Analyse von Deep-Learning-Verfahren zur semantischen Segmentierung von photogrammetrischen Punktwolken aus Luftbildern

MARKUS HÜLSEN[1,2]

Zusammenfassung: Die semantische Segmentierung von Punktwolken stellt eine herausfordernde Aufgabe dar, die in dieser Arbeit untersucht wird. Der Fokus liegt auf der Anwendung von Deep-Learning-Verfahren zur semantischen Segmentierung von photogrammetrischen Punktwolken aus Luftbildern. Es wird ein Arbeitsablauf zur Erstellung eines solchen Modells vorgestellt, welcher die Annotation einer Punktwolke mittels verschiedener Verfahren des Machine Learnings sowie die Anwendung von PointNet++ umfasst. Mit dem gewählten Ansatz wird auf einem Validierungsdatensatz eine Genauigkeit von 96,5% erreicht. Zusätzlich wird jedem Punkt die Wahrscheinlichkeit der Klassenzugehörigkeit zugeordnet. Das Modell bietet eine geeignete Grundlage für weitere Punktwolkenverarbeitungen, wie z. B. die Ableitung von digitalen Geländemodellen.

1 Einleitung

Punktwolken haben sich im Fachbereich der Geodäsie zu einem der Standardprodukte etabliert. Die Erhebung von Punktwolken ist ausgereift und weitgehend erforscht. Die effiziente Weiterverarbeitung ist jedoch Gegenstand aktueller Forschung. Insbesondere die semantische Segmentierung, also die punktweise Klassifikation, gewinnt in verschiedenen Bereichen zunehmend an Bedeutung. Anwendung findet die semantische Segmentierung beispielweise bei der Ableitung von Gebäudemodellen, der Berechnung von Digitalen Geländemodellen (DGM) oder bei Veränderungsanalysen. Während die manuelle Segmentierung äußerst zeitaufwendig ist und geometriebasierte Algorithmen schnell an ihre Grenzen stoßen, erzielen verschiedene Deep-Learning-Verfahren vielversprechende Fortschritte. Forschungsbedarf besteht jedoch bei der semantischen Segmentierung von topographischen Punktwolken, wie sie durch Airborne Laserscanning oder durch *Dense Image Matching* (DIM) von Luftbildern erhoben werden. Insbesondere das Potenzial der photogrammetrischen DIM-Punktwolken wird bisher nur selten ausgeschöpft, obwohl diese Punktwolken mit geringem Mehraufwand während der Orthophoto-Produktion erzeugt werden können.

Die Generierung eines trainierten Deep-Learning-Modells zur semantischen Segmentierung umfasst im Allgemeinen drei Hauptaufgaben: (1) die Beschaffung einer geeigneten Ground Truth für Training und Validierung, (2) die Auswahl und Anwendung einer Deep-Learning-Architektur und (3) die Bestimmung einer optimalen Trainingskonfiguration. Im Rahmen dieser Arbeit wird eine Möglichkeit zur Annotation einer geeigneten Ground Truth aufgezeigt. Des Weiteren werden die Schritte zur Anwendung einer Deep-Learning-Architektur näher beschrieben. Nach einer kurzen Erläuterung von Deep-Learning-Verfahren zur semantischen

[1] Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN), Regionaldirektion Otterndorf, Am Großen Specken 7, D-21762 Otterndorf,
E-Mail: markus.huelsen@lgln.niedersachsen.de
[2] Jade Hochschule Oldenburg, Ofener Straße 16/19, D-26121 Oldenburg

 DOI: 10.24407/KXP:1885272189

Segmentierung von Punktwolken werden nachfolgend die angestrebte Methodik und die gewählte Implementierung zur praxisnahen Bearbeitung der Aufgabenstellung vorgestellt. Anschließend werden die Ergebnisse des Trainings evaluiert und ein Fazit gezogen.

2 (Semantische) Segmentierung von Punktwolken

Bei der (semantischen) Segmentierung wird die Punktwolke in Regionen mit ähnlichen Eigenschaften aufgeteilt und einer (ggf. semantischen) Klasse zugeordnet. Dazu werden verschiedene Verfahren des maschinellen Lernens oder andere Algorithmen eingesetzt, darunter lernbasierte und nicht-lernbasierte Ansätze. Zu den nicht-lernbasierten Verfahren zählen klassische Machine-Learning-Verfahren wie *K-Means* (MACQUEEN 1967) oder *DBSCAN* (ESTER et al. 1996), aber auch speziell für Punktwolken entwickelte Verfahren wie der *Cloth-Simulation-Filter* (CSF) (ZHANG et al. 2016), der zwischen Boden- und Nicht-Bodenpunkten unterscheiden kann.
In den letzten Jahren wurden verschiedene Deep-Learning-Architekturen zur semantischen Segmentierung von Punktwolken entwickelt. Eine kategorische Einordnung bestehender Deep-Learning-Ansätze für die semantische Segmentierung von Punktwolken wird beispielsweise in HE et al. (2023) oder GUO et al. (2021) vorgestellt. Erste Architekturen überführten die Punktwolken in strukturierte Gitter, wie 3D-Voxelgitter oder zweidimensionale Multi-View-Repräsentationen. Erst mit der Entwicklung von *PointNet* (QI et al. 2016) wurde eine direkte Verarbeitung ohne vorherige Diskretisierung möglich. Allerdings ist PointNet nicht in der Lage, lokale Strukturen zu erfassen, was die Fähigkeit einschränkt, feine Strukturen zu erkennen und komplexe Szenen zu generalisieren. Um den Problemen von PointNet entgegenzuwirken wurde *PointNet++* (QI et al. 2017) vorgestellt. Es handelt sich hierbei um ein hierarchisches neuronales Netz, das PointNet rekursiv auf überlappende Partitionen von Eingangspunkten anwendet. Die Grundidee von PointNet++ basiert auf *Convolutional Neural Networks* (CNNs), bei denen Merkmale schrittweise in immer größeren Maßstäben entlang einer Hierarchie mit mehreren Auflösungen erfasst werden. PointNet++ teilt die Punktwolke auf Basis der Distanzmetrik in sich überlappende lokale Regionen auf und extrahiert mittels PointNet lokale Merkmale, die feine geometrische Strukturen erfassen. Diese Merkmale werden weiter zu größeren Einheiten gruppiert und prozessiert, um Merkmale auf höherer Ebene zu erzeugen.

3 Methodik

Um ein Modell zur semantischen Segmentierung von Punktwolken zu generieren, müssen zunächst geeignete Ground-Truth-Daten für das Modelltraining generiert und eine Deep-Learning-Architektur ausgewählt werden. Die in der Arbeit angestrebte Methodik ist im Folgenden erläutert.

3.1 Generierung von Trainingsdaten

Insbesondere für Deep-Learning-Modelle ist eine große Menge an Trainingsdaten notwendig, um das Modell effizient zu trainieren. Dabei ist nicht nur die Datenmenge entscheidend, sondern auch die Homogenität der Klassenzuordnung sowie die Vielfalt an Objekten innerhalb einer Klasse. Die Trainingsdaten sollten die verschiedenen Variationen und Randfälle enthalten, die in der realen Anwendung auftreten können. Die korrekte und konsistente Klassenzuordnung ist entscheidend, da das Modell auf diesen Daten basiert. Gleichzeitig führt eine zu

starke Heterogenität innerhalb einer semantischen Klasse zu einer geringeren Performance. Die Objekte innerhalb einer semantischen Klasse sollten daher möglichst ähnliche Eigenschaften aufweisen, um eine optimale Modellperformance zu erreichen.
Für die Annotation von Punktwolken stehen drei Vorgehensweisen zur Verfügung: manuell, semi-automatisch oder durch Generierung synthetischer Punktwolken. Bei der manuellen Annotation werden die Punkte manuell selektiert und einer Klasse zugeordnet. Abhängig von der Datenmenge ist diese Methode sehr zeitaufwendig, selbst wenn spezielle Software verwendet wird. Bei der semiautomatischen Annotation werden unüberwachte Lernverfahren eingesetzt, um Punkte mit ähnlichen Eigenschaften zu gruppieren. Die resultierenden Cluster müssen dann manuell der richtigen semantischen Klasse zugeordnet werden. Dies kann den Zeitaufwand gegenüber der manuellen Annotation verringern, erfordert jedoch die Auswahl geeigneter Algorithmen und Merkmalskombinationen. Von synthetisch generierten Punktwolken wird gesprochen, wenn die Erhebung von Punktwolken in einem 3D-Modell simuliert wird, häufig auf Basis von Laserscanning. Eine Pipeline zur synthetischen Generierung von annotierten photogrammetrischen Punktwolken aus Luftbildern wurde beispielsweise von CHEN et al. (2022) vorgestellt. Diese Verfahren befinden sich weitgehend in der Entwicklung und erfordern entsprechende 3D-Modelle.

3.2 Wahl einer Deep-Learning-Architektur

Viele punktbasierte Deep-Learning-Architekturen erzielen zufriedenstellende Ergebnisse für Indoor-Datensätze. Für topographische Punktwolken, wie sie mittels Airborne Laserscanning (ALS) erhoben werden, besteht bisher ein Forschungsrückstand (WIDYANINGRUM et al. 2021:2). Dies gilt insbesondere für Dense Image Matching (DIM) Punktwolken aus Luftbildern. Es existieren jedoch einige Publikationen, die unterschiedliche Architekturen zur semantischen Segmentierung von ALS-Punktwolken verwenden, die näherungsweise ähnliche Eigenschaften wie DIM-Punktwolken aufweisen. WIDYANINGRUM et al. (2021) adaptieren das DGCNN (*Dynamic Graph Convolutional Neural Network*) (WANG et al. 2019) für die semantische Segmentierung von ALS-Punktwolken und erreichen eine Gesamtgenauigkeit von 93,3%. WINIWARTER & MANDLBURGER (2019) untersuchten die Eignung von PointNet++ für die punktweise Klassifikation von ALS-Punktwolken und erreichten eine Gesamtgenauigkeit von 95,8% für urbane Gebiete im Voralberg-Datensatz. Des Weiteren evaluierten KADA & KURAMIN (2021) die Anwendung von PointNet++ und *KPConv* (THOMAS et al. 2019) auf ALS-Punktwolken und erreichten Gesamtgenauigkeiten von 93,1% mit PointNet++ und 95,5% mit KPConv.
In dieser Arbeit wird PointNet++ verwendet, um ein Modell für die semantische Segmentierung von Punktwolken aus photogrammetrischen Luftbildern zu erstellen. Obwohl PointNet++ mittlerweile in vielen Experimenten von aktuelleren Netzwerken übertroffen wird, bleibt das Modell relevant, da sein Mechanismus zur Merkmalsextraktion und seine Architektur häufig als Baustein für andere punktwolkenbasierte Netzwerke dienen. Darüber hinaus zeichnet sich PointNet++ aufgrund seiner simplen Architektur durch eine effiziente Klassifikation aus, was es besonders für große Datenmengen geeignet macht.

4 Implementierung

Mit der angestrebten Methodik im Hintergrund kann nun mit der Umsetzung begonnen werden. Für das Training des Modells wird zunächst eine umfangreiche Punktwolke annotiert, wobei

ein semi-automatischer Ansatz aus verschiedenen Machine-Learning-Verfahren entwickelt wird. Anschließend wird die Punktwolke in Trainings- und Validierungsdaten aufgeteilt, ein geeignetes Batching durchgeführt und eine optimale Trainingskonfiguration bestimmt. Schließlich wird die gewählte Deep-Learning-Architektur angewendet und der Trainingsprozess gestartet. Der gesamte Quellcode für die Annotation der Ground-Truth-Punktwolke und die Anwendung der PointNet++ Architektur ist in einem *GitHub Repository* unter https://github.com/mhuelsen/PointNet2ForDIM-PC verfügbar.

4.1 Annotation der Trainingsdaten

Das Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN) hält intern ein Testgebiet vor, welches u. a. DIM-Punktwolken aus dem Jahr 2022 beinhaltet, die mit der Software *SURE* von *nFrames/Esri* aus Luftbildern prozessiert wurden. Neben den räumlichen Koordinaten (X, Y, Z) und der Farbinformation (R, G, B bzw. $Intensity,\ Hue,\ Saturation$; IHS) sind für jeden Punkt weitere photogrammetrische Attribute verfügbar, wie die Punktpräzision ($\sigma \cdot 10.000/GSD$) und die Anzahl der Stereomodelle (NFRAMES o.J.). Die Punktwolken umfassen ein umfangreiches, heterogenes Gebiet von 12 km² mit 66,7 Mio. Punkten. Im Rahmen einer noch in Arbeit befindlichen Dissertation wurde die Punktwolke auf Basis der Geometrie semantisch in folgende Klassen segmentiert: *Gelände*, *Nicht-Gelände*, *Gebäude* und *Brücken*. Die Geländeklasse wurde anhand eines Schwellwertes des vertikalen Abstands zu einem DGM aus 2016 bestimmt, während die Gebäude- und Brückenklasse durch Verschneidung mit den entsprechenden zweidimensionalen ALKIS-Objekten entstand. Alle übrigen Punkte wurden der Klasse *Nicht-Gelände* zugeordnet. Eine visuelle Darstellung der erzielten Segmentierung ist in Abb. 1 gegeben.

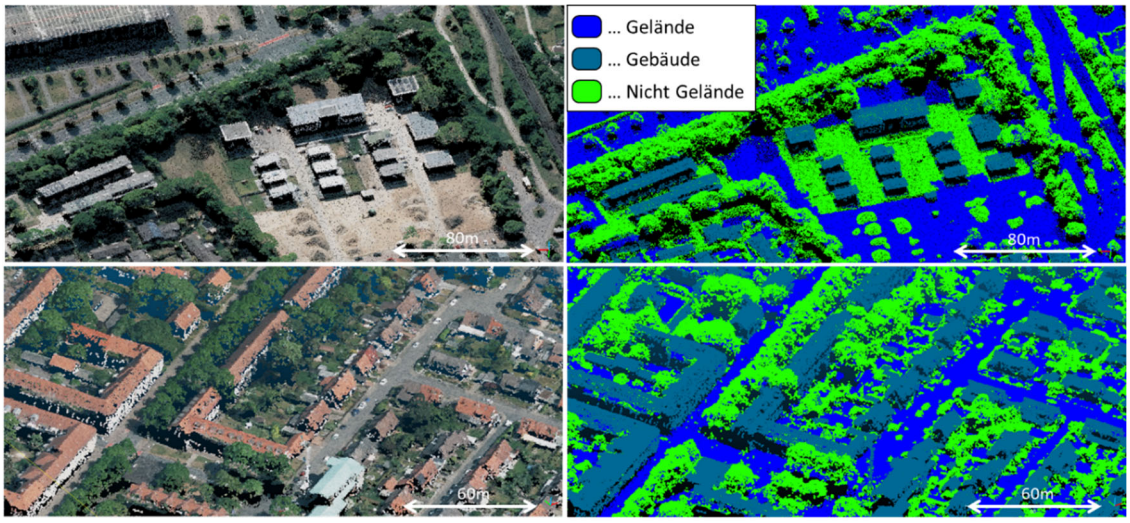

Abb. 1: Klassifikation auf Basis geometrischer Eigenschaften. Links: RGB; Rechts: Klassifikation

Obwohl diese Annotation einen guten Ausgangspunkt darstellt, ist eine Optimierung erforderlich. Zum einen ist die Qualität der Klassenzuordnung unzureichend und zum anderen enthält die Klasse *Nicht-Gelände* verschiedene Objekte mit sehr unterschiedlichen Eigenschaften, z. B. Fahrzeuge und Vegetation. Da die Performance des Deep-Learning-Modells stark von der Homogenität der Klassenzuordnung innerhalb einer Klasse abhängt, wird diese Klasse in die

Klassen „*Vegetation*“ und „*Künstliche Objekte*“ (engl. *humanmade*) aufgeteilt. Zur Durchführung dieser Optimierung wird ein semi-automatischer Workflow entwickelt, der sowohl unüberwachte als auch überwachte Lernverfahren beinhaltet und in Abb. 2 visualisiert ist.

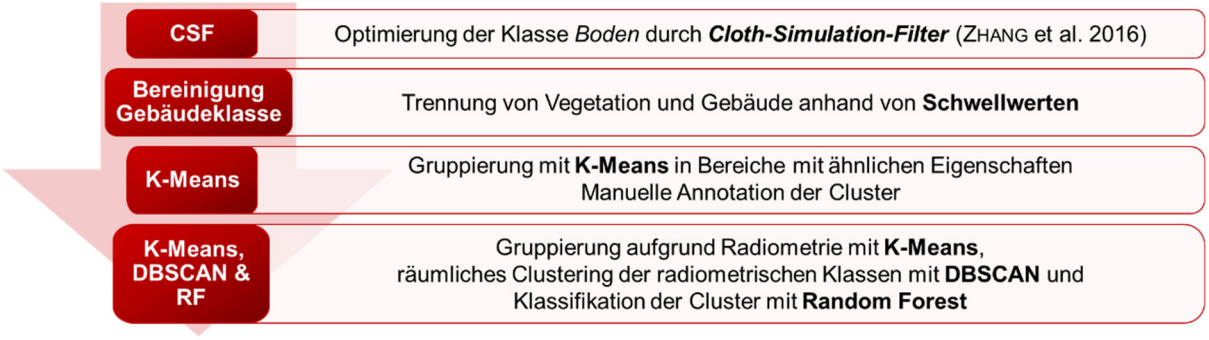

Abb. 2: Semi-automatischer Workflow zur Optimierung der Annotation

In einem ersten Schritt wird die Bodenklasse mit dem Cloth-Simulation-Filter (ZHANG et al. 2016) optimiert. Um Vegetation, die über Gebäude ragt, aus der Klasse *Gebäude* zu separieren, werden Schwellwerte für verschiedene Merkmale in Kombination definiert, sodass diese der Vegetationsklasse zugeordnet werden können. Anschließend wird das K-Means-Clustering mehrfach mit verschiedenen Merkmalskombinationen angewendet, um die Klasse *Nicht-Gelände* in *Vegetation* und *Künstliche Objekte* aufzuteilen. Mit diesem Vorgehen werden bereits ca. 98% der Punktwolke einer semantischen Klasse zugeordnet. Um die verbleibenden Punkte abschließend zu klassifizieren, wird eine Kombination aus Machine-Learning-Verfahren eingesetzt: Zunächst wird mittels K-Means auf Basis der Radiometrie geclustert und anschließend mittels DBSCAN anhand der Geometrie. Auf diese Weise entstehen räumliche Cluster aus Punkten die nahe beieinander liegen („Punkt-Inseln“) mit ähnlichen radiometrischen Eigenschaften. Diese werden abschließend mit einem Random-Forest-Klassifikator klassifiziert. Die verbleibendenden Rauschpunkte des DBSCAN-Algorithmus werden der häufigsten Klasse in der Nachbarschaft zugeordnet.

Die Auswahl und Kombination geeigneter Algorithmen und entsprechender Parameter ist abhängig vom Inhalt der Punktwolke und der Charakteristik. Durch die beschriebene Vorgehensweise kann mit der hier vorliegenden Punktwolke eine Ground Truth erzeugt werden, die für ein Training hinreichend korrekt erscheint. Die hier verwendeten Algorithmen und insbesondere die gewählten Parameter sind jedoch nicht allgemeingültig, sondern an die vorliegende Punktwolke angepasst. Wie Abb. 3 zeigt, beinhaltet die resultierende Ground-Truth-Punkwolke verschiedenste Topographien, sodass eine gute Allgemeingültigkeit zu erwarten ist. Die Daten beinhalten sowohl ländliche Bereiche mit Wäldern und Wiesen als auch urbane Bereiche mit verschiedensten Baustilen. Zudem sind auch Besonderheiten, wie Baustellen mit Kränen, ein Schienen- und Straßenbahnnetz und Autobahnen enthalten. Der generierte Datensatz enthält diverse heterogene Objekte und sollte eine geeignete Grundlage für das Training darstellen.

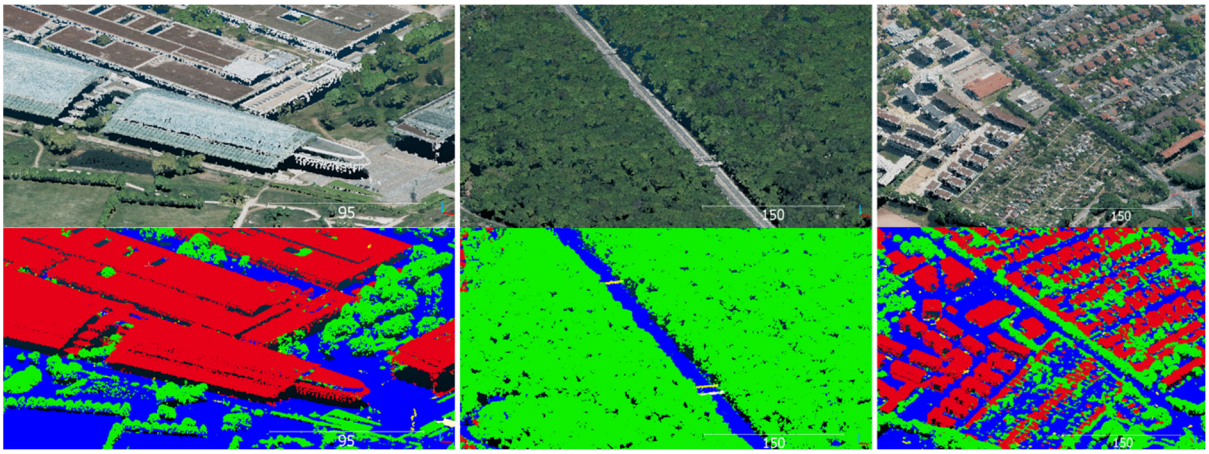

Abb. 3: Beispielhafte Ausschnitte des Ground-Truth-Datensatzes. Oben RGB; Unten Klassifikation

In Tab. 1 sind die absoluten und relativen Anteile der Punkte pro Klasse dargestellt. Hier ist zu erkennen, dass die Klassen *Brücken* und *künstliche Objekte* stark unterrepräsentiert sind, während nahezu die Hälfte der Punkte auf die Klasse Boden entfällt.

Tab. 1: Absoluter und relativer Anteil an Punkten pro Klasse im gesamten Ground-Truth-Datensatz

	Boden	**Gebäude**	**Vegetation**	**Künstliche Objekte**	**Brücken**
Punktanzahl	31.463.945	12.384.375	22.498.089	330.857	101.342
Relativer Anteil	47,1%	18,5%	33,7%	0,5%	0,2%

4.2 Aufteilung der Trainingsdaten & Batching der Punktwolke

Der Ground-Truth-Datensatz ist in zwölf Patches von je 1 km² unterteilt. Um die Leistungsfähigkeit des Modells beurteilen zu können, wird eines der Patches nur für die Validierung verwendet, so dass 11 km² für das Training verbleiben. Bei der Auswahl der Validierungspunktwolke ist darauf zu achten, dass sie sich nicht mit den Trainingsdaten überlappt und alle Objektklassen enthält. Außerdem sollten die Validierungsdaten eine ähnliche Topographie wie die Trainingsdaten aufweisen – Sonderfälle, wie spezielle Gebäudeformen, sollten nicht in den Daten enthalten sein.

Da es aufgrund des Speicherbedarfs nicht möglich ist, große Punktwolken in einem Prozessierungs-schritt zu verarbeiten, ist eine weitere Unterteilung der Punktwolke notwendig, was als Batching bezeichnet wird. WINIWARTER & MANDLBURGER (2019) schlagen vor, kreisförmig überlappende Batches zu verwenden, indem für bestimmte Gitterpunkte eine feste Anzahl nächster Punkte (K) gewählt wird. Durch die resultierende Kreisform kann Isotropie gewährleistet werden und durch die Überlappung wird derselbe Punkt mehrfach in unterschiedlichen Kontexten klassifiziert, wodurch eine Mittelung der Klassenwahrscheinlichkeiten möglich wird. In dieser Arbeit wird eine Punktanzahl von $K = 100.000$ gewählt, sodass 4.287 Batches für das Training und 400 Batches für die Validierung entstehen. Eine beispielhafte Darstellung eines resultierenden Batches ist in **Fehler! Verweisquelle konnte nicht gefunden werden.** dargestellt. Mit dem gewählten Parameter wird jeder Punkt in mindestens drei Batches abgebildet; 44% der Punkte werden in sieben Batches abgebildet.

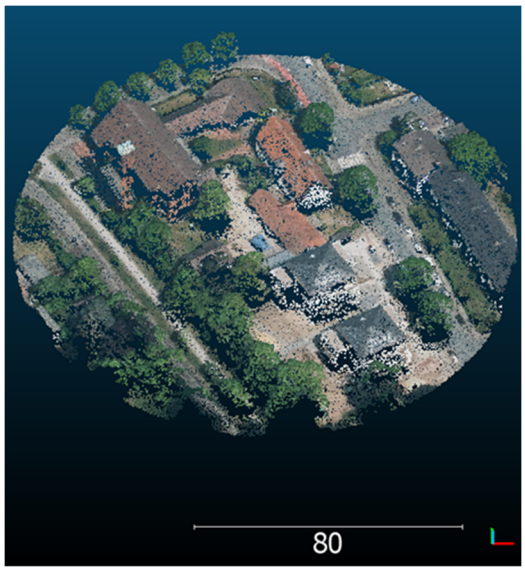

Abb. 4: Beispiel-Batch mit 100.000 Punkten

4.3 Anwendung des Deep-Learning-Modells PointNet++

Die Anwendung von PointNet++ ist im Python-Framework *PyTorch* umgesetzt. In der Grundkonfiguration werden drei *Set-Abstraction (SA)-Layer*, sowie drei *Feature-Propagation (FP)-Layer* verwendet. Abb. 5 zeigt den Aufbau der Architektur und die verwendeten Parameter. Die Auswahl der Parameter erfolgt hierbei in Anlehnung an WINIWARTER & MANDLBURGER (2019).

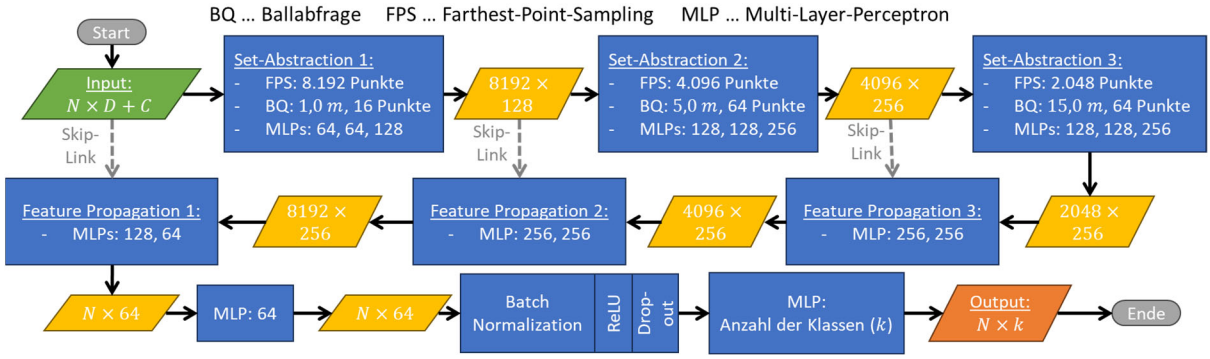

Abb. 5: Prozessablauf des PointNet++ Modells inklusive der gewählten Parameter

Im Folgenden wird der Prozessablauf von PointNet++ kurz zusammengefasst. Zunächst werden drei SA-Layer durchgeführt. In einem SA-Layer wird mittels *Farthest-Point-Sampling* (FPS) eine bestimmte Anzahl von Punkten anhand der Distanz selektiert. Für jeden dieser Punkte wird eine Ballabfrage (engl. *Ball Query*, BQ) durchgeführt, die alle Punkte innerhalb eines festen Radius gruppiert. Die resultierenden Punktnachbarschaften werden anschließend an *Multi-Layer-Perceptrons* (MLPs) mit geteilten Gewichten und einer bestimmten Anzahl an Filtern übergeben, um lokale Regionsmerkmale zu dekodieren. Diese werden als 2D-Faltung mit einer Kernelgröße von eins implementiert. Das Ergebnis eines SA-Layers ist eine Matrix, welche für jeden Punkt des FPS Merkmale auf einer höheren Ebene beinhaltet. Diese werden wiederum an einen weiteren SA-Layer übergeben um Merkmale in einer größeren Nachbarschaft zu berechnen. Gefolgt von den drei SA-Layern, werden drei FP-Layer angewendet, um die Merkmale zurück auf die Punkte der niedrigeren Ebenen zu propagieren. Im FP-Layer werden die Merkmale mittels einer distanzbasierten Interpolation der drei nächstgelegenen Punkte

auf die Punkte des vorhergehenden Layers interpoliert und mit den Merkmalen des entsprechenden SA-Layers verkettet („*Skip-Link*“). Die resultierende Matrix wird dann an eine vorgegebene Anzahl von MLPs übergeben, welche ebenfalls als 2D-Faltung mit einer vorgegebenen Anzahl von Filtern implementiert sind. Nachdem durch Feature-Propagation für alle Punkte der Eingangspunktwolke Merkmale berechnet sind, wird ein weiteres MLP mit 64 Filtern durchgeführt, gefolgt von einer *Batch-Normalisierung*, der *ReLU*-Aktivierungsfunktion und einem *Dropout*-Layer mit 50% Dropout. Im finalen Output-MLP werden schließlich die Klassenwahrscheinlichkeiten für jeden Punkt zurückgegeben.

5 Finales Training & Evaluierung

Zur Bestimmung der optimalen Trainingskonfiguration werden verschiedene Tests durchgeführt. Als Verlustfunktion hat sich der *Focal Loss* (LIN et al. 2017) mit der Wahl von $\gamma = 4$ als vorteilhaft erwiesen. Gleichzeitig wird eine Lernrate von $0{,}001$ verwendet, wobei diese ab der zehnten Epoche durch einen *Learning-Rate-Scheduler* exponentiell reduziert wird. Unter verschiedenen Dataset-Augmentation-Methoden hat sich nur die Rotation um die Z-Achse als sinnvoll erwiesen. Darüber hinaus konnte in Experimenten gezeigt werden, dass die Verwendung der zusätzlichen photogrammetrischen Informationen (Anzahl der Stereomodelle und Punktpräzision) zu keiner Leistungssteigerung führt. Für das Modelltraining werden daher nur die Geometrie (XYZ-Koordinaten) und die Radiometrie (IHS-Werte) verwendet. Bei der Rückführung der einzelnen Batches zu einer Gesamtpunktwolke werden die Klassenwahrscheinlichkeiten jedes Punktes gemittelt. Mit der beschriebenen Konfiguration wird eine Validierungsgenauigkeit von 96,5% erreicht. Abb. 6 zeigt beispielhaft den Vergleich zwischen der Ground Truth und der vorhergesagten Klassifikation der Validierungspunktwolke, sowie die Wahrscheinlichkeiten der Klassenzugehörigkeit.

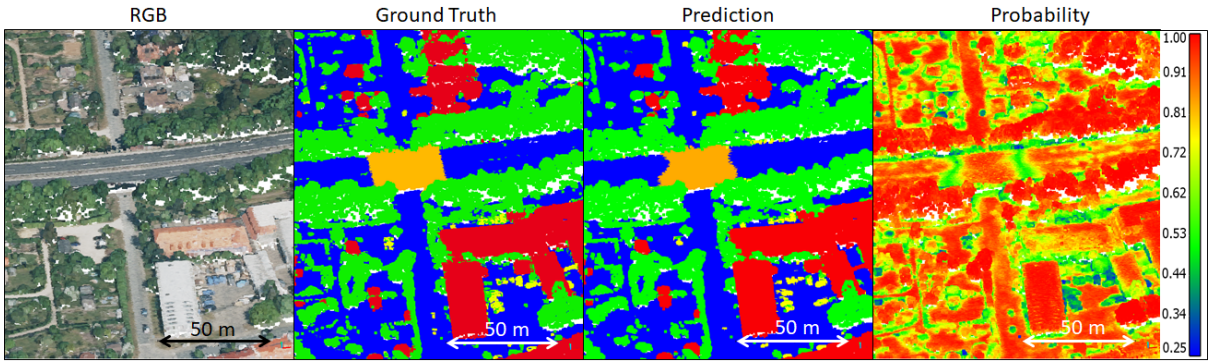

Abb. 6: Beispielhafte Visualisierung einer Klassifikation mit zugehöriger Wahrscheinlichkeit ($0 = 0\%$, $1.00 = 100\%$)

Zunächst zeigt die Abbildung, dass rein visuell eine gute Klassifikation erzielt wird – vom Modell falsch klassifizierte Punkte sind auch für den Menschen schwer zuzuordnen. Im Vergleich zeigen sich nur wenige Unterschiede zwischen Ground Truth und Prädiktion. Teilweise zeigt die Prädiktion Ungenauigkeiten in der Ground Truth auf. Lediglich bei der Brückenklasse wirkt der Übergang zwischen Straße und Brücke abgerundet. Es zeigt sich auch, dass für die Klasse *Künstliche Objekte* generell geringere Wahrscheinlichkeiten erzielt werden.
In Abb. 7 sind die Validierungsgenauigkeiten in einer Konfusionsmatrix sowie die ROC-Kurven (Receiver Operating Characteristics) der einzelnen Klassen dargestellt. Mit Ausnahme der

Klasse *Künstliche Objekte* werden für alle Klassen Genauigkeiten von > 90% erreicht. Die höchste Genauigkeit wird von der Klasse Boden mit 98,6% erreicht, gefolgt von Gebäuden mit 95,9% und Vegetation mit 93,9%. Trotz der starken Unterrepräsentation in den Trainingsdaten wird für die Klasse *Brücken* eine Genauigkeit von 90,5% erreicht. Auch die ROC-Kurve zeigt für diese Klasse einen nahezu optimalen Verlauf. Die geringste Genauigkeit erzielt die Klasse *Künstliche Objekte* mit 38,1% und auch die ROC-Kurve zeigt für diese Klasse einen weniger optimalen Verlauf.

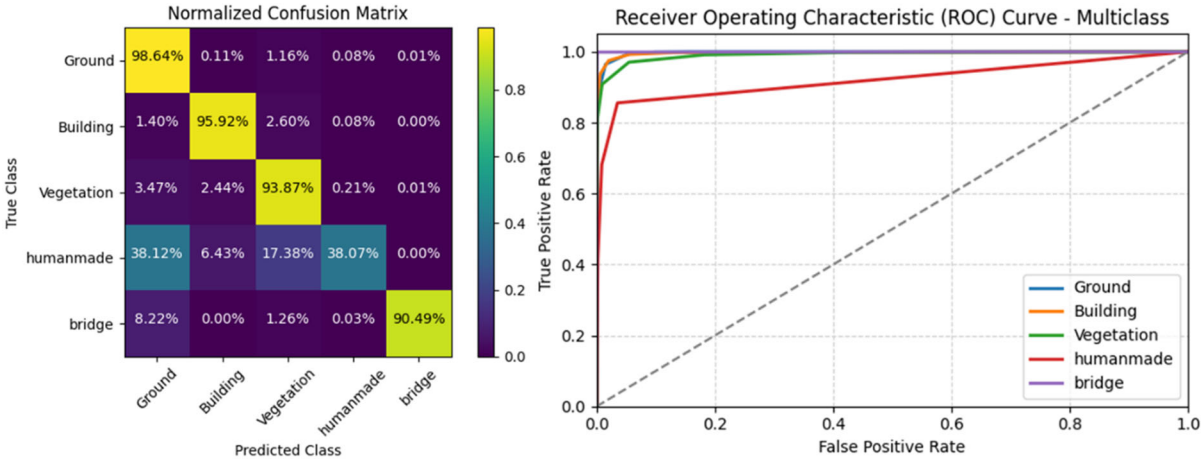

Abb. 7: Konfusionsmatrix und ROC-Kurven des finalen Trainings

Die mangelhafte Performance bei der Klasse *Künstliche Objekte* kann auf verschiedene Ursachen zurückgeführt werden: Die Klasse ist in den Trainingsdaten deutlich unterrepräsentiert, Objekte innerhalb der Klasse sind kleinräumig und sehr heterogen und die Klassenzuordnung in den Trainingsdaten weist zudem ein hohes Rauschen auf. Die Konfusionsmatrix in Abb. 7 zeigt, dass 38% der Klasse *Künstliche Objekte* fälschlicherweise als Bodenpunkte klassifiziert werden. Abb. 8 zeigt, dass vor allem Fahrzeuge als Bodenpunkte klassifiziert werden.

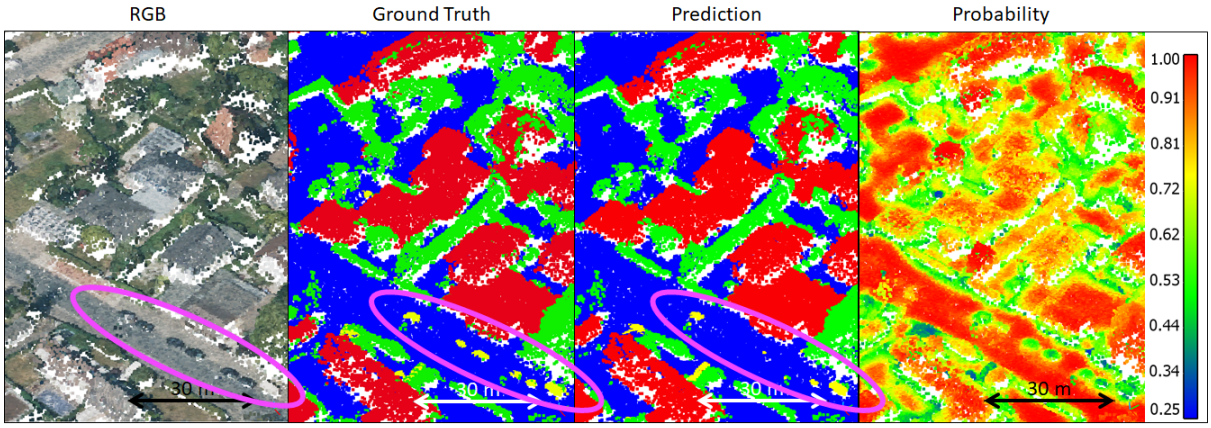

Abb. 8: Fehlerhafte Klassifikation von Fahrzeugen als Boden

Bei der Betrachtung der Wahrscheinlichkeiten in Abb. 8 wird ersichtlich, dass den tatsächlichen Fahrzeugen nur eine relativ geringe Wahrscheinlichkeit zugeordnet wird – das Modell scheint sich hier unsicher zu sein. Gleichzeitig zeigt die Abbildung, dass die True-Positiv-Rate der Bodenklasse weiter gesteigert werden kann, indem eine höhere Wahrscheinlichkeit als Schwellwert verwendet wird.

6 Fazit

Zusammenfassend wird in dieser Arbeit der gesamte Workflow zur Erstellung eines Deep-Learning-Modells zur semantischen Segmentierung von Punktwolken durchlaufen. Zunächst wird mittels verschiedener unüberwachter Verfahren ein umfangreicher Ground-Truth-Datensatz generiert, welcher eine elementare Grundlage für das Training des Modells darstellt. Durch die Kombination verschiedener Machine-Learning-Verfahren konnte ein zufriedenstellender Kompromiss zwischen Qualität und Zeitaufwand erzielt werden. Der auf diese Weise erzeugte Datensatz wird anschließend zum Training und zur Validierung einer PointNet++ Architektur verwendet. Durch verschiedene Experimente wird die optimale Trainingskonfiguration ermittelt, mit der dann eine Validierungsgenauigkeit von 96,5% erreicht wird, was im Bereich des aktuellen Forschungsstandes liegt. Insbesondere der Focal Loss mit $\gamma = 4$ hat sich als geeignete Verlustfunktion herausgestellt. Weiterhin konnte gezeigt werden, dass die zusätzlichen photogrammetrischen Informationen (Punktpräzision und Anzahl der Stereomodelle) hier keinen Mehrwert für das Deep-Learning-Modell bieten. Mit Ausnahme der Klasse *Künstliche Objekte* werden für alle Klassen Genauigkeiten von $> 90\%$ erreicht. Die relativ geringe Genauigkeit dieser Klasse ist vor allem auf die Heterogenität der Objekte innerhalb der Klasse zurückzuführen. Zudem sind Objekte hier deutlich kleinräumiger und die Klassenzuordnung in den Ground-Truth-Daten weist ein hohes Rauschen auf. Die visuelle Betrachtung der Klassifikation des Validierungsdatensatzes zeigt eine hohe Zuverlässigkeit der Klassifikation. Vom Modell falsch klassifizierte Punkte sind auch für den Menschen schwer zuzuordnen.

7 Diskussion und Ausblick

Die Ergebnisse dieser Arbeit zeigen, dass die semantische Segmentierung von photogrammetrischen Punktwolken möglich ist und dass mit Hilfe der PointNet++ Architektur Ergebnisse erzielt werden können, die dem aktuellen Stand der Forschung entsprechen. Allerdings kann im Rahmen dieser Arbeit nicht untersucht werden, ob PointNet++ die optimale Architektur für diese Aufgabenstellung ist, oder ob eine andere Deep-Learning-Architektur, wie z. B. KPConv (THOMAS et al. 2019), bessere Ergebnisse liefern kann. Gleichzeitig erfordert der Einsatz komplexerer Architekturen eine intensivere Rechenleistung, was sie bei großen Datenmengen ineffizient macht. Zudem ist die Klassenzuordnung der Ground-Truth-Daten verrauscht und keineswegs perfekt. Erst eine Optimierung der Ground-Truth-Daten würde den Einsatz komplexerer Strukturen rechtfertigen. Zudem wird das trainierte Modell nur anhand eines Validierungsdatensatzes getestet, der in seiner Topographie dem Trainingsdatensatz sehr ähnlich ist. Es ist zu erwarten, dass das Modell schlechtere Ergebnisse liefert, wenn die Topographie von den Trainingsdaten abweicht. Dennoch bietet das generierte Modell eine geeignete Grundlage, um photogrammetrische Punktwolken aus Luftbildern semantisch zu segmentieren und weitere Verarbeitungen wie die Ableitung von Geländemodellen oder Dachformen durchzuführen.

8 Literaturverzeichnis

CHEN, M., HU, Q., YU, Z., THOMAS, H., FENG, A., HOU, Y., MCCULLOUGH, K., REN, F. & SOIBELMAN, L., 2022: STPLS3D: A Large-Scale Synthetic and Real Aerial Photogrammetry 3D Point Cloud Dataset. Proceedings of the 33rd British Machine Vision Conference, https://doi.org/10.48550/arXiv.2203.09065.

ESTER, M., KRIEGEL H.-P., SANDER, J. & XU, X., 1996: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the 2nd Internat. Conference on Knowledge Discovery and Data Mining, AAAI Press, 226-231.

GUO, Y., WANG, H., HU, Q., LIU, L. & BENNAMOUN, M., 2021: Deep Learning for 3D Point Clouds: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, **43**(12), 4338-4364. https://doi.org/10.1109/TPAMI.2020.3005434.

HE, Y., YU, H., LIU, X., YANG, Z., SUN, W., & MIAN, A., 2023: Deep Learning based 3D Segmentation: A Survey. https://doi.org/10.48550/arXiv.2103.05423.

KADA, M & KURAMIN, D., 2021: ALS Point Cloud Classification using PointNet++ and KPConv with prior knowledge. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, **XLVI**-4/W4-2021, 91-96.

LIN, T.-Y., GOYAL, P., GIRSHICK, R., HE, K. & DOLLÁR, P., 2020: Focal Loss for Dense Object Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, **42**(2), 318-327, https://doi.org/10.1109/TPAMI.2018.2858826.

MACQUEEN, J., 1967: Some Methods for Classification and analysis of multivariate observations. Proceedings of the 5th Berkeley Symposium on Mathematical Statistics & Probability, Le Cam, L. M. & J. Neyman (Hrsg.), University of Carolina Press, 281-297.

NFRAMES, o.J.: Las format definition. https://docs.nframes.com/input-&-output/output-formats/las-format-definition/, letzter Zugriff 15.01.24.

QI, C. R., SU, H., MO, K. & GUIBAS, L.J., 2017: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 77-85, https://doi.org/10.1109/CVPR.2017.16.

QI, C. R., YI, L., SU, H. & GUIBAS L. J., 2017: PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. 31st Conference on Neural Information Processing Systems (NIPS 2017). 04.-09. December 2017 in Long-Beach (USA).

THOMAS H., QI, C. R., DESCHAUD, J.-E., MARCOTEGUI, B., GOULETTE, F. & GUIBAS, L., 2019: KPConv: Flexible and Deformable Convolution for Point Clouds. IEEE/CVF International Conference on Computer Vision (ICCV), 6410-6419, https://doi.org/10.1109/ICCV.2019.00651.

WANG, Y., SUN, Y., LIU, Z., SARMA, S. E., BRONSTEIN, M. M. & SOLOMON, J. M., 2019: Dynamic Graph CNN for Learning on Point Clouds. ACM Transactions on Graphics, **38**(5), 1-12, https://doi.org/10.1145/3326362.

WIDYANINGRUM, E., BAI, Q., FAJARI, M. K. & LINDENBERGH, R. C., 2021: Airborne Laser Scanning Point Cloud Classification Using the DGCNN Deep Learning Method. Remote Sensing, **13**(5), 859.

WINIWARTER, L & MANDLBURGER, G., 2019: Classification of 3D Point Clouds using Deep Neural Networks. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., **28**, 663-674.

ZHANG, W., QI, J., WAN, P., WANG, H., XIE, D., WANG, X. & YAN, G., 2016: An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. Remote Sensing, **8**(6), https://doi.org/10.3390/rs8060501.