersuchungsgegenstand der Versuche ist zum einen die Bestimmung des Einflusses der beiden *Sampling* Methoden auf die Datensatz- und Modellgüte. Zum anderen wird als Hauptbestandteil der Testreihe die Modellleistung für die drei vorliegenden Landbedeckungsmodelle lbakt, lnakt und CORINE *Land Cover* evaluiert. Im Detail untersucht werden dazu erstens Unterschiede der Modellgüte zwischen den drei Modellen, zweitens klassenspezifische Qualitätsunterschiede innerhalb der Modelle, sowie drittens das Potenzial einer möglichen Klassenaggregierung am Beispiel von lbakt.

Tabelle 4.1.: Durchgeführte Experimente mit variierendem Landbedeckungsmodell und variierender Sampling Methode. Für lbakt wird zusätzlich ein Aggregierungsschema getestet.

Experiment	Landbedeckungsmodell	Samplingmethode
11	lbakt	Random
12	lbakt	Stratifiziert
13	lbakt (aggregiert)	Random
14	lnakt	Random
15	lnakt	Stratifiziert
16	CLC	Random
17	CLC	Stratifiziert

4.2. Metriken zur Bewertung der Modellgüte

Zur Bewertung der Modellergebnisse werden mehrere Metriken betrachtet, die sowohl eine Einschätzung der Gesamtmodellgüte als auch eine klassenspezifische Beurteilung erlauben.

Die jeweilige Modellgüte wird während des Trainings nach jeder Epoche mit der Validierungs-IoU bewertet. Das Modell mit dem besten IoU-Wert wird gespeichert und nach dem Abschluss des Trainings für die Klassifizierung mit dem Testdatensatz verwendet. Die Bewertung der Modellgüte im Rahmen der Klassifizierung erfolgt anhand mehrerer Metriken. Für eine Gesamtbewertung des Modells werden die Gesamtgenauigkeit (*Overall Accuracy*) sowie Cohen's Kappa auf Pixelebene bestimmt. Die Gesamtgenauigkeit ist definiert als

$$OA = \frac{TP \cdot TN}{Gesamtpixelanzahl} \tag{4.1}$$

mit TP und TN als Summe der korrekt klassifizierten Pixel (*True Positives* und *True Negatives*). Sie gibt das Verhältnis der korrekt klassifizierten Pixel zur Gesamtpixelanzahl an (Bruce u. a., 2021). Der Kappa-Koeffizient ist definiert als

$$k = \frac{p_o - p_e}{1 - p_e} \tag{4.2}$$

mit p_o als beobachtete Übereinstimmung und p_1 als erwartete Übereinstimmung. Die Gesamtgenauigkeit allein ist für eine Modellbewertung bei auftretenden Klassenungleichgewichten nicht ausreichend. Werden beispielsweise nur sehr kleine Klassen schlecht vom Modell widergegeben, große werden aber präzise klassifiziert, ist dies nicht in der Gesamtgenauigkeit erkennbar, die trotzdem hoch bleibt. Um eine präzisere Aussage unter Einbeziehung der Klassenungleichgewichte für die Gesamtmodellgüte treffen zu können, wird zusätzlich Cohen's Kappa (Landis u. a., 1977) berechnet. Cohen's Kappa gibt an, wie viel besser das Modell verglichen mit einem Modell abschneidet, das per Zufallsprinzip entsprechend der Häufigkeit der einzelnen Klassen vorgeht. Der Wert ist dimensionslos und kleiner oder gleich 1, die Interpretation erfolgt nach Wertebereichen: Wenn Cohen's Kappa kleiner als 0 ist, liegt keine Übereinstimmung vor, 0-0,20 bedeutet eine geringfügige, 0,21-0,40 eine moderate, 0,41-0,60 eine mäßige, 0,61-0,80 eine erhebliche und 0,81-1 eine nahezu perfekte Übereinstimmung.

Für die klassenspezifische Bewertung der Modellgüte wird eine Konfusionsmatrix genutzt (Abbildung 4.1). Auf deren Basis werden für jede Klasse *Precision*, *Recall* und f1-Wert berechnet.

Actual \ Predicted	Class 1	Class 2	Class 3
Class 1	TP	FN	FN
Class 2	FP	TN	
Class 3	FP		TN

Abbildung 4.1.: Konfusionsmatrix zur Berechnung der klassenspezifischen Gütemaße. Bezogen auf die Klasse 1 mit TP (True Positives), TN (True Negatives), FP (False Positives), FN (False Negatives).

Die *Precision* (Relevanz) gibt das Verhältnis der korrekt in die Klasse n klassifizierten Pixel zur Gesamtanzahl der als Klasse n klassifizierten Pixel an (Bruce u. a., 2021). Sie ist definiert als

$$Precision = \frac{TP}{TP + FP} \tag{4.3}$$

Je höher der Wert, desto weniger Pixel sind fälschlicherweise als Klasse n klassifiziert worden. Der *Recall* (Sensitivität) bestimmt das Verhältnis der korrekt als zur Klasse n zugehörig identifizierten Pixel zur tatsächlichen Gesamtpixelanzahl der Klasse n (Bruce u. a., 2021). Er ist definiert als

$$Recall = \frac{TP}{TP + FN} \tag{4.4}$$

Je höher der Wert, desto mehr Pixel der Klasse n sind als der Klasse n zugehörig identifiziert worden. Mit dem f1-Wert wird außerdem das gewichtete harmonische Mittel aus *Precision* und *Recall* berechnet:

$$f1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4.5}$$

Damit lässt sich eine Aussage zur Gesamtgüte einer betrachteten Klasse treffen.

4.3. Einfluss der Samplingmethode auf Datensatz- und Modellgüte

Für die drei Landbedeckungsmodelle lbakt, lnakt und CLC werden Experimente mit beiden *Sampling* Methoden durchgeführt, um den Einfluss der jeweiligen Klassenungleichgewichte auf die Datensatz- und Modellqualität abzuschätzen und gegebenfalls durch die Wahl der passenden *Sampling* Methode auszugleichen.

Bezüglich der Qualität der Teildatensätze lassen sich sowohl für lbakt, als auch für lnakt und CLC Unterschiede zwischen den Samplingmethoden feststellen. Diese werden nachfolgend exemplarisch an den Ergebnissen mit lbakt erläutert (Abbildung 4.2). Die *Sampling* Eergebnisse für lnakt und CLC sind im Anhang (A.1) hinterlegt. In die folgende Auswertung nicht miteinbezogen werden die Ergebnisse der kleinsten Klasse Brandfläche, vorkommend in lbakt und CLC, da die Klasse nur auf einer Szene vertreten ist und aus diesem Grund in keinem Fall sinnvoll gesampelt werden kann.

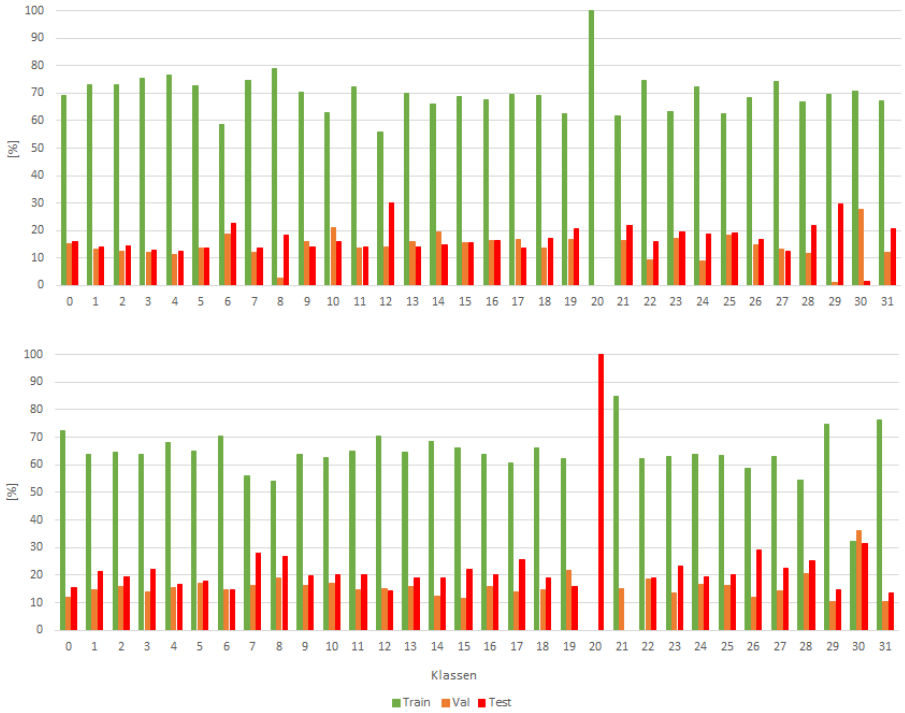

Abbildung 4.2.: Ergebnisse des *Samplings* für lbakt. Pixelanteile [%] für Trainings- Validierungs- und Testdatensatz. Oben: *Random Sampling*. Unten: Stratifiziertes *Sampling*.

Die Zuweisung der Szenen durch die *Random Sampling* Methode ergibt mehrere Unregelmäßigkeiten; für die lbakt-Klassen 8, 22 und 24, sowie 29 werden sehr kleine Validierungsdatensätze generiert, die jeweils einen Pixelanteil unter 10% aufweisen (Abbildung 4.2 oben). Im Testdatensatz ist die Klasse 30 nur unzureichend vertreten. Im Vergleich hierzu können mit einem stratifizierten *Sampling* die kleinen Klassen des Datensatzes gezielter in die Teildatensätze aufgeteilt werden (Abbildung 4.2 unten).

Alle relevanten Klassen sind zu mindestens 10% in den Validierungs- und Testdatensätzen vertreten.

Bei lnakt fallen die Unterschiede zwischen den *Sampling* Methoden geringer aus. Zurückzuführen ist dieses Ergebnis auf die deutlich geringere Klassenanzahl mit gleichzeitig schwächeren Klassenungleichgewichten. Nur eine Klasse ist mit *Random Sampling* mit weniger als 10% Pixelanteil im Validierungsdatensatz vertreten, was im stratifizierten *Sampling* behoben werden kann. Die CLC-Klassen weisen, bedingt durch die starken Klassenungleichgewichte im Gesamtdatensatz, mit der *Random Sampling* Methode Unregelmäßigkeiten ähnlich denen von lbakt auf. Durch die Anwendung des stratifizierten *Samplings* kann diese unausgewogene Repräsentation kleiner Klassen in den Teildatensätzen zunächst behoben werden; allerdings ergeben sich stellenweise neue Unregelmäßigkeiten, wenn Klassen in den Validierungs- und Testdatensatz *oversampled* werden und dadurch der Trainingsdatensatz unverhältnismäßig klein wird. Dies ist der Fall, wenn Klassen wie in Kapitel 3.1.3 dargestellt, insgesamt nur auf wenigen Szenen vertreten sind und die Pixelanzahlen pro Szene stark variieren.

Anders als bei der Qualität der Teildatensätze kann für die beiden *Sampling* Methoden in der initialen Bewertung der Modellleistung kein signifikanter Unterschied festgestellt werden. Wie in Tabelle 4.2 dargestellt, unterscheiden sich die Ergebnisse der beiden *Sampling* Methoden hinsichtlich der Validierungs-IoU und der Gesamtgenauigkeit nicht mehr als um vier Prozentpunkte. Cohen's Kappa weicht maximal um 0,02 ab.

Tabelle 4.2.: Gütemaße für die Experimente mit LBM-DE.

Landbedeckungsmodell	Samplingmethode	Val-IoU	Gesamtgenauigkeit	Kappa
lbakt	Random	51%	71%	0,66
lbakt	Stratifiziert	52%	69%	0,64
lnakt	Random	75%	84%	0,78
lnakt	Stratifiziert	71%	84%	0,77
CLC	Random	54%	71%	0,66
CLC	Stratifiziert	54%	67%	0,64

Zusammenfassend lässt sich aus der Wahl der *Sampling* Methode in der Gesamtbetrachtung ein Einfluss auf die Klassenverteilung in den Teildatensätzen feststellen, ein signifikanter Einfluss auf die Gesamtmodellgüte resultiert daraus aber nicht. Aus diesem Grund werden nachfolgend die Klassifizierungsergebnise für lbakt, lnakt und CLC jeweils exemplarisch an den Experimenten basierend auf dem *Random Sampling* dargestellt. Die Ergebnisse der Experimente 12, 15 und 17 mit stratifiziertem *Sampling* sind im Anhang der Arbeit hinterlegt. Treten hinsichtlich einzelner Klassen signifikante Unterschiede zwischen den *Sampling* Methoden auf, wird an entsprechender Stelle darauf hingewiesen.

4.4. Landbedeckungsmodell (lbakt)

Mit den Experimenten 11 und 12 wird die Modellleistung bezüglich der Identifikation von Landbedeckung evaluiert. Die allgemeine Bewertung des in Experiment 11 trainierten Modells ergibt eine Validierungs-IoU von 51%. Im Test erreicht das Modell eine Gesamtgenauigkeit von 71%, Cohen's Kappa liegt bei 0,66. In der Gesamtbetrachtung der klassenspezifischen Ergebnisse (Abbildung 4.3) werden fast alle Klassen überwiegend innerhalb ihrer Klassengruppe (bebaute und versiegelte Flächen, Anbauflächen, Grünland, etc.) klassifiziert. Dieser Trend wird durch die visuellen Ergebnisse (Abbildung 4.4 A und B) bestätigt: Das Modell ist dazu in der Lage, semantische Klassengruppen wie Bebauung, Ackerland, Grünland, Baumbestand und Gewässer klar voneinander zu unterscheiden.

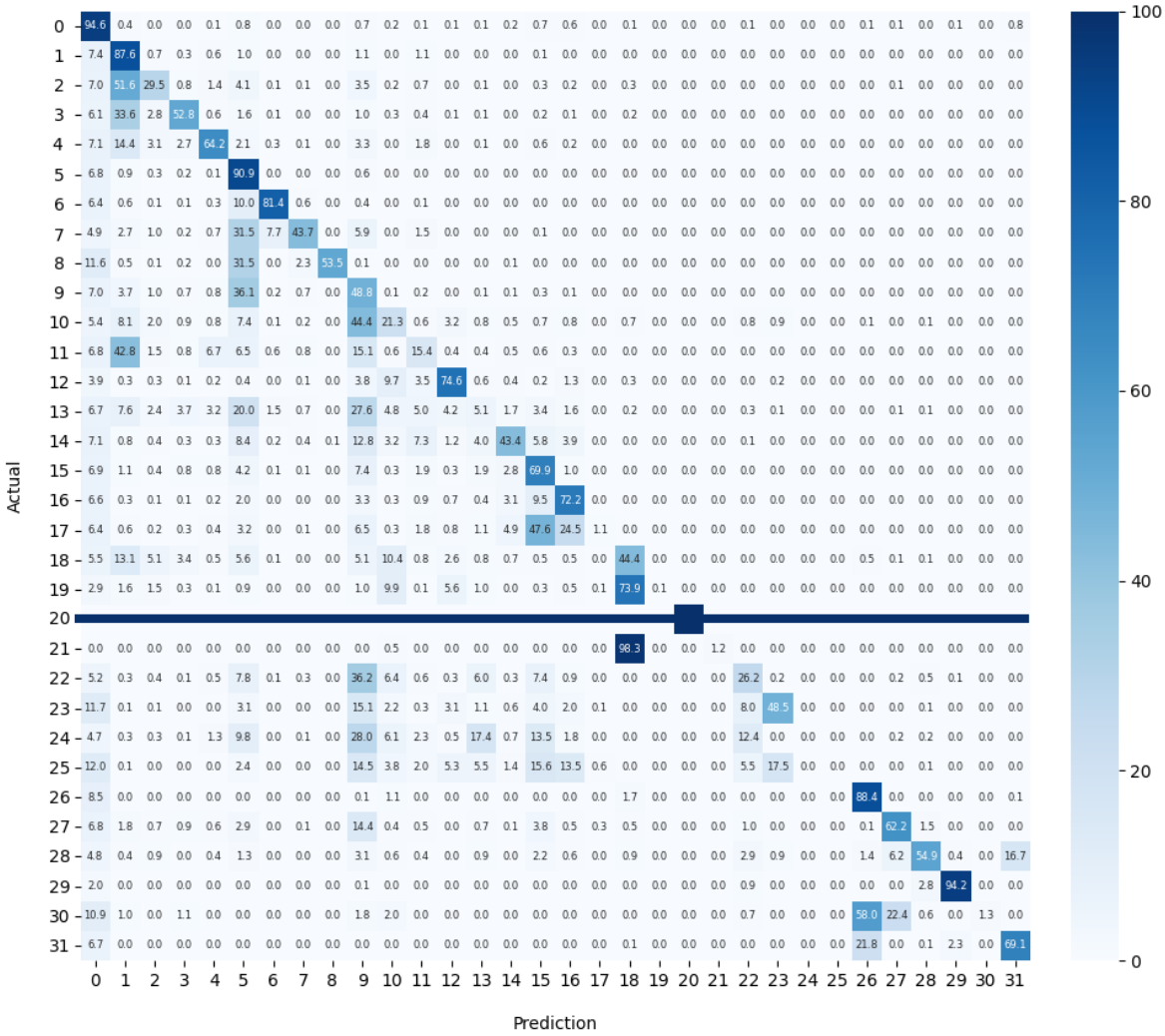

Abbildung 4.3.: Konfusionsmatrix für Experiment 11. Klasse 20 (blau durchgestrichen) ist nicht im Testdatensatz enthalten.

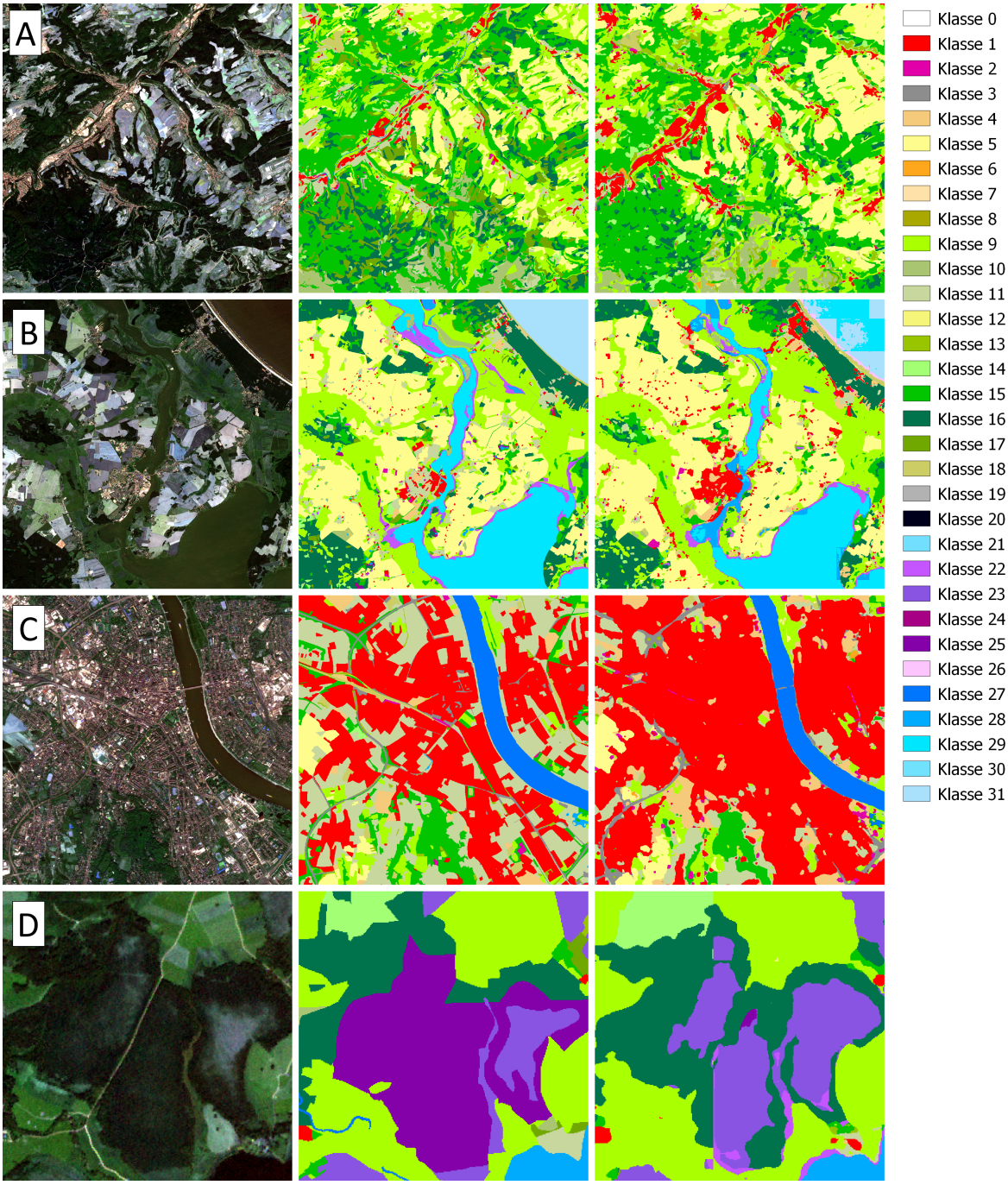

Abbildung 4.4.: Beispiele der Klassifizierungsergebnisse des Modells aus Experiment 11. Links: Aufnahme aus dem RapidEye2015 Datensatz. Mitte: Landbedeckungsmodell lbakt. Rechts: Modellvorhersage. Mit dem Modell können Bedeckungsstrukturen (Siedlung, Grünland, Baumbestand, Gewässer) identifiziert werden. Fehlklassifizierungen treten beispielsweise in Form von Klassenüberschätzungen in der Bebauung (C) oder bei Mischklassen wie Sumpf- und Moorflächen mit Baumbestand (D) auf.

Größere Abweichungen treten nur in wenigen Klassen wie Klasse 11 (Grasland mit Bäumen), Klasse 13 (Büsche und Sträucher), sowie 22 bis 25 (Moor- und Sumpfflächen) auf. 42,8% der Pixel der Klasse 11 werden als Bebauung klassifiziert (Abbildung 4.4 C). Dabei handelt es sich um im Landbedeckungsmodell als kleinräumige Grünflächen im städtischen Raum ausgewiesene Objekte, in Form von Innenhöfen, Gärten, Grünstreifen usw. Diese werden im Modell als bebaute Flächen klassifiziert. Die der Klasse 13 zugehörigen Pixel werden in 20% der Fälle als Klasse 5 (Ackerland) und zu 27,6% als Klasse 9 (Homogenes Grünland) identifiziert. Die Klassen 22 bis 25 werden überwiegend in die Klasse 9 klassifiziert. Klasse 24 und 25, bei denen es sich um Sumpf- und Moorflächen mit Baumbestand handelt, werden zudem noch in die Klassen 13, 15 und 16 (Baumbestandsarten) klassifiziert. Auch hier führt eine hohe Übereinstimmung spektraler Information von verschiedenen Klassen zu Fehlklassifizierungen (Abbildung 4.4 D). Die Ergebnisse der Gesamtbetrachtung werden durch die klassenspezifischen Metriken (Abbildung 4.5) bestätigt.

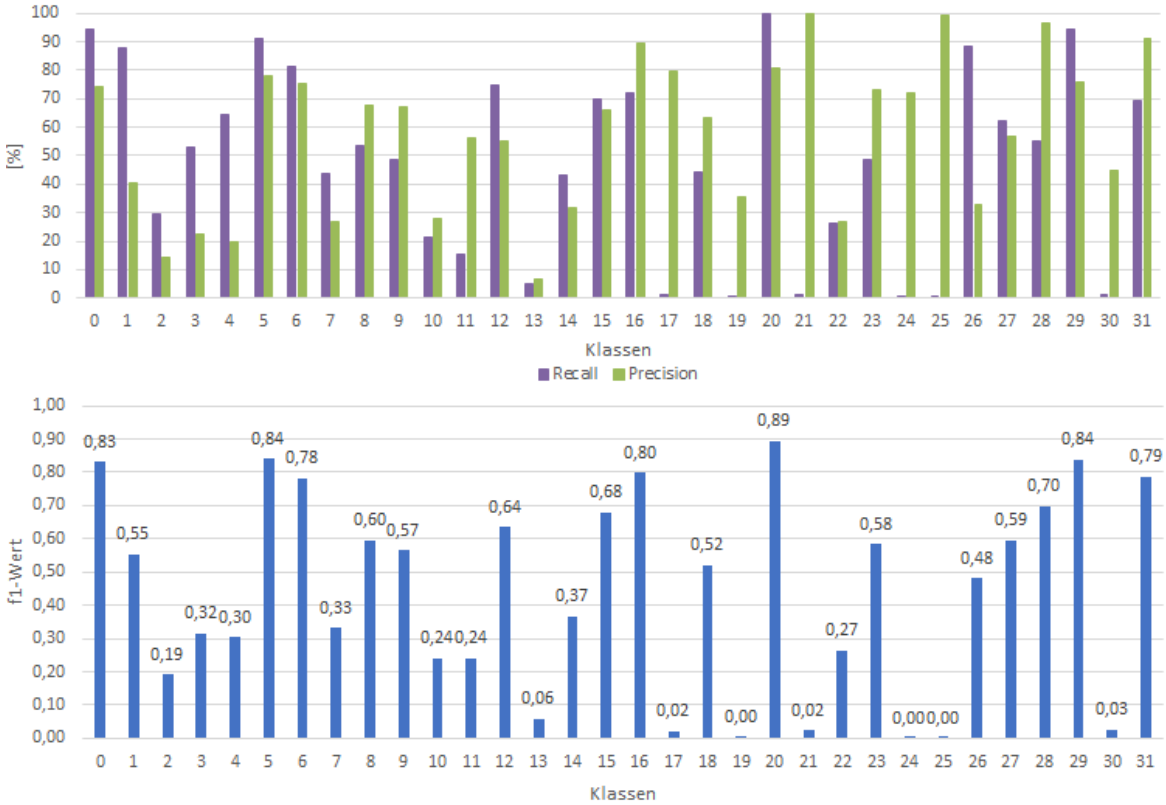

Abbildung 4.5.: Klassenspezifische Metriken für das Modell aus Experiment 11. Oben: *Recall* und *Precision*. Unten: f1-Wert. Die Ergebnisse der Klasse 20 werden nicht berücksichtigt, da die Klasse nicht im Testdatensatz vertreten war.

Die höchsten f1-Werte größer 0,8 erzielen die Klassen 0, 5 und 29. Die Klassen 13, 17, 19, 21, 24, 25 und 30 erhalten sehr niedrige f1-Werte, überwiegend verursacht durch einen niedrigen Recall. Tendiert der f1-Wert einer Klasse gegen Null, wird die Klasse durch das Netzwerk nur schlecht bis gar nicht erkannt. Es werden keine Pixel

fälschlicherweise in die betreffende Klasse identifiziert (Precision = 100%) Die Klasse bleibt jeoch insgesamt leer, da auch diejenigen Pixel, die der Klasse angehören, nicht als solche erkannt werden (Recall = 0%). Das ist beispielsweise bei den Klassen 21 und 25 der Fall. Die entsprechenden Ergebnisse für Experiment 12 sind im Anhang (A.2) zu finden.

Insgesamt ist eine valide Klassifizierung der Landbedeckung für die Bedeckungsarten Bebauung, Ackerland, Grünland, Baumbestand und Gewässer umsetzbar. Fehlklassifizierungen treten besonders bei kleineren Klassen und Mischklassen auf. Diese werden anderen Klassen zugeordnet, was zu einer Überklassifizierung an anderer Stelle führt.

4.5. Aggregiertes Landbedeckungsmodell (lbakt)

Die Ergebnisse der Experimente 11 und 12 deuten darauf hin, dass durch eine Klassenaggregierung die Modellleistung gesteigert werden kann. Fehlklassifizierung mit den Landbedeckungsmodell erfolgen überwiegend innerhalb semantisch ähnlicher Klassengruppen. Die mit Experiment 13 getestete Aggregierung fasst deshalb Klassen entsprechend der Ergebnisse der Konfusionsmatrizen der Experimente 11 und 12 zusammen. Beginnend bei Klasse 31 werden die Klassen absteigend auf eine Gesamtzahl von 15 aggregiert (Abbildung 4.6).

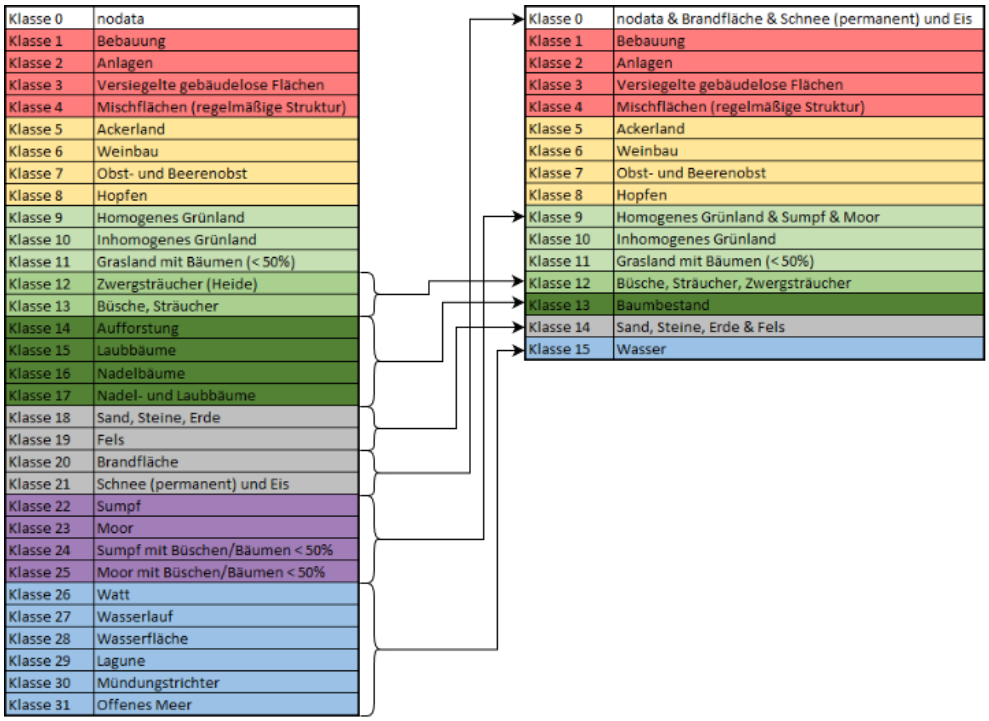

Abbildung 4.6.: Aggregierungsschema für Experiment 13, basierend auf dem Landnutzungsmodell lbakt. Die ursprünglich 31 Klassen werden auf 15 reduziert.

Dazu werden die Klassen 26 bis 31 zur Klasse Wasser zusammengefasst. Die Sumpf- und Moorklassen werden aufgrund ihrer hohen Fehlklassifikationen in die Klasse 9 und der semantischen Ähnlichkeit dieser Klasse zugefügt. Die Klassen 20 und 21 werden aufgrund ihrer geringen Größe aus dem Training ausgeschlossen und der Klasse 0 zugefügt. Die Klassen 18 und 19 werden zu einer Klasse zusammengefasst, ebenso die Klassen 14 bis 17 und 12 und 13. Die restlichen Klassen bleiben bestehen. Das Experiment soll mit dieser Methode eine erste Einschätzung des Potenzials einer Aggregierung liefern und bietet keine optimierte Aggregierungsstrategie.

Das in Experiment 13 trainierte Modell hat eine Validierungs-IoU von 62%. Im Test erreicht das Modell eine Gesamtgenauigkeit von 74%, Cohen's Kappa liegt bei 0,68. Insgesamt lässt sich ein Einfluss der Aggregierung auf die Ergebnisse erkennen (Abbildung 4.7).

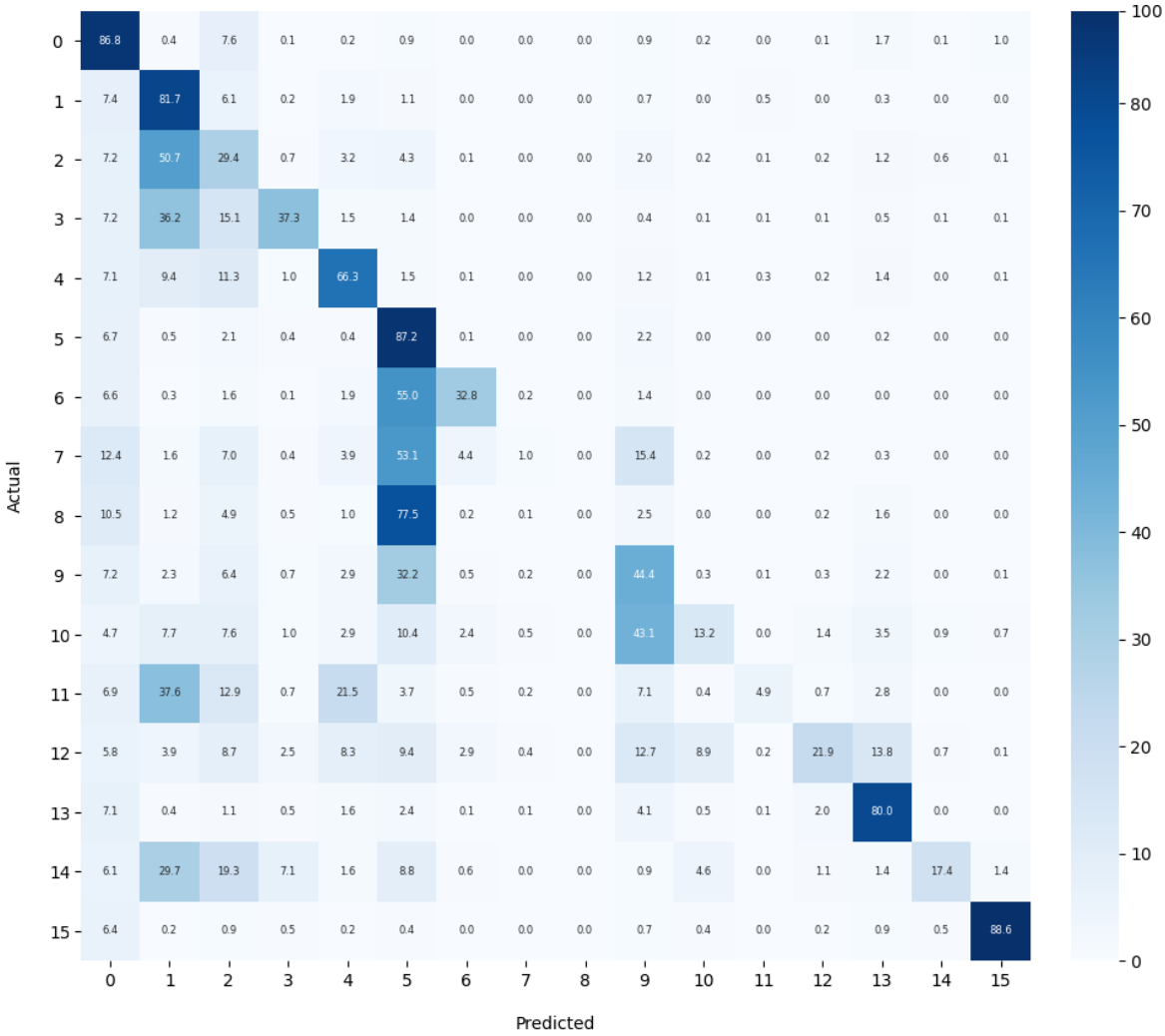

Abbildung 4.7.: Konfusionsmatrix für Experiment 13.

Sehr gute Ergebnisse liefert das Modell für die aggregierten Klassen 13 (Baumbestand) mit 88,6% und 15 (Gewässer) mit 80,0% korrekt klassifizierter Pixel. Die aggregierten Klassen 0, 9, 12 und 14 zeigen keine Verbesserung gegenüber Experiment 11 und 12. Die von der Aggregierung nicht betroffenen Klassen 7, 8 und 11 haben sich zudem verschlechtert. Die Klassen 1 und 5 werden am stärksten überschätzt. Diese Erkenntnisse werden durch die visuellen Ergebnisse bestätigt (Abbildung 4.8): Wasserflächen und Baumbestand werden treffend klassifiziert. Ebenfalls erkennbar ist die Überschätzung von Anbau- und bebauten Flächen.

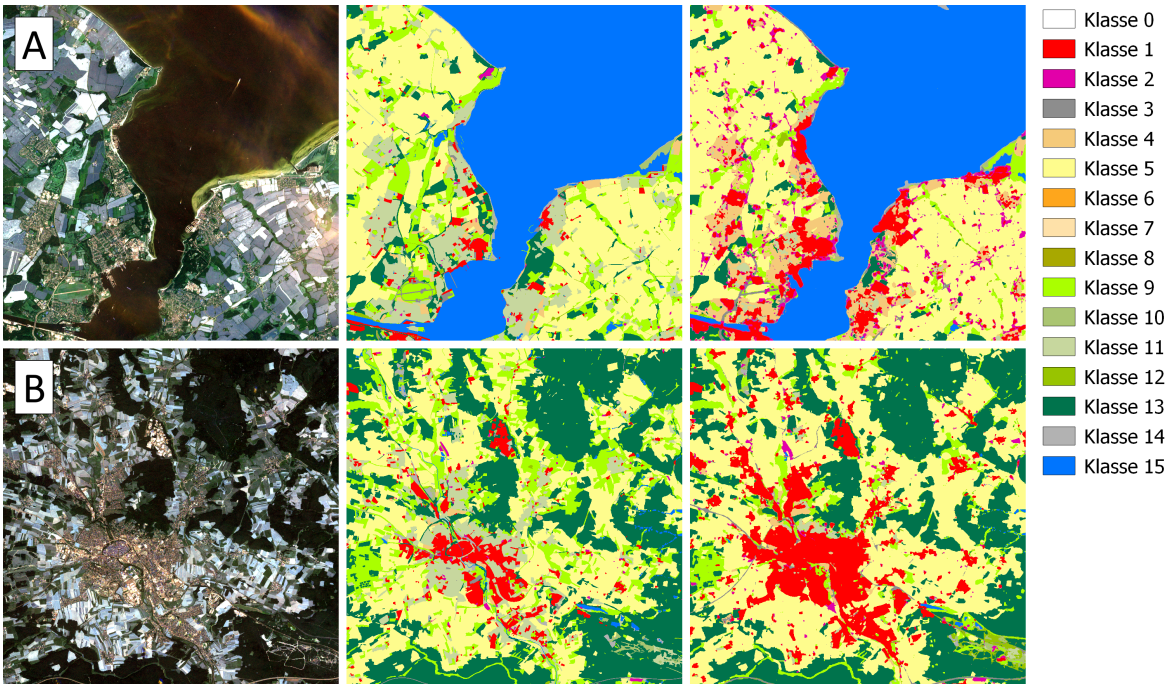

Abbildung 4.8.: Beispiele der Klassifizierungsergebnisse des Modells aus Experiment 13. Links: Aufnahme aus dem RapidEye2015 Datensatz. Mitte: Aggregiertes Landbedeckungsmodell lbakt. Rechts: Modellvorhersage. Das Modell klassifiziert Wasserflächen und Baumbestand treffend. Ebenso erkennbar ist die Überklassifizierung von Anbau- und bebauten Flächen.

Die klassenspezifischen Metriken zeigen die höchsten f1-Werte für die Klassen 0, 5, 13 und 15 (Abbildung 4.9). Die Klasse 5 (Ackerland) erhält obwohl sie große Anteile falsch klassifizierter Pixel der Klassen 6 bis 9 enthält, einen hohen *Precision*-Wert, da es sich bei den Klassen 6 bis 8 um kleine Klassen handelt, deren Fehlklassifizierung die Güte nicht stark beeinflussen. Die Klassen 2, 7, und 8 erhalten sehr niedrige f1-Werte, dabei liegen sowohl *Precision* als auch *Recall* sehr tief.

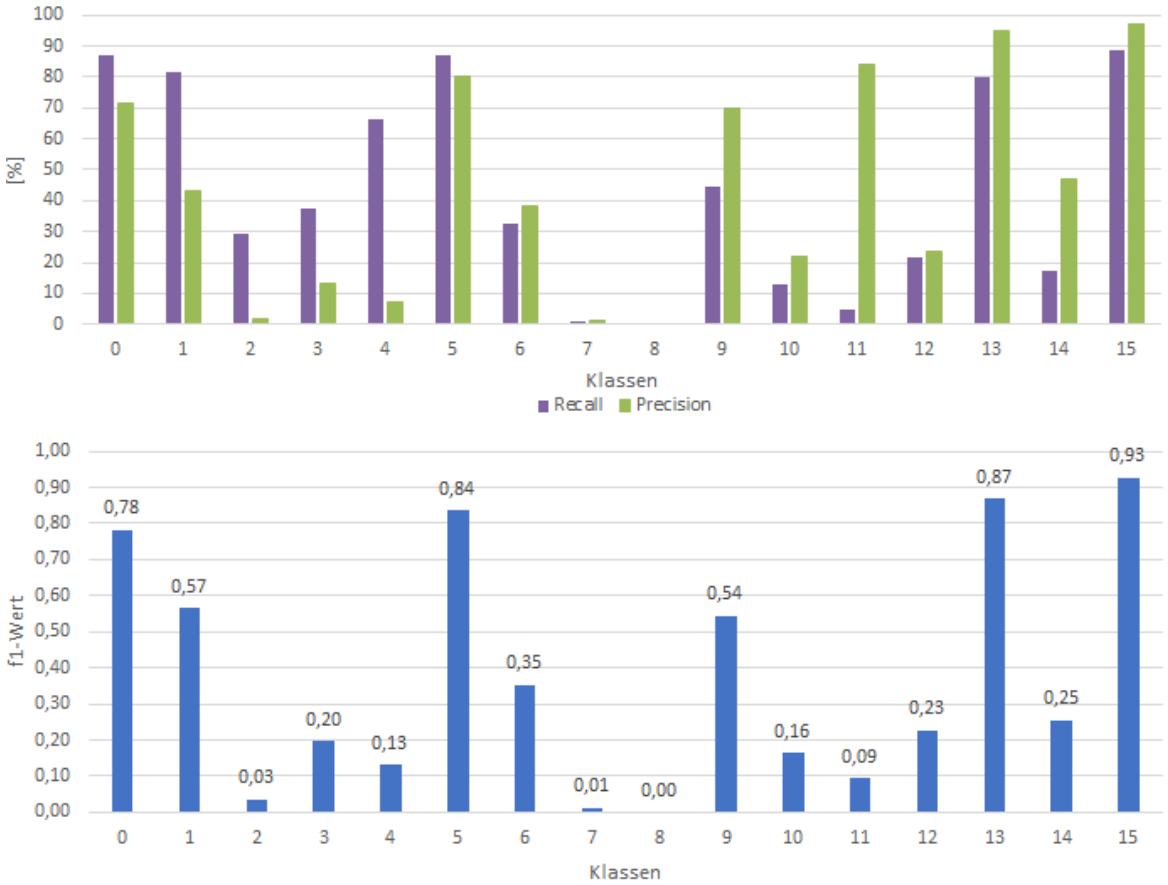

Abbildung 4.9.: Klassenspezifische Metriken für das Modell aus Experiment 13. Oben: *Recall* und *Precision*. Unten: f1-Wert.

Mit Experiment 13 kann die Klasssifizierung der aggregierten Klassen 13 und 15 mit hohen Genauigkeitsmaßen durchgeführt werden. Andererseits zeigen sich durch die Aggregierungen auch Verschlechterungen, sowohl hinsichtlich aggregierter Klassen als auch hinsichtlich der Klassen, die nicht in das Aggregierungsschema miteinbezogen werden.

4.6. Landnutzungsmodell (Inakt)

Mit den Experimenten 14 und 15 wird die Modellleistung bezüglich der Identifikation von Landnutzung evaluiert. Die allgemeine Bewertung des in Experiment 14 trainierten Modells ergibt eine Validierungs-IoU von 75%. Im Test erreicht das Modell eine Gesamtgenauigkeit von 84,36%, Cohen's Kappa liegt bei 0,78. In der Gesamtbetrachtung der klassenspezifischen Ergebnisse (Abbildung 4.10) werden Pixel aller Klassen in die Klasse 0 klassifiziert. Bei den größten Klassen 1, 5, 12 und 14 sowie der Klasse 8 tritt das Problem abgeschwächt auf. Daraus ergeben sich die hohen Werte der allgemeinen Gütemaße, da ein großer Anteil der Pixel korrekt klassifiziert wird.

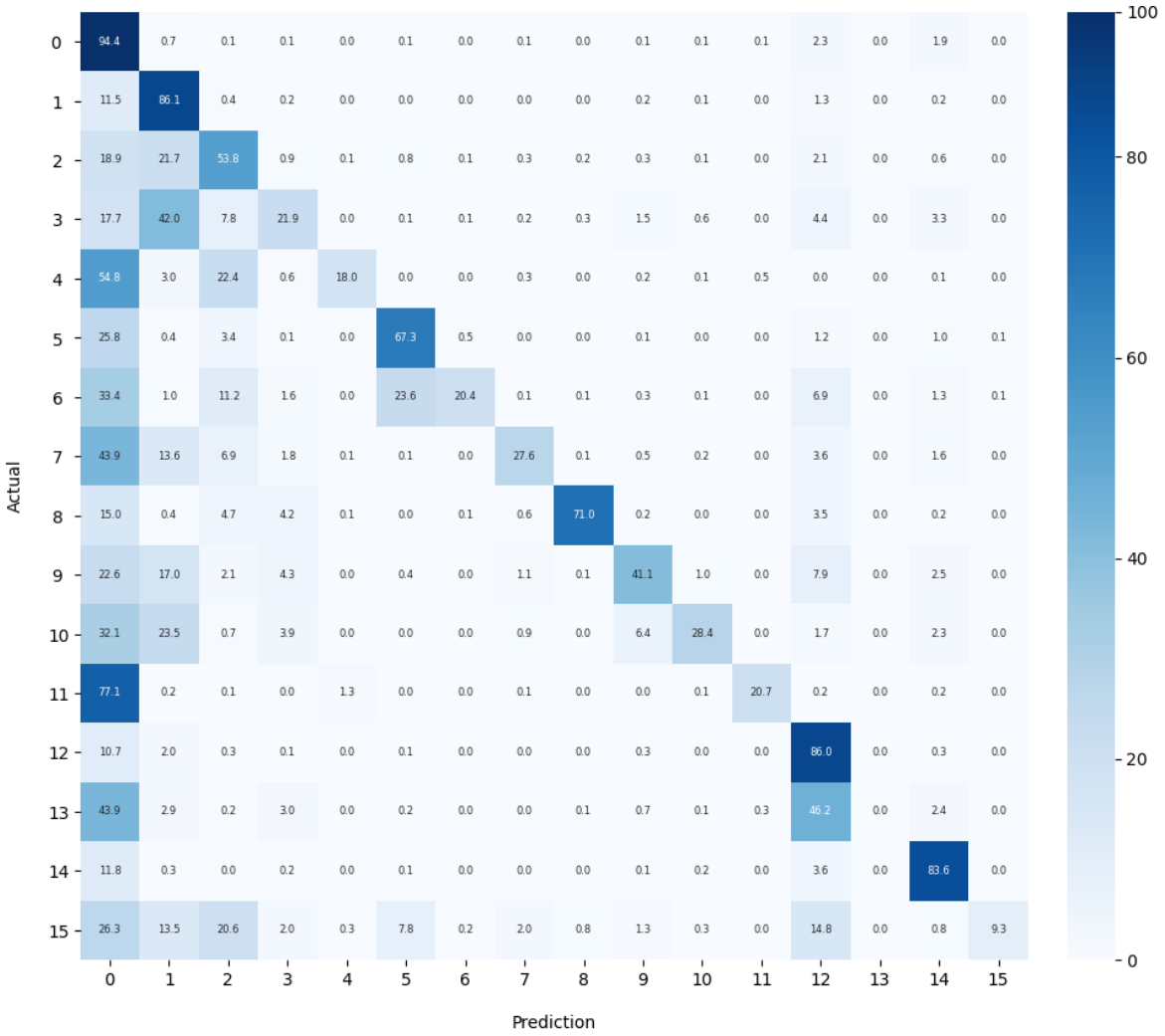

Abbildung 4.10.: Konfusionsmatrix für Experiment 14.

Die Erkenntnisse werden mit der visuellen Ergebnisauswertung verdeutlicht (Abbildung 4.11): Die im Datensatz am häufigsten auftretenden Klassen Wohnen (Klasse 1), intensive Landwirtschaft (Klasse 12) und Forstwirtschaft (Klasse 14) werden gut visualisiert. Erkennbar ist auch die Überklassifizierung der Klasse 0.

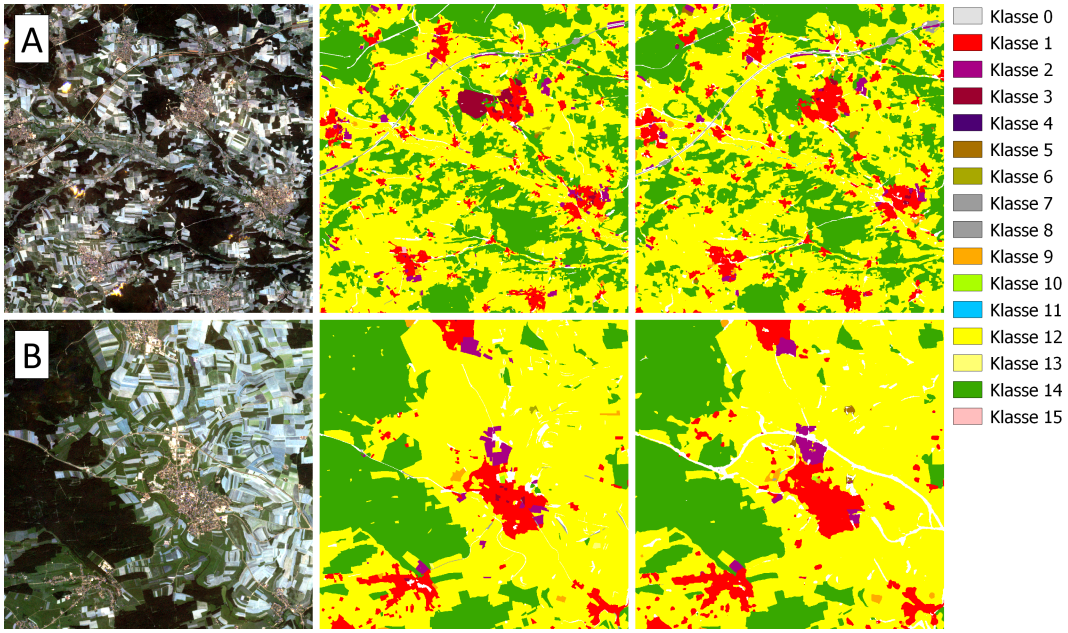

Abbildung 4.11.: Beispiele der Klassifizierungsergebnisse des Modells aus Experiment 14. Links: Aufnahme aus dem RapidEye2015 Datensatz. Mitte: Landnutzungsmodell lnakt. Rechts: Modellvorhersage. Die besten Ergebnisse werden für die Klassen Wohnen, intensive Landwirtschaft und Forstwirtschaft erzielt. Ebenfalls erkennbar ist die Überklassifizierugn der Klasse 0.

Die klassenspezifischen Metriken (Abbildung 4.12) liegen dementsprechend unter anderem für die Klassen 1, 12 und 14 am höchsten. Auch der f1-Wert der Klasse 0 liegt mit 0,80 trotz ihrer Überklassifizierung sehr hoch. Die fälschlicherweise in Klasse 0 klassifizierten Pixel machen im Verhältnis zur Größe der Klasse nur einen Bruchteil aus, wodurch die Klassengenauigkeit kaum beeinflusst wird. Klasse 13 hat den niedrigsten f1-Wert von 0. Sie wird vom Modell nicht erkannt und ist leer.

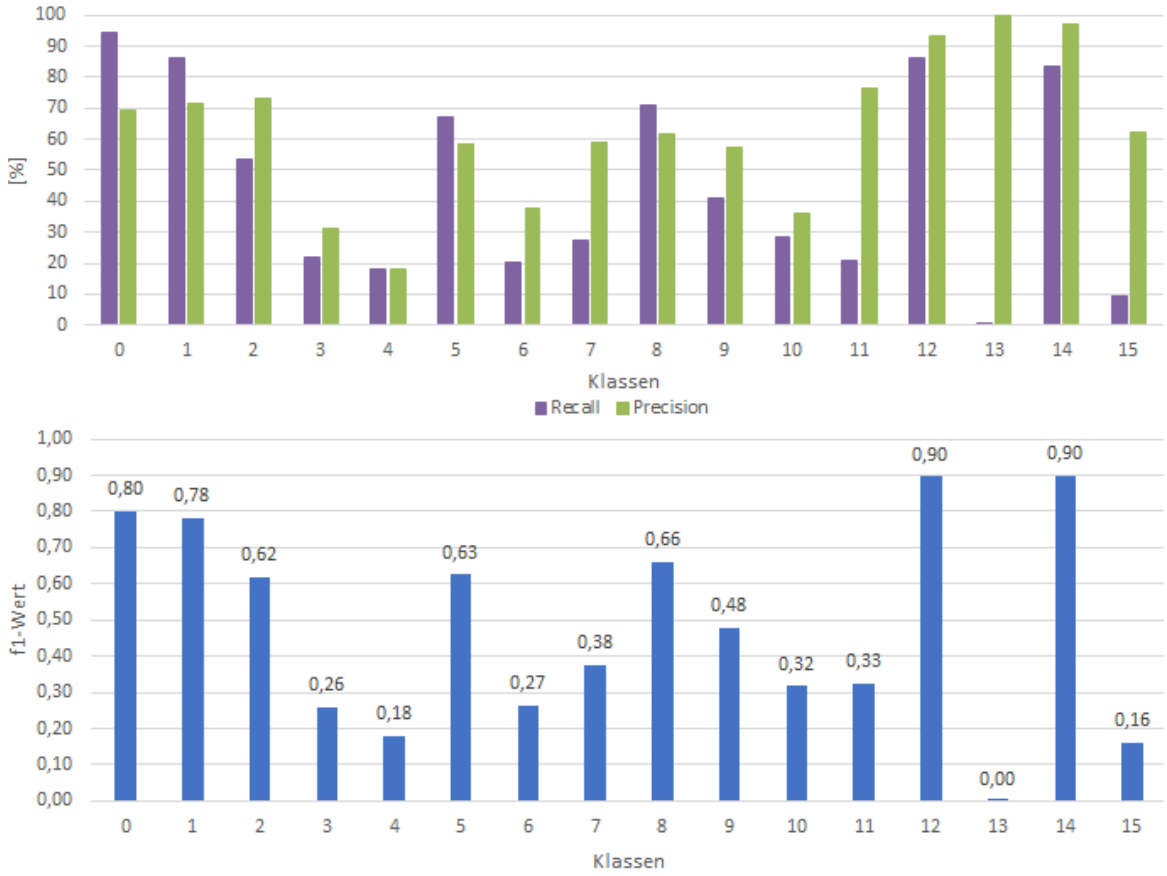

Abbildung 4.12.: Klassenspezifische Metriken für das Modell aus Experiment 14. Oben: *Recall* und *Precision*. Unten: f1-Wert.

Die Klassifizierung der Landnutzung kann für die größten Klassen Wohnen, intensive Landwirtschaft und Forstwirtschaft gut durchgeführt werden. Kleinere Klassen werden überwiegend in Klasse 0 fehlklassifiziert. Die entsprechenden Ergebnisse für Experiment 15 sind im Anhang (A.3) zu finden.

4.7. CORINE Land Cover (CLC)

Mit den Experimenten 16 und 17 wird die Modellleistung für die Identifikation der CORINE *Land Cover* evaluiert. Die allgemeine Bewertung des in Experiment 16 trainierten Modells ergibt eine Validierungs-IoU von 54%. Im Test erreicht das Modell eine Gesamtgenauigkeit von 71%, Cohen's Kappa liegt bei 0,64. Die klassenspezifischen Ergebnisse (Abbildung 4.13) spiegeln in ihrer Gesamtbetrachtung die Resultate der Landbedeckungs- und Landnutzungsmodelle wider. Die meisten Fehlklassifizierungen bewegen sich, wie bei lbakt, innerhalb der fünf Überkategorien Bebaute Flächen, Landwirtschaftliche Flächen, Wälder und naturnahe Flächen, Feuchtflächen und Wasserflächen. Abweichungen treten bei Mischflächen, wie den Klassen 15 und 16, 26 und 27, 29, sowie 32, 34 und 36 auf. Am besten werden die Klassen 0, 1 und 12 getroffen.

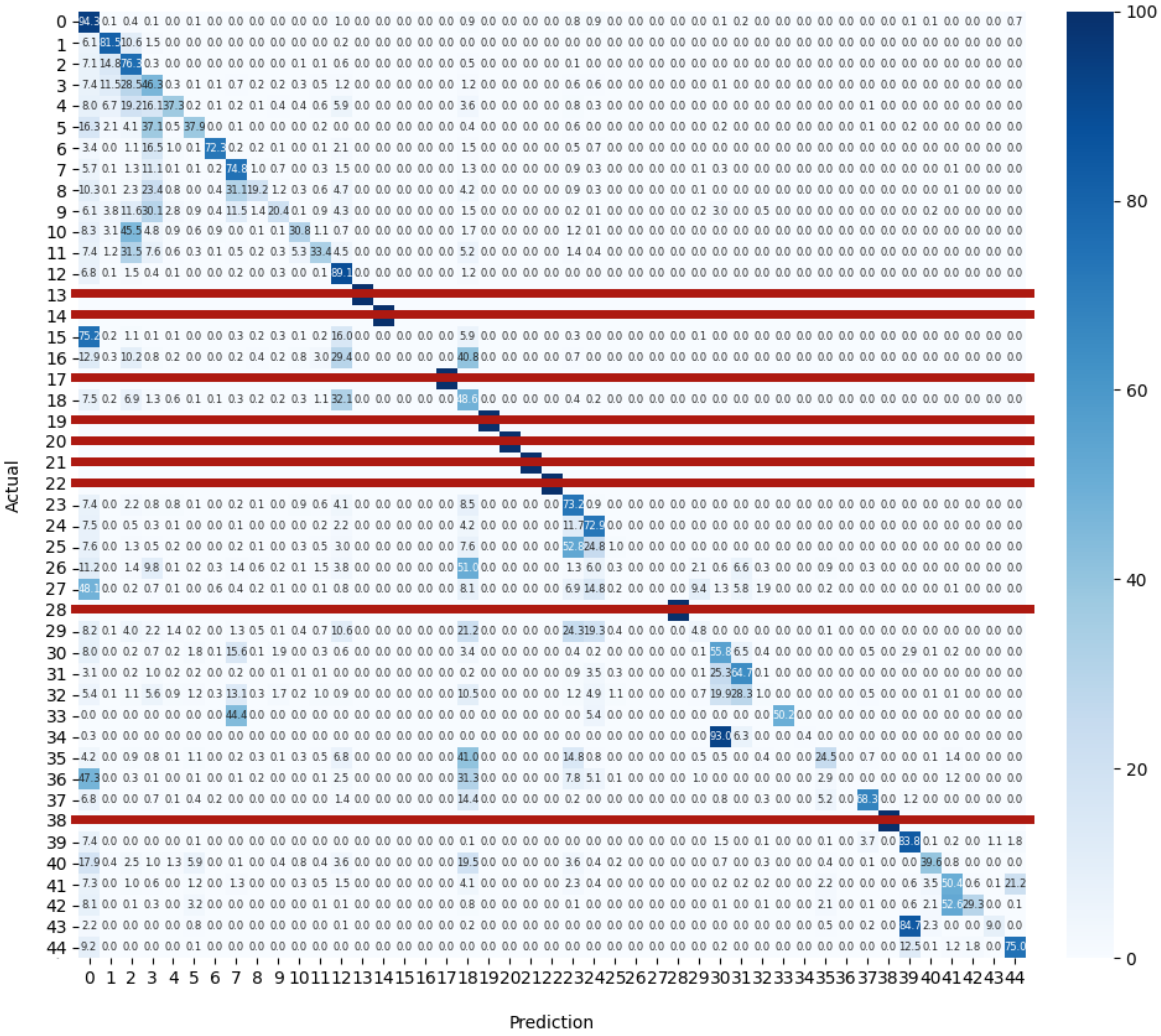

Abbildung 4.13.: Konfusionsmatrix für Experiment 16. Die rot durchgestrichenen Klassen kommen im Datensatz für Deutschland nicht vor.

Abbildung 4.14 zeigt exemplarische Klassifizierungsergebnisse des Modells. Verschiedene Arten bebauter Flächen werden von dem Modell gut erkannt. Ebenso gute Ergebnisse liefert die Klassifizierung von Wäldern und naturnahen Flächen (Abbildung 4.14 A). Anfälliger für Fehlklassifizierungen ist die spezifische Identifizierung kleinerer landwirtschaftlicher Klassen (Abbildung 4.14 A) und die korrekte Erkennung konkreter Feucht- und Wasserflächen-Klassen (Abbildung 4.14 B).

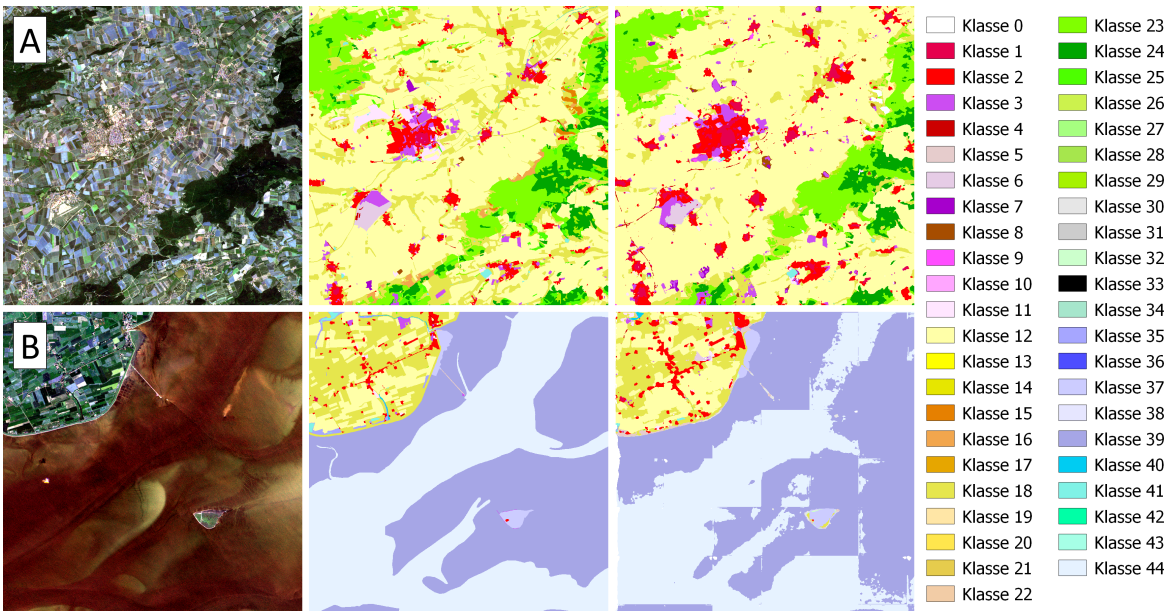

Abbildung 4.14.: Beispiele der Klassifizierungsergebnisse des Modells aus Experiment 16. Links: Aufnahme aus dem RapidEye2015 Datensatz. Mitte: CORINE *Land Cover*. Rechts: Modellvorhersage. Die besten Ergebnisse werden für bebaute Flächen erzielt. Ebenso gut erkennbar sind Wälder und naturnahe Flächen. Hearusfordernder ist die Identifizierung kleinerer landwirtschaftlicher Klassen (A) und die korrekte Erkennung von Feucht- und Wasserflächen (B).

Die klassenspezifischen Metriken (Abbildung 4.15) liegen für die Klassen 0, 12 (Nicht bewässertes Ackerland), 24 (Nadelwälder) und 44 (Meere und Ozeane) am höchsten. Die Klassen 15, 16, 26, 27 und 36 haben einen f1-Wert von 0. Sie werden vom Modell nicht erkannt und sind leer.

Mit dem stratifizierten *Sampling* treten in der Klassifizierung klassenspezifische Unterschiede auf. Eine starke Überschreitung der *Sampling* Schwellenwerte führt in den Klassen 42 und 43 zu einer Änderung des Ergebnisses gegenüber dem *Random Sampling*. Durch das gezielte Zuweisen der Klassen an die Test- und Validierungsdatensätze wird der Trainingsdatensatz unverhältnismäßig klein. Daraus ergeben sich schlechtere Klassengenauigkeiten: Für Klasse 42 werden 13.2% der Pixel korrekt klassifiziert (*Random Sampling* 29,3%), für die Klasse 43 0,0% (*Random Sampling* 9,0%).

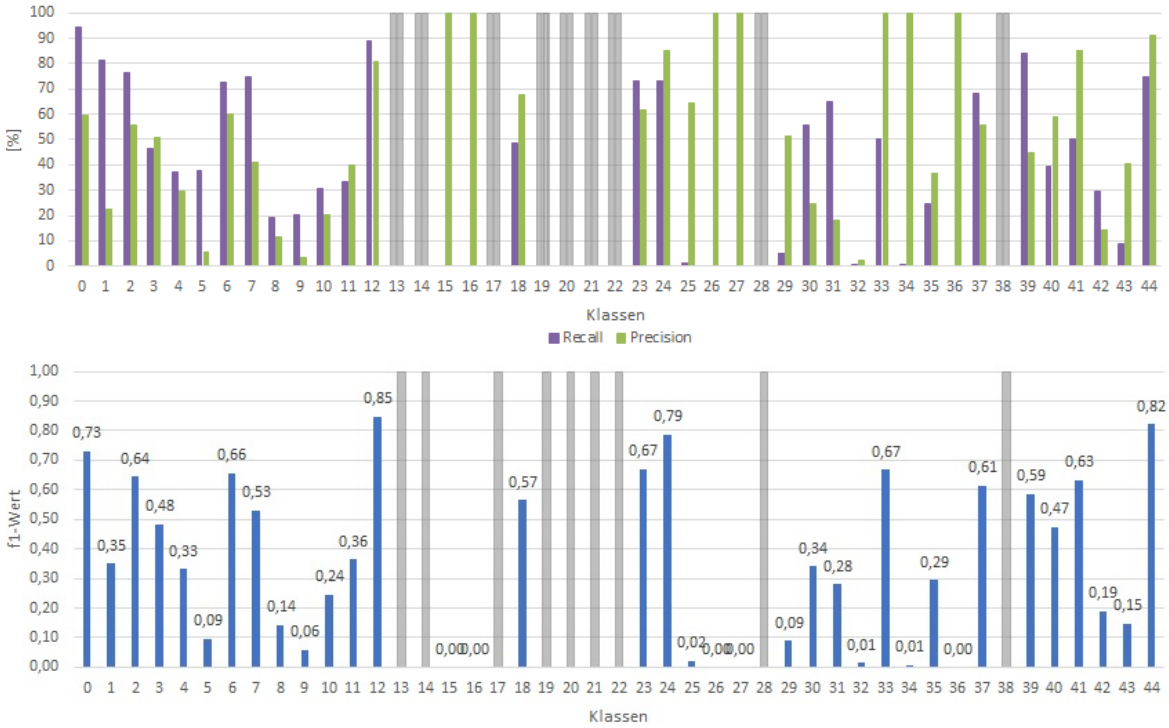

Abbildung 4.15.: Klassenspezifische Metriken für das Modell aus Experiment 16. Oben: *Recall* und *Precision*. Unten: f1-Wert. Grau dargestellte Klassen kommen im Datensatz für Deutschland nicht vor.

Die Klassifizierung der CORINE *Land Cover* liefert für die Klassen 1 (durchgängig städtische Prägung) und 12 (nicht bewässertes Ackerland) die besten Ergebnisse. Insgesamt folgen Fehlklassifizierungen dem Rahmen der fünf Überkategorien. Die kleinsten Klassen zeigen die größte Fehlerbehaftung beziehungsweise werden vom Modell nicht erlernt. Die CORINE *Land Cover* folgt dahingehend den Erkenntnissen der Landbedeckungs- und Landnutzungsklassifizierung. Die entsprechenden Ergebnisse für Experiment 17 sind im Anhang (A.3) zu finden.

5. Diskussion

Nachfolgend werden die Ergebnisse der Arbeit diskutiert. Kapitel 5.1 fasst die Erkenntnisse zum Einfluss der *Sampling* Methode auf Datensatz- und Modellqualität zusammen. In Kapitel 5.2 werden anschließend die Ergebnisse der durchgeführten Experimente analysiert und diskutiert.

5.1. Einfluss der Sampling Methode

Der Einfluss der *Sampling* Methode auf die Güte der Teildatensätze sowie mögliche Einflüsse auf die Modellleistung werden umfangreich getestet. Die *Random Sampling* Methode zeigt Schwächen in der Klassenverteilung bei auftretenden Klassenungleichgewichten mit besonders kleinen Klassen für lbakt und CLC. Durch die Einführung eines stratifizierten Samplings kann die Aufteilung in die Teildatensätze überwiegend verbessert werden. Verschlechterungen treten vereinzelt für kleine Klassen der CORINE *Land Cover* auf. Die Limitierung liegt dabei in der szenenbasierten Funktionsweise. Ein aus der Sampling Methode resultierender signifikanter Einfluss auf die Gesamtleistung des Modells lässt sich nicht feststellen. Beide *Sampling* Methoden sind praktikabel, die Anwendungsentscheidung hängt von den spezifischen Charakteristika der genutzten Landbedeckungsmodelle ab. Entscheidende Faktoren sind Klassenungleichgewichte und damit einhergehend das Auftreten besonders kleiner Klassen, die durch das *Sampling* gegebenfalls nicht erreicht werden können. In jedem Fall sollte für die Auswertung der Modellleistung die Zusammensetzung der Teildatensätze mit untersucht werden.

5.2. Klassifizierungsergebnisse

Mit allen durchgeführten Experimenten können Modelle trainiert werden, die in variierender Qualität in der Lage sind, die Eingangsdaten wiederzugeben.

Sehr gute Ergebnisse werden in der Landbedeckungsklassifizierung (Experiment 11 und 12) erzielt. Die Identifizierung übergeordnete Klassengruppen wie Wasserflächen oder Baumbestand kann umgesetzt werden. Fehlklassifizierungen erfolgen überwiegend innerhalb dieser Klassengruppen und sind auf die starke Ähnlichkeit der Klassen zurückzuführen. So bleibt beispielsweise die Unterscheidung in spezifische Wasserklassen fehlerbehaftet. Fehlklassifizierungen außerhalb der Klassengruppen treten einerseits

für Klassen auf, deren spektrale Information sehr heterogen ist, da es sich um Mischklassen handelt, wie zum Beispiel für Sumpf- und Moorflächen mit Baumbestand. Andererseits können Klassen mit zu geringer *Sample*-Anzahl, wie es bei den Klassen Brandfläche sowie Schnee und Eis der Fall ist, vom Modell nicht erlernt werden. Insgesamt sind die Ergebnisse der Landbedeckungsklassifizierung sehr vielversprechend, bedürfen zukünftig aber noch einer weiterführenden Optimierung.

Aus den Erkenntnissen der Landbedeckungsklassifizierung entsteht die Überlegung, inwiefern den auftretenden Herausforderungen entgegengewirkt werden kann. Mithilfe der Klassenaggregierung in Experiment 13 werden die Klassengruppen des Landbedeckungsmodells genutzt, um ähnliche Klassen zusammenzuführen. Die Klassenaggregierung zeigt sowohl positive als auch negative Effekte. Die Gesamtgüte des Modells hat sich gegenüber den Experimenten 11 und 12 verbessert. Durch die Zusammenfassung semantisch ähnlicher Klassen kann die Güte der Klassifizierung von Wasserflächen und Baumbeständen gesteigert werden. Keine Verbesserung ist bei den Grünland- und Strauchklassen sowie der aggregierten Klasse für Sand, Steine, Erde und Fels erkennbar. Durch das Zusammenführen von bereits in vorangegangenen Experimenten schlecht identifizierbarer Klassen wird die Streuung der Ergebnisse verstärkt. Aggregierte Klassen können zudem die Qualität anderer Klassen beeinflussen, indem die klassenspezifische spektrale Information durch die Aggregierung variabler wird: Im Vergleich zu den Experimenten 11 und 12 nimmt die Genauigkeit der kleineren Anbauklassen stark ab, die Fehlklassifizierung als Ackerland steigt an. Eine Klassenaggregierung kann die Modellqualität folglich positiv beeinflussen. Entscheidend ist eine optimale Zusammenführung geeigneter Klassen. Auch der Einfluss auf nicht von der Aggregierung betroffene Klassen muss bei der weiteren Untersuchung berücksichtigt werden. Eine weitere Prüfung möglicher Aggregierungen ist für eine optimale Anwendung nötig.

Charakteristisch für die Ergebnisse der Landnutzungsklassifizierung (Experiment 14 und 15) ist die für jede Klasse auftretende Überklassifizierung der Klasse 0. Diese lässt sich auf die Nomenklatur der Klasse und damit ihr Auftreten im Datensatz zurückführen (Abbildung 5.1). Anders als bei lbakt oder CLC enthält die Klasse 0 per Definition nicht nur *nodata* Pixel, sondern auch Flächen mit dem Merkmal "keine Nutzung erkennbar". Die spektrale Information dieser Klasse wird dadurch sehr unspezifisch.

Außerdem sind Klassen wie Straßen- und Bahnverkehr nicht konsistent im Landnutzungsmodell enthalten. Dadurch klassifiziert das Modell stellenweise bezogen auf die Eingabedaten falsch, die Ausgabe stimmt aber präziser mit den Fernerkundungsdaten überein. Ebenfalls problematisch in ihrer Erkennung und Abgrenzung ist die Klasse Wasser. Gewässer sind nur innerhalb genutzter Flächen als solche im Datensatz enhalten. Große Teile von Wasserkörpern sind dadurch als Klasse 0 deklariert (Abbildung 5.1). Eine weitere Herausforderung in der Landnutzungsklassifizierung besteht im hohen Anteil von Mischklassen. Die am besten klassizierbaren Klassen Wohnen, Intensive Landwirtschaft und Forstwirtschaft weisen eine verglichen mit Klassen wie Öffentlichkeit oder Deponie eindeutigere spektrale Information auf, die im Netzwerk leichter

Abbildung 5.1.: Landbedeckungs- und Landnutzungsinformation für eine exemplarische Rapideye-Szene. Links: Szene aus dem RapidEye2015-Datensatz. Mitte: Landbedeckungsinformation. Rechts: Landnutzungsinformation. Während die Landbedeckungsinformation flächendeckend vorliegt, zeigt die Landnutzungsinformation große Bereiche, die mit der Information "keine Nutzung erkennbar"(in weiß) deklariert sind.

trainiert werden kann. Die Heterogenität der spektralen Information von Landnutzungsklassen führt dadurch zu einer verminderten Modellleistung. Insgesamt betrachtet können viele Landnutzungsklassen nicht durch die spektrale Information allein definiert werden. Für eine Klassifizierung sind weitere Merkmale notwendig, wie beispielsweise Nachbarschaftsbeziehungen, die räumliche Lage und die zeitliche Dynamik der betreffenden Flächen.

Die Experimente 16 und 17 mit CLC-Daten spiegeln in Teilen die vorangegangenen Resultate der Landbedeckung und Landnutzung wider. Trotz der sehr großen Klassenanzahl liefern diese Modelle vergleichbar gute Ergebnisse. Wie bei der Landbedeckung, können Klassengruppen semantisch ähnlicher Klassen gut identifiziert werden. Die spezifische Zuweisung ist auch hier fehlerbehaftet. Ebenso werden die kleinsten Klassen des Datensatzes aufgrund uneindeutiger spektraler Informationen beziehungsweise einem Mangel an zur Verfügung stehenden Trainingsflächen vom Netzwerk nicht erlernt. Die Landnutzungsklassen zeigen die selben Herausforderungen wie die der Landnutzungsbedeckung. Aufgrund des heterogenen Charakters von Mischklassen kommt es zu Fehlklassifizierungen.

In allen Experimenten werden semantisch starke Klassen wie Bebauung, Ackerland, Baumbestand und Wasserflächen am besten klassifiziert. Ebenso tritt während aller Experimente eine Überschätzung der Bebauung auf. In den Landbedeckungsmodellen treten in Siedlungsgebieten hauptsächlich die Klassen Bebauung und Grasland mit Bäumen auf. Zweitere kann dabei Bebauungselemente wie Wohngebäude enthalten, die durch ihre vergleichsweise geringe Größe im Vergleich zur umliegenden Grünfläche (in Form von Gärten, Parks, etc.) auf Basis des Mehrheitsprinzips dieser Klasse zugefügt werden.

Insgesamt können die Fehlklassifikationen der Experimente auf drei Faktoren zurückgeführt werden: Fehlklassifikationen treten auf, wenn Trainingsdaten nicht ausreichend spektrale Information für eine Klasse liefern, die Klassen untereinander zu hohe Gemeinsamkeiten aufweisen oder einer Klasse zugehörige Flächen aus unterschiedlichen Landbedeckungen zusammengesetzt sind und es sich folglich um Mischflächen handelt.

Die daraus resultierenden Herausforderungen bestehen hinsichtlich dreier Hauptaspekte: Erstens werden in allen drei Landschaftsmodellen Mischklassen nur schlecht klassifiziert. Beispiele sind die Moor- und Sumpfflächen bei lbakt und CLC sowie Sport- und Freizeitflächen bei lnakt; die spektrale Information der Klasse ist in diesen Fällen heterogen und weist Überschneidungen mit anderen Klassen auf. Dadurch kann keine präzise Zuordnung erfolgen. Zweitens treten ähnliche Effekte bei der Klassifizierung semantisch ähnlicher Klassen auf. Das Modell ist dazu in der Lage, Wasserflächen eindeutig von Land zu unterscheiden. Für eine genaue Zuordnung in eine der sechs Wasserklassen der Landbedeckung reichen die spektralen Informationen der Wasserflächen oft nicht aus. Das führt besonders bei kleineren Klassen wie Wattflächen oder Lagunen zu einer Fehlklassifizierung in eine semantisch ähnliche, größere Klasse wie beispielsweise offenes Meer. Drittens zeigt sich, dass die Klassifizierung von Landnutzung zu weit mehr Fehlklassifizierungen führt als die von Landbedeckung. Hauptursache dafür ist die spektral stark heterogene Klasse 0.

6. Zusammenfassung

In der vorliegenden Arbeit werden anhand einer Reihe von Experimenten Methoden der künstlichen Intelligenz systematisch evaluiert. Das Ziel besteht in der Einschätzung, inwieweit sich die Erzeugung von Landbedeckung und Landnutzung aus fernerkundungsgestützten Bilddaten für Deutschland durch eine *Deep Learning* Anwendung mit semantischer Segmentierung automatisieren lässt.

Die Haupterkenntnisse der Arbeit sind:

1. Auf Grundlage der gewählten Architektur und den evaluierten Methoden zu Datenaufbereitung und Hyperparametertuning kann ein Modell trainiert werden, das Klassengruppen mit semantisch ähnlichen Klassen der Landbedeckung (Bebauung, Landwirtschaft, Grünland, Baumbestand, Wasserflächen) identifiziert.
2. Die spezifische Identifizierung semantisch ähnlicher Klassen mit vergleichbarer spektraler Information führt zu höheren Fehlklassifizierungen.
3. Landbedeckung ist aufgrund ihrer auf Klassenebene spezifischeren spektralen Information besser klassifizierbar als Landnutzung. Zweitere weist durch eine spektrale Heterogenität der Klassen größere Fehlerraten auf.
4. Eine zunehmende Klassenanzahl kann die klassenspezifische Modellgüte verschlechtern; gleichzeitig führt eine Klassenaggregierung nicht zwingend zu besseren Ergebnissen. Hier ist eine genaue Evaluierung geeigneter Aggregierungsschemata nötig.

Die Ergebnisse der Experimente hinsichtlich der Klassifizierung von Landbedeckung mittels automatisierter Methoden aus dem *Deep Learning* Bereich sind äußerst vielversprechend. Die Klassifizierung von Landnutzung stellt mit der gewählten Methode eine größere Herausforderung dar. Ein weiterführende Evaluierung bezüglich geeigneter Methoden zur verbesserten Identifikation von Mischklassen bietet das Potenzial, die Modellleistung weiter zu steigern. Außerdem offen bleibt die Erarbeitung eines praktikablen Ansatzes zur Klassenaggregierung, um die Qualität des Modells weiter zu steigern. Zukünftige Experimente mit dem erarbeiteten Setup prüfen zudem die Übertragbarkeit des Modells, um Anwendungsziele, wie beispielsweise Änderungsanalysen in Landbedeckung und Landnutzung erfüllen zu können.

A. Anhang

A.1. Einfluss der Samplingmethode auf die Qualität der Teildatensätze

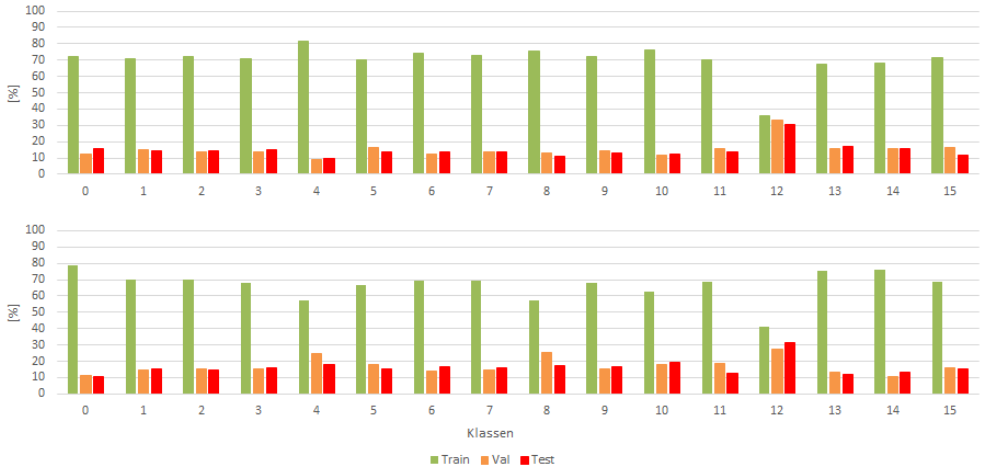

Abbildung A.1.: Ergebnisse des *Samplings* für lnakt. Pixelanteile [%] für Trainings- Validierungs- und Testdatensatz. Oben: *Random Sampling*. Unten: Stratifiziertes *Sampling*.

Abbildung A.2.: Ergebnisse des *Samplings* für CLC. Pixelanteile [%] für Trainings- Validierungs- und Testdatensatz. Oben: *Random Sampling*. Unten: Stratifiziertes *Sampling*.

A.2. Landbedeckungsmodell (lbakt) - Stratifiziertes Sampling

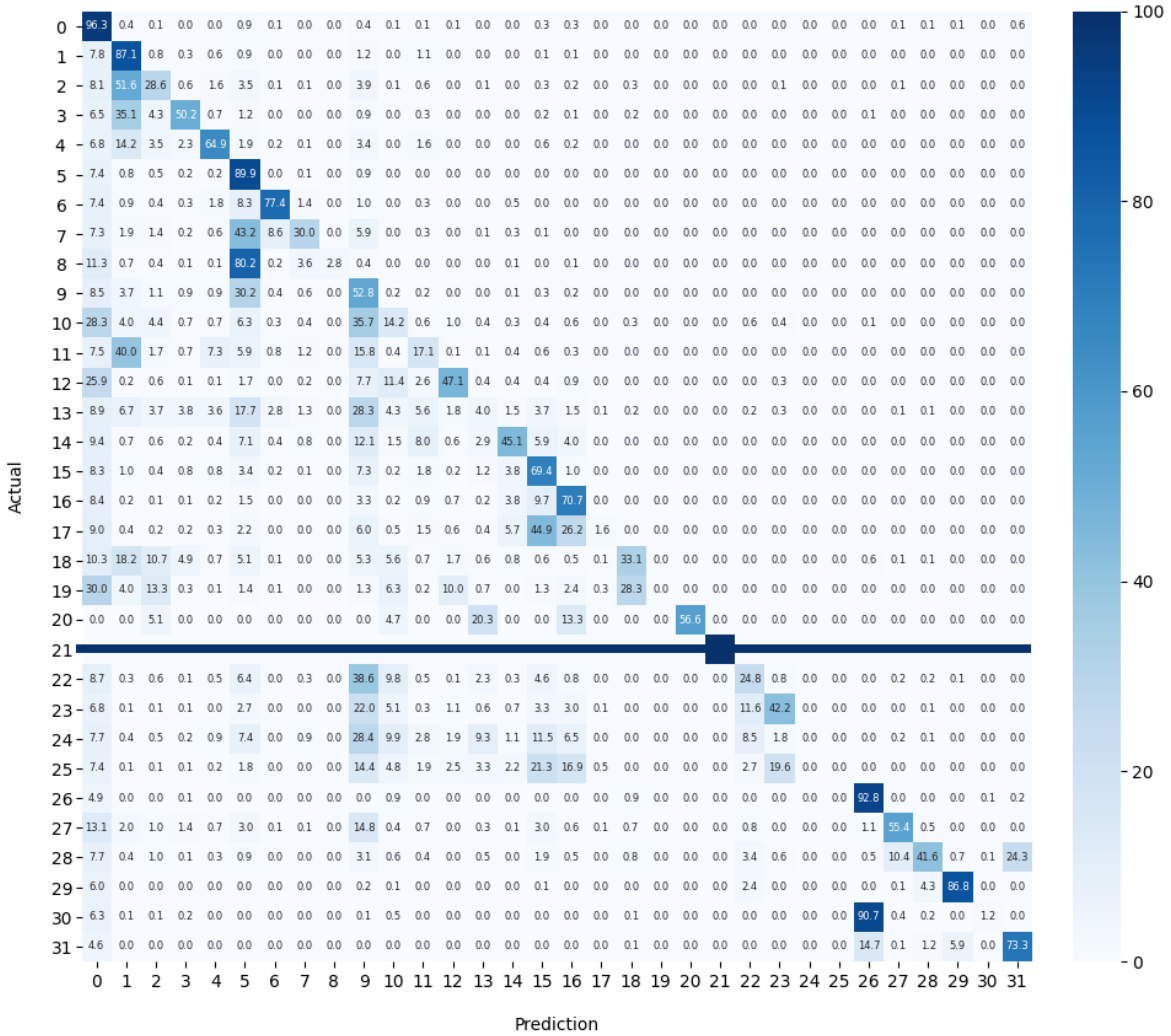

Abbildung A.3.: Konfusionsmatrix für Experiment 12. Klasse 21 (blau durchgestrichen) ist nicht im Testdatensatz enthalten.

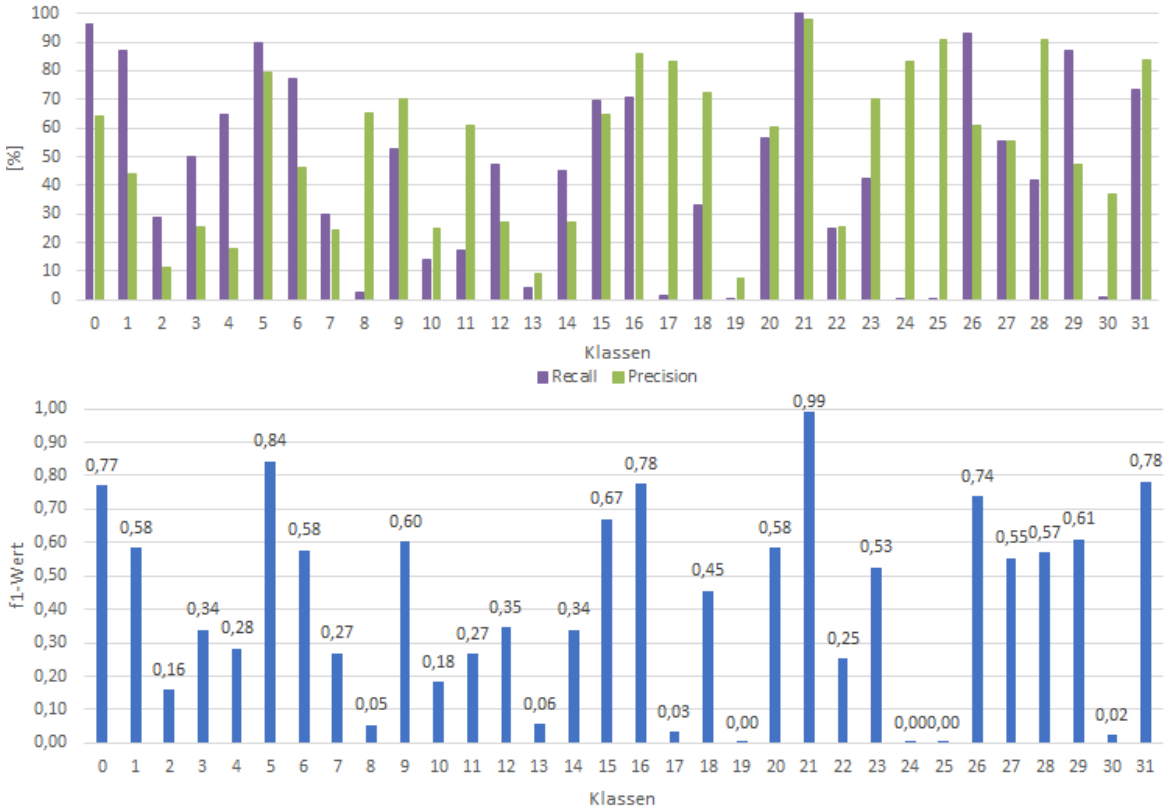

Abbildung A.4.: Klassenspezifische Metriken für das Modell aus Experiment 12. Oben: *Recall* und *Precision*. Unten: f1-Wert. Die Ergebnisse der Klasse 21 werden nicht berücksichtigt, da die Klasse nicht im Testdatensatz vertreten war.

A.3. Landnutzungsmodell (Inakt) - Stratifiziertes Sampling

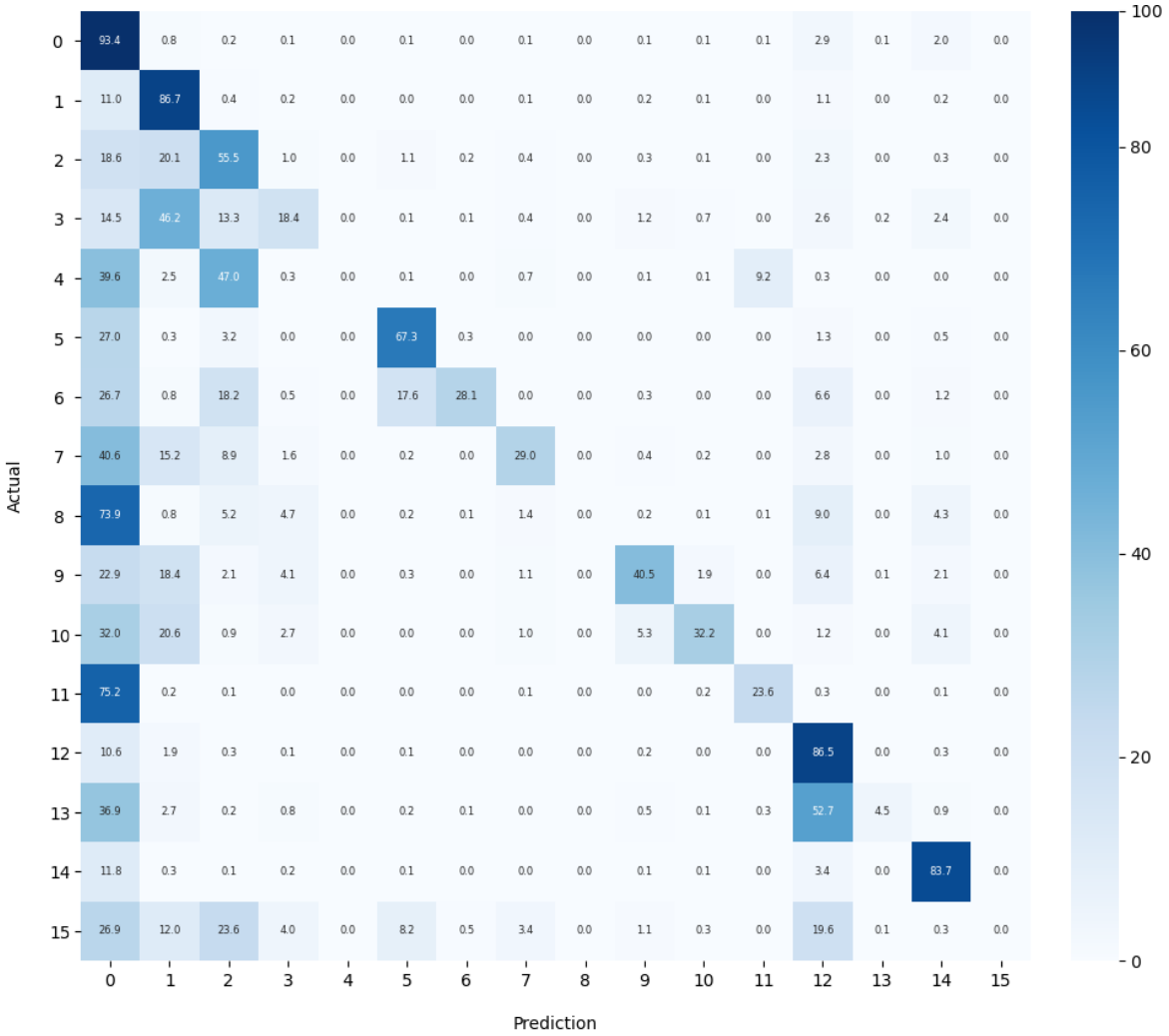

Abbildung A.5.: Konfusionsmatrix für Experiment 15.

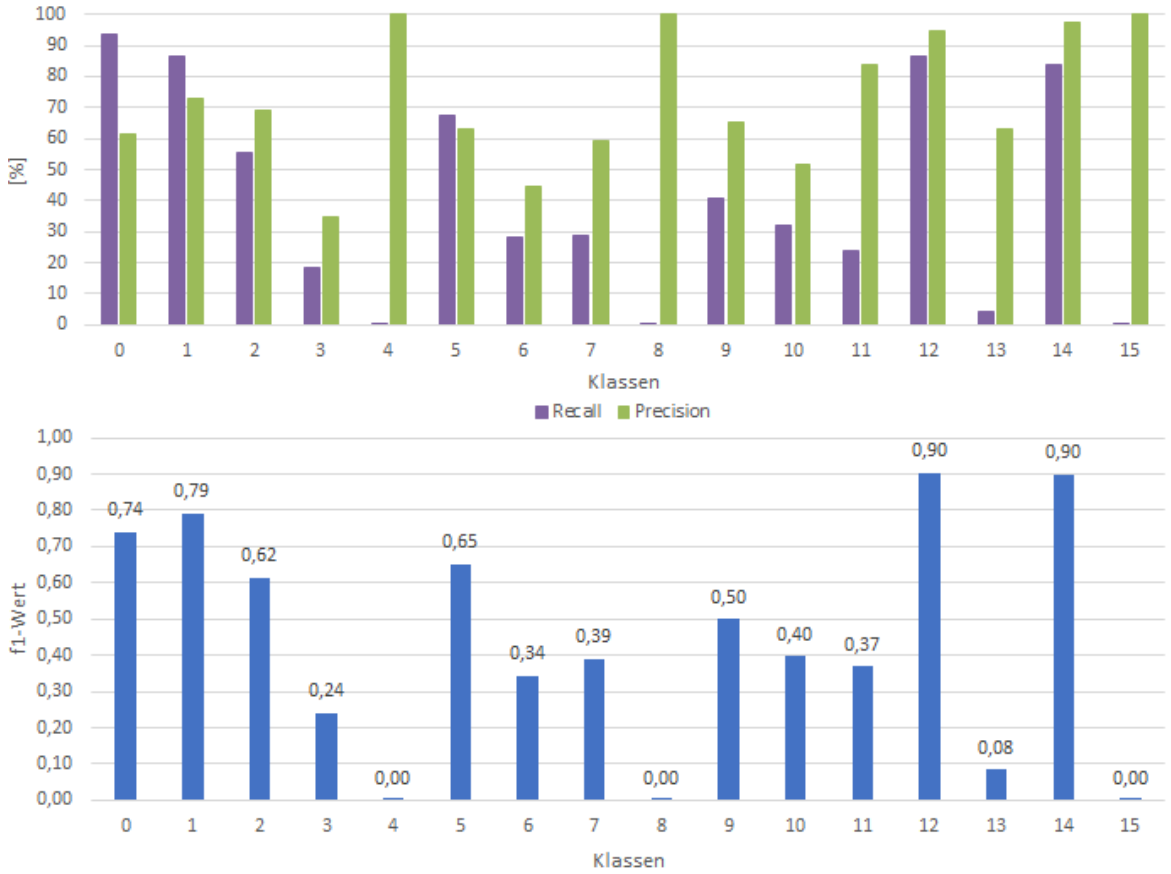

Abbildung A.6.: Klassenspezifische Metriken für das Modell aus Experiment 15. Oben: *Recall* und *Precision*. Unten: f1-Wert.

A.4. CORINE Land Cover (CLC) - Stratifiziertes Sampling

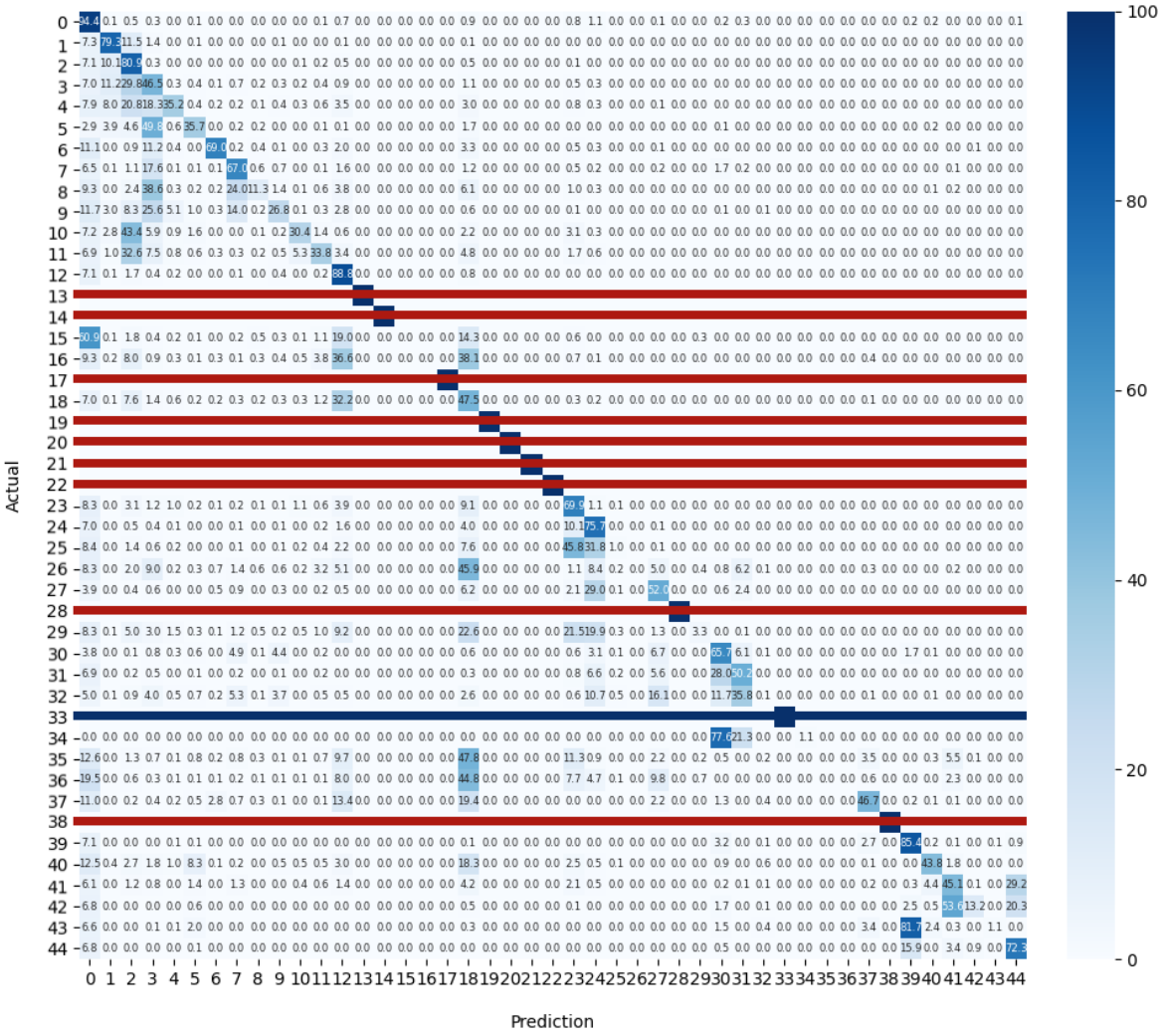

Abbildung A.7.: Konfusionsmatrix für Experiment 17. Die rot durchgestrichenen Klassen kommen im Datensatz für Deutschland nicht vor. Klasse 33 (blau durchgestrichen) ist nicht im Testdatensatz enthalten.

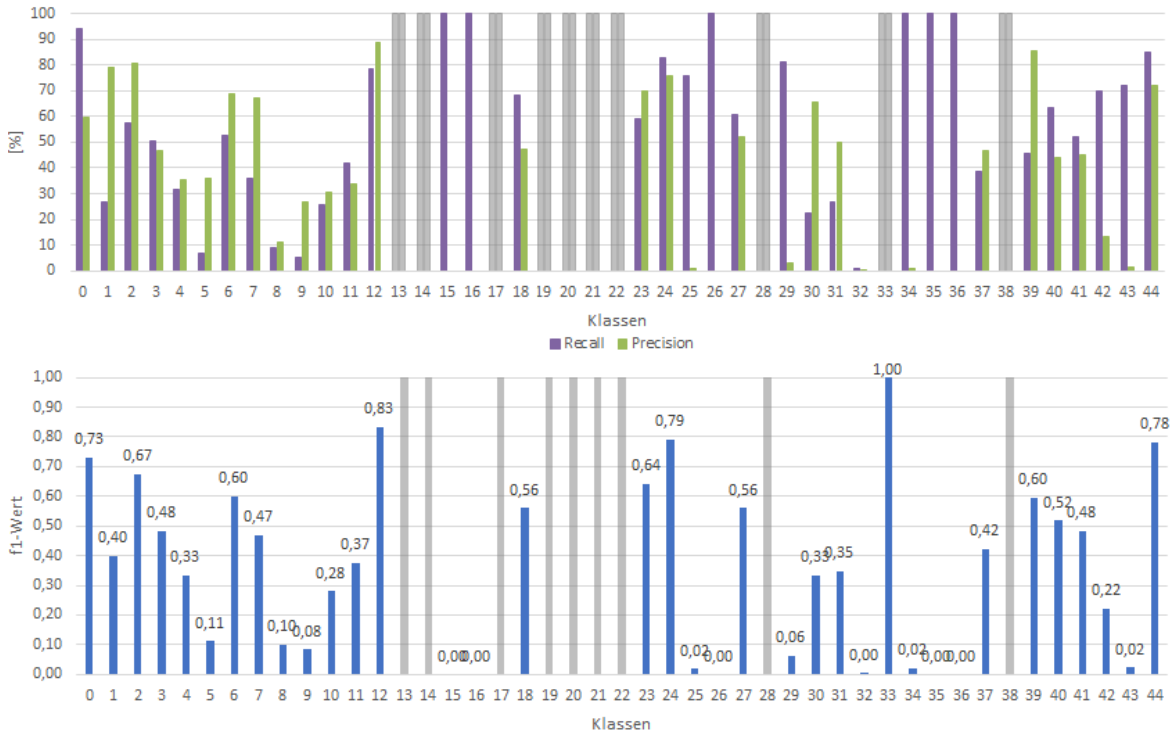

Abbildung A.8.: Klassenspezifische Metriken für das Modell aus Experiment 17. Oben: *Recall* und *Precision*. Unten: f1-Wert. Grau dargestellte Klassen kommen im Datensatz für Deutschland nicht vor. Die Ergebnisse der Klasse 33 werden nicht berücksichtigt, da die Klasse nicht im Testdatensatz vertreten war.

Literatur

Alp, G. und E. Sertel (2022). „Deep Learning Based Patch-Wise Land Cover Land Use Classification: A New Small Benchmark Sentinel-2 Image Dataset“. In: *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, S. 3179–3182. DOI: 10.1109/IGARSS46834.2022.9883715.

Bruce, P., A. Bruce und P. Gedeck (2021). *Praktische Statistik für Data Scientists*. O'Reilly. ISBN: 978-3-96009-153-0.

Bruzzone, L. und B. Demir (2014). „A Review of Modern Approaches to Classification of Remote Sensing Data“. In: *Land Use and Land Cover Mapping in Europe. Practices Trends*. Hrsg. von I. Manakos und M. Braun. Bd. 18. Remote Sensing and Digital Image Processing. Springer, S. 127–143. DOI: 10.1007/978-94-007-7969-3_9.

Bundesamt für Kartographie und Geodäsie (BKG) (2021). *Dokumentation. Digitales Landbedeckungsmodell für Deutschland LBM-DE2015*. URL: https://sg.geodatenzentrum.de/web_public/gdz/dokumentation/deu/lbm-de2015.pdf.

Buslaev, A., V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin und A. A. Kalinin (2020). „Albumentations: Fast and Flexible Image Augmentations“. In: *Information* 11(2). DOI: 10.3390/info11020125.

Büttner, G. (2014). „CORINE Land Cover and Land Cover Change Products“. In: *Land Use and Land Cover Mapping in Europe. Practices & Trends*. Hrsg. von I. Manakos und M. Braun. Bd. 18. Remote Sensing and Digital Image Processing. Springer, S. 55–74. DOI: 10.1007/978-94-007-7969-3_5.

Camps-Valls, G., X. X. Zhu, D. Tuia und M. Reichstein (2021). „Introduction“. In: *Deep Learning for the Earth Sciences. A Comprehensive Approach to Remote Sensing, Climate Science and Geosciences*. Hrsg. von G. Camps-Valls, X. X. Zhu, D. Tuia und M. Reichstein. Wiley, S. 1–11. ISBN: 978-1-1196-4614-3.

Castelluccio, M., G. Poggi, C. Sansone und L. Verdoliva (2015). *Land Use Classification in Remote Sensing Images by Convolutional Neural Networks*. DOI: 10.48550/ARXIV.1508.00092.

Copernicus (2022a). *About Copernicus*. URL: https://www.copernicus.eu/en/about-copernicus (besucht am 18. 10. 2022).

– (2022b). *CORINE Land Cover*. URL: https://land.copernicus.eu/pan-european/corine-land-cover (besucht am 18. 10. 2022).

DeFries, R. S. und J. R. Townshend (1999). „Global land cover characterization from satellite data: from research to operational implementation?“ In: *Global Ecology and Biogeography* 8, S. 367–379.

Deutsche Raumfahrtagentur (2008). *RapidEye*. URL: https://www.dlr.de/rd/desktopdefault.aspx/tabid-2440/3586_read-5336 (besucht am 11. 10. 2022).

Goodfellow, I., Y. Bengio und A. Courville (2016). *Deep Learning*. MIT Press. URL: http://www.deeplearningbook.org.

Goyal, P., P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia und K. He (2017). *Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour*. DOI: 10.48550/ARXIV.1706.02677.

Hansen, M. C. und T. R. Loveland (2012). „A review of large area monitoring of land cover change using Landsat data“. In: *Remote Sensing of Environment* 122. Landsat Legacy Special Issue, S. 66–74. DOI: https://doi.org/10.1016/j.rse.2011.08.024.

He, K., X. Zhang, S. Ren und J. Sun (2016). „Deep Residual Learning for Image Recognition“. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, S. 770–778. DOI: 10.1109/CVPR.2016.90.

Hovenbitzer, M., F. Emig, C. Wende, S. Arnold, M. Bock und S. Feigenspan (2014). „Digital Land Cover Model for Germany - DLM-DE“. In: *Land Use and Land Cover Mapping in Europe. Practices Trends*. Hrsg. von I. Manakos und M. Braun. Bd. 18. Remote Sensing and Digital Image Processing. Springer, S. 255–282. DOI: 10.1007/978-94-007-7969-3_16.

Hu, F., G.-S. Xia, J. Hu und L. Zhang (2015). „Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 7(11), S. 14680–114707. DOI: 10.3390/rs71114680.

Ioffe, S. und C. Szegedy (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. DOI: 10.48550/ARXIV.1502.03167.

Karra, K., C. Kontgis, Z. Statman-Weil, J. C. Mazzariello, M. Mathis und S. P. Brumby (2021). „Global land use / land cover with Sentinel 2 and deep learning“. In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, S. 4704–4707. DOI: 10.1109/IGARSS47720.2021.9553499.

Ketkar, N. und J. Moolayil (2021). *Deep Learning with Python. Learn Best Practices of Deep Learning Models with PyTorch*. Apress. ISBN: 978-1-4842-5363-2.

Kingma, D. P. und J. Ba (2014). *Adam: A Method for Stochastic Optimization*. DOI: 10.48550/ARXIV.1412.6980.

Kirillov, A., R. Girshick, K. He und P. Dollár (2019). *Panoptic Feature Pyramid Networks*. DOI: 10.48550/ARXIV.1901.02446.

Krizhevsy, A., I. Sutskever und G. E. Hinton (2012). „Imagenet classification with deep convolutional neural networks“. In: *Advances in Neural Information Processing Systems*, S. 1097–1105. DOI: 10.1145/3065386.

Kussul, N., M. Lavreniuk, S. Skakun und A. Shelestov (2017). „Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data“. In: *IEEE Geoscience and Remote Sensing Letters* 14(5), S. 778–782. DOI: 10.1109/LGRS.2017.2681128.

Landis, J. R. und G. G. Koch (1977). „The Measurement of Observer Agreement for Categorical Data“. In: *Biometrics* 33(1), S. 159–174. DOI: 10.2307/2529310.

Li, R., C. Duan, S. Zheng, C. Zhang und P. M. Atkinson (2020). *MACU-Net for Semantic Segmentation of Fine-Resolution Remotely Sensed Images*. DOI: 10.48550/ARXIV.2007.13083.

Lin, T.-Y., P. Dollár, R. Girshick, K. He, B. Hariharan und S. Belongie (2016). *Feature Pyramid Networks for Object Detection*. DOI: 10.48550/ARXIV.1612.03144.

Long, J., E. Shelhamer und T. Darrell (2014). *Fully Convolutional Networks for Semantic Segmentation*. DOI: 10.48550/ARXIV.1411.4038.

Mansour, D., S.-A. Souiah und M. El Amin Larabi (2021). „Built-Up Area Extraction Through Deep Learning“. In: *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, S. 6805–6808. DOI: 10.1109/IGARSS47720.2021.9554694.

Marmanis, D., K. Schindler, J. D. Wegner, S. Galliani, M. Datcu und U. Stilla (2018). „Classification with an edge: Improving semantic segmentation with boundary detection“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 135, S. 158–172. DOI: 10.1016/j.isprsjprs.2017.11.009.

McGlinchy, J., B. Johnson, B. Muller, M. Joseph und J. Diaz (2019). „Application of UNet Fully Convolutional Neural Network to Impervious Surface Segmentation in Urban Environment from High Resolution Satellite Imagery“. In: *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, S. 3915–3918. DOI: 10.1109/IGARSS.2019.8900453.

MIT Technology Review (2022). *10 Breakthrough Technologies 2013*. URL: https://www.technologyreview.com/10-breakthrough-technologies/2013/ (besucht am 03. 11. 2022).

Müller, R., S. Kornblith und G. Hinton (2019). *When Does Label Smoothing Help?* DOI: 10.48550/ARXIV.1906.02629.

Nakamura, K., B. Derbel, K.-J. Won und B.-W. Hong (2021). „Learning-Rate Annealing Methods for Deep Neural Networks“. In: *Electronics* 10. DOI: doi.org/10.3390/electronics10162029.

Pal, K. K. und K. S. Sudeep (2016). „Preprocessing for image classification by convolutional neural networks“. In: *2016 IEEE International Conference on Recent Trends in Electronics, Information Communication Technology (RTEICT)*, S. 1778–1781. DOI: 10.1109/RTEICT.2016.7808140.

Pedrayes, O., D. Lema, F. D. Garcia, R. Usamentiaga und A. Alonso (2021). „Evaluation of Semantic Segmentation Methods for Land Use with Spectral Imaging Using Sentinel-2 and PNOA Imagery“. In: *Remote Sensing* 13, S. 2292. DOI: 10.3390/rs13122292.

Pointer, I. (2021). *PyTorch für Deep Learning. Anwendungen für Bild-, Ton- und Textdaten entwickeln und deployen*. O'Reilly. ISBN: 978-3-96009-134-9.

Reichstein, M., G. Camps-Valls, B. Stevens, M. Jung, J. Denzler, N. Carvalhais und M. Prabhat (2019). „Deep learning and process understanding for data-driven Earth system science“. In: *Nature* 566, S. 195. DOI: 10.1038/s41586-019-0912-1.

Richards, J. A. (2022). *Remote Sensing Digital Image Analysis*. 6. Aufl. Springer. ISBN: 978-3-030-82326-9.

Ronneberger, O., P. Fischer und T. Brox (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. DOI: 10.48550/ARXIV.1505.04597.

Shane, J. (2021). *Künstliche Intelligenz. Wie sie funktioniert und wann sie scheitert*. O'Reilly. ISBN: 978-3-96009-160-8.

Simonyan, K. und A. Zisserman (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. DOI: 10.48550/ARXIV.1409.1556.

Stanford University (2022). *CS231n Convolutional Neural Networks for Visual Recognition*. URL: https://cs231n.github.io/convolutional-networks/#fc (besucht am 18. 10. 2022).

Stanford Vision Lab (2020). *ImageNet*. URL: https://www.image-net.org/challenges/LSVRC/index.php (besucht am 15. 10. 2022).

Stark, T., M. Wurm, X. X. Zhu und H. Taubenböck (2020). „Satellite-Based Mapping of Urban Poverty With Transfer Learned Slum Morphologies". In: *IEEE Journal of Selected Topics in Applied Earth Observations an Remote Sensing* 13, S. 5251–5263. DOI: 10.1109/JSTARS.2020.3018862.

Stoian, A., V. Poulain, J. Inglada, V. Poughon und D. Derksen (2019). „Land Cover Maps Production with High Resolution Satellite Image Time Series and Convolutional Neural Networks: Adaptations and Limits for Operational Systems". In: *Remote Sensing* 11(17). DOI: 10.3390/rs11171986.

Storie, C. D. und C. J. Henry (2018). „Deep Learning Neural Networks for Land Use Land Cover Mapping". In: *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, S. 3445–3448. DOI: 10.1109/IGARSS.2018.8518619.

Sugimoto, Y. und M. Aono (2021). „Semantic Segmentation based on Extended MobileNet with FPN". In: *2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*. DOI: 10.1109/ICAICTA53211.2021.9640256.

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke und A. Rabinovich (2014). *Going Deeper with Convolutions*. DOI: 10.48550/ARXIV.1409.4842.

The European Space Agency (ESA) (2022a). *RapidEye*. URL: https://earth.esa.int/eogateway/missions/rapideye (besucht am 17. 10. 2022).

– (2022b). *RapidEye*. URL: https://www.eoportal.org/satellite-missions/rapideye#downlink-analysis-for-rapideye-data (besucht am 05. 11. 2022).

Tuia, D., D. Marcos, K. Schindler und B. Le Saux (2021). „Deep Learning-based Semantic Segmentation in Remote Sensing". In: *Deep Learning for the Earth Sciences. A Comprehensive Approach to Remote Sensing, Climate Science and Geosciences*. Hrsg. von G. Camps-Valls, X. X. Zhu, D. Tuia und M. Reichstein. Wiley, S. 46–66. ISBN: 978-1-1196-4614-3.

Voelsen, M., M. Teimouri, F. Rottensteiner und C. Heipke (2022). „Investigating 2D and 3D Convolutions for Multitemporal Land Cover Classification using Remote Sensing Images". In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 3, S. 271–279. DOI: 10.5194/isprs-annals-V-3-2022-271-2022.

Wurm, M., A. Droin, T. Stark, C. Geiß, W. Sulzer und H. Taubenböck (2021). „Deep Learning-Based Generation of Building Stock Data from Remote Sensing for Urban Heat Demand Modeling“. In: *ISPRS International Journal of Geo-Information* 10. DOI: 10.3390/ijgi10010023.

Wurm, M., T. Stark, X. X. Zhu, M. Weigand und H. Taubenböck (2019). „Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks“. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 150, S. 59–69. DOI: 10.1016/j.isprsjprs.2019.02.006.

Zhang, X., L. Du, S. Tan, F. Wu, L. Zhu, Y. Zeng und B. Wu (2021). „Land Use and Land Cover Mapping Using RapidEye Imagery Based on a Novel Band Attention Deep Learning Method in the Three Gorges Reservoir Area“. In: *Remote Sensing* 13(6). DOI: 10.3390/rs13061225.

Zhu, X. X., D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu und F. Fraundorfer (2017). „Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources“. In: *IEEE Geoscience and Remote Sensing Magazine* 5(4), S. 8–36. DOI: 10.1109/MGRS.2017.2762307.

Eidesstattliche Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat.

Alle Ausführungen der Arbeit, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Augsburg, 18. November 2022

(Julia Maria Holzner)