KI-gestützte Fernerkundung und Datenintegration öffentlicher Quellen zur Ableitung von Nachhaltigkeitsindikatoren auf Gebäude- und Parzellenebene

Masterarbeit im Fach Informatik

vorgelegt von

Daniel Kurtz

[redacted]

angefertigt am

Lehrstuhl Informatik 7
-
Rechnernetze und Kommunikationssysteme

Department Informatik
Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)

Betreuer: **Prof. Dr.-Ing. Reinhard German**
Dr.-Ing. Peter Bazan
Alexander Sommer, M. Sc.
Christian Berger, M. Sc.

Abgabe der Arbeit: **15. Juni 2026**

Erklärung

Ich versichere, dass ich die 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 und von dieser als Teil einer Prüfungsleistung angenommen wurde.
Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Declaration

I declare that the work is entirely my own and was produced with no assistance from third parties.
I certify that the work has not been submitted in the same or any similar form for assessment to any other examining body and all references, direct and indirect, are indicated as such and have been cited accordingly.

(Daniel Kurtz)
Erlangen, 15. Juni 2026

Danksagung

Mein Dank gilt Herrn Prof. Dr. Reinhard German und Herrn Dr. Peter Bazan für die formelle Begutachtung dieser Arbeit seitens der Universität. Ebenso danke ich Alexander Sommer für die wertvollen inhaltlichen Anregungen und die konstruktive Begleitung am Lehrstuhl.

Darüber hinaus möchte ich mich bei BUILD.ING Consultants + Innovators für die Möglichkeit bedanken, diese Masterarbeit im Rahmen des gemeinsamen Forschungsprojekts deQarb in einem so spannenden praxisnahen Umfeld verfassen zu dürfen. Ein ganz besonderer Dank richtet sich hierbei an meinen betrieblichen Betreuer Christian Berger für die kontinuierliche fachliche Unterstützung, die offenen Diskussionen und die wertvollen Einblicke in die Unternehmenspraxis.

Nicht zuletzt gilt mein Dank auch den Werkstudierenden der Abteilung, die mich bei der aufwendigen Datenannotation für das KI-Modell tatkräftig unterstützt haben.

Abstract

The sustainable transformation of urban spaces requires spatially precise indicators such as the degree of land sealing and solar potential; however, collecting this data manually is hardly scalable. Existing AI methods typically fail in practice due to high annotation requirements or a lack of physical relevance for real-world applications. Building on this, the present study investigates which segmentation architecture is best suited for fine-grained classification of urban areas when training data is severely limited, and how accurately pixel-based predictions can be translated into spatial indicators at the parcel level.

To this end, an end-to-end pipeline is developed that automatically converts 20-cm aerial images (DOP20) and supplementary elevation and 3D geodata into urban ecological metrics and solar potential, evaluated on a custom-created 15-class dataset for Middle Franconia.

A preliminary architecture search on the FLAIR proxy dataset shows that a self-supervised pre-trained foundation model (DINOv3) with a lightweight SegFormer decoder outperforms supervised CNN and Transformer baselines and achieves over 80 % of its saturation performance with just 75 training images. When applied to the Bavarian dataset, the pixel metric remains moderate (mIoU: 49.89 %); however, the key point is that these local errors largely cancel each other out during aggregation at the parcel level: Ratio-based indicators such as the runoff coefficient (MAE: 0.0215) and unused solar potential (nMAE: 5.12 %) are consistently derived, while quasi-binary metrics remain more sensitive. These values demonstrate low error propagation but do not replace validation against real measurement data.

This study thus demonstrates that foundation models drastically reduce the annotation effort and that the practical suitability of such methods is measured not by pixel metrics but by the accuracy of the derived indicators. The developed framework does not replace field surveys but serves as a resource-efficient tool for pre-qualification.

Kurzfassung

Die nachhaltige Transformation städtischer Räume erfordert räumlich präzise Indikatoren wie Versiegelungsgrad und Solarpotenzial, deren manuelle Erhebung jedoch kaum skalierbar ist. Bestehende KI-Verfahren scheitern in der Praxis meist an hohem Annotationsbedarf oder fehlendem physikalischen Anwendungsbezug. Daran anknüpfend untersucht diese Arbeit, welche Segmentierungsarchitektur sich bei stark begrenzten Trainingsdaten am besten zur feingranularen Klassifikation urbaner Flächen eignet und wie verlustarm sich die pixelbasierten Vorhersagen in raumbezogene Indikatoren auf Parzellenebene überführen lassen.

Hierzu wird eine Ende-zu-Ende-Pipeline entwickelt, die 20 cm-Luftbilder (DOP20) und ergänzende Höhen- und 3D-Geodaten automatisiert in stadtökologische Kennzahlen und Solarpotenziale überführt, evaluiert auf einem eigens erstellten 15-Klassen-Datensatz für Mittelfranken.

Eine vorgeschaltete Architektursuche auf dem Stellvertreterdatensatz FLAIR zeigt, dass ein selbstüberwacht vortrainiertes Basismodell (DINOv3) mit leichtgewichtigem SegFormer-Decoder überwachten CNN- und Transformer-Basislinien überlegen ist und bereits mit nur 75 Trainingsbildern über 80 % seiner Sättigungsleistung erreicht. Bei der Anwendung auf den bayerischen Datensatz bleibt die Pixelmetrik zwar moderat (mIoU: 49,89 %); entscheidend ist jedoch, dass sich diese lokalen Fehler bei der Aggregation auf Parzellenebene weitgehend kompensieren: Verhältnisbasierte Indikatoren wie der Abflussbeiwert (MAE: 0,0215) und das ungenutzte Solarpotenzial (nMAE: 5,12 %) werden konsistent abgeleitet, während quasi-binäre Zielgrößen empfindlicher bleiben. Diese Werte belegen eine geringe Fehlerfortpflanzung, ersetzen aber keine Validierung gegen reale Messdaten.

Die Arbeit zeigt damit, dass Basismodelle den Annotationsaufwand drastisch senken und sich die Praxistauglichkeit solcher Verfahren nicht an der Pixelmetrik, sondern an der Genauigkeit der abgeleiteten Indikatoren bemisst. Das entwickelte Framework ersetzt die fachliche Erhebung nicht, sondern dient als ressourceneffizientes Werkzeug zur Vorqualifizierung.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Vor dem Hintergrund des globalen Klimawandels und der fortschreitenden Urbanisierung gewinnt die nachhaltige Transformation städtischer Räume zunehmend an Bedeutung. Während gesetzliche Rahmenwerke wie das Gebäudeenergiegesetz [1] primär die energetischen und bauphysikalischen Eigenschaften von Gebäuden adressieren, rücken übergreifende Klimaanpassungsstrategien und Zertifizierungssysteme wie die der Deutschen Gesellschaft für Nachhaltiges Bauen (DGNB) [2] das Quartier als Ganzes in den Fokus. Für eine zielgerichtete planerische Umsetzung sind daher räumlich präzise Nachhaltigkeitsindikatoren unerlässlich. Dies betrifft einerseits die Identifikation von Dachflächen für den solaren Ausbau im Sinne der Energiewende, andererseits die Bewertung stadtökologischer Parameter wie Flächenversiegelung und Begrünungspotenziale zur Erhöhung der urbanen Klimaresilienz.

Dank aktueller Open-Data-Strategien und rechtlicher Rahmenbedingungen [3] stehen hochauflösende Geobasisdaten wie Orthophotos, Laserpunktwolken und amtliche Gebäudemodelle heute flächendeckend zur Verfügung. Sie bieten ein enormes Potenzial für detaillierte Analysen urbaner Räume bis auf Parzellenebene. Die praktische Nutzbarmachung dieser heterogenen Datenquellen stellt jedoch weiterhin eine erhebliche Herausforderung dar: Die manuelle Erfassung relevanter Gebäude- und Flächenmerkmale ist äußerst zeitaufwändig und für große Gebiete kaum skalierbar.

Bestehende, auf Künstlicher Intelligenz (KI) basierende Ansätze fokussieren sich häufig auf makroskopische Auswertungen [4] oder optimieren theoretische Metriken auf idealisierten Benchmark-Datensätzen [5], [6], ohne die Ergebnisse in physikalische Bewertungsverfahren zu überführen. Angewandte Studien setzen hingegen meist massive annotierte Trainingsdaten voraus [7], [8]. Genau dieser Mangel an umfangreichen Annotationen erschwert jedoch den praktischen Einsatz moderner Deep-Learning-Methoden in der Fernerkundung maßgeblich [9], [10]. Für eine praxisnahe Nachhaltigkeitsbewertung im datenarmen Umfeld fehlen daher bis-

lang integrierte Ende-zu-Ende-Verfahren, die hochauflösende Fernerkundungsdaten automatisiert verarbeiten und direkt in räumliche Indikatoren übersetzen.

An dieser Forschungslücke setzt die vorliegende Masterarbeit an, die in Kooperation mit der Firma BUILD.ING Consultants + Innovators GmbH im Rahmen des Forschungsprojekts deQarb[1] entstanden ist. Ziel ist die Entwicklung eines automatisierten, KI-gestützten Frameworks zur Ableitung von Nachhaltigkeitsindikatoren auf Gebäude- und Parzellenebene.

Um dies zu erreichen, wird eine Ende-zu-Ende-Pipeline konzipiert, die multimodale Geodaten integriert und Methoden der semantischen Segmentierung mit nachgelagerten georäumlichen Analysen verknüpft. Auf diese Weise lassen sich urbane Flächenstrukturen automatisiert erfassen und Schlüsselindikatoren wie der Versiegelungsgrad, der Anteil nachhaltig genutzter Dachflächen und das Solarpotenzial präzise ableiten. Ein besonderer methodischer Fokus liegt dabei auf dem effizienten Einsatz von Deep Learning unter den Bedingungen stark begrenzter Trainingsdaten.

Um die formulierten Ziele systematisch zu bearbeiten, ist die vorliegende Arbeit wie folgt gegliedert: Auf diese Einleitung aufbauend, vermittelt Kapitel 2 die theoretischen Grundlagen der künstlichen neuronalen Netze, der Geoinformatik sowie der Fernerkundung und ordnet die Arbeit in den aktuellen Forschungsstand ein. Kapitel 3 beschreibt die Methodik der konzipierten Ende-zu-Ende-Pipeline, von der Datenaufbereitung über das Training der Segmentierungsmodelle bis hin zur Ableitung der räumlichen Indikatoren. In Kapitel 4 erfolgt die quantitative und qualitative Analyse der Modelle sowie die Evaluierung der extrahierten Nachhaltigkeitsindikatoren anhand konkreter Anwendungsfälle. Abschließend fasst Kapitel 5 die zentralen Erkenntnisse der Arbeit zusammen und gibt einen Ausblick auf zukünftige Forschungspotenziale sowie die praktische Übertragbarkeit der Ergebnisse.

Aus Gründen der sprachlichen Präzision und Lesbarkeit wird in der vorliegenden Arbeit bei international etablierten Fachbegriffen auf eine künstliche Eindeutschung verzichtet. Um dennoch eine durchgehende Verständlichkeit zu gewährleisten, wird bei der jeweils ersten Nennung dieser spezifischen Begriffe eine deutsche Entsprechung oder Erläuterung angeführt.

[1] Vollständiger Projekttitel: KMU-innovativ – Verbundprojekt Klimaschutz: Digitales Toolset zur Effizienzsteigerung und Dekarbonisierung in komplexen Gebäude-, Portfolio- und Quartiersenergiesystemen (deQarb).

Kapitel 2

Grundlagen und verwandte Arbeiten

Dieses Kapitel legt das theoretische Fundament für die vorliegende Arbeit. Zunächst werden die mathematischen und informatischen Grundlagen künstlicher neuronaler Netze sowie die spezifischen Herausforderungen der pixelgenauen Klassifizierung im Bereich Computer Vision erläutert. Darauf aufbauend folgt eine detaillierte Beschreibung der verwendeten Fernerkundungsdaten und Geoinformationssysteme, die als Datengrundlage für die Solarpotenzialanalyse dienen, sowie der Verfahren zu deren Annotation. Abschließend werden aktuelle Forschungsarbeiten sowie die technische Arbeitsumgebung vorgestellt, um die methodische Einordnung der Arbeit zu ermöglichen.

2.1 Künstliche Neuronale Netze und Deep Learning

Künstliche neuronale Netze (engl. *Artificial Neural Networks*) bilden das algorithmische Fundament des modernen maschinellen Lernens. Im Gegensatz zu klassischen, regelbasierten Ansätzen lernen sie eigenständig hierarchische Repräsentationen direkt aus den zur Verfügung gestellten Daten [11].

Die elementare Architekturform bildet das mehrschichtige Perzeptron (engl. *Multilayer Perceptron*), bei dem künstliche Neuronen in einer Eingabeschicht, einer oder mehreren verborgenen Schichten (engl. *Hidden Layers*) und einer Ausgabeschicht vollständig miteinander vernetzt sind. Jedes Neuron berechnet dabei eine gewichtete Summe seiner Eingaben, addiert einen Bias-Wert und wendet eine nicht-lineare Aktivierungsfunktion an, um komplexe Zusammenhänge abbilden zu können.

Das Lernen dieser komplexen Zusammenhänge erfolgt durch die iterative Anpassung der Netzwerkgewichte. Wie in Abbildung 2.1 schematisch dargestellt, gliedert sich dieser Trainingszyklus in mehrere Phasen: Im sogenannten Vorwärtsdurchlauf (engl. *Forward Pass*) durchlaufen die Eingabedaten (engl. *Batch*) die Schichten des Netzwerks, welches daraus eine Modellvorhersage erzeugt. Eine Verlustfunktion

quantifiziert anschließend die Abweichung zwischen dieser Vorhersage und der idealen Referenzmaske (engl. *Ground Truth*) zu einem Fehlerwert.

Um diesen Fehler zu minimieren, kommt im Rückwärtsdurchlauf der *Backpropagation*-Algorithmus zum Einsatz. Er berechnet die Gradienten des Fehlers bezüglich aller Netzwerkgewichte. Auf Basis dieser Informationen führt ein adaptiver Optimierer, wie beispielsweise AdamW [12], die finale Gewichtsaktualisierung durch. Dieser Zyklus wiederholt sich für alle Batches über mehrere Trainingsepochen hinweg, bis das Modell konvergiert.

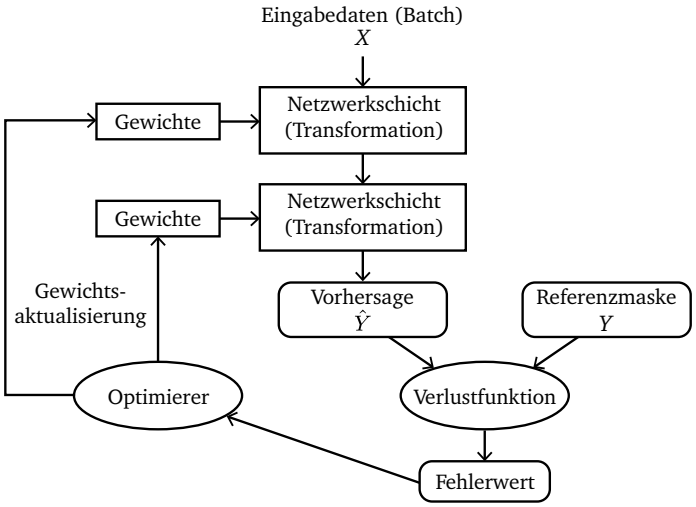


Abbildung 2.1 – Schematische Darstellung des iterativen Trainingszyklus eines künstlichen neuronalen Netzes. Der Vorwärtsdurchlauf generiert aus den Eingabedaten eine Vorhersage, deren Abweichung zur Referenzmaske durch die Verlustfunktion quantifiziert wird. Basierend auf diesem Fehlerwert aktualisiert der Optimierer die Netzwerkgewichte. Eigene Darstellung nach [13].

Mehrschichtige Perzeptronen sind jedoch für die direkte Verarbeitung hochdimensionaler Bilddaten ungeeignet, da sie räumliche Strukturen nicht berücksichtigen und bei hochauflösenden Eingaben eine nicht handhabbare Parameteranzahl erfordern. Diese Limitierung motiviert den Einsatz von faltenden neuronalen Netzen (engl. *Convolutional Neural Networks*, CNNs), die im folgenden Abschnitt beschrieben werden.

2.2 Deep-Learning-Architekturen in der Bildverarbeitung

Die automatische Extraktion von Informationen aus Bilddaten hat durch den Einsatz tiefer künstlicher neuronaler Netze erhebliche Fortschritte erzielt. Während traditionelle Ansätze der Bildverarbeitung auf algorithmisch beschreibbaren Merkmalen,

wie beispielsweise der Kantenerkennung, basierten, lernen moderne Architekturen diese Repräsentationen hierarchisch und datengetrieben direkt aus den Eingabebildern [11]. Im Folgenden werden die grundlegenden Konzepte und Architekturen vorgestellt, die für die in dieser Arbeit durchgeführte pixelgenaue Klassifizierung von Luftbildern relevant sind.

2.2.1 Faltungsnetze

Zur Verarbeitung hochdimensionaler Bilddaten haben sich CNNs als zentraler Architekturtyp etabliert. Im Deep Learning werden solche Daten typischerweise als Tensoren repräsentiert. Ein Tensor ist dabei eine mathematische Verallgemeinerung von Vektoren und Matrizen auf eine beliebige Anzahl von Dimensionen. Ein gewöhnliches Farbbild wird folglich als dreidimensionaler Tensor aufgefasst, der sich aus der räumlichen Höhe, der Breite und den Farbkanälen (z. B. Rot, Grün, Blau) zusammensetzt. Im Gegensatz zu vollständig vernetzten Netzen berücksichtigen CNNs die räumliche Struktur dieser Tensoren explizit und ermöglichen so eine effiziente Extraktion visueller Merkmale [11].

CNNs basieren auf den Prinzipien der lokalen Konnektivität und der Gewichtsteilung. Anstatt jedes Neuron mit allen Eingabewerten zu verbinden, operieren Faltungsschichten auf lokalen Bildausschnitten. Dabei wird ein lernbarer Filter (Kernel) über das Eingabebild bewegt und erzeugt sogenannte Merkmalskarten (engl. *Feature Maps*), die das Auftreten spezifischer Muster wie Kanten, Texturen oder Formen kodieren.

Durch das Stapeln mehrerer Faltungsschichten entsteht eine hierarchische Repräsentation: Frühere Schichten erfassen einfache, lokale Strukturen, während tiefere Schichten zunehmend komplexe und semantische Merkmale abstrahieren. Ein zentrales Konzept ist hierbei das rezeptive Feld, das beschreibt, welcher Bereich des Eingabebildes in die Aktivierung eines Neurons einfließt. Mit zunehmender Netzwerktiefe wächst dieses Feld, sodass globale Kontextinformationen berücksichtigt werden können.

Zur Reduktion der räumlichen Auflösung und des Rechenaufwands werden häufig Pooling-Operationen eingesetzt. Diese verdichten die Merkmalskarten und erhöhen gleichzeitig die Invarianz gegenüber kleinen Verschiebungen im Eingabebild.

Für viele Aufgaben der Computer Vision, insbesondere solche mit räumlich aufgelösten Vorhersagen, stellen durch tiefe neuronale Architekturen gelernte hierarchische Merkmalsrepräsentationen eine zentrale Grundlage dar. CNNs zählen dabei zu den etabliertesten und historisch dominierenden Architekturtypen in der Bildverarbeitung.

2.2.2 Einordnung der semantischen Segmentierung

Aufbauend auf den zuvor beschriebenen Eigenschaften von CNNs stellt die semantische Segmentierung eine der anspruchsvollsten Aufgaben der Computer Vision dar. Um ihre Komplexität einordnen zu können, ist eine Abgrenzung zu verwandten Paradigmen der maschinellen Bildverarbeitung notwendig, wie in Abbildung 2.2 veranschaulicht:

- **Bildklassifikation (engl. *Image Classification*):** Dies ist die grundlegendste Aufgabe, bei der einem gesamten Eingabebild eine einzelne, globale Kategorie zugewiesen wird (z. B. „Das Bild zeigt einen Parkplatz“) [14]. Lokale räumliche Informationen gehen dabei vollständig verloren (siehe Abbildung 2.2a).

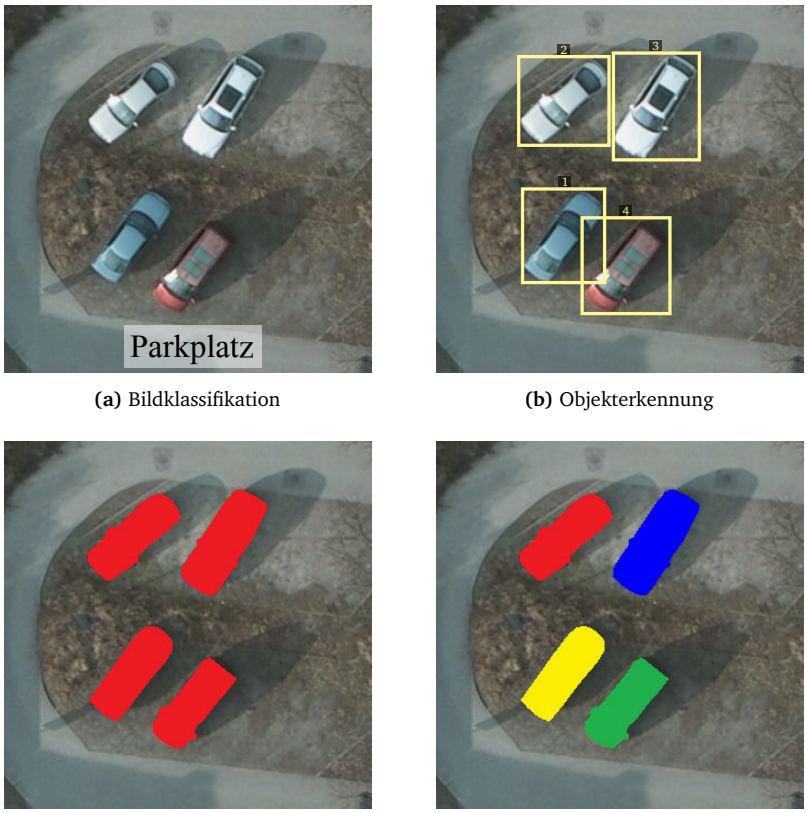
- **Objekterkennung (engl. *Object Detection*):** Hierbei werden Objekte im Bild nicht nur klassifiziert, sondern auch grob lokalisiert [15]. Die Ausgabe erfolgt typischerweise in Form von umschließenden Rechtecken (engl. *Bounding Boxes*). Die genaue Form oder Kontur des Objekts wird jedoch nicht erfasst (siehe Abbildung 2.2b).

- **Semantische Segmentierung (engl. *Semantic Segmentation*):** Im Gegensatz zur Bildklassifikation verfolgt die semantische Segmentierung das Ziel, eine dichte Vorhersage (engl. *Dense Prediction*) zu erzeugen. Dabei wird jedem einzelnen Pixel des Eingabebildes eine semantische Klasse zugeordnet [16]. Wichtig ist hierbei, dass nicht zwischen einzelnen Instanzen derselben Klasse unterschieden wird. Stehen beispielsweise zwei Autos im Bild direkt nebeneinander, verschmelzen sie in der Vorhersagemaske zu einer zusammenhängenden Fläche. Doch auch wenn sie räumlich getrennt sind, werden alle zugehörigen Pixel schlicht derselben Klasse *Auto* zugeordnet und identisch markiert (siehe Abbildung 2.2c).

- **Instanz-Segmentierung (engl. *Instance Segmentation*):** Diese Methode kombiniert Objekterkennung und semantische Segmentierung. Sie identifiziert einzelne Objekte und weist ihnen pixelgenaue Masken zu, unterscheidet aber strikt zwischen separaten Instanzen (z. B. *Auto 1* und *Auto 2*) [17] (siehe Abbildung 2.2d).

Formal gibt ein neuronales Netz für ein Eingabebild der Dimension $H \times W \times D$ nicht direkt die finalen Klassen aus, sondern generiert einen Ausgabe-Tensor der Dimension $H \times W \times C$, wobei C der Anzahl der definierten Zielklassen entspricht [11]. Die Rohwerte dieses Tensors werden als *Logits* bezeichnet.

Um von diesem globalen Tensor zu einer Klassifikation zu gelangen, wird jeder räumliche Pixel einzeln betrachtet.



(a) Bildklassifikation **(b)** Objekterkennung

(c) Semantische Segmentierung **(d)** Instanz-Segmentierung

Abbildung 2.2 – Gegenüberstellung grundlegender Paradigmen der maschinellen Bildverarbeitung. Während die Bildklassifikation (a) ein globales Label vergibt und die Objekterkennung (b) Objekte grob lokalisiert, ordnet die semantische Segmentierung (c) jedem Pixel eine Klasse zu, ohne Instanzen zu trennen. Die Instanz-Segmentierung (d) separiert zusätzlich einzelne Objekte der gleichen Klasse. Eigene Darstellung auf Basis von [18].

Für einen bestimmten Pixel an der Position (h, w) bilden die Werte der C Zielklassen einen Vektor $\mathbf{z}$. Da die Werte dieses Logit-Vektors unbeschränkte reelle Zahlen sind, die auch negative Werte annehmen können, werden sie für die finale Klassenzuweisung (sowie die spätere Berechnung der Verlustfunktion) meist durch eine Softmax-Aktivierungsfunktion transformiert.

$$\text{Softmax}(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \quad \text{für } i = 1, \ldots, C \tag{2.1}$$

Diese Operation wandelt den Logit-Vektor $\mathbf{z}$ in eine diskrete Wahrscheinlichkeitsverteilung um, deren Werte zwischen 0 und 1 liegen. Dabei ist entscheidend, dass sich diese Wahrscheinlichkeiten für jeden Pixel isoliert betrachtet über alle

C Klassen hinweg zu 1 summieren. Die finale Segmentierungsmaske ergibt sich anschließend durch die *Argmax-Operation*, welche für jeden Pixel separat die Klasse mit der höchsten Wahrscheinlichkeit auswählt.

Da die semantische Segmentierung eine pixelgenaue Zuordnung verlangt, ergeben sich spezifische architektonische Herausforderungen, die klassische Klassifikationsnetzwerke nicht bewältigen müssen:

1. **Verlust räumlicher Auflösung:** Wie im vorherigen Abschnitt beschrieben, nutzen tiefe neuronale Netze häufig Pooling-Operationen, um den Rechenaufwand zu senken und ein größeres rezeptives Feld zu generieren. Dies führt jedoch zu einem drastischen Verlust der räumlichen Auflösung. Für eine dichte Vorhersage müssen diese komprimierten Merkmalskarten in nachgelagerten Schichten wieder auf die ursprüngliche Bildgröße hochskaliert werden, ohne dass feine Objektdetails an den Rändern verloren gehen.

2. **Integration von lokalem und globalem Kontext:** Um einen Pixel korrekt zu klassifizieren, benötigt das Netzwerk sowohl stark lokalisierte Informationen (z. B. Kanten und Texturen zur exakten Abgrenzung) als auch globalen Kontext (z. B. das Wissen, dass ein Schornsteinpixel wahrscheinlich von Dachpixeln umgeben ist). Die Fusion dieser unterschiedlichen Informationsebenen erfordert spezialisierte Netzwerkarchitekturen.

3. **Klassenungleichgewicht (engl. *Class Imbalance*):** In realen Datensätzen sind Klassen oft extrem ungleich verteilt. Während Hintergrundklassen (wie Straßen oder Vegetation) große Bildbereiche einnehmen, machen kleine Objekte (wie Autos oder einzelne Bäume) oft nur einen Bruchteil der Pixel aus. Das Netzwerk tendiert ohne entsprechende Gegenmaßnahmen (wie angepasste Verlustfunktionen) dazu, die dominanten Klassen stark zu bevorzugen.

Diese spezifischen Anforderungen haben zur Entwicklung der *Encoder-Decoder-Architekturen* geführt, welche heute die Grundlage fast aller modernen Segmentierungsnetzwerke bilden [19].

2.2.3 Encoder-Decoder-Paradigma

Encoder-Decoder-Architekturen adressieren die zuvor beschriebenen Anforderungen, indem sie globale Kontextinformation und räumliche Präzision in einer gemeinsamen Modellstruktur vereinen. Sie stellen heute den Standard für dichte Bildvorhersagen dar und wurden insbesondere durch Architekturen wie U-Net [19] und SegNet [20] geprägt.

Wie der schematische Aufbau in Abbildung 2.3 zeigt, gliedert sich eine solche Architektur in zwei primäre Komponenten:

- **Encoder (Kontraktionspfad):** Der Encoder (Kodierer) nimmt das hochauflösende Eingabebild auf und wendet sukzessive Faltungs- und Pooling-Operationen an. Dadurch wird die räumliche Dimension der Merkmalskarten schrittweise reduziert, während die Tiefe (Anzahl der Kanäle) zunimmt. Dieser Prozess filtert feingranulare Details heraus und abstrahiert zunehmend komplexe, globale Muster. An der tiefsten Stelle des Netzwerks, dem sogenannten Flaschenhals (engl. *Bottleneck*), liegt die Information maximal verdichtet vor und verfügt über das größte rezeptive Feld.

- **Decoder (Expansionspfad):** Der Decoder (Dekodierer) skaliert die semantisch dichten, niedrig aufgelösten Merkmalskarten schrittweise auf die ursprüngliche Bildauflösung zurück. Dies geschieht entweder durch bilineare Interpolation oder durch lernbare transponierte Faltungen. Letztere können jedoch bei ungünstiger Konfiguration Schachbrettmuster-Artefakte erzeugen, weshalb bilineare Interpolation gefolgt von einer regulären Faltungsschicht in der Praxis häufig bevorzugt wird.

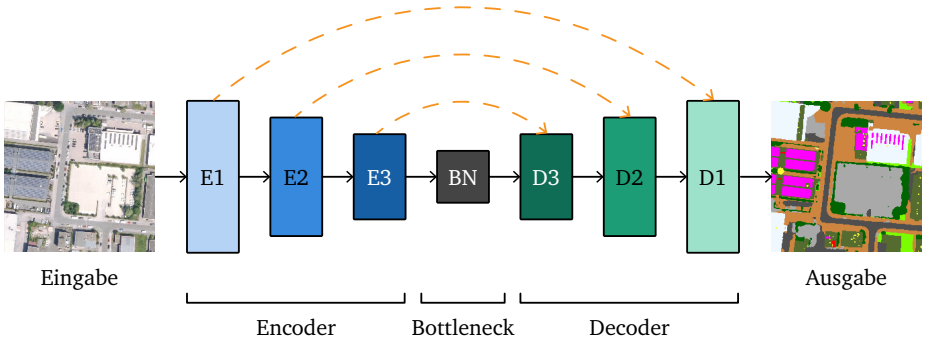


Abbildung 2.3 – Schematische Darstellung einer Encoder-Decoder-Architektur. Der Encoder reduziert schrittweise die räumliche Auflösung der Eingabe (dargestellt durch die abnehmende Blockhöhe), um globale Merkmale zu extrahieren. Der Decoder rekonstruiert anschließend die ursprüngliche Bildgröße. Die orange gestrichelten Pfeile markieren die Skip-Connections, welche hochauflösende Detailinformationen aus dem Encoder direkt in den Decoder überführen. Eigene Darstellung in Anlehnung an [20].

Ein zentrales Problem klassischer Encoder-Decoder-Architekturen ist der Verlust hochfrequenter räumlicher Informationen durch wiederholte Pooling-Operationen im Encoder. Dies erschwert insbesondere die präzise Rekonstruktion von Objektgrenzen.

Zur Lösung dieses Problems wurden in Architekturen wie dem U-Net sogenannte *Skip-Connections* (Querverbindungen) eingeführt [19]. Wie in Abbildung 2.3 veranschaulicht, leiten diese Verbindungen hochauflösende Merkmalskarten aus den frühen Encoder-Schichten direkt an korrespondierende Decoder-Schichten weiter, wo sie mit den hochskalierten, semantischen Merkmalen fusioniert werden. Da-

durch wird eine effektive Kombination aus lokal präzisen Strukturinformationen und globalem semantischem Kontext ermöglicht.

Neben der Verbesserung der räumlichen Detailtreue tragen Skip-Connections auch zu einem stabileren Trainingsverhalten bei, da sie den Gradientenfluss durch das Netzwerk erleichtern. Encoder-Decoder-Architekturen bilden die Grundlage für die in dieser Arbeit eingesetzten Modelle zur semantischen Segmentierung von Luftbildern.

2.2.4 Encoder-Architekturen

In der modernen Computer Vision hat sich für den Encoder-Teil eines Netzwerks der Begriff *Backbone* etabliert. Seine primäre Aufgabe besteht in der diskriminativen Merkmalsextraktion aus dem Eingabebild. Um den Trainingsprozess zu beschleunigen und die Generalisierungsfähigkeit zu erhöhen, werden diese Encoder in der Regel auf massiven Bilddatensätzen (wie ImageNet) vortrainiert und anschließend an die spezifische Segmentierungsaufgabe angepasst. Die theoretischen Hintergründe und spezifischen Strategien dieses Vortrainings werden in Abschnitt 2.3.3 detailliert behandelt. Im Rahmen dieser Arbeit werden verschiedene Encoder-Architekturen evaluiert, die historisch gewachsene und aktuelle Paradigmen repräsentieren:

- **ResNet (Residual Networks):** Als klassischer Vertreter der CNNs adressiert ResNet das Problem verschwindender Gradienten in sehr tiefen Netzwerken durch die Einführung von *Residual Blocks*. Diese integrieren Skip-Connections innerhalb des Backbones, wodurch das Netzwerk lediglich die Residuen (Restfehler) anstelle der gesamten zugrundeliegenden Abbildung lernen muss [14]. ResNet dient in vielen Segmentierungsaufgaben als etablierte Basislinie für den Encoder [6], [21].

- **ViT (Vision Transformer):** Ursprünglich für die Verarbeitung natürlicher Sprache entwickelt, hat die Transformer-Architektur durch den ViT auch in der Bildverarbeitung Einzug gehalten. Anstelle von Faltungsoperationen zerlegt der ViT das Bild in nicht-überlappende Bereiche (engl. *Patches*), beispielsweise der Größe 16×16 Pixel, und verarbeitet diese als Sequenz. Durch den sogenannten *Self-Attention*-Mechanismus kann das Modell globale Abhängigkeiten zwischen allen Bildbereichen bereits in frühen Schichten erfassen [22]. Ein Nachteil für dichte Vorhersagen ist jedoch das Fehlen hierarchischer Merkmalskarten, da die Auflösung über alle Schichten hinweg konstant bleibt.

- **ViT-Adapter:** Obwohl es sich hierbei nicht um eine eigenständige Backbone-Architektur im klassischen Sinne handelt, stellt der ViT-Adapter eine essenzielle Erweiterung des Standard-ViT für dichte Bildvorhersagen dar. Wie zuvor beschrieben, liefert der reguläre ViT aufgrund seiner konstanten Token-Auflösung

keine hierarchischen Merkmalskarten. Anstatt die interne Architektur des Transformers zu modifizieren, erweitert der ViT-Adapter das Netzwerk um externe Module. Über ein sogenanntes *Spatial Prior Module* werden dem Netzwerk lokale räumliche Informationen zugeführt. Parallel dazu extrahiert der Adapter aus verschiedenen Tiefen des ViT Informationen und fusioniert sie zu mehrskaligen Merkmalskarten. Dieser Ansatz ermöglicht es, die leistungsstarken, auf massiven Datensätzen vortrainierten Gewichte reiner Vision-Transformer direkt und effizient für komplexe Segmentierungsaufgaben zu adaptieren [23].

- **Swin Transformer (Shifted Window Transformer):** Um die Limitierungen des ViT für Computer-Vision-Aufgaben zu überwinden, führt der Swin Transformer eine hierarchische Struktur ein, die der von CNNs ähnelt. Die Self-Attention-Berechnung wird hierbei auf lokale, verschobene Fenster (engl. *Shifted Windows*) beschränkt. Dies reduziert die quadratische Rechenkomplexität klassischer Transformer auf ein lineares Maß bezüglich der Bildgröße und generiert mehrskalige Merkmalskarten, die für Segmentierungsaufgaben essenziell sind [24].

- **InternImage:** Als Reaktion auf die Dominanz der Vision-Transformer demonstriert InternImage, dass auch rein faltungsbasierte Architekturen modernste Ergebnisse erzielen können. Der Kern dieser Architektur basiert auf deformierbaren Faltungen (engl. *Deformable Convolutions v3*), die das rezeptive Feld dynamisch an die räumliche Struktur der Objekte im Bild anpassen können. Dadurch vereint InternImage den globalen Kontext von Transformern mit der effizienten induktiven Vorprägung (engl. *Inductive Bias*) klassischer CNNs [25].

2.2.5 Decoder-Architekturen

Der Decoder, in der aktuellen Literatur häufig als Segmentierungskopf (engl. *Segmentation Head*) bezeichnet, empfängt die vom Encoder extrahierten, oftmals mehrskaligen Merkmalskarten und transformiert diese in die finale, pixelgenaue Klassifikationsmaske. Je nach Art der vom Encoder gelieferten Informationen kommen unterschiedliche Decoder-Konzepte zum Einsatz, deren Leistungsfähigkeit in dieser Arbeit vergleichend untersucht wird:

- **UperNet (Unified Perceptual Parsing Network):** UperNet ist ein etablierter, hochgradig flexibler Decoder, der ein *Feature Pyramid Network* (FPN) mit einem *Pyramid Pooling Module* (PPM) kombiniert. Es wurde explizit dafür entwickelt, hierarchische Merkmale aus unterschiedlichen Netzwerkschichten effizient zu fusionieren. Dadurch eignet es sich besonders als standardisierter Decoder für den Vergleich verschiedener hierarchischer Backbones [26].

- **Atrous Spatial Pyramid Pooling (DeepLabV3+):** Als weitreichend erprobter faltungsbasierter Standard nutzt diese Architektur sogenannte *Atrous Convolutions* (dilatierte Faltungen). Dadurch kann das Modell das rezeptive Feld flexibel vergrößern, ohne an räumlicher Auflösung zu verlieren. Dies ist insbesondere bei hochauflösenden Luftbildern von Vorteil, um feine räumliche Strukturen wie schmale Gebäudeabstände oder kleine Dachaufbauten präzise aufzulösen [27].

- **UNetFormer:** Diese Architektur wurde speziell für die Herausforderungen der Fernerkundung konzipiert. Der Decoder greift die bewährte makrostrukturelle Form des U-Nets auf, integriert jedoch komplexe *Global-Local-Transformer-Block*-Module in den Expansionspfad. Dies ermöglicht eine verbesserte Rekonstruktion feiner Objektränder, wie sie bei Gebäuden oder Solaranlagen in Luftbildern entscheidend sind [5].

- **SegFormer:** Im Gegensatz zu komplexen Decodern wie UperNet verfolgt SegFormer einen minimalistischen Ansatz. Er verzichtet auf rechenintensive Module zur Kontextaggregation und besteht lediglich aus leichtgewichtigen mehrschichtigen Perzeptronen. Dieser Ansatz ist möglich, weil der zugehörige Transformer-Backbone bereits hochgradig semantisch angereicherte, mehrskalige Merkmale liefert. SegFormer zeichnet sich daher durch eine hohe Effizienz und Stabilität aus [28].

- **Mask2Former:** Dieser Ansatz stellt einen Paradigmenwechsel in der Segmentierung dar. Anstatt jedem Pixel isoliert eine Klasse zuzuweisen (Pixel-Klassifikation), formuliert Mask2Former das Problem als Masken-Klassifikation. Mittels Transformer-Decodern und sogenannten *Object Queries* lernt das Netzwerk, direkt binäre Masken und zugehörige Klassenwahrscheinlichkeiten für auftretende Objekte oder Flächen vorherzusagen. Diese Architektur hat sich als äußerst leistungsstark erwiesen und vereint Konzepte der semantischen sowie der Instanz-Segmentierung in einem universellen Framework [29].

2.3 Trainingsstrategien im Low-Data-Regime

Obwohl moderne Deep-Learning-Architekturen wie CNNs und ViT beeindruckende Ergebnisse in der pixelgenauen Segmentierung erreichen, sind sie extrem datenhungrig. Sie erfordern für ein erfolgreiches Training in der Regel Zehntausende bis Millionen von annotierten Beispielen, um repräsentative Merkmale zu lernen und eine ausreichende Generalisierungsfähigkeit zu entwickeln. In vielen anwendungsspezifischen Domänen der Fernerkundung liegt jedoch ein sogenanntes *Low-Data-Regime* vor. Hochwertige, manuell erstellte Segmentierungsmasken sind zeit-

und kostenintensiv in der Beschaffung, weshalb die Menge an verfügbaren Trainingsdaten stark limitiert ist. Dieser Abschnitt beleuchtet die daraus resultierenden Herausforderungen und stellt etablierte Strategien vor, um tiefe neuronale Netze auch mit begrenzten Datenmengen stabil zu trainieren.

2.3.1 Herausforderungen und Verlustfunktionen

Die zwei primären Herausforderungen beim Training mit wenigen Daten sind die Überanpassung (engl. *Overfitting*) und das extreme Klassenungleichgewicht. Überanpassung beschreibt das Phänomen, bei dem ein Modell die Trainingsdaten gewissermaßen auswendig lernt, anstatt zugrundeliegende, generalisierbare Muster zu abstrahieren. Dies führt zu einer sehr hohen Genauigkeit auf den Trainingsdaten, aber einer unzureichenden Leistung auf ungesehenen Testdaten [11].

Zusätzlich weisen Fernerkundungsdatensätze für spezifische Klassen oft ein drastisches Klassenungleichgewicht auf. Bei der Suche nach Solaranlagen beispielsweise nimmt die Hintergrundklasse (Gebäude, Straßen, Vegetation) den weitaus größten Teil der Pixel ein, während die Zielklasse nur einen Bruchteil ausmacht. Wird ein solches Modell mit einer Standard-Kreuzentropie (engl. *Cross-Entropy Loss*) trainiert, dominiert der Fehler der Hintergrundklasse den gesamten Gradienten. Das Netzwerk tendiert dann dazu, ein triviales Modell zu erlernen, das einfach alle Pixel als Hintergrund klassifiziert.

Um dieses Problem anzugehen, kommen spezialisierte Verlustfunktionen zum Einsatz. Die gewichtete Kreuzentropie (engl. *Weighted Cross-Entropy Loss*) weist den verschiedenen Klassen vordefinierte Gewichte zu, wodurch Fehlklassifikationen der unterrepräsentierten Klassen stärker bestraft werden und das Netzwerk somit gezwungen wird, diese besser zu erlernen [19]. Außerdem wird in der Segmentierung häufig die Dice-Verlustfunktion verwendet. Sie basiert auf dem Sørensen-Dice-Koeffizienten und maximiert direkt die räumliche Überlappung zwischen der Vorhersage und der Referenzmaske, unabhängig von der absoluten Größe der vorhergesagten Fläche. Dadurch ist sie von Natur aus weniger anfällig für starke Klassenungleichgewichte [30].

Während spezialisierte Verlustfunktionen primär dem Problem des Klassenungleichgewichts entgegenwirken, erfordert die Vermeidung von Überanpassung im Low-Data-Regime gesonderte Strategien auf Daten- und Architekturebene.

2.3.2 Datenaugmentierung und Regularisierung

Eine der effektivsten und am weitesten verbreiteten Methoden zur Bekämpfung von Überanpassung ist die Datenaugmentierung (engl. *Data Augmentation*). Ihr Ziel ist es, die Größe und Varianz des Trainingsdatensatzes künstlich zu erhöhen,

indem während des Trainingsprozesses stochastische Transformationen auf die Eingabedaten angewendet werden [31]. Bei den Transformationen wird grundlegend zwischen geometrischen und photometrischen Methoden unterschieden, was eine strikte Unterscheidung in der Handhabung von Bild und Referenzmaske erfordert:

- **Geometrische Transformationen:** Diese verändern die räumliche Struktur der Daten. Typische Operationen umfassen das Skalieren (engl. *Resize*), zufälliges Beschneiden (engl. *Random Crop*), Spiegeln (horizontal sowie vertikal) und Rotieren. Da sich hierbei die Position und Form der Objekte im Bild ändert, müssen diese Transformationen zwingend synchron auf das Eingabebild *und* die korrespondierende Segmentierungsmaske angewendet werden. Nur so bleibt die pixelgenaue Zuordnung zwischen Bildmerkmal und Klassenlabel erhalten. Diese Methoden machen das Modell invariant gegenüber unterschiedlichen Aufnahmewinkeln, Ausrichtungen und Maßstäben der gesuchten Objekte.

- **Photometrische Transformationen:** Diese Methoden, oft auch als Pixeltransformationen bezeichnet, verändern ausschließlich die Farb- und Intensitätswerte der Pixel, ohne deren räumliche Geometrie zu beeinflussen. Folglich werden sie nur auf das Eingabebild angewendet, während die Referenzmaske unverändert bleibt. Zu den gängigen Techniken gehören die stochastische Variation von Helligkeit, Kontrast, Farbsättigung und Farbton (engl. *Photometric Distortion*) sowie das Hinzufügen von Bildrauschen (z. B. Gaußsches Rauschen) oder Unschärfe. In der Fernerkundung helfen diese Augmentierungen, das Netzwerk unempfindlich gegenüber unterschiedlichen Beleuchtungsverhältnissen, atmosphärischen Störungen oder variierender Sensorqualität zu machen.

Ergänzt werden diese klassischen Ansätze zunehmend durch fortgeschrittene Regularisierungsmethoden auf Datenebene, wie beispielsweise das *Cutout* [32]. Hierbei werden stochastisch gewählte, rechteckige Bildbereiche maskiert, indem ihre Pixelwerte auf null gesetzt werden. Um das Netzwerk nicht fälschlicherweise für Vorhersagen in diesen zerstörten Bereichen zu bestrafen, wird den korrespondierenden Pixeln in der Referenzmaske ein spezifischer Ignorierwert (engl. *Ignore Index*) zugewiesen, der sie von der Verlustberechnung ausschließt. Solche Verdeckungstechniken zwingen das Modell dazu, sich nicht auf isolierte, lokal stark ausgeprägte Merkmale zu verlassen, sondern den breiteren, globalen Kontext eines Objekts zu erlernen.

Neben diesen datengetriebenen Ansätzen kommen ergänzend auch Regularisierungstechniken auf Modellebene zum Einsatz. Dazu zählen beispielsweise *Dropout*, bei dem während des Trainings zufällig Neuronen deaktiviert werden, sowie *Weight Decay*, welches große Gewichtswerte bestraft und so die Modellkomplexität reduziert. Auch das vorzeitige Abbrechen (engl. *Early Stopping*) stellt eine effektive

Maßnahme dar, indem das Training basierend auf der Validierungsleistung frühzeitig beendet wird, bevor es zur Überanpassung kommt. Diese Methoden wirken komplementär zur Datenaugmentierung und tragen zusätzlich zur Verbesserung der Generalisierungsfähigkeit im Low-Data-Regime bei.

2.3.3 Transferlernen

Der historisch etablierteste Ansatz zur Bewältigung von Datenknappheit ist das Transferlernen (engl. *Transfer Learning*). Die grundlegende Idee besteht darin, ein Modell nicht von Grund auf mit zufällig initialisierten Gewichten zu trainieren, sondern Wissen, das auf einem großen und datenreichen Ausgangsdatensatz erlernt wurde, auf die eigentliche, datenarme Zielaufgabe (Zieldomäne) zu übertragen [33].

In der Praxis erfolgt dies meist durch ein überwachtes Vortraining (engl. *Supervised Pre-training*). Standardmäßig werden hierfür gewaltige Bilddatensätze wie ImageNet [34] genutzt, die Millionen von annotierten Alltagsbildern umfassen. Während dieses Vortrainings lernen die flachen Schichten des Netzwerks universelle, fundamentale Bildmerkmale wie Kanten, Farbverläufe und einfache Texturen. Die tieferen Schichten extrahieren zunehmend komplexere und klassenspezifische semantische Konzepte [35].

Ein wichtiger praktischer Aspekt des Transferlernens in der semantischen Segmentierung besteht darin, dass sich das Vortraining in der Regel ausschließlich auf den Encoder (Backbone) beschränkt. Dieser lernt generische und domänenübergreifend übertragbare Merkmalsrepräsentationen. Der Decoder hingegen ist an die konkrete Zielaufgabe gebunden, da seine Struktur und insbesondere die Anzahl der Ausgabekanäle direkt von den zu segmentierenden Klassen abhängen. Entsprechend wird der Segmentierungskopf in der Praxis meist zufällig initialisiert und ausschließlich auf dem Zieldatensatz trainiert. Diese Trennung zwischen einem vortrainierten Encoder und einem aufgabenspezifisch optimierten Decoder stellt ein zentrales Designprinzip moderner Segmentierungsarchitekturen dar. Abbildung 2.4 veranschaulicht diesen zweistufigen Prozess.

Wird dieser vortrainierte Backbone nun für die Zieldomäne im Low-Data-Regime adaptiert, spricht man von der sogenannten Feinabstimmung (engl. *Fine-Tuning*). Hierbei werden die initialen Gewichte aus dem Vortraining übernommen und mit einer reduzierten Lernrate auf dem kleinen Zieldatensatz weitertrainiert. Dies beschleunigt nicht nur die Konvergenz des Trainingsprozesses erheblich, sondern wirkt auch als starke Regularisierung gegen Überanpassung, da das Modell bereits in einer fundierten und generalisierbaren Merkmalsrepräsentation verankert ist.

In der Fernerkundung stößt das klassische, auf ImageNet basierende Vortraining jedoch auf spezifische Grenzen. Es existiert eine erhebliche Domänenlücke (engl. *Domain Gap*) zwischen gewöhnlichen Fotografien und Satelliten- oder Luftbildern.

Fernerkundungsdaten zeichnen sich durch eine strikte Vogelperspektive (Nadiransicht), abweichende Objektmaßstäbe, eine hohe Dichte an kleinen Objekten sowie die häufige Nutzung multispektraler Kanäle jenseits des sichtbaren RGB-Spektrums aus. Die auf ImageNet erlernten Merkmale lassen sich daher oft nur suboptimal auf die komplexen Strukturen der Fernerkundung übertragen.

Um diese Domänenlücke zu schließen, bedarf es idealerweise eines Vortrainings direkt auf Fernerkundungsdaten. Da hierfür jedoch, wie eingangs beschrieben, großflächige Annotationen fehlen, rücken zunehmend unüberwachte Paradigmen in den Fokus. Auch in diesen Ansätzen wird primär der Encoder vortrainiert, der anschließend als generischer Merkmalsextraktor für nachgelagerte Aufgaben dient.

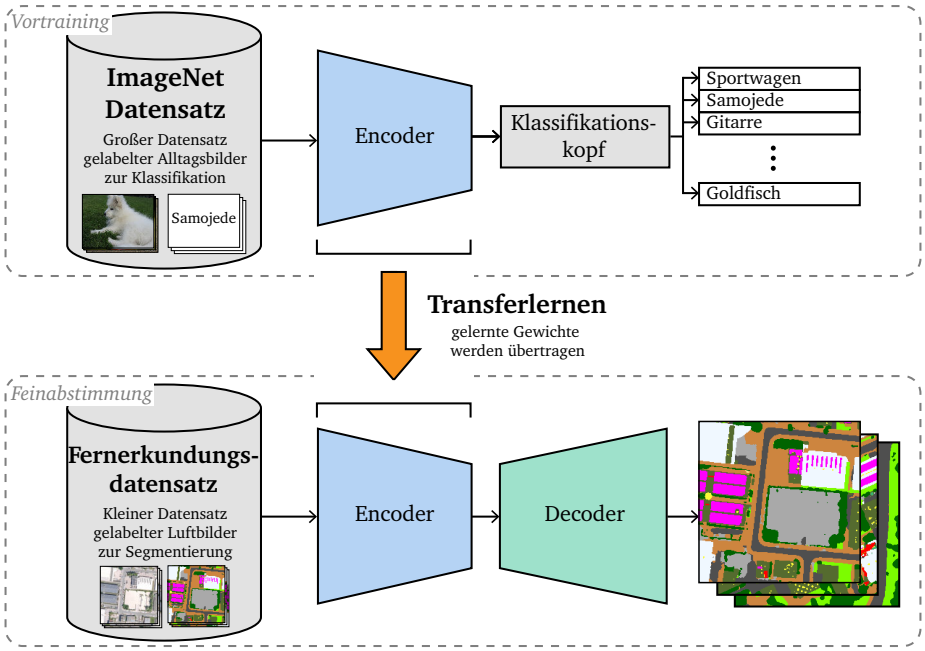


Abbildung 2.4 – Zweistufiges Transferlernen: Im Vortraining wird ein Encoder auf dem ImageNet-Datensatz [34] zur Bildklassifikation trainiert. Anschließend wird der Klassifikationskopf verworfen. Die gelernten Gewichte des Encoders werden per Transferlernen auf die Zieldomäne übertragen, wo ein neu initialisierter Decoder gemeinsam mit dem Encoder per Feinabstimmung auf einem beispielhaften, kleinen Fernerkundungsdatensatz zur semantischen Segmentierung trainiert wird. Eigene Darstellung in Anlehnung an [36].

2.3.4 Selbstüberwachtes Lernen

Eine der derzeit vielversprechendsten methodischen Realisierungen solcher Paradigmen ist das selbstüberwachte Lernen (engl. *Self-Supervised Learning*, SSL). Es bietet einen eleganten Lösungsansatz, um die massive Verfügbarkeit von Rohdaten in der Fernerkundung (z. B. nicht annotierte Luft- oder Satellitenbilder) für das Vortraining

nutzbar zu machen, ohne durch den Flaschenhals manuell erstellter Segmentierungsmasken limitiert zu sein. Anstatt auf externe Annotationen zurückzugreifen, generiert der Algorithmus die Zielvorgaben in einem vorgeschalteten Trainingsschritt automatisiert aus den Daten selbst. Dadurch kann dem Netzwerk ein grundlegendes, tiefgreifendes Verständnis für die visuellen und domänenspezifischen Strukturen der Erdbeobachtung beigebracht werden, bevor die finale Feinabstimmung auf den wenigen annotierten Zieldaten erfolgt.

In der aktuellen Bildverarbeitung haben sich mehrere Ansätze etabliert. Im Rahmen dieser Arbeit werden insbesondere zwei Paradigmen betrachtet:

- **Generatives Lernen durch Maskierung (Masked Autoencoders):** Dieses Paradigma, im Folgenden als Masked AE abgekürzt, basiert auf dem Prinzip des *Masked Image Modeling*. Hierbei werden signifikante Anteile des Eingabebildes maskiert, und das Netzwerk muss lernen, die fehlenden Bildbereiche bzw. deren Repräsentationen aus dem verbleibenden sichtbaren Kontext zu rekonstruieren [37]. Durch diese anspruchsvolle Voraufgabe ist das Modell gezwungen, ein tiefes semantisches Verständnis für die räumlichen und strukturellen Zusammenhänge in den Bilddaten zu entwickeln.
- **Wissensdestillation und Basismodelle (DINO):** Im Gegensatz zur pixelgenauen Rekonstruktion zielen Ansätze der DINO-Familie darauf ab, emergente semantische Repräsentationen zu erlernen [38]. Das Modell lernt hierbei in einer Lehrer-Schüler-Architektur (Selbstdistillation ohne Annotationen), für unterschiedlich augmentierte Ausschnitte desselben Bildes konsistente und hochdimensionale Merkmalsvektoren zu erzeugen. Moderne Iterationen wie DINOv3 [39], die auf massiven und diversifizierten Datensätzen trainiert wurden, fungieren heute als leistungsstarke Basismodelle (engl. *Foundation Models*). Sie liefern hochgradig generalisierbare und semantisch reiche Bildmerkmale und eignen sich daher ideal als vortrainierte Backbone-Architektur für anspruchsvolle Segmentierungsaufgaben.

2.4 Grundlagen der Evaluierung

Aufbauend auf den zuvor beleuchteten architektonischen und datengetriebenen Strategien, die ein stabiles Training von Segmentierungsmodellen auch unter limitierter Datenverfügbarkeit ermöglichen, stellt sich unweigerlich die Frage nach der objektiven Qualitätsmessung. Um die Leistungsfähigkeit der trainierten Netzwerke sowie den tatsächlichen Mehrwert vortrainierter Merkmalsextraktoren für die konkrete Zielaufgabe valide quantifizieren zu können, bedarf es einer strukturierten Evaluierung. Der folgende Abschnitt widmet sich daher der Evaluierungsmethodik und stellt die verwendeten Metriken zur Leistungsbeurteilung vor.

2.4.1 Metriken

Zur quantitativen Bewertung der trainierten Modelle werden standardisierte Metriken verwendet, die in der Literatur zur semantischen Segmentierung verbreitet sind [16], [40]. Die Evaluierung basiert dabei auf dem Vergleich zwischen der vorhergesagten Maske und der Referenzmaske.

Grundlage für alle Metriken ist die Konfusionsmatrix, die jeden Pixel einer von vier Kategorien zuordnet: Richtig-Positiv (engl. *True Positive*, TP) und Richtig-Negativ (engl. *True Negative*, TN) repräsentieren korrekte Klassifikationen, während Falsch-Positiv (engl. *False Positive*, FP) und Falsch-Negativ (engl. *False Negative*, FN) die jeweiligen Fehlerfälle beschreiben.

Da es sich bei der semantischen Segmentierung meist um ein Mehrklassenproblem handelt, wird jede Klasse für die Evaluierung einzeln betrachtet. Hierbei wird die Zielklasse als positiv definiert, während alle übrigen Klassen temporär als Hintergrund (negativ) zusammengefasst werden. Die meisten globalen Metriken ergeben sich anschließend durch die Mittelung der klassenweisen Ergebnisse. Eine Ausnahme bildet die Pixelgenauigkeit, die direkt über alle Pixel berechnet wird.

Pixelgenauigkeit

Die einfachste Metrik ist die Pixelgenauigkeit (engl. *Pixel Accuracy*, PA). Sie gibt den Anteil der korrekt klassifizierten Pixel an der Gesamtzahl aller Pixel an.

$$\mathrm{PA} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{2.2}$$

Obwohl diese Metrik intuitiv ist, ist sie bei unausgeglichenen Datensätzen oft irreführend. Besteht ein Bild beispielsweise zu 90 % aus Hintergrund, würde ein Modell, das alles pauschal als Hintergrund klassifiziert, eine Pixelgenauigkeit von 90 % erreichen, ohne jedoch irgendetwas tatsächlich erkannt zu haben.

Präzision und Sensitivität

Zwei grundlegende Kennzahlen, auf denen weitere Metriken aufbauen, sind die Präzision (engl. *Precision*) und die Sensitivität (engl. *Recall*). Die Präzision gibt an, welcher Anteil der als positiv vorhergesagten Pixel tatsächlich positiv ist, während die Sensitivität den Anteil der korrekt erkannten unter allen tatsächlich positiven Pixeln misst.

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \qquad \mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.3}$$

F1-Score (Dice-Koeffizient)

Der F1-Score, auch als *Dice-Koeffizient* bekannt, stellt das harmonische Mittel aus Präzision und Sensitivität dar. Er bietet eine balancierte Einschätzung, da er sowohl FP als auch FN gleichermaßen bestraft.

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} = \frac{2 \cdot \mathrm{TP}}{2 \cdot \mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{2.4}$$

Intersection over Union (Jaccard-Index)

Der *Jaccard-Index*, im Deep Learning meist als Intersection over Union (IoU) bezeichnet, gilt als die wichtigste Metrik für Segmentierungsaufgaben. Er beschreibt das Verhältnis der Schnittmenge zur Vereinigungsmenge von Vorhersage und Referenzmaske.

$$\mathrm{IoU} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN}} \tag{2.5}$$

Die IoU bestraft sowohl FP als auch FN stark. In der Regel wird der Mittelwert über alle Klassen als Mean Intersection over Union (mIoU) angegeben.

Im direkten Vergleich fällt die IoU bei identischer Segmentierungsqualität mathematisch bedingt tendenziell etwas geringer aus als der F1-Score. Dies liegt daran, dass beim F1-Score der Term der korrekten Vorhersagen (TP) doppelt gewichtet wird. Dennoch korrelieren beide Metriken stark miteinander und führen meist zu einer ähnlichen Rangfolge bei der Modellbewertung.

2.4.2 Datenaufteilung

Um die Leistungsfähigkeit und Generalisierungsfähigkeit eines maschinellen Lernmodells objektiv beurteilen zu können, ist eine strikte Trennung der zur Verfügung stehenden Daten unerlässlich. Das zentrale Ziel beim Training künstlicher neuronaler Netze ist es nicht, die Trainingsdaten auswendig zu lernen, sondern zugrunde liegende Muster zu erkennen, die sich auf neue, ungesehene Daten anwenden lassen.

In der Praxis erfolgt die Aufteilung des Gesamtdatensatzes in der Regel in drei disjunkte Teilmengen:

- **Trainingsdatensatz:** Diese Teilmenge wird ausschließlich für das aktive Training des Modells verwendet. Der Optimierungsalgorithmus berechnet basierend auf diesen Daten den Fehler und passt die internen Parameter des Netzwerkes iterativ an.

- **Validierungsdatensatz:** Dieser Datensatz wird während des Trainingszyklus regelmäßig zur Evaluierung der Modellleistung auf ungesehenen Daten genutzt und dient primär der Überwachung des Trainingsfortschritts sowie der

Vermeidung von Überanpassung. Zudem wird er zur Optimierung der Hyperparameter genutzt, also jener Konfigurationsvariablen, die vorab manuell festgelegt werden müssen. Er erlaubt es, frühzeitig zu erkennen, wenn das Modell beginnt, die Trainingsdaten überanzupassen, und ermöglicht somit Regularisierungstechniken wie das vorzeitige Abbrechen.

- **Testdatensatz:** Diese Teilmenge wird strikt zurückgehalten und erst ganz am Ende des Entwicklungsprozesses verwendet. Sie dient der finalen, unverfälschten Evaluierung des fertigen Modells und simuliert den Einsatz unter realen Bedingungen.

Eine feste Aufteilung (beispielsweise 70 % Training, 15 % Validierung, 15 % Test) ist bei großen Datensätzen statistisch belastbar und üblich. Im Low-Data-Regime führt so eine Aufteilung jedoch zu statistischen Unsicherheiten. Da die Validierungs- und Testdatensätze in diesem Fall extrem klein ausfallen, kann die gemessene Modellleistung stark davon abhängen, welche spezifischen Bilder zufällig in diese Teilmengen sortiert wurden (hohe Varianz).

In solchen Szenarien kommt oftmals die k-fache Kreuzvalidierung (engl. *k-Fold Cross-Validation*) zum Einsatz. Bei diesem Verfahren wird der Datensatz in k annähernd gleich große, disjunkte Teilmengen unterteilt. Das Modell wird dann iterativ k-mal von Grund auf neu trainiert. In jeder der k Iterationen fungiert genau eine Teilmenge als Validierungsdatensatz, während die verbleibenden $k-1$ Teilmengen das Trainingsset bilden. Die finale Leistungsbewertung des Modells ergibt sich dann aus dem Durchschnitt der Evaluierungsergebnisse aller k Iterationen.

Dieser Ansatz garantiert, dass jeder Datenpunkt exakt einmal zur Validierung herangezogen wird. Zwar vervielfacht die k-fache Kreuzvalidierung den Rechenaufwand linear um den Faktor k, sie liefert im Low-Data-Regime jedoch eine deutlich zuverlässigere und aussagekräftigere Metrik für die tatsächliche Generalisierungsfähigkeit.

Bei der Verarbeitung von Geodaten und Luftbildern reicht eine rein zufällige Aufteilung der Daten in Teilmengen oftmals nicht aus, um die tatsächliche Generalisierungsfähigkeit eines Modells zu bewerten. Aufgrund der sogenannten räumlichen Autokorrelation sind sich geographisch nah beieinander liegende Bildausschnitte sehr ähnlich. Eine zufällige Trennung benachbarter Bildkacheln in Trainings- und Validierungsdatensätze kann zu einem Informationsleck (engl. *Data Leakage*) führen: Das Modell neigt dazu, stark lokalisierte Merkmale (wie spezifische Architekturstile, Sensorrauschen oder lokale Beleuchtungsverhältnisse) auswendig zu lernen, was zu überoptimistischen Evaluierungsergebnissen führt.

Um die Übertragbarkeit eines Netzwerks auf ungesehene Regionen verlässlich zu messen, wird die räumliche Kreuzvalidierung eingesetzt. Hierbei werden Daten vor der Aufteilung in die k Teilmengen anhand räumlicher Einheiten (wie beispielsweise

Städte oder Nachbarschaften) gruppiert. In jeder Iteration wird das Modell auf einer Menge von Regionen trainiert und auf einer geographisch strikt getrennten, neuen Region validiert. Dieser Ansatz stellt sicher, dass die finalen Metriken die Leistung des Modells unter realistischen Bedingungen widerspiegeln.

2.4.3 Hyperparameter-Optimierung

Die Leistungsfähigkeit tiefer neuronaler Netze hängt maßgeblich von einer Vielzahl an Hyperparametern ab. Im Gegensatz zu den Modellparametern (wie den Gewichten innerhalb der Faltungsschichten), die das Netzwerk während des Trainings durch Backpropagation selbstständig lernt, werden Hyperparameter vor dem eigentlichen Trainingsprozess festgelegt [11]. Dazu gehören beispielsweise die Lernrate (engl. *Learning Rate*), die Batch-Größe (engl. *Batch Size*) oder Regularisierungsfaktoren.

Da das Training moderner Deep-Learning-Architekturen, insbesondere im Bereich der hochauflösenden Bildverarbeitung, extrem rechenintensiv ist, stoßen klassische Optimierungsverfahren schnell an ihre Grenzen. Wie Abbildung 2.5a veranschaulicht, testet die Rastersuche (engl. *Grid Search*) alle möglichen Kombinationen von Hyperparameter-Werten in einem vordefinierten Raster systematisch durch, was bei einer wachsenden Anzahl an Hyperparametern zu einer kombinatorischen Explosion der benötigten Rechenzeit führt. Die in Abbildung 2.5b abgebildete Zufallssuche (engl. *Random Search*) wählt Konfigurationen hingegen zufällig aus und findet oft schneller gute Lösungen als die Rastersuche [41]. Ein wesentlicher Grund hierfür ist die oft ungleiche Wichtigkeit der Hyperparameter, was Abbildung 2.5 anhand der Randverteilungen verdeutlicht: Da die Zufallssuche nicht an ein starres Raster gebunden ist, tastet sie die einflussreichen Dimensionen deutlich feinkörniger ab. Sie verschwendet jedoch ebenfalls enorme Ressourcen, da sie keine Erkenntnisse aus vergangenen, schlechten Trainingsläufen zieht.

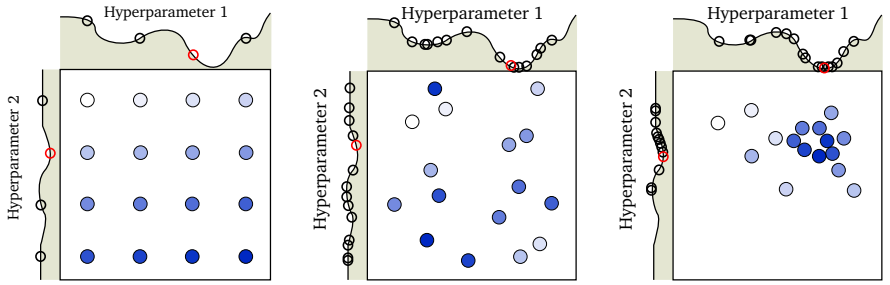
Um diese Ineffizienz zu überwinden, kommen fortschrittliche Suchstrategien zum Einsatz, die im Folgenden skizziert werden.

Bayessche Optimierung

Die Bayessche Optimierung behandelt das Optimierungsproblem als einen iterativen Lernprozess [42]. Anstatt Konfigurationen blind zu testen, nutzt die Bayessche Optimierung die Ergebnisse bereits evaluierter Hyperparameter, um ein probabilistisches Modell der zugrundeliegenden Zielfunktion aufzubauen. Dieses Modell wird als Surrogatmodell bezeichnet.

Der Ablauf folgt einem klaren Schema: Das Surrogatmodell schätzt ab, wie gut bestimmte Hyperparameter-Kombinationen voraussichtlich abschneiden werden, und bewertet gleichzeitig die Unsicherheit dieser Schätzung. Eine sogenannte Erfassungsfunktion (engl. *Acquisition Function*) entscheidet daraufhin, welcher Punkt

im Suchraum als Nächstes evaluiert wird. Dabei wägt der Algorithmus kontinuierlich zwischen der Erkundung unsicherer Bereiche (engl. *Exploration*) und der Ausbeutung vielversprechender Regionen (engl. *Exploitation*) ab. Abbildung 2.5c demonstriert dies anhand der progressiven Konzentration der Evaluierungen im vielversprechendsten Bereich des Suchraums. Ein großer Nachteil der klassischen Bayesschen Optimierung im Deep-Learning-Kontext ist jedoch, dass sie jede vorgeschlagene Konfiguration über den gesamten Trainingszyklus evaluieren muss, was bei großen Netzwerken den Prozess stark verlangsamt.



(a) Rastersuche **(b)** Zufallssuche **(c)** Bayessche Optimierung

Abbildung 2.5 – Vergleich verschiedener Strategien zur Exploration eines zweidimensionalen Hyperparameter-Raums. Der Farbverlauf der Punkte von Weiß nach Blau verdeutlicht die zeitliche Abfolge der Evaluierungen. Während die Rastersuche (a) feste Intervalle systematisch abtastet, wählt die Zufallssuche (b) zufällige Wertekombinationen. Die Bayessche Optimierung (c) konzentriert die Evaluierungen progressiv in den vielversprechendsten Bereichen. Die Kurven an den Rändern visualisieren die jeweilige eindimensionale Verlustfunktion mitsamt den projizierten Evaluierungspunkten, wobei das gefundene Minimum rot markiert ist. Daran zeigt sich exemplarisch, dass Hyperparameter 1 in diesem Szenario einen deutlicheren Einfluss auf das Ergebnis hat als Hyperparameter 2. Eigene Darstellung in Anlehnung an [43].

Hyperband

Um den Rechenaufwand drastisch zu reduzieren, fokussiert sich der Hyperband-Algorithmus auf die intelligente Zuteilung von Ressourcen (Budgets) [44]. Das Budget kann beispielsweise die Anzahl der Trainingsepochen oder die Menge der verwendeten Trainingsdaten definieren.

Hyperband basiert auf dem Prinzip der sukzessiven Halbierung (engl. *Successive Halving*). Der Algorithmus startet mit einer großen Anzahl zufällig generierter Konfigurationen und weist jeder zunächst nur ein sehr kleines Trainingsbudget zu (z. B. nur drei Epochen). Nach Ablauf dieses Budgets evaluiert der Algorithmus alle Konfigurationen und bricht das Training für einen definierten Anteil der schlechtesten Modelle (meist die untere Hälfte) sofort ab. Die verbleibenden Kandidaten erhalten

ein größeres Budget und trainieren weiter. Dieser Prozess wiederholt sich iterativ, bis nur noch die beste Konfiguration mit dem maximalen Budget übrig bleibt. Hyperband identifiziert schlechte Hyperparameter somit extrem früh und spart durch das frühe Abbrechen erheblich Rechenzeit, wählt die initialen Konfigurationen jedoch rein zufällig aus.

BOHB: Bayesian Optimization and Hyperband

Das BOHB-Verfahren vereint die Stärken beider Ansätze in einem robusten und hochgradig effizienten Algorithmus [45]. BOHB nutzt die dynamische Ressourcenverteilung und das frühzeitige Abbrechen unprofitabler Läufe von Hyperband, ersetzt aber dessen zufällige Wahl der Hyperparameter-Konfigurationen durch die intelligente, modellbasierte Suche der Bayesschen Optimierung.

Konkret trainiert BOHB während des Optimierungsprozesses ein mehrdimensionales probabilistisches Modell (meist basierend auf einer Kernel-Dichte-Schätzung) als Surrogatmodell. Dieses Modell lernt kontinuierlich, welche Hyperparameter-Kombinationen bei den bisherigen Evaluierungen gut oder schlecht abschnitten. Wenn Hyperband nun in einer neuen Iteration frische Konfigurationen anfordert, zieht BOHB diese nicht mehr blind aus dem Suchraum, sondern nutzt das aufgebaute Modell, um gezielt vielversprechende Kandidaten vorzuschlagen.

Durch dieses Zusammenspiel erreicht BOHB eine schnelle Konvergenz zu optimalen Hyperparametern bei gleichzeitiger Maximierung der Ressourceneffizienz. Damit eignet sich der Algorithmus besonders für die rechenintensiven Anforderungen der in dieser Arbeit durchgeführten semantischen Segmentierung.

2.5 Fernerkundung und Geodaten

Nachdem in den vorangegangenen Abschnitten die algorithmischen Grundlagen der maschinellen Bildverarbeitung sowie die Methodik zur Evaluierung tiefer neuronaler Netze etabliert wurden, widmet sich dieser Abschnitt der anwendungsspezifischen Domäne. Die Leistungsfähigkeit von Segmentierungsmodellen im raumbezogenen Kontext hängt maßgeblich von der Qualität, Auflösung und Struktur der zugrundeliegenden Geodaten ab. Im Folgenden werden daher die spezifischen Charakteristika der verwendeten Fernerkundungsdaten und amtlichen Geobasisdaten vorgestellt, welche das essenzielle Fundament für diese Arbeit bilden.

2.5.1 Urbane Nachhaltigkeitsindikatoren

Vor dem Hintergrund der fortschreitenden Urbanisierung und der nationalen Klimaschutzziele rückt die messbare Nachhaltigkeit von Quartieren und bestehenden

Gebäuden zunehmend in den Fokus der Stadtplanung. In Deutschland bildet das Gebäudeenergiegesetz den zentralen ordnungsrechtlichen Rahmen. Es definiert nicht nur energetische Mindeststandards für den Gebäudebestand, sondern forciert auch maßgeblich die Integration erneuerbarer Energien zur Dekarbonisierung der Wärme- und Stromversorgung [1].

Ergänzt wird diese gesetzliche Basis durch etablierte, ganzheitliche Zertifizierungssysteme wie das der DGNB. Das DGNB-System bewertet Quartiere und Gebäude anhand eines umfassenden Kriterienkatalogs, der neben der reinen Ökobilanzierung auch Aspekte des Mikroklimas, der Biodiversität und der Flächeninanspruchnahme quantifiziert [2]. Um diese normativen Vorgaben in der planerischen Praxis umzusetzen, bedarf es konkreter, räumlich messbarer Parameter, sogenannter Nachhaltigkeitsindikatoren. Für die datengetriebene urbane Analyse sind dabei insbesondere zwei Indikatoren von zentraler Bedeutung:

- **Versiegelungsgrad und Flächeninanspruchnahme:** Die zunehmende Bodenversiegelung durch Bebauung und Infrastruktur verstärkt den städtischen Wärmeinseleffekt und behindert die natürliche Versickerung von Niederschlagswasser. Die präzise Ermittlung versiegelter Flächen (oft in Anlehnung an die Grundflächenzahl) sowie deren Verhältnis zu Vegetationsflächen ist ein entscheidender Indikator für die klimatische Resilienz eines Quartiers.
- **Solares Potenzial:** Die flächendeckende Installation von Photovoltaikanlagen (PV) auf bereits existierenden Dachflächen stellt eine der wichtigsten Ressourcen der urbanen Energiewende dar. Die Erschließung dieses Potenzials ermöglicht die Erzeugung erneuerbarer Energie direkt am Verbrauchsort, ohne zusätzlichen Flächenverbrauch zu generieren.

Die präzise Erhebung dieser Indikatoren erfordert detaillierte Kenntnisse über die Gebäudegeometrie, Dachaufbauten, Verschattungseffekte sowie die exakte Verteilung von versiegelten und unversiegelten Flächen im direkten Gebäudeumfeld. Traditionelle, manuelle Erhebungsverfahren oder die Nutzung veralteter, rein zweidimensionaler Katasterdaten stoßen bei gesamtstädtischen Analysen an ihre Grenzen. Hier bietet die automatisierte semantische Auswertung von aktuellen, hochauflösenden Fernerkundungsdaten den entscheidenden Hebel, um Indikatoren wie den Versiegelungsfaktor oder das Solarpotenzial skalierbar, flächendeckend und objektiv abzuleiten.

2.5.2 KI in der Geodatenanalyse

Die Ermittlung der zuvor beschriebenen Nachhaltigkeitsindikatoren hat durch die Fusion von Geoinformationssystemen (GIS) und Methoden der KI einen Paradigmenwechsel erfahren [46].

Während klassische GIS-Analysen zur Merkmalsextraktion stark auf manuell definierten Heuristiken oder pixelbasierten Schwellenwerten (wie dem *Normalized Difference Vegetation Index*) basierten, ermöglichen moderne Deep-Learning-Verfahren eine datengetriebene und kontextbezogene Interpretation von Geodaten. CNNs und Vision Transformer sind in der Lage, komplexe semantische und räumliche Muster, wie den architektonischen Fußabdruck eines Gebäudes oder die Textur von Solarpaneelen, direkt aus hochauflösenden Orthophotos zu lernen [47].

Diese Automatisierung schließt die Lücke zwischen der bloßen Verfügbarkeit von Rohdaten und der Ableitung handlungsrelevanter, raumbezogener Informationen. Geo-KI ermöglicht es somit, mikroskopische Analysen, die historisch auf einzelne Quartiere beschränkt waren, recheneffizient auf ganze Stadtgebiete oder Landkreise zu skalieren, ohne dabei an geometrischer Detailtiefe zu verlieren.

Obwohl moderne Deep-Learning-Verfahren die semantische Auswertung von Geodaten signifikant automatisiert haben, unterliegt die Analyse optischer Luftbilder spezifischen, domäneninhärenten Störfaktoren. Neben dem in Abschnitt 2.3.1 beschriebenen Klassenungleichgewicht erschweren insbesondere physikalische und umweltbedingte Varianzen die konsistente Merkmalsextraktion durch künstliche neuronale Netze:

- **Verdeckung und Schattenwurf (Okklusion):** Im urbanen Raum kommt es häufig zu Verdeckungen relevanter Zielobjekte. Überhängende Baumkronen können Dachflächen, Schornsteine oder kleinformatige PV-Anlagen physisch für den optischen Sensor verdecken. Zusätzlich erzeugen aufragende Gebäude und dichte Vegetation je nach Sonnenstand (Azimut und Elevation) starke Schlagschatten. Diese Schattenbereiche weisen eine drastisch reduzierte radiometrische Intensität auf und verschlucken feine Texturmerkmale, wodurch das Netzwerk Objektkanten und Materialstrukturen oftmals nur noch unzureichend auflösen kann.

- **Saisonale Varianz und Phänologie:** Fernerkundungsdaten unterliegen einer starken zeitlichen Dynamik, da die Befliegungen zur Erstellung amtlicher Orthophotos zu unterschiedlichen Jahreszeiten stattfinden. Dies führt zu einer hohen Varianz: Während im laubfreien Zustand (Winter- und Frühjahrsbefliegungen) der Boden und tieferliegende Strukturen gut detektierbar sind, dominieren im belaubten Zustand (Sommerbefliegungen) weitreichende Vegetationskronen das Bild. Zudem variieren die Beleuchtungsverhältnisse und Schattenlängen durch den jahreszeitlich bedingten Sonnenstand erheblich.

Diese Herausforderungen verdeutlichen, dass für eine verlässliche urbane Analyse nicht nur leistungsstarke KI-Architekturen, sondern auch präzise und konsistente Datengrundlagen essenziell sind.

2.5.3 Datengrundlage

Die Datengrundlage umfasst zwei komplementäre Quellen: amtliche Geobasisdaten zur Repräsentation der realen Gegebenheiten sowie etablierte Referenzdatensätze für Training und Evaluierung der Segmentierungsmodelle.

Amtliche Geobasisdaten auf Länder- und Kommunalebene

Im Rahmen der Open-Data-Strategie stellt das Bayerische Landesamt für Digitalisierung, Breitband und Vermessung (LDBV) eine Vielzahl amtlicher georeferenzierter Geobasisdaten frei zur Verfügung. Diese Datensätze zeichnen sich durch hohe geometrische Genauigkeit, flächendeckende Verfügbarkeit sowie regelmäßige Aktualisierungszyklen aus. Im Folgenden werden die für diese Arbeit relevanten Datensätze charakterisiert:

Digitale Orthophotos (DOP) Geometrisch korrigierte und maßstabsgetreue Luftbilder der Erdoberfläche. In Bayern liegen diese standardmäßig als sogenannte *True-Orthophotos* mit einer Bodenauflösung (engl. *Ground Sampling Distance*, GSD) von 20 cm vor. Im Gegensatz zu klassischen Orthophotos werden diese unter Einbezug eines detaillierten Oberflächenmodells orthogonal entzerrt. Dadurch werden perspektivische Verzerrungen vollständig eliminiert, sodass eine exakte Nadiransicht ohne verdeckende Fassaden oder perspektivisch bedingte Abschattungen entsteht, wie in Abbildung 2.6 veranschaulicht wird. Dies ermöglicht eine präzise Überlagerung mit anderen Geodaten. Die bayerischen DOP20 umfassen in der Regel vier Spektralkanäle: Rot, Grün und Blau (RGB) sowie Nahinfrarot (NIR) [48].

Digitales Oberflächenmodell (DOM) Ein hochauflösender Rasterdatensatz, der die absolute Höhe der Erdoberfläche inklusive aller darauf befindlichen Objekte (wie Gebäude und Vegetation) repräsentiert. Das bayerische DOM20 besitzt eine GSD von 20 cm und ist damit räumlich konsistent mit den Orthophotos [49].

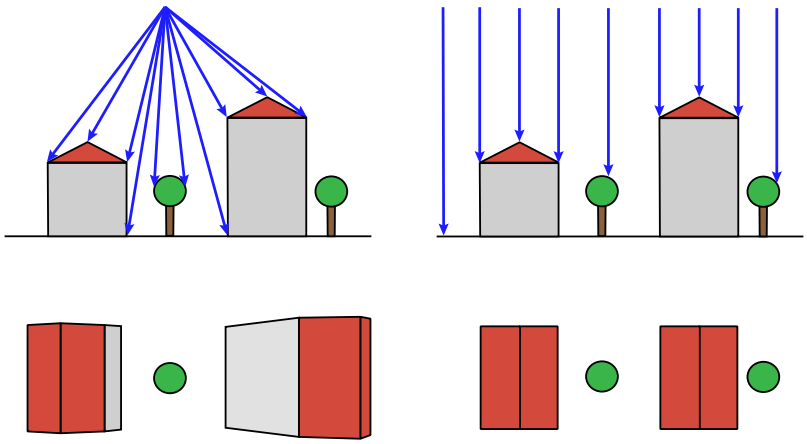
Digitales Geländemodell (DGM) Ein georeferenziertes Gitter von Höhenwerten, das die reale Erdoberfläche ohne Vegetation und Gebäude beschreibt. Das DGM1 weist eine GSD von 1 m auf und bildet die Grundlage für geomorphologische Analysen sowie die Ableitung von Höhenlinien [50].

Dreidimensionale Gebäudemodelle Vektorbasierte 3D-Repräsentationen von Gebäuden im Detaillierungsgrad 2 (engl. *Level of Detail 2*, LoD2). Im Gegensatz zu reinen Blockmodellen (LoD1) enthalten LoD2-Modelle neben der Gebäudehöhe auch generalisierte Dachstrukturen, bei denen beispielsweise Elemente wie Gauben, Schornsteine oder Balkone abstrahiert oder weggelassen werden [51].

Hausumringe 2D-Polygone, die die exakten Außengrenzen der Gebäudegrundrisse beschreiben. Sie werden direkt aus dem amtlichen Liegenschaftskataster abgeleitet und sind somit nicht durch perspektivische Verzerrungen oder Verdeckungen beeinflusst. Sie bilden eine verlässliche geometrische Grundlage für zweidimensionale Gebäudeanalysen [52].

Laserpunktdaten Hochpräzise 3D-Punktwolken, die mittels *Light Detection and Ranging* (LiDAR) durch flugzeuggestütztes Laserscanning erfasst werden. Die Punktwolken enthalten dreidimensionale Koordinaten reflektierender Oberflächenpunkte und bilden eine wichtige Grundlage für die Ableitung von Höhenmodellen und Oberflächenstrukturen [53].

Amtliche Katasterdaten Das Amtliche Liegenschaftskataster-Informationssystem führt geometrische Daten der Flurkarte und beschreibende Sachdaten in einem bundeseinheitlichen, objektstrukturierten Vektormodell zusammen. Es enthält hochdetaillierte Informationen zur „Tatsächlichen Nutzung“ (z. B. Wohnbaufläche, Ackerland, Wald) und dient als zentrale Grundlage für raumbezogene Verwaltungsprozesse und Geoinformationssysteme [54].



(a) Standard-Orthophoto **(b)** True-Orthophoto

Abbildung 2.6 – Prinzipbedingte Unterschiede in der Projektion, jeweils dargestellt durch den Strahlengang (oben) und das resultierende Abbild (unten): (a) Bei einem klassischen Orthophoto führt die Zentralprojektion bei aufragenden Gebäuden zu einem radialen Versatz (Umklappeffekt). Fassaden werden sichtbar und verdecken angrenzende Bereiche. (b) Das True-Orthophoto nutzt ein Oberflächenmodell für eine strikte Orthogonalprojektion, wodurch eine exakte, lagegenaue Nadiransicht der Dächer ohne Verdeckungen entsteht. Eigene Darstellung in Anlehnung an [55].

Datensätze zur semantischen Segmentierung

Während die zuvor beschriebenen amtlichen Geobasisdaten eine hochpräzise und verlässliche Repräsentation der Realität liefern, werden für das Training und die objektive Evaluierung von Methoden des maschinellen Lernens speziell aufbereitete Standard-Referenzdatensätze benötigt. Datensätze für die semantische Segmentierung bestehen typischerweise aus Paaren von Eingabebildern und korrespondierenden Referenzmasken, in denen jedem einzelnen Pixel eine semantische Klasse zugewiesen ist.

Für die Entwicklung und Evaluierung der in dieser Arbeit vorgestellten Modelle werden vier etablierte Datensätze verwendet, die sich insbesondere in ihrer GSD und Szenenkomplexität unterscheiden: die ISPRS-Referenzdatensätze (Vaihingen und Potsdam), OpenEarthMap sowie der FLAIR-Datensatz. Im Folgenden werden die spezifischen Eigenschaften dieser Datensätze kurz charakterisiert.

ISPRS 2D Semantic Labeling Challenge Die zwei Datensätze der *International Society for Photogrammetry and Remote Sensing* (ISPRS) gelten als Standard-Referenzdatensätze für die semantische Segmentierung hochauflösender Luftbilder. Beide Datensätze teilen sich das gleiche Klassenschema, bestehend aus sechs Klassen: *Impervious Surfaces* (versiegelte Flächen), *Building*, *Low Vegetation*, *Tree*, *Car* und *Clutter* (Hintergrund/Störobjekte).

- **Vaihingen:** Dieser Datensatz besteht aus True-Orthophotos mit einer GSD von ca. 9 cm. Besonderheit ist die spektrale Zusammensetzung: Es stehen drei Kanäle zur Verfügung: NIR, Rot und Grün. Durch den fehlenden Blaukanal und die Infrarot-Informationen eignet sich dieser Datensatz besonders zur Vegetationsanalyse [56].
- **Potsdam:** Im Gegensatz zu Vaihingen bietet dieser Datensatz eine noch feinere GSD von 5 cm und umfasst vier Kanäle (RGB-NIR). Er bildet eine historische Altstadt ab und stellt durch die dichte Bebauung und komplexe Strukturen hohe Anforderungen an die Kantenschärfe der Segmentierung [18].

OpenEarthMap Während die ISPRS-Datensätze lokal begrenzt sind, zielt *OpenEarthMap* (OEM) auf globale Diversität ab [57]. Der Datensatz umfasst 5 000 Luft- und Satellitenbilder aus 97 Regionen weltweit. Die GSD variiert dabei zwischen 25 und 50 cm. Diese geographische Varianz stellt eine Herausforderung für die Generalisierung dar, da sich Beleuchtung, Baustile und Vegetationstypen stark unterscheiden. OEM definiert 8 Landbedeckungsklassen: *Bareland* (Brachland), *Rangeland* (Grasland), *Developed Space* (erschlossene Fläche), *Road*, *Tree*, *Water*, *Agriculture Land* und *Building*.

FLAIR Der FLAIR-Datensatz (*French Land Cover from Aerospace ImageRy*) [6] ist einer der umfangreichsten Datensätze für die semantische Segmentierung von Luftbildern. Er deckt weite Teile des französischen Staatsgebiets ab (ca. 800 km^2) und bietet eine GSD von 20 cm, was exakt der Auflösung moderner Befliegungen in Deutschland (DOP20) entspricht. Der Datensatz zeichnet sich durch eine hohe Heterogenität in Bezug auf Aufnahmezeitpunkte und Landschaftstypen aus. Das Klassenschema ist mit 19 Klassen (bzw. 13 nach offizieller Klassenneuzuordnung) deutlich detaillierter als das der ISPRS-Referenzdatensätze und unterscheidet beispielsweise zwischen *Coniferous* (Nadelwald) und *Deciduous* (Laubwald) sowie verschiedenen Bebauungsarten.

Abbildung 2.7 verdeutlicht die Unterschiede in Auflösung und Szenenkomplexität anhand eines einheitlichen Bildausschnitts von 30 × 30 m.

(a) Vaihingen **(b)** Potsdam **(c)** OpenEarthMap **(d)** FLAIR

Abbildung 2.7 – Visueller Vergleich der verwendeten Bilddatensätze. Alle Ausschnitte repräsentieren denselben räumlichen Bereich von **30 × 30 m**. (a) und (b) zeigen die hohe Auflösung der ISPRS-Luftbilder, wobei (a) die Kanäle NIR-R-G verwendet. (c) verdeutlicht die gröbere Auflösung der OpenEarthMap-Bilder. FLAIR (d) entspricht der Zielauflösung dieser Arbeit. Bildquellen: [6], [18], [56], [57].

2.5.4 Methoden der 3D-Punktwolkenverarbeitung

Die bisher betrachteten Architekturen und Datensätze beziehen sich ausschließlich auf die zweidimensionale, pixelbasierte Auswertung optischer Bilddaten. Für die in dieser Arbeit angestrebte Solarpotenzialanalyse ist diese flächenhafte Information allein jedoch nicht ausreichend, da der nutzbare Ertrag maßgeblich von der dreidimensionalen Dachgeometrie, insbesondere von Neigung und Ausrichtung der einzelnen Flächen, bestimmt wird. Die hierfür notwendige Höheninformation liefern unter anderem die zuvor beschriebenen Laserpunktdaten in Form unstrukturierter 3D-Punktwolken. Deren Verarbeitung erfordert jedoch spezialisierte Algorithmen, die mit großen, unstrukturierten Datenmengen und inhärentem Messrauschen umgehen können. Um aus fehlerbehafteten 3D-Punktwolken primitive geometrische

Formen oder topologische Strukturen abzuleiten, haben sich in der Geoinformatik verschiedene Kernalgorithmen etabliert.

Für die Schätzung mathematischer Modelle in verrauschten Daten ist der RANSAC-Algorithmus (*Random Sample Consensus*) ein Standardverfahren [58]. Der Algorithmus iteriert über zufällige Teilmengen der Daten, um die Parameter eines Modells (beispielsweise einer Ebene) zu berechnen. Raumpunkte, die innerhalb einer vordefinierten Distanztoleranz zu diesem Modell liegen, werden als konsistente Datenpunkte (engl. *Inlier*) klassifiziert. Durch die Maximierung der Inlier-Anzahl ist RANSAC besonders robust gegenüber Ausreißern (engl. *Outliers*) und ermöglicht eine präzise Anpassung theoretischer Modelle an empirische Messungen.

Zur Identifikation lokaler Anomalien oder zur Segmentierung von Objektgruppen finden dichtebasierte Clustering-Verfahren Anwendung. Der DBSCAN-Algorithmus (*Density-Based Spatial Clustering of Applications with Noise*) gruppiert Punkte ausschließlich basierend auf ihrer lokalen räumlichen Dichte [59]. Die Clusterbildung wird durch zwei Parameter gesteuert: eine maximale Suchdistanz (ϵ) und eine minimale Punktzahl (*MinPts*) innerhalb dieses Radius. Der wesentliche Vorteil für komplexe Punktwolken liegt darin, dass DBSCAN weder die Anzahl der Cluster a priori benötigt, noch an bestimmte Clusterformen gebunden ist. Punkte in Regionen zu geringer Dichte isoliert der Algorithmus zuverlässig und deklariert sie als Rauschen.

Um bei der Anwendung derartiger Algorithmen effiziente topologische Nachbarschaftsanalysen zu gewährleisten, werden häufig KD-Bäume (engl. *k-dimensional trees*) eingesetzt [60]. Diese binären Suchbäume partitionieren den mehrdimensionalen Raum und beschleunigen Distanz- und Nachbarschaftsabfragen auf logarithmische Zeitkomplexität.

Zur Bestimmung der äußeren Begrenzung einer Punktmenge werden sogenannte Alpha-Shapes herangezogen [61]. Im Gegensatz zu einer einfachen konvexen Hülle, welche die äußersten Punkte lediglich konvex umschließt, erlaubt ein Alpha-Shape durch den namensgebenden Parameter α die geometrische Abbildung konkaver Konturen. Dadurch lassen sich komplexe, asymmetrische Ränder aus unstrukturierten Punktmengen detailgetreu nachbilden und vektorisieren.

2.5.5 Modellierung solarer Einstrahlung und Verschattung

Die vorgestellten Algorithmen ermöglichen die geometrische und topologische Auswertung dreidimensionaler Höhendaten. Diese räumliche Information bildet zugleich die Grundlage für die Modellierung der solaren Einstrahlung. Die präzise Quantifizierung des solaren Potenzials im urbanen Raum erfordert die Berücksichtigung lokaler Verschattungseffekte durch Topografie, Vegetation oder benachbarte Bebauung. Eine gängige Methode zur Schattenmodellierung ist die Berechnung sogenannter Hori-

zontwinkelkarten (engl. *Horizon Angle Maps*). Hierbei tasten Raytracing-Algorithmen ein DOM ab, indem sie von einem Zielpunkt aus Suchstrahlen in diskrete Azimutrichtungen aussenden. Entlang jedes Strahls wird der maximale Elevationswinkel (Horizontwinkel) ermittelt, ab dem ein physisches Hindernis den Lichteinfall blockiert [62], wie Abbildung 2.8 schematisch veranschaulicht. Durch den Abgleich der stündlichen Sonnenposition mit diesem richtungsspezifischen Höhenprofil lässt sich geometrisch bestimmen, ob ein Dachpunkt direkt bestrahlt wird oder im Schatten liegt.

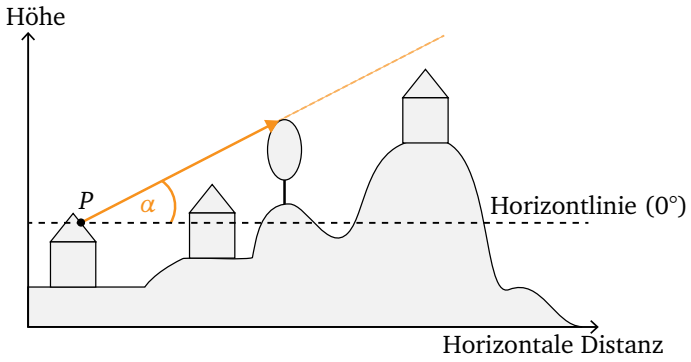


Abbildung 2.8 – Schematische Darstellung des Horizontwinkels α für einen spezifischen Azimut. Der Winkel wird am Referenzpunkt P als maximaler Elevationswinkel zwischen der horizontalen Referenzlinie und der Sichtlinie zum höchsten sichtbaren Hindernis bestimmt. Eigene Darstellung in Anlehnung an [62].

Die auf eine geneigte Dachfläche treffende globale Einstrahlung setzt sich physikalisch aus direkter Strahlung, diffuser Himmelsstrahlung und Bodenreflexion zusammen. Um die Umrechnung der horizontalen Messwerte auf geneigte Flächen realitätsnah abzubilden, wird häufig das Perez-Modell [63] angewendet. Als anisotropes Himmelsmodell berücksichtigt es, im Gegensatz zu einfacheren isotropen Ansätzen, die physikalische Realität einer ungleichmäßigen Himmelsleuchtdichte, insbesondere die erhöhte Strahlungsintensität in unmittelbarer Sonnennähe (Zirkumsolarstrahlung) sowie am Horizont (Horizontaufhellung). Liegt ein Modulpunkt im Schatten, wird die direkte Strahlungskomponente blockiert und der nutzbare Ertrag aus der verbleibenden, durch den *Sky-View*-Faktor (Himmelsansichtsfaktor) reduzierten, diffusen Strahlung abgeleitet.

Zur Einordnung und Validierung derartiger hochaufgelöster physikalischer Simulations-Pipelines dienen in der Praxis normative Referenzwerte. Etablierte Regelwerke wie die DIN V 18599 definieren standardisierte Verfahren für die energetische Bewertung von Gebäuden. Sie bieten einen Referenzrahmen zur Ertragsabschätzung von PV-Systemen, der auf tabellierten klimatischen Zonen, festen Systemnutzungsgraden und standardisierter altersbedingter Moduldegradation basiert [64], [65].

Gemäß DIN V 18599 wird die monatliche Netto-Stromproduktion aus dem PV-System $Q_{\mathrm{f,prod,PV,i}}$ nach folgender Gleichung berechnet:

$$Q_{\mathrm{f,prod,PV,i}} = \frac{E_{\mathrm{sol}} \cdot P_{\mathrm{pk,m}} \cdot f_{\mathrm{perf}}}{I_{\mathrm{ref}}} \tag{2.6}$$

Dabei beschreibt E_{sol} die monatliche solare Bestrahlungsenergie auf das PV-System in $\mathrm{kW\,h/m^2}$, $P_{\mathrm{pk,m}}$ die mittlere Peakleistung in kW über einen Zeitraum von 25 Jahren (unter Norm-Prüfbedingungen und Berücksichtigung der Degradation), f_{perf} den Systemleistungsfaktor (Standardwerte gemäß Norm) und I_{ref} die Referenzsolarbestrahlungsstärke ($1\,\mathrm{kW/m^2}$).

2.6 Methoden der Datenannotation

Für das Training von überwachten Lernverfahren (engl. *Supervised Learning*) sind große Mengen an annotierten Daten notwendig. In der Praxis stellt die Beschaffung dieser Referenzdaten oft den größten Engpass bei der Entwicklung neuer Modelle dar [10]. Dabei beeinflusst die Qualität der Annotationen die Leistung des späteren Modells direkt, da Ungenauigkeiten oder Fehler, auch als Annotationsrauschen (engl. *Label Noise*) bezeichnet, vom Modell als fehlerhafte Muster gelernt werden können [66].

Grundsätzlich lassen sich bei der Erstellung von Referenzmasken drei verschiedene Herangehensweisen unterscheiden:

1. **Manuelle Annotation:** Bei diesem Ansatz zeichnen Menschen die Objektgrenzen händisch nach, entweder pixelgenau oder durch das Setzen von Polygonen. Dieses Vorgehen liefert sehr präzise Annotationen, ist jedoch mit einem hohen zeitlichen Aufwand verbunden. Die Autoren des *Cityscapes*-Datensatzes, eines Referenzdatensatzes für straßenbasierte Szenen, berichten, dass die feingranulare Annotation einer einzigen Straßenszene durchschnittlich 1,5 Stunden in Anspruch nahm [67]. Bei Fernerkundungsdaten ist dieser Aufwand oft noch höher: Für den *OpenEarthMap*-Datensatz geben die Autoren an, dass die Annotation einer Kachel (1024×1024 Pixel) durchschnittlich sogar 2,5 Stunden erforderte [57].

2. **Crowdsourcing:** Beim Crowdsourcing werden Annotationsaufgaben auf eine große Anzahl von Laien-Annotatoren verteilt. Während dies die kostengünstige und schnelle Annotation großer Datensätze ermöglicht, erfordert die Qualitätssicherung besondere Aufmerksamkeit [68]. Insbesondere bei komplexen oder domänenspezifischen Daten fehlt Laienannotatoren häufig das notwendige Fachwissen und der Kontext, um ohne explizite Qualitätskontrollen expertentaugliche Annotationen zu erzeugen [69].

3. **KI-gestützte Annotation (*Human-in-the-loop*):** Dieser hybride Ansatz nutzt ein Modell zur Generierung von Vorannotationen, die anschließend von menschlichen Annotatoren nur noch überprüft und korrigiert werden müssen. Die korrigierten Annotationen können genutzt werden, um das Modell iterativ weiter zu verbessern, wodurch sich die Qualität der Vorannotationen erhöht [70]. Dieses Vorgehen kann den Annotationsaufwand deutlich reduzieren, funktioniert aber nur, wenn bereits ein halbwegs passendes Basismodell existiert. Zudem besteht die Gefahr, dass Annotatoren dem Modell zu sehr vertrauen und vorhandene Fehler einfach übersehen, was eine systematische Voreingenommenheit (Bias) in den Datensatz induziert [71].

Neben der Datenkomplexität beeinflusst auch die Wahl des Annotationswerkzeugs die Qualitätssicherung maßgeblich. Während lokale Tools wie *LabelMe* [72] für Einzelnutzer konzipiert sind, bieten serverbasierte Lösungen wie *CVAT (Computer Vision Annotation Tool)* [73] Funktionen für kollaboratives Arbeiten. Dies ermöglicht die Verteilung von Annotationsaufgaben im Team sowie integrierte Kontrollprozesse zur Validierung der erstellten Annotationen. Abbildung 2.9 zeigt die Benutzeroberfläche von CVAT am Beispiel der in dieser Arbeit durchgeführten Annotation.

Das in dieser Arbeit eingesetzte Verfahren wird in Abschnitt 3.1.2 detailliert beschrieben.

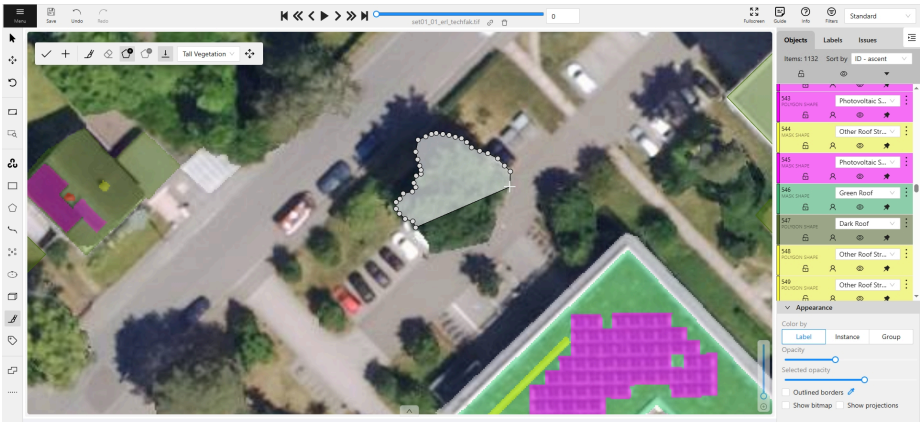


Abbildung 2.9 – Benutzeroberfläche des Annotationstools CVAT während des aktiven Annotationsprozesses. Die Bildschirmaufnahme zeigt die Erfassung eines Baumes mithilfe des Polygon-Werkzeugs.

2.7 Verwandte Arbeiten

Die automatisierte semantische Auswertung von Fernerkundungsdaten bildet die Schnittstelle zwischen der modernen maschinellen Bildverarbeitung und der angewandten Geoinformatik. Um die methodische Einordnung und den wissenschaftlichen Beitrag der vorliegenden Arbeit zu fundieren, beleuchtet dieser Abschnitt den aktuellen Stand der Forschung. Zunächst werden die jüngsten architektonischen Entwicklungen sowie Strategien zur Bewältigung von Datenknappheit diskutiert (Abschnitt 2.7.1). Anschließend wird evaluiert, wie diese Methoden derzeit zur Erhebung urbaner Nachhaltigkeitsindikatoren eingesetzt werden (Abschnitt 2.7.2), bevor die Arbeit in den Kontext nationaler Referenzprojekte eingeordnet wird (Abschnitt 2.7.3).

2.7.1 Methodische Entwicklungen in der Fernerkundung

Die methodische Entwicklung folgt dabei einem Spannungsfeld zwischen immer leistungsfähigeren Architekturen und der Notwendigkeit, mit begrenzten Trainingsdaten auszukommen.

Evolution der Architekturen zur semantischen Segmentierung

In den vergangenen Jahren hat sich in der maschinellen Bildverarbeitung von Fernerkundungsdaten ein Paradigmenwechsel von rein faltungsbasierten Netzwerken hin zu Transformer-basierten und hybriden Architekturen vollzogen. Während rein faltungsbasierte Netzwerke bei sehr hochauflösenden (engl. *Very High Resolution*, VHR) Luftbildern oft an ihre Grenzen stoßen, wenn es darum geht, weitreichende globale semantische Zusammenhänge zu erfassen, überwinden moderne Ansätze diese Limitierung, indem sie das bewährte Encoder-Decoder-Paradigma um explizite Aufmerksamkeitsmechanismen erweitern.

Einen signifikanten Fortschritt demonstrieren Wang, Li, Zhang u. a. [5] mit der Einführung des *UNetFormers*. Durch die Kombination eines ResNet-basierten Encoders mit einem Transformer-Decoder, der globale und lokale Kontexte fusioniert, erreicht das Modell auf Referenzdatensätzen wie ISPRS Vaihingen eine neue Bestleistung. Hanyu, Yamazaki, Tran u. a. [74] erweitern diesen hybriden Ansatz mit dem *AerialFormer*, welcher einen hierarchischen Transformer-Encoder mit einem mehrfach dilatierten CNN-Decoder verknüpft. Der Ansatz zeigt insbesondere bei Datensätzen mit extremem Klassenungleichgewicht und komplexen städtischen Hintergründen eine konstantere Segmentierungsleistung. Parallel dazu etabliert sich das *Mask2Former*-Framework [29], welches das Problem nicht als Pixelklassifikation, sondern als direkte Maskenvorhersage formuliert. Wie aktuelle Studien zeigen, ermöglicht eine Adaption dieses Frameworks mit Swin-Transformer-Backbone

die großflächige und hochpräzise Extraktion von Gebäudegrundrissen aus VHR-Satellitenbildern, da es stark variierende Gebäudeformen effektiv auflösen kann [75].

Trotz dieser architektonischen Durchbrüche fokussieren sich die meisten Arbeiten primär auf Standard-Referenzdatensätze oder klassische Gebäudeextraktion. Die Autoren des FLAIR-Datensatzes [6] evaluieren die Übertragbarkeit auf heterogene, landesweite Datensätze und demonstrieren, dass KI-Modelle konsistente Landbedeckungskarten über vielfältige Ökosysteme hinweg generieren können. Eine gemeinsame Limitierung dieser Arbeiten ist jedoch die implizite Annahme großer Mengen an manuell erstellten Trainingsdaten. Das Verhalten dieser modernen Architekturen im extremen Low-Data-Regime bleibt weitgehend unerforscht.

Überwindung des Low-Data-Regimes: Geo-Basismodelle und SSL

Um der chronischen Datenknappheit in der Erdbeobachtung zu begegnen, verschiebt sich der Fokus zunehmend auf das selbstüberwachte Lernen und den Einsatz von Basismodellen. Diese Modelle zielen darauf ab, die Domänenlücke zwischen gewöhnlichen Fotografien (ImageNet) und der Vogelperspektive optischer Sensoren durch unüberwachtes Vortraining auf massiven Geodatenarchiven zu überwinden.

Mit *SatMAE* [76] wurde das Paradigma der Masked AE erfolgreich auf multitemporale und multispektrale Satellitendaten übertragen. Die Studie belegt, dass sich durch SSL erzeugte Repräsentationen bei nachgelagerten Segmentierungsaufgaben durch eine höhere Dateneffizienz auszeichnen. Einen Meilenstein in dieser Entwicklung markiert *Prithvi*, ein von NASA und IBM entwickeltes Basismodell, das auf über einem Terabyte harmonisierter Landsat- und Sentinel-2-Daten vortrainiert wurde [77]. Die Evaluierung von Prithvi zeigt, dass die Modellleistung selbst bei einer drastischen Reduktion der feinabgestimmten Annotationsdaten weitgehend erhalten bleibt, ein essenzieller Vorteil für Low-Data-Anwendungen.

Eine systematische Forschungslücke in der Literatur zu Geo-Basismodellen war bis zuletzt deren räumliche Auflösung. Modelle wie Prithvi oder SatMAE sind primär auf Satellitendaten mittlerer Auflösung (10 bis 30 m) optimiert. Einen Durchbruch zur Schließung dieser Lücke markiert die jüngste Generation selbstüberwachter Basismodelle, insbesondere *DINOv3* [39]. Neben einem universellen Modell, das auf rund 1,7 Milliarden Alltagsbildern aus dem Internet trainiert wurde (LVD-1689M), umfasst DINOv3 ein spezialisiertes Fernerkundungsmodell (SAT-493M). Dieses wurde auf knapp 500 Millionen optischen Maxar-Satellitenbildern mit einer Bodenauflösung von 0,6 m trainiert. Inwiefern dieses domänenspezifische Modell trotz der verbleibenden Auflösungsdifferenz zu den hier verwendeten 20 cm-Luftbildern einen Vorteil gegenüber dem massiven, generischen Web-Modell bietet, wird im Rahmen dieser Arbeit vergleichend evaluiert.

Fortgeschrittene Adaptionsmethoden: Meta-Lernen und Testzeit-Diffusion

Neben den etablierten Ansätzen des Transferlernens und den Basismodellen rücken zunehmend Methoden in den Fokus, die das Paradigma des Lernens unter Datenknappheit grundlegend erweitern. Ein prominenter Ansatz hierfür ist das Meta-Lernen (Lernen zu lernen). Im Gegensatz zur klassischen Feinabstimmung vortrainierter Gewichte optimieren Meta-Lern-Algorithmen, wie beispielsweise das modellunabhängige Meta-Lernen (engl. *Model-Agnostic Meta-Learning*), die Initialisierung eines Netzwerks explizit darauf, sich mit einer minimalen Anzahl zukünftiger Lernschritte an neue Zieldomänen anzupassen [78], [79]. Während solche Ansätze in theoretischen Szenarien mit extrem wenigen Trainingsbeispielen beachtliche Erfolge erzielen, ist ihr Einsatz bei hochauflösenden Fernerkundungsdaten und großen Transformer-Architekturen in der Praxis oft mit einem prohibitiv hohen Speicherbedarf verbunden, da für die Optimierung Ableitungen zweiter Ordnung berechnet werden müssen.

Eine weitere hochaktuelle Entwicklung zur Steigerung der Vorhersagestabilität bildet die Integration generativer Diffusionsmodelle. Obwohl diese ursprünglich zur Bildsynthese entwickelt wurden, eignen sie sich hervorragend als datengetriebenes Vorwissen, um Segmentierungsmasken nachträglich zu verfeinern [80]. Im Rahmen der sogenannten Testzeit-Diffusion (engl. *Test-Time Diffusion*) können solche Modelle während der Inferenz eingesetzt werden, um Rauschen in den Vorhersagen zu glätten und Domänenverschiebungen auszugleichen, ohne das Basismodell neu trainieren zu müssen [81]. Da dieser iterative Entrauschungsprozess jedoch pro Bildausschnitt zahlreiche Vorwärtsdurchläufe erfordert, limitiert die enorme Rechenintensität derzeit die Skalierbarkeit für flächendeckende, stadtweite Analysen. Für den operativen Einsatz auf großen Geodaten bilden daher speichereffiziente Strategien, wie die Datenaugmentierung zur Testzeit sowie morphologische Operationen, oftmals den pragmatischen Kompromiss zwischen Vorhersagequalität und Rechenaufwand.

2.7.2 Ableitung urbaner Nachhaltigkeitsindikatoren

Die Ableitung konkreter Nachhaltigkeitsindikatoren aus Fernerkundungsdaten konzentriert sich in der kommunalen Praxis auf wenige, besonders aussagekräftige Kenngrößen. Zwei davon stehen im Mittelpunkt der folgenden Betrachtung: der Versiegelungsgrad als Indikator für die klimatische Resilienz urbaner Räume und das solare Potenzial städtischer Dachflächen als Grundlage der Energiewende.

Erfassung von Versiegelungsgraden und urbaner Landbedeckung

Die präzise Kartierung versiegelter Flächen ist ein zentraler Indikator für die klimatische Resilienz von Quartieren. Auf europäischer Ebene hat sich hierfür der Copernicus

Land Monitoring Service (CLMS) mit dem *High Resolution Layer Imperviousness* als operationaler Standard etabliert. Dabei wird auf Basis von Sentinel-2-Zeitreihen und Regressionsmodellen ein kontinuierlicher Versiegelungsgrad (0–100 %) in einem 10 m-Raster abgeleitet [82], [83].

Eine umfassende Übersichtsstudie von Mahyoub, Rhinane, Mansour u. a. [84] zeigt jedoch auf, dass in der aktuellen Forschung Deep-Learning-Ansätze solche traditionellen, index- oder regressionsbasierten Methoden (wie NDVI-Schwellenwerte) zunehmend ablösen. Klassische Ansätze scheitern oftmals an Verschattungen oder spektral ähnlichen Materialien (z. B. dunklen Dächern und asphaltierten Straßen). Tiefe neuronale Netze hingegen beziehen den räumlichen Kontext in die Klassifikationsentscheidung mit ein.

Für großflächige, landesweite Erhebungen dominieren daher mittlerweile Ansätze, die multimodale oder multiskalige Daten fusionieren. So nutzen Li, Zhang, Sun u. a. [85] ein Terrain-gesteuertes Fusionsnetzwerk für Sentinel-Daten, um topographische Einflüsse auszugleichen. Frameworks wie *CISC* verschneiden hochauflösende 2 m-Satellitenbilder mit 30 m-Daten, um landesweite Versiegelungskarten für China zu generieren [86].

Die Limitierung sowohl der operationalen Dienste wie Copernicus als auch der aktuellen Deep-Learning-Arbeiten für den kommunalen Einsatz liegt jedoch in ihrer Granularität und ihrem Endzweck. Meist verbleibt die räumliche Auflösung im groben Satellitenbereich (10 m bis 30 m) oder die Versiegelung wird lediglich als diskrete Landbedeckungsklasse abgebildet. Eine direkte, gebäudescharfe und parzellengenaue Ableitung kontinuierlicher Versiegelungsfaktoren, wie sie für moderne Quartierszertifizierungen (z. B. DGNB) auf Blockebene gefordert wird, findet kaum statt.

KI-gestützte Solarpotenzialanalyse und PV-Detektion

Die automatisierte Erfassung des solaren Potenzials hat sich von einer rein zweidimensionalen Bildklassifikation hin zu integrativen 3D-Ansätzen weiterentwickelt. Frühe Frameworks wie das von der Stanford University entwickelte *DeepSolar* [7] demonstrierten die Machbarkeit der flächendeckenden 2D-Detektion, scheiterten jedoch an der Erfassung geometrischer Anlagenparameter.

Den aktuellen Stand der Technik zur Erfassung bestehender PV-Anlagen markiert der *3D-PV-Locator* [8]. Der Ansatz kombiniert Deep-Learning-basierte 2D-Masken mit amtlichen 3D-Gebäudemodellen, um Azimut und Neigungswinkel großflächig zu extrahieren. Dies reduziert die Fehlerquote bei der Kapazitätsschätzung im Vergleich zu reinen 2D-Verfahren drastisch.

Geht es hingegen um die Abschätzung potenzieller Erträge noch unbelegter Dachflächen, erfordert dies die Kopplung von Segmentierung und physikalischer

Modellierung. Zhong, Zhang, Chen u. a. [87] präsentieren einen Workflow, der CNN-basierte Dachsegmentierungen mit LiDAR-Daten verschneidet, um stadtweite Einstrahlungssimulationen durchzuführen. Dass hierbei insbesondere die Detailliertheit der Segmentierung ausschlaggebend ist, belegen Li, Wang und Xu [88]: Durch die explizite Berücksichtigung von Dachaufbauten (wie Gauben) auf Quartiersebene kann das tatsächlich installierbare PV-Potenzial signifikant genauer abgeleitet werden als durch pauschale Flächenberechnungen.

Trotz dieser vielversprechenden Pipelines basieren die eingesetzten Modelle primär auf ressourcenintensiv trainierten Standard-CNNs. Die Nutzung von Basismodellen zur Reduktion des Annotationsaufwands wird im Kontext der geometrisch komplexen Dachsegmentierung bisher nicht ausgeschöpft.

2.7.3 Systemische Ansätze und nationale Referenzprojekte

Die Notwendigkeit einer automatisierten Informationsgewinnung zum Gebäudebestand wird auch von offizieller Seite betont. So unterstreicht das Deutsche Zentrum für Luft- und Raumfahrt in seiner Konzeptstudie *G-DAT DE* die Dringlichkeit, Fernerkundungsdaten mittels moderner maschineller Lernverfahren auszuwerten, um valide Datengrundlagen für den Klimaschutz und die Stadtplanung zu schaffen [89].

Ein prominentes Referenzprojekt im deutschen Kontext ist ETHOS.BUILDA (Forschungszentrum Jülich), welches eine umfassende Datenbank für den deutschen Gebäudebestand bereitstellt [4]. Solche Systeme nutzen maschinelle Lernverfahren primär zur Vorhersage statistischer Metadaten auf nationaler Ebene (makroskopische Bewertung). Im Vergleich dazu besteht derzeit eine Diskrepanz zwischen der bloßen Verfügbarkeit hochauflösender Basisdaten (wie den bayerischen DOP20) und deren systematischer, semantischer Erschließung für konkrete, gebäudescharfe Planungsprozesse.

2.7.4 Forschungsanschluss und methodische Einordnung

Aus der Betrachtung der verwandten Arbeiten ergeben sich deutliche Forschungslücken. Bisherige KI-gestützte Geodatenprojekte weisen typischerweise mindestens eine der folgenden drei Limitierungen auf: Erstens agieren Basismodelle bisher vorrangig auf mittelauflösenden Satellitendaten (Landsat/Sentinel) und wurden kaum für den operativen Einsatz auf 20 cm-Luftbildern operationalisiert. Zweitens verbleiben Modelle oft auf der reinen Evaluierungsebene (Pixelklassifikation auf Referenzdatensätzen), ohne die Ergebnisse in physikalische 3D-Berechnungsverfahren zu überführen. Drittens wird das Low-Data-Regime in städtebaulichen Anwendungsszenarien methodisch vernachlässigt.

Diese Arbeit schließt diese Lücken durch ein ganzheitliches Lösungskonzept, das die folgenden drei Kernaspekte integriert:

1. **Einsatz von Geo-Basismodellen:** Um die praktische Skalierbarkeit für Kommunen zu demonstrieren, wird evaluiert, inwiefern sich moderne Architekturen und das Prinzip des selbstüberwachten Lernens adaptieren lassen, um auch mit minimalem manuellem Annotationsaufwand expertentaugliche Vorhersagen zu generieren.

2. **Ganzheitliche Prozesskette:** Die extrahierten 2D-Klassen (Dachflächen, Störobjekte, Bodenbedeckung) fließen unmittelbar in gekoppelte 3D-Simulationen ein. Das System überführt die Segmentierungen in eine topologische Solarsimulation und nutzt sie parallel zur Berechnung stadtökologischer Flächenindikatoren.

3. **Bedarfsgesteuerte Detailanalyse:** Anstelle einer statischen Stapelverarbeitung auf Bundesebene liefert der Ansatz eine flexible Auswertungsroutine, die die physische Realität auf Basis aktueller hochauflösender Fernerkundungsdaten lokal und detailliert abbildet.

Damit liefert die vorliegende Arbeit einen methodischen Machbarkeitsnachweis, wie anspruchsvolle Verfahren der maschinellen Bildverarbeitung ressourceneffizient in operative stadtplanerische Fragestellungen integriert werden können.

2.8 Arbeitsumgebung

Die Entwicklung und Evaluierung von Deep-Learning-Modellen zur semantischen Segmentierung stellt hohe Anforderungen an die Rechenressourcen, insbesondere an den Grafikspeicher. Während Forschungsprojekte auf dem aktuellen Stand der Technik häufig auf verteilten Rechenclustern mit mehreren leistungsstarken Grafikprozessoren trainiert werden, um massive Batch-Größen und kurze Trainingszeiten zu realisieren, beschränkt sich diese Arbeit auf die Nutzung einer einzelnen Grafikkarte.

Obwohl das Training von Segmentierungsmodellen stark von der massiv parallelen Architektur typischer Forschungscluster profitiert, wird in dieser Arbeit untersucht, wie sich die Modelle unter den Restriktionen eines einzelnen lokalen Grafikprozessors trainieren und evaluieren lassen. Im Folgenden wird die für diese Arbeit genutzte Hardware- und Software-Infrastruktur beschrieben.

2.8.1 Hardware-Umgebung

Das für das Training verwendete System ist mit einer Grafikkarte des Typs *NVIDIA RTX 4000 SFF Ada Generation* ausgestattet, welche über 20 GB Grafikspeicher verfügt. Zusätzlich umfasst der Server 64 GB Arbeitsspeicher sowie NVMe-Speichermedien,

um den Datendurchsatz bei der Verarbeitung der großskaligen Geo-Rasterdaten zu gewährleisten.

Da bei der semantischen Segmentierung die räumliche Auflösung der Eingabebilder durch das Netzwerk hindurch verarbeitet oder rekonstruiert werden muss, resultiert daraus im Vergleich zu reinen Klassifikationsaufgaben ein höherer Speicherbedarf pro Bild [16]. Die Beschränkung auf 20 GB Grafikspeicher sowie die Nutzung einer einzelnen Grafikkarte erfordern daher folgende methodische Anpassungen für den experimentellen Aufbau:

- **Limitierung der Batch-Größe:** Um Speicherüberläufe zu vermeiden, wird die physische Batch-Größe reduziert. Dies macht entsprechende Anpassungen im Trainingsprozess erforderlich, um stabile Gradienten zu erhalten.

- **Fokussierung der Experimente:** Da das Training auf einer einzelnen Grafikkarte und nicht auf einem Rechencluster erfolgt, ergeben sich verlängerte Trainingszeiten. Dies limitiert den in dieser Arbeit realisierbaren Explorationsumfang. Anstelle einer breiten automatisierten Suche über diverse Modellarchitekturen, Vortrainingsmethoden und Hyperparameter erfordert die Hardware-Restriktion eine gezielte Selektion und Fokussierung auf die vielversprechendsten, umsetzbaren Konfigurationen.

2.8.2 Software-Frameworks und Bibliotheken

Aufgrund des breiten Ökosystems an effizienten Bibliotheken für maschinelles Lernen und Geodatenverarbeitung erfolgt die Implementierung der gesamten Pipeline in Python. Als zentrales Deep-Learning-Framework dient dabei PyTorch [90]. Es zeichnet sich durch einen dynamischen Berechnungsgraphen aus und stellt die notwendigen Schnittstellen für eine effiziente Hardwarebeschleunigung bereit, weshalb es die Basis für die vorliegende Arbeit bildet.

Deep Learning und Modelltraining

Modellarchitekturen und Toolboxes Für die semantische Segmentierung existieren verschiedene etablierte Frameworks, wie etwa *Detectron2* (Meta) oder die *MMSegmentation*-Toolbox (OpenMMLab) [91]. Die Wahl fällt auf das OpenMMLab-Ökosystem aufgrund seines modularen Aufbaus, der eine systematische und flexible Kombination verschiedenster Encoder- und Decoder-Architekturen ermöglicht. Die Nutzung standardisierter Bibliotheken gewährleistet die Reproduzierbarkeit der Ergebnisse und ermöglicht faire Benchmark-Vergleiche, ohne komplexe Trainings-Pipelines vollständig neu implementieren zu müssen. Ergänzend dazu integriert das Framework *MMPretrain* [92] nahtlos

Methoden des selbstüberwachten Lernens, wie den Masked AE. Für spezialisierte Architekturen, die noch nicht in Standard-Bibliotheken verfügbar sind, wird auf die offiziellen Repositories der Autoren zurückgegriffen, wie beispielsweise für das in der Vorstudie evaluierte UNetFormer-Modell [5].

Hyperparameter-Optimierung Zur systematischen Suche nach optimalen Trainingsparametern stehen Bibliotheken wie *Ray Tune* oder *Optuna* [93] zur Verfügung. In dieser Arbeit kommt *Optuna* zum Einsatz, da es leichtgewichtig ist und effiziente Pruning-Strategien bietet. Diese brechen unvorteilhafte Trainingsläufe frühzeitig ab, was die begrenzten Ressourcen schont.

Geodatenverarbeitung und Solarsimulation

Räumliche Analyse und Datenverarbeitung Anstatt Geometrie-Operationen in ressourcenintensiven Desktop-GIS-Programmen durchzuführen, erfolgt die Verarbeitung direkt im Python-Code. Dies garantiert eine durchgehende und automatisierbare Pipeline. Die performante Verarbeitung hochauflösender Rasterdaten wird durch *Rasterio* gelöst, welches eine Python-Schnittstelle zur C++-Bibliothek *GDAL* darstellt. Für Vektoroperationen ist *Shapely* der etablierte Standard. Komplexe räumliche Algorithmen, wie die Delaunay-Triangulierung oder KD-Bäume, sowie Methoden des unüberwachten maschinellen Lernens (beispielsweise DBSCAN für das Clustering von Störobjekten) werden auf die Bibliotheken *SciPy* [94] und *Scikit-Learn* [95] ausgelagert.

Solarsimulation Während in der Industrie oft proprietäre Simulationssoftware wie *PV*SOL* oder *PVSyst* verwendet wird, erfordert eine automatisierte Pipeline den Einsatz programmierbarer Schnittstellen. Hierfür gilt die Open-Source-Bibliothek *Pvlib* [96] als ein umfassendes und wissenschaftlich validiertes Werkzeug, weshalb sie für die physikalische Modellierung der Sonnenstände und Einstrahlung genutzt wird.

Geoinformationssysteme Zur manuellen Datenaufbereitung, visuellen Validierung der berechneten Vektorgeometrien und zur kartographischen Darstellung wird ergänzend ein Desktop-Geoinformationssystem herangezogen. Zwar dominiert *ArcGIS* als proprietäre Lösung den kommerziellen Markt, für diese Arbeit wird jedoch das Open-Source-Pendant *QGIS* [97] gewählt. Diese Entscheidung unterstreicht den Open-Science-Gedanken der Arbeit. Durch den konsequenten Verzicht auf lizenzpflichtige Software und proprietäre Datenformate wird sichergestellt, dass die entwickelte Toolchain von Dritten kostenfrei reproduziert und weiterentwickelt werden kann.

Kapitel 3

Methodik

Dieses Kapitel beschreibt das methodische Vorgehen zur Entwicklung der semantischen Segmentierungs-Pipeline sowie der darauf aufbauenden stadtökologischen Analysen. Um die Forschungsziele systematisch anzugehen, gliedert sich der praktische Teil der Arbeit in vier aufeinander aufbauende Phasen, deren konzeptioneller Ablauf in Abbildung 3.1 veranschaulicht ist.

Den Beginn bildet eine Voruntersuchung zur Identifikation der optimalen Trainingsdomäne und Datenbasis. Darauf aufbauend erfolgt eine gezielte Architekturoptimierung unter den restriktiven Bedingungen eines Low-Data-Regimes, um das effizienteste Modell für die weitere Verarbeitung zu ermitteln. Die dritte Phase umfasst die systematische Erstellung und Annotation eines eigenen, hochauflösenden Anwendungsdatensatzes, um den Transfer auf die finale Zieldomäne zu realisieren. Den Abschluss bildet die Ableitung praxisrelevanter Flächenindikatoren sowie die Berechnung des Solarpotenzials auf Basis der generierten Segmentierungsmasken.

Aus Gründen der Übersichtlichkeit beschränkt sich die nachfolgende Darstellung auf die wesentlichen architektonischen Entscheidungen, algorithmischen Konzepte und entscheidenden Hyperparameter. Die vollständigen und exakten Konfigurationen sämtlicher Trainingsläufe und Pipelines sind aus Gründen der Reproduzierbarkeit dem Code-Repositorium zu entnehmen, das dieser Arbeit beiliegt.

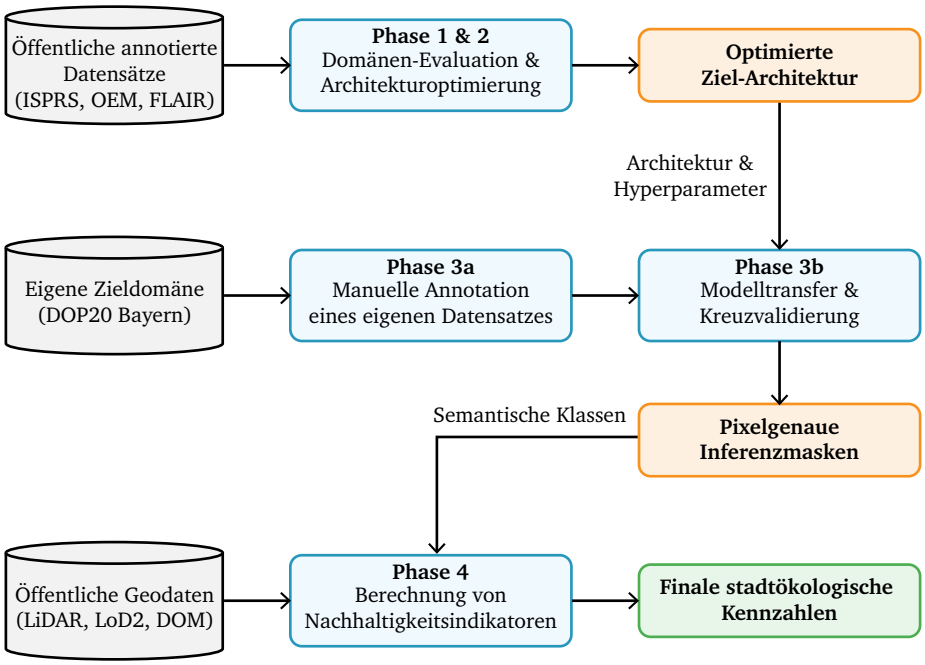


Abbildung 3.1 – Detaillierte Übersicht der methodischen Pipeline. Der Workflow koppelt die Architekturoptimierung auf öffentlichen Stellvertreterdaten (Phase 1 & 2) mit dem Transfer auf die manuell annotierten DOP20-Bildausschnitte (Phase 3). Die resultierenden Inferenzmasken fließen anschließend gemeinsam mit öffentlichen Geodaten in die finale Indikator- und Solarpotenzial-Berechnung (Phase 4).

3.1 Übergreifende methodische Konzepte

Bevor die folgenden Abschnitte die konkreten experimentellen Schritte zur Architektur- und Datenauswahl detailliert beschreiben, legt dieser Abschnitt das methodische Fundament der praktischen Umsetzung. Aufbauend auf der in Abschnitt 2.8 beschriebenen Arbeitsumgebung definiere ich Evaluierungsprotokolle, die Vorgehensweise bei der Datenannotation sowie die Inferenzstrategie für großflächige Bilddaten.

Da diese Konzepte als universelle Grundlage sowohl für die initialen Voruntersuchungen als auch für das finale Modelltraining dienen, fasst dieser Abschnitt sie hier zentral zusammen, um eine redundanzfreie Diskussion der nachfolgenden Experimente zu ermöglichen.

3.1.1 Evaluierungsprotokoll, Metriken und Verlustfunktion

Da Abschnitt 2.4.1 die Evaluierungsmetriken bereits theoretisch einführt, fokussiert sich dieser Abschnitt auf ihre anwendungsspezifische Rolle in der vorliegenden Pipeline.

Als primäres Optimierungs- und Entscheidungskriterium über alle experimentellen Phasen hinweg dient die mIoU. Da diese Metrik sowohl Über- als auch Untersegmentierung bestraft, erweist sie sich bei den stark unbalancierten urbanen Datensätzen dieser Arbeit als robustester Indikator für die räumliche Genauigkeit. Zur Bestimmung des besten Modellzwischenstands während der Trainingsphase wertet der Algorithmus diese Metrik kontinuierlich auf dem jeweils internen Validierungsdatensatz aus.

Zusätzlich nutzen alle Trainingsläufe, unabhängig von der gewählten Architektur oder Phase, dieselbe Verlustfunktion. Um das Training auf die spezifischen Herausforderungen hochauflösender Luftbilder abzustimmen, nutzt die Pipeline eine Kombination aus Dice-Verlust und gewichteter Kreuzentropie. Der Dice-Verlust-Term optimiert dabei direkt die räumliche Überlappung der Vorhersagen, während die Kreuzentropie-Komponente die pixelweise Klassifikationssicherheit maximiert. Um dem starken Klassenungleichgewicht (z. B. Dominanz von Dächern gegenüber kleinen Dachaufbauten) entgegenzuwirken, berechnet das System die Gewichtung der Kreuzentropie dynamisch: Jede Klasse erhält ein Gewicht basierend auf ihrer inversen Flächenhäufigkeit innerhalb des jeweils aktiven Trainingsdatensatzes.

3.1.2 Systematik der Datenannotation

Die Erstellung präziser semantischer Segmentierungsmasken auf Basis von 20 cm-Orthophotos stellt aufgrund der reinen Nadirperspektive, projektionsbedingter Verzerrungen sowie spektraler und struktureller Ähnlichkeiten zwischen Objektklassen eine generelle methodische Herausforderung dar. Wie anspruchsvoll diese Aufgabe ist, zeigt ein Vergleich der Annotationskonsistenz in etablierten Referenzdatensätzen: Während menschliche Annotatoren bei Straßenszenen (Cityscapes) eine Übereinstimmung von 96 % erreichten [67], lag dieser Wert bei den Luftbildern von OEM nur bei 78 % [57]. Die korrekte Interpretation aus der Vogelperspektive erfordert demnach spezifisches Erfahrungswissen [47] und eine stringente Methodik.

Für einen kontrollierten und effizienten Ablauf aller manuellen, pixelgenauen Annotationen kommt die Open-Source-Plattform *CVAT* zum Einsatz. Um die Datensouveränität zu wahren und Latenzzeiten bei der Verarbeitung der hochauflösenden Bilddaten zu vermeiden, stellt das Unternehmensnetzwerk hierfür eine dedizierte Instanz der Software bereit.

Um über alle Datensätze hinweg eine hohe semantische und geometrische Konsistenz zu gewährleisten, gelten folgende universelle Annotationsregeln:

- **Visuelle Interpretation:** Bei unscharfen Kantenverläufen gilt die Mitte des Übergangsbereichs als Grenze.

- **Umgang mit Verdeckungen:** Da die Segmentierung auf optischen Daten basiert, erfasst die Annotation stets die physikalisch sichtbare Oberfläche. Verdeckt beispielsweise ein Baum einen Teil eines Gebäudes, erhält diese Stelle die Klasse *Tree* und nicht *Building*.
- **Umgang mit Schattenwurf:** Da Schatten lediglich die Beleuchtungsverhältnisse, nicht aber die physische Bodenbedeckung verändern, ordnet die Systematik verschattete Bereiche stets der tatsächlich darunterliegenden Oberfläche zu. Wirft beispielsweise eine Baumkrone einen Schatten auf eine Straße, erhält dieser abgedunkelte Bereich die Klasse der Straße und nicht die des Baumes.
- **Topologische Konsistenz:** Die Annotationslogik stellt sicher, dass keine Lücken oder unbeabsichtigten Überlappungen zwischen angrenzenden Klassen entstehen.

Zur Qualitätssicherung durchlaufen alle annotierten Daten einen zweistufigen Prozess: Auf die initiale Erfassung folgt stets ein systematischer visueller Überprüfungsdurchlauf, in dem ich gezielt semantische Fehlklassifikationen (z. B. Falschzuweisungen) identifiziere und korrigiere.

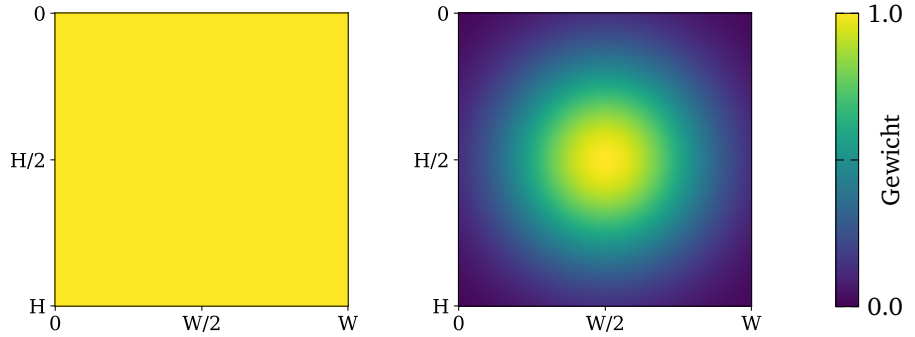
3.1.3 Gleitfenster-Inferenz für großflächige Bilddaten

Da die Segmentierung der großflächigen DOP20-Aufnahmen in dieser Arbeit die Kapazität des Grafikspeichers bei Weitem übersteigt, verarbeitet das System die Untersuchungsgebiete nicht als Gesamtbild. Stattdessen erfordert dies eine kachelbasierte Verarbeitung mittels Gleitfenster-Inferenz (engl. *Sliding Window Inference*).

Standardmäßig nutzen Segmentierungs-Frameworks wie *MMSegmentation* hierfür eine gleichmäßige Gewichtung. Dabei mittelt das Framework die Vorhersagen überlappender Bildausschnitte an den Schnittmengen arithmetisch, unabhängig davon, ob ein Pixel im Zentrum oder am äußersten Rand eines Bildausschnitts liegt. Dieses Vorgehen erweist sich jedoch als suboptimal: Insbesondere bei den verwendeten ViTs verfügen Pixel am Rand eines Bildausschnitts über einen stark asymmetrischen und eingeschränkten lokalen Kontext. Folglich fallen ihre Vorhersagen unzuverlässiger aus als die von Pixeln im Zentrum. In der Praxis führt die gleichmäßige Gewichtung daher häufig zu sichtbaren, geradlinigen Kantenartefakten an den Grenzen der Bildausschnitte in der finalen Segmentierungsmaske.

Um diese fehlerhaften Übergänge zu eliminieren, ersetzt das Verfahren die Standard-Inferenz durch eine Gauß-gewichtete Akkumulation (engl. *Gaussian-Weighted Sliding Window Inference*), wie sie ähnlich auch in modernsten kachelbasierten Segmentierungsarchitekturen [98] zum Einsatz kommt. Hierbei gewichtet das Verfahren den Beitrag jedes Pixels in der Vorhersage eines Bildausschnitts basierend auf seiner räumlichen Distanz zum Zentrum (siehe Abbildung 3.2). Pixel im Zentrum, die

über den vollständigen Bildkontext verfügen, erhalten das höchste Gewicht, während der Einfluss der Randpixel entsprechend einer zweidimensionalen Normalverteilung abfällt. Die Standardabweichung σ skaliert dabei dynamisch mit der gewählten Größe eines Bildausschnitts ($\sigma = \text{Bildausschnittsgröße}/4$).



(a) Gleichmäßige Gewichtung **(b)** Gaußsche Gewichtung

Abbildung 3.2 – Gewichtungskarten für die Gleitfenster-Inferenz. Bei gleichmäßiger Gewichtung tragen alle Pixelpositionen gleich bei, während die Gaußsche Gewichtung Vorhersagen nahe dem Zentrum höher gewichtet, da dort der vollständige Kontext verfügbar ist.

Obwohl diese Maßnahme globale Metriken wie die mIoU kaum signifikant verändert, verbessert sie die topologische Integrität der Vorhersagen maßgeblich. Da die generierten Segmentierungsmasken in dieser Arbeit als direkte geometrische Grundlage für die nachgelagerte Solarpotenzialanalyse und Flächenextraktion (vgl. Abschnitt 3.5) dienen, ist die Vermeidung künstlicher Bruchkanten auf zusammenhängenden Flächen von entscheidender Bedeutung.

3.2 Voruntersuchung zur Eignung verschiedener Trainingsdomänen

Nachdem der vorangegangene Abschnitt die methodischen Gemeinsamkeiten etabliert hat, die als universelle Grundlage für alle weiteren Phasen dienen, rückt nun die konkrete Modellentwicklung in den Fokus. Den Startpunkt bildet dabei die Frage nach der optimalen Datenbasis.

Dieses Vorexperiment evaluiert daher verschiedene Trainingsstrategien für die Inferenz auf hochauflösenden DOP20-Aufnahmen [48]. Da der geplante Hauptteil dieser Arbeit die Übertragung von Modellen auf dieses Zielformat vorsieht, untersucht dieser Abschnitt, welche Eigenschaften der Trainingsdaten, insbesondere das Zusammenspiel aus GSD und Szenenvarianz, die Generalisierungsfähigkeit am stärksten

begünstigen. Dabei stehen sich zwei Paradigmen gegenüber: Das Training auf hochauflösenden, aber geographisch begrenzten Datensätzen (ISPRS Potsdam/Vaihingen) und das Training auf moderat aufgelösten, aber geographisch diversen Datensätzen (OEM).

Als Referenzarchitektur fungiert hierfür der UNetFormer in seiner leistungsstärkeren Variante FT-UNetFormer, da dieser auf gängigen Referenzdatensätzen (Vaihingen/Potsdam) eine hohe Leistungsfähigkeit aufweist [5]. Im Rahmen des Versuchsaufbaus trainiert die Pipeline das Modell separat auf Datensätzen mit unterschiedlichen Auflösungen und evaluiert es anschließend auf einem einheitlichen, unabhängigen Validierungsdatensatz (DOP20).

3.2.1 Auswahl und Harmonisierung der Trainingsdaten

Das Training basiert auf drei Datensätzen, die sich signifikant in ihrer räumlichen Auflösung unterscheiden. Tabelle 3.1 gibt einen Überblick über die technischen Spezifikationen und verdeutlicht die Auflösungsunterschiede zum Zielformat.

Tabelle 3.1 – Übersicht der im Vorexperiment verwendeten Trainings- und Validierungsdatensätze. Zu beachten sind die signifikanten Unterschiede in GSD, Flächenabdeckung und geographischer Diversität zwischen den Quellen.

Datensatz	GSD	Kanäle	Klassen (Orig.)	Fläche
Training				
ISPRS Potsdam [18]	5 cm	RGB-NIR	6	3.42 km^2
ISPRS Vaihingen [56]	9 cm	NIR-R-G	6	1.36 km^2
OpenEarthMap [57]	25–50 cm	RGB	8	80 km^2
Validierung				
DOP20 [48]	20 cm	RGB-NIR	5	1 km^2

Die Datensätze repräsentieren einen klassischen Zielkonflikt: Während die ISPRS-Datensätze eine extrem hohe Detailtiefe und spektrale Konsistenz bieten, beschränken sie sich auf einzelne Städte. Im Gegensatz dazu bietet OEM trotz geringerer Auflösung eine weitaus höhere Varianz an urbanen Szenarien. Ein geographischer Filter hält die Variabilität der baulichen Strukturen (Domänenunterschiede) bei OEM dennoch kontrollierbar. Ein erster Durchgang basiert ausschließlich auf Teilbereichen aus Deutschland. Ein zweiter Schritt erweitert den Datensatz anschließend um Regionen aus Österreich und Polen, da sie eine ähnliche urbane Bauweise und Architektur aufweisen.

Um die unterschiedlichen Klassenstrukturen der Datensätze anzugleichen, harmonisiert der nächste Schritt die Datensätze. Das Zielschema orientiert sich hierbei an den etablierten ISPRS-Referenzdatensätzen. Tabelle 3.2 stellt die Abbildung der OEM-Klassen auf dieses Schema dar. Die Annotation des Validierungsdatensatzes

orientiert sich an der Annotationslogik des OEM-Datensatzes [57]. Da dieser Datensatz primär auf Landbedeckungsklassen fokussiert und keine Klasse für mobile Objekte wie Fahrzeuge definiert, ordnet die eigene Annotation diese analog zur OEM-Systematik der darunterliegenden versiegelten Fläche zu.

Da die auf ISPRS-Daten trainierten Referenzmodelle jedoch die Klasse *Car* explizit vorhersagen, entsteht ein systematischer Konflikt: Eine naive Auswertung wertet ein korrekt als Auto erkanntes Objekt als Fehler (Falsch-Positiv für *Car* bzw. Falsch-Negativ für *Impervious Surface*).

Um diese Verzerrung zu verhindern, erfolgt eine deterministische Zuordnung für die Vorhersagen: Pixel, die das Modell als *Car* klassifiziert, ordnet der Auswertungsalgorithmus der Klasse *Impervious Surface* zu. Dies folgt der semantischen Logik, dass sich Fahrzeuge im urbanen Raum fast ausschließlich auf versiegelten Flächen befinden. Die Detektion eines Fahrzeugs gilt somit als valider Indikator für das Vorhandensein einer versiegelten Fläche.

Hinsichtlich der spektralen Kanäle nutzen alle Modelle eine einheitliche Eingangsgröße von drei Kanälen. Bei den Datensätzen OEM und ISPRS Potsdam dienen hierzu die RGB-Kanäle. Da für ISPRS Vaihingen kein Blaukanal vorliegt, trainiert das Verfahren das Modell stattdessen auf der Kanalkombination Nahinfrarot-Rot-Grün (NIR-R-G). Zur Inferenz auf dem DOP20-Datensatz extrahiert der Algorithmus jeweils die Kanäle, die der spektralen Konfiguration des trainierten Modells entsprechen.

Tabelle 3.2 – Harmonisierung der Klassenschemata. Abbildung der Quellklassen aus den Datensätzen ISPRS [56] und OpenEarthMap [57] auf gemeinsame Zielklassen.

Zielklasse	**ISPRS Original**	**OpenEarthMap Original**
Impervious Surfaces (Versiegelte Flächen)	Impervious Surfaces, Car	Road, Developed Space
Building (Gebäude)	Building	Building
Low Vegetation (Niedrige Vegetation)	Low Vegetation	Rangeland, Bareland, Agriculture Land
Tree (Bäume)	Tree	Tree
Clutter / Background (Hintergrund)	Clutter	Water

3.2.2 Erstellung der Referenzdaten für die Validierung

Zur unabhängigen Überprüfung der Modellleistung auf der Zielauflösung dient ein räumlich zusammenhängendes Testgebiet von 1 km^2 Fläche in Erlangen. Der gewählte Ausschnitt zeichnet sich durch eine heterogene Bebauungsstruktur aus,

die sowohl dicht besiedelte Wohngebiete mit Giebeldächern als auch universitäre Einrichtungen mit großflächigen Flachdächern umfasst. Dies gewährleistet eine repräsentative Abdeckung der unterschiedlichen urbanen Klassen.

Die Erstellung der Referenzmaske erfolgt gemäß der in Abschnitt 3.1.2 definierten Systematik. Der Arbeitsaufwand für diesen 1 km^2 großen Ausschnitt beläuft sich auf ca. 40 Stunden. Gemäß dem übergreifenden Qualitätssicherungskonzept folgt auf die initiale Erfassung der zweite Überprüfungsdurchlauf zur Bereinigung semantischer Fehler. Abbildung 3.3 visualisiert den so erstellten Validierungsdatensatz.

Die Analyse der Referenzmaske zeigt ein starkes Ungleichgewicht der Klassenverteilung. Während versiegelte Flächen und Vegetation dominieren, ist die Klasse *Clutter* mit einem Flächenanteil von unter 0,01 % nahezu gar nicht vertreten. Auf Basis dieses Ausschnitts lassen sich daher nur eingeschränkt statistisch belastbare Aussagen über die Detektionsleistung dieser spezifischen Klasse treffen. Während der Inferenz ordnet das Modell Pixel weiterhin regulär der Klasse *Clutter* zu, da diese integraler Bestandteil des gelernten Klassenschemas ist. Bei der anschließenden Berechnung der Evaluierungsmetriken (wie der mIoU) wird der spezifische IoU-Wert dieser Hintergrundklasse jedoch systematisch ignoriert. Dieses Vorgehen stellt sicher, dass die quantitative Evaluierung nicht durch eine anwendungsirrelevante Klasse verzerrt wird, sondern sich strikt auf die Leistungsbewertung der relevanten urbanen Objektklassen fokussiert.

3.2.3 Trainingskonfiguration

Alle experimentellen Trainingsläufe verwenden die FT-UNetFormer-Architektur. Die Originalautoren stellen diese Modifikation als leistungsstärkere Variante vor, die statt eines CNN-Encoders einen Swin Transformer als Backbone nutzt [5]. Um von den bereits erlernten Merkmalsrepräsentationen der Original-Implementierung zu profitieren, nutzt das Setup zur Initialisierung die von den Autoren bereitgestellten Modellgewichte. Diese trainierten die Autoren sowohl auf ImageNet-21K [34] als auch auf dem ADE20K-Datensatz [99] vor.

Um die interne Vergleichbarkeit der Ergebnisse zu gewährleisten, basieren alle Trainings auf der gleichen Modellkonfiguration und denselben Hyperparametern. Die Trainingskonfiguration orientiert sich dabei grundsätzlich an den Empfehlungen der Originalautoren, passt diese jedoch hinsichtlich Batch-Größe und Thread-Anzahl an die vorliegende Hardware-Umgebung an.

Die Optimierung nutzt eine schichtweise Lernratenstrategie: Das weitere Training des bereits vortrainierten Backbones läuft mit einer deutlich reduzierten Lernrate von 6×10^{-5} ab, während für den Decoder eine Lernrate von 6×10^{-4} gilt. Ein Cosine-Annealing-Scheduler mit Warm Restarts steuert die Lernrate dynamisch über den Trainingsverlauf.

(a) Übersicht – Original

(b) Übersicht – Referenzmaske

(c) Detailausschnitt – Original

(d) Detailausschnitt – Referenzmaske

Abbildung 3.3 – Visualisierung des manuell annotierten Validierungsdatensatzes (DOP20). (a) und (b) zeigen den gesamten 1 km^2 großen Ausschnitt; das eingezeichnete Rechteck markiert den in (c) und (d) vergrößert dargestellten Bereich.

Zur Erhöhung der Generalisierungsfähigkeit wendet das Verfahren neben geometrischen Transformationen auch Mosaik-Augmentierung an. Tabelle B.1 fasst die wesentlichen Trainingsparameter zusammen.

Die Unterteilung der Datensätze folgt den offiziellen Vorgaben des jeweiligen Referenzdatensatzes für Trainings- und Validierungsdaten. Zur Validierung des Modells während der Trainingsphase (z. B. zur Bestimmung des besten Modellzwischenstands) dient die interne Teilmenge des jeweiligen Datensatzes. Die abschließende Evaluierung der Generalisierungsleistung findet hingegen ausschließlich auf dem unabhängigen DOP20-Datensatz statt.

Um das Generalisierungspotenzial der Quelldatensätze in ihrer nativen Form zu überprüfen, verzichte ich bewusst auf eine Skalierung der Trainingsdaten auf eine einheitliche GSD. Die Modelle lernen somit auf der jeweils datensatzspezifischen GSD und den inhärenten Merkmalen der Domäne, während die Inferenz einheitlich auf der 20 cm-GSD des DOP20-Datensatzes abläuft. Dies impliziert, dass die Modelle nicht nur unterschiedliche Texturen, sondern auch drastisch unterschiedliche Objektgrößen lernen und auf den mittleren Maßstab der DOP20-Daten generalisieren müssen.

3.3 Architekturoptimierung im Low-Data-Regime

Nachdem die Voruntersuchung die grundlegende Datenstrategie geklärt hat, widmet sich dieser Abschnitt der systematischen Ermittlung der leistungsstärksten Modellarchitektur. Da die finale Zieldomäne eine hohe semantische Detailtiefe bei gleichzeitig begrenzten Annotationsressourcen erfordert, findet diese Optimierung unter den realistischen Bedingungen eines stark limitierten Datensatzes statt. Die nachfolgenden Abschnitte begründen zunächst die Wahl der hierfür genutzten Stellvertreterdaten, definieren den experimentellen Rahmen und beschreiben abschließend die mehrstufige Strategie zur Modellselektion.

3.3.1 Ableitung der Datengrundlage

Die Ergebnisse der Voruntersuchung (siehe Abschnitt 4.1) bilden die Grundlage für die Auswahl der Datenbasis im Hauptteil dieser Arbeit. Sie zeigen, dass für eine erfolgreiche Generalisierung auf DOP20-Aufnahmen nicht zwingend die höchstmögliche Auflösung (5 cm GSD) ausschlaggebend ist, sondern vielmehr die Ähnlichkeit der Domänencharakteristika zwischen Trainings- und Zielverteilung. Der OEM-Datensatz erzielt hierbei aufgrund seiner passenden Auflösung (25–50 cm GSD) und seiner hohen geographischen Diversität die besten Generalisierungsergebnisse auf dem Erlanger Testgebiet.

Allerdings weist der OEM-Datensatz eine entscheidende Limitierung im Hinblick auf das Gesamtziel dieser Arbeit auf: Mit nur 8 Landbedeckungsklassen deckt er die

semantische Komplexität des angestrebten Zielschemas (15 Klassen) nicht hinreichend ab. Um differenzierte urbane Strukturen (z. B. Unterscheidung verschiedener Vegetationstypen oder detaillierte Gebäudeeigenschaften) zu modellieren, ist eine feinere Klassenunterteilung notwendig.

Auf Grundlage dieser Erkenntnisse, der Notwendigkeit einer 20 cm-Auflösung, einer hohen räumlichen Diversität sowie einer tieferen semantischen Granularität, fällt die Wahl für die weiteren Analysen auf den FLAIR-Datensatz (French Land cover from Aerospace ImageRy) [6]. Dieser erfüllt die im Vorexperiment identifizierten Anforderungen besser:

- **Übereinstimmende Auflösung:** Mit einer GSD von 20 cm entspricht FLAIR exakt der Auflösung der bayerischen DOP20-Zielaufnahmen, was den Skalierungsfehler minimiert.

- **Hohe Diversität:** Der Datensatz deckt 50 verschiedene Domänen in ganz Frankreich ab und bietet damit eine vergleichbare, wenn nicht höhere Varianz als OEM, was das Risiko lokaler Überanpassungen minimiert.

- **Erweiterter Klassenraum:** Im Gegensatz zu den ISPRS-Referenzdatensätzen oder OEM bietet FLAIR ein differenziertes Klassenschema (über 10 Klassen), welches dem geplanten Schema der eigenen bayerischen Referenzdaten deutlich näher kommt und somit ein realistischeres Vortrainingsszenario für komplexe semantische Aufgaben ermöglicht.

Die Wahl von FLAIR ergibt sich direkt aus der Beobachtung, dass die Kombination aus auflösungskonformen Daten und hoher Szenenvarianz die besten Voraussetzungen für die Übertragbarkeit auf neue, ungesehene Gebiete (wie die spätere Zieldomäne) schafft.

Der FLAIR-Datensatz dient im weiteren Verlauf dieser Arbeit somit als Stellvertreterdatensatz, um Architekturentscheidungen für den finalen Anwendungsfall zu validieren, ohne bereits auf den noch zu erstellenden Datensatz angewiesen zu sein.

3.3.2 Definition des Low-Data-Regimes und Sampling-Strategie

Um den FLAIR-Datensatz als geeignete Stellvertreterumgebung für die spätere Anwendung auf bayerische Siedlungsstrukturen zu verwenden, extrahiert ein Samplingverfahren eine gezielte Teilmenge. Ziel ist es, ein Szenario mit stark begrenzter Trainingsmenge zu simulieren, ohne dabei die semantische und strukturelle Varianz des Gesamtdatensatzes zu verlieren.

Vorfilterung und Klassenharmonisierung

Der originale FLAIR-Datensatz umfasst diverse Landschaftstypen; neben urbanen Szenen enthält er auch großflächige landwirtschaftliche und bewaldete Gebiete. Da der Fokus dieser Arbeit auf der semantischen Segmentierung urbaner Räume liegt, filtert ein automatisierter Prozess die Daten zunächst vor. Dieser Schritt schließt Bildkacheln ohne signifikante Bebauung systematisch aus. Als Grenzwert dient hierfür eine Mindestanzahl von 15 000 Pixeln der Klasse *Building* (entspricht ca. 5,7 % der Bildfläche). Um das Rauschen durch ein extremes Klassenungleichgewicht zu minimieren, fasst das neue Schema zudem extrem seltene Klassen (Flächenanteil nach Filterung $< 0,3\,\%$) in der Sammelklasse *Other* zusammen. Das resultierende Schema umfasst 10 Klassen und ist in Tabelle 3.3 dargestellt. Diese Schritte stellen sicher, dass die resultierende Teilmenge überwiegend urbane Szenen enthält und für die angestrebte Anwendung repräsentativ ist.

Tabelle 3.3 – Klassenharmonisierung des FLAIR-Datensatzes. Seltene Klassen (Flächenanteil $< 0,3\,\%$) fasst das Schema in der Sammelklasse *Other* zusammen. Der angegebene Flächenanteil bezieht sich auf die Trainingsdaten nach der Vorfilterung.

Neue Klasse	Ursprüngliche Klasse(n)	Anteil (%)
Building (Gebäude)	Building	25,5
Pervious Surface (Unversiegelte Flächen)	Pervious Surface	6,8
Impervious Surface (Versiegelte Flächen)	Impervious Surface	29,6
Water (Gewässer)	Water	1,1
Coniferous (Nadelbäume)	Coniferous	0,9
Deciduous (Laubbäume)	Deciduous	10,5
Brushwood (Gebüsch)	Brushwood	3,4
Herbaceous Vegetation (Grünflächen)	Herbaceous Vegetation	19,0
Agricultural Land (Agrarflächen)	Agricultural Land	2,0
Other (Sonstige)	Bare Soil, Vineyard, Plowed Land, Swimming Pool, Snow, Clear Cut, Mixed, Ligneous, Greenhouse, Other	1,1

Definition des Low-Data-Regimes

Für die Architektursuche führe ich bewusst kein Training auf dem vollen Datensatz durch, sondern definiere ein Low-Data-Regime. Dabei lege ich die Trainingsgröße auf $N = 75$ Bilder fest. Bei einer GSD von 0,2 m und einer Bildgröße von 512×512 Pixeln entspricht dies einer effektiv annotierten Fläche von lediglich:

$$A_{train} = 75 \cdot (512 \cdot 0,2\,\text{m})^2 \approx 0,79\,\text{km}^2 \tag{3.1}$$

Diese starke Reduktion der Datenmenge dient drei methodischen Zielen:

1. **Diskriminativer Stresstest:** Das Low-Data-Regime dient als Test für die Stichprobeneffizienz (engl. *Sample Efficiency*): Es macht sichtbar, welche Modelle mit den im Vortraining gelernten Merkmalen auch unter Datenknappheit zuverlässig generalisieren. Zahlreiche Arbeiten zeigen, dass vortrainierte Modelle mit semantisch hochwertigen, z. B. selbstüberwacht gelernten Repräsentationen gerade in Szenarien mit wenigen Labels deutlich stabiler generalisieren als rein überwacht trainierte Backbones [100].

2. **Simulation eines Kaltstart-Szenarios:** In der realen Anwendung (z. B. Training auf neuen bayerischen Daten) stehen initial oft nur wenige, teilweise zudem fehlerhafte Labels zur Verfügung, während große Mengen unannotierter Bilddaten vorliegen [101]. Der Versuchsaufbau simuliert diese Ressourcenknappheit realistisch.

3. **Hypothese der Rangfolgenkonsistenz:** Basierend auf Forschungsergebnissen zur *Neural Architecture Search* nehme ich an, dass die relative Rangfolge der Modelle auf dem kleinen Datensatz stark mit der Rangfolge auf größeren Datensätzen korreliert (d. h. eine hohe Rangkorrelation). Dies ermöglicht eine effiziente Selektion des besten Modells ohne Ressourcenaufwand eines Trainings auf einem größeren Datensatz [102].

Datenselektion und Sampling-Strategie

Um trotz der geringen Datenmengen die semantische Varianz des Gesamtdatensatzes akkurat abzubilden, entwickle ich ein mehrstufiges, inhaltsbasiertes Selektionsverfahren. Dieser Algorithmus verbessert die statistische Repräsentativität durch eine iterative Optimierung, deren Architektur in Abbildung 3.4 schematisch dargestellt ist. Der logische Ablauf gliedert sich in folgende Stufen:

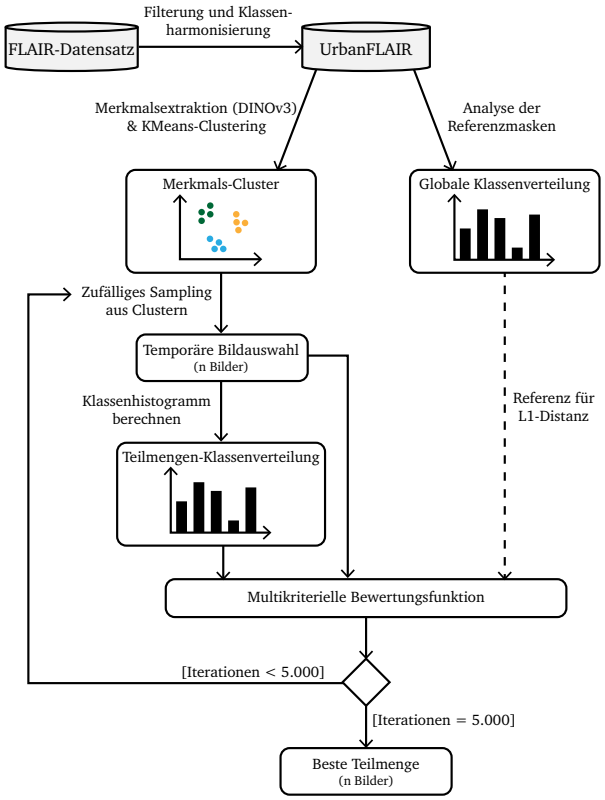


Abbildung 3.4 – Schematische Darstellung des Selektionsalgorithmus zur Konstruktion repräsentativer Teilmengen einer vorgegebenen Größe N. Für die Architektursuche erzeugt das Verfahren fünf unabhängige Trainings-Teilmengen ($N = 75$); der Validierungsdatensatz ($N = 250$) wird nach demselben Verfahren aus dem offiziellen Testdatensatz gezogen.

1. **Erfassung der visuellen Diversität:** Zur Charakterisierung des Bildinhalts extrahiert das Basismodell *DINOv3* (`facebook/dinov3-vitl16-pretrain-sat493m`) [39] zunächst Merkmalsrepräsentationen, da dieses semantische Zusammenhänge gut abbildet. Anschließend unterteilt der Algorithmus die Bilder basierend auf diesen Merkmalen mittels k-Means ($k = 50$) in Cluster. Ähnliche, diversitätsorientierte Strategien finden auch in der Literatur zur Datenselektion Anwendung, um repräsentative Teilmengen zu konstruieren [103]. Das Ergebnis sind Gruppen von visuell ähnlichen Szenen (z. B. bilden dicht bebaute Stadtkerne ein Cluster, große Industriehallen ein anderes). Um die Qualität dieses Schrittes zu verifizieren, visualisiert Abbildung 3.5 die extrahierten Merkmale. Wie in der Abbildung dargestellt, erzeugt der DINOv3-Merkmalsextraktor räumlich separierte Cluster. Bei der qualitativen Betrachtung der Clusterinhalte (siehe Beispielbilder) zeigt sich oft eine grobe

visuelle Kohärenz. Auffällig ist jedoch, dass die maschinell gelernten Repräsentationen nicht immer strikt der menschlichen Semantik entsprechen. So finden sich beispielsweise vereinzelt Bilder von Vorstädten in Clustern, die vornehmlich Industrieanlagen enthalten, vermutlich aufgrund ähnlicher maschinell extrahierter Merkmale wie Kantenfrequenzen oder Dachstrukturen. Für das Ziel dieser Arbeit ist eine perfekte semantische Reinheit der Cluster jedoch nicht erforderlich. Das k-Means-Clustering dient hierbei als datengetriebene Heuristik: Indem gezielt aus jedem der $k = 50$ Cluster Vertreter gezogen werden, lässt sich die Varianz des Datenraums systematisch abbilden und eine deutlich höhere Diversität im finalen Trainingsset gewährleisten als durch ein rein zufälliges Sampling.

2. **Filterung auf Informationsgehalt (nur Training):** Speziell für die Trainingsdaten sortiert ein Vorfilter Bilder, die von einer einzelnen Klasse dominiert werden (Dominanzanteil $> 0,75$, z. B. eine einzelne Halle, die das gesamte Bild abdeckt), vorab systematisch aus. Ziel ist es, den Informationsgehalt pro Pixel zu maximieren, damit das Modell trotz der geringen Bildanzahl möglichst diverse Merkmale lernt.

3. **Optimierte Zufallsauswahl (Monte-Carlo-Suche):** Statt einer naiven Zufallsauswahl kommt eine iterative Suche zum Einsatz. Der Algorithmus generiert 5 000 zufällige Vorschläge für eine Teilmenge. Dabei stellt das Verfahren sicher, dass jede dieser Zufalls-Teilmengen Bilder aus möglichst vielen unterschiedlichen Clustern enthält (statt nur aus einem einzigen Bereich).

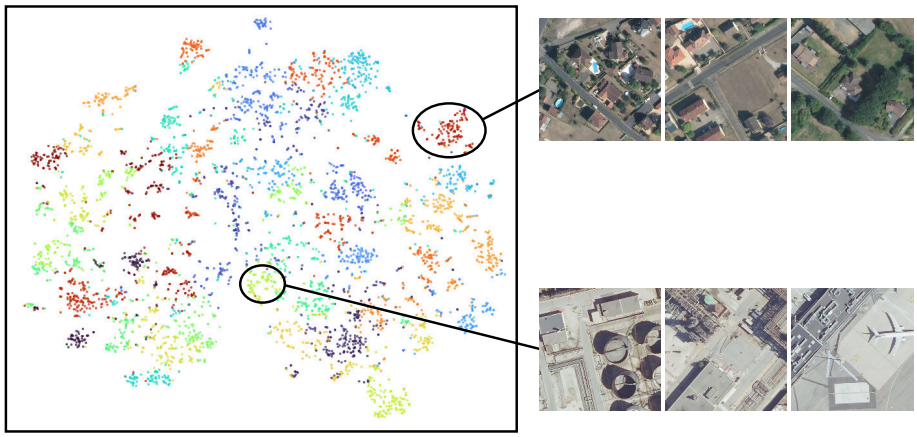


Abbildung 3.5 – Merkmalsrepräsentationen des gefilterten FLAIR-Trainingspools, dargestellt mittels t-SNE [104]. Die Färbung zeigt die k-Means-Clusterzugehörigkeit ($k = 50$). Zur Veranschaulichung sind für zwei exemplarisch markierte Cluster jeweils drei Beispielbilder abgebildet, die einen qualitativen Eindruck der visuellen Ähnlichkeiten vermitteln.

Aus diesen 5 000 Vorschlägen wählt das System schließlich denjenigen aus, der die Eigenschaften des Gesamtdatensatzes am besten widerspiegelt. Eine definierte Bewertungsfunktion evaluiert die Teilmengen dabei wie folgt:

$$S_{\text{score}} = -\underbrace{||\mathbf{h}_{\text{subset}} - \mathbf{h}_{\text{global}}||_1}_{\text{Verteilungsanpassung}} + \lambda_1 \cdot \underbrace{C_{\text{cov}}}_{\text{Diversität}} - \lambda_2 \cdot \underbrace{R_{\text{miss}}}_{\text{Vollständigkeit}} \tag{3.2}$$

Der erste Term minimiert die L_1-Distanz zwischen der Klassenverteilung der Teilmenge und der des Gesamtdatensatzes. C_{cov} belohnt eine hohe Abdeckung der visuellen Cluster, während R_{miss} das vollständige Fehlen von Klassen bestraft.

Der Algorithmus kommt differenziert zum Einsatz, um den unterschiedlichen Anforderungen gerecht zu werden:

- **Training:** Für die Architektursuche generiert das Verfahren aus dem offiziellen FLAIR-Trainingsdatensatz fünf unabhängige Teilmengen der Größe $N = 75$. Um statistische Unabhängigkeit zu gewährleisten, startet der Optimierungsprozess (Schritt 3) für jede dieser Teilmengen mit einem unterschiedlichen Zufallsstartwert. Dies stellt sicher, dass die Teilmengen zwar verschiedene Bilder enthalten, aber hinsichtlich ihrer statistischen Repräsentativität vergleichbar hochwertig sind.
- **Validierung:** Für die Validierung während des Trainings extrahiert das Setup aus dem offiziellen FLAIR-Testdatensatz einen separaten Validierungsdatensatz ($N = 250$). Um eine realistische Verteilung zu gewährleisten, verzichte ich hierbei bewusst auf die Filterung homogener Bilder (Schritt 2).

Die verbleibenden Daten des offiziellen Testdatensatzes bleiben unberührt und dienen ausschließlich der finalen Evaluierung der selektierten Architektur, um eine unvoreingenommene Einschätzung der Modellgüte zu gewährleisten.

3.3.3 Experimentelles Design

Um eine optimale Architektur für die semantische Segmentierung der hochauflösenden DOP20-Aufnahmen zu ermitteln, wähle ich einen mehrstufigen Evaluierungsansatz. Da die Leistungsfähigkeit eines Segmentierungsmodells maßgeblich von der Qualität der extrahierten Bildmerkmale sowie deren räumlicher Rekonstruktion abhängt, entkoppelt das Vorgehen die Architektursuche in drei aufeinanderfolgende Phasen: In der ersten Phase liegt der Fokus auf der Evaluierung verschiedener Merkmalsextraktoren (Backbones), um die leistungsstärkste Basisrepräsentation zu ermitteln. Die zweite Phase vergleicht unterschiedliche Decoder-Architekturen (Heads), die diese Merkmale auf die finale Pixel-Ebene projizieren. Den Abschluss bildet eine umfassende Hyperparameter-Optimierung der leistungsstärksten Gesamtkombination. Um in den ersten beiden Phasen die Architekturunterschiede isoliert

betrachten zu können, bleiben die Trainingsparameter sowie die Datengrundlage (das in Abschnitt 3.3.2 definierte Low-Data-Regime) streng konstant.

Backbone-Auswahl und Merkmalsharmonisierung

Der Backbone fungiert als primärer Merkmalsextraktor und bestimmt, wie gut das Modell sowohl lokale Texturen als auch globale semantische Kontexte erfasst. Um ein breites Spektrum aktueller Paradigmen abzubilden, vergleicht diese Phase fünf grundlegend unterschiedliche Architektur- und Vortrainingsansätze. Um eine unvoreingenommene Evaluierung der Backbones zu gewährleisten, koppelt das Setup alle Kandidaten an einen identischen Segmentierungskopf. Die Wahl fällt hierbei auf die UperNet-Architektur [26], da diese mehrskalige Merkmale effektiv verarbeitet und in der Literatur als bewährter Standard zur Evaluierung neuer Vision-Backbones herangezogen wird [24], [25], [37].

Ein methodisches Hindernis beim Vergleich stellt jedoch die Struktur der ausgegebenen Merkmalskarten dar. Wie in Abschnitt 2.2.4 beschrieben, liefern reine ViTs lediglich einskalige Merkmale, während der UperNet-Decoder zwingend hierarchische Pyramidenstrukturen voraussetzt. Um diese Lücke zu schließen, kommt der *ViT-Adapter* [23] zum Einsatz.

Da die direkte Anwendung der Originalimplementierung dieses Adapters auf das Basismodell DINOv3 jedoch zu architektonischen Konflikten führt, erfordert der experimentelle Aufbau zwei essenzielle Modifikationen am Adapter:

- **Umgang mit Register-Tokens:** DINOv3 nutzt vier explizite *Register-Tokens*, die als Informationssenke für irrelevante Hintergrundbereiche dienen und so Artefakte in den Aufmerksamkeitskarten verhindern [39], [105]. Der originale ViT-Adapter erwartet jedoch eine rein räumliche Token-Sequenz ($N = H \times W$). Ein naives Abschneiden dieser Register-Tokens würde das Modell zwingen, das Rauschen auf die regulären Bild-Patches zu verteilen, was die semantische Reinheit der Merkmalskarten beeinträchtigt. Der Adapter wird daher dahingehend modifiziert, dass er diese zusätzlichen Tokens bei der räumlichen Rekonstruktion korrekt isoliert und verarbeitet, um die Absorptionsfunktion des Modells für irrelevante Hintergrundinformationen zu erhalten.

- **Entfernung der absoluten Positionskodierung:** Während Standard-ViTs absolute, lernbare Positionseinbettungen (`pos_embed`) addieren, verwendet DINOv3 *Rotary Positional Embeddings* (RoPE) [39], [106]. Hierbei wird die räumliche Information mathematisch durch Rotation in den Aufmerksamkeitsschichten kodiert. Der Standard-ViT-Adapter fügt jedoch standardmäßig ein zufällig initialisiertes, absolutes `pos_embed` hinzu. Da DINOv3 hierfür keine vortrainierten Gewichte besitzt, würde das Modell ein zufälliges Rauschmuster

lernen und die geometrische Wahrnehmung stören. Um dies zu verhindern, entfällt diese Addition vollständig aus der Architektur des Adapters.

Aufbauend auf dieser harmonisierten Schnittstelle treten fünf methodisch unterschiedliche Backbone-Strategien (deren theoretische Grundlagen in Abschnitt 2.2.4 behandelt wurden) gegeneinander an:

1. **Klassische CNN-Basislinie (ResNet):** Initialisiert mit überwacht trainierten ImageNet-Gewichten [14], fungiert dieser Ansatz als Basislinie, um den messbaren Mehrwert modernerer Transformer-Architekturen zu quantifizieren.
2. **Moderne CNN-Architektur (InternImage):** Um zu prüfen, ob fortschrittliche faltungsbasierte Modelle mit deformierbaren Faltungen [25] durch ihren induktiven Bias im Low-Data-Regime stabiler generalisieren als reine Transformer, wird dieser Ansatz evaluiert.
3. **Hierarchische Transformer (Swin):** Ebenfalls mit ImageNet-Gewichten initialisiert, repräsentiert dieses Modell [24] den Transfer von faltungsähnlichen, hierarchischen Strukturen in das Attention-Paradigma.
4. **Basismodelle (DINOv3):** Um den Einfluss der Vortrainingsdomäne bei massiven, selbstüberwachten Modellen [39] zu untersuchen, treten zwei Varianten gegeneinander an: das auf Alltagsbildern trainierte Standardmodell (*LVD-1689M*) sowie die domänenspezifische Variante (*SAT-493M*), die auf optischen Maxar-Satellitenaufnahmen (0,6 m GSD) trainiert wurde.
5. **Domänenspezifisches SSL-Vortraining:** Um eine maximale Domänennähe zu erreichen, wird ein eigener ViT-Backbone von Grund auf neu mittels Masked AE [37] auf unannotierten bayerischen DOP20-Luftbildern trainiert. Als Datengrundlage dienen hierfür Kacheln, die laut Katasterdaten mindestens 400 Gebäude aufweisen (siehe Abbildung A.1).

Dieser Versuchsaufbau ermöglicht eine systematische Evaluierung, ob für die vorliegende Aufgabe eher die Architektur (CNN vs. ViT), die Art des Vortrainings (Supervised vs. SSL) oder die Domänennähe der Trainingsdaten (ImageNet vs. DOP20) ausschlaggebend für die spätere Segmentierungsqualität ist.

Decoder-Auswahl

Nachdem die erste Phase der Architektursuche den optimalen Merkmalsextraktor identifiziert hat, fokussiert sich die zweite Phase auf die Evaluierung der Decoder-Architektur. Der Decoder hat die Aufgabe, die räumlich verdichteten und semantisch angereicherten Merkmalskarten des Backbones auf die ursprüngliche Bildauflösung hochzuskalieren und jedem Pixel eine finale Klassenzugehörigkeit zuzuweisen. Um

den isolierten Einfluss des Decoders zu quantifizieren, koppelt das experimentelle Setup in dieser Phase alle Decoder-Kandidaten an den in der vorherigen Phase ermittelten leistungsstärksten Encoder sowie ergänzend an die effizienteste Modellalternative.

Um ein repräsentatives Spektrum an Dekodierungsstrategien abzudecken, vergleicht diese Untersuchung vier etablierte Paradigmen, deren architektonische Eigenschaften in Abschnitt 2.2.5 vorgestellt wurden:

- **Pyramid Pooling Baseline (UperNet):** Wie bereits in der Backbone-Evaluierung als konstanter Segmentierungskopf genutzt, dient UperNet [26] hier als methodische Basislinie für die hierarchische Merkmalsfusion.
- **Atrous Spatial Pyramid Pooling (DeepLabV3+):** Dieser Ansatz [27] wird evaluiert, um die Leistungsfähigkeit klassischer dilatierter Faltungen bei der Rekonstruktion feiner Dach- und Versiegelungsstrukturen zu prüfen.
- **Leichtgewichtiger MLP-Decoder (SegFormer):** Um die architektonische Effizienz in den Fokus zu rücken, prüft das Setup, ob die rein linearen Schichten dieses Decoders [28] ausreichen, um die bereits hochgradig semantischen Merkmale des Backbones fehlerfrei auf die Bildebene zu projizieren.
- **Maskenklassifikations-Paradigma (Mask2Former):** Als Vertreter des aktuellen Stands der Technik testet diese Variante [29], inwiefern die Umformulierung der Segmentierung in ein Maskenklassifikationsproblem konkrete Vorteile bei der Unterscheidung feingliedriger urbaner Klassen im Low-Data-Regime bietet.

Durch diese gezielte Auswahl lassen sich klassische faltungsbasierte Ansätze, extrem leichtgewichtige lineare Decoder und komplexe, aufmerksamkeitsbasierte Maskierungsstrategien direkt miteinander vergleichen. Die Evaluierung zeigt auf, welches Aggregationsparadigma unter dem vorliegenden Low-Data-Regime die extrahierten Merkmale am effizientesten und genauesten in die finale Inferenzmaske überführt.

Hyperparameter-Optimierung

Aufbauend auf den in Abschnitt 2.4.3 dargelegten theoretischen Vorteilen der Kombination aus Bayesian Optimization und Hyperband (BOHB) nutzt die praktische Umsetzung dieser Arbeit das Framework *Optuna* [93]. Während das ursprüngliche BOHB-Verfahren auf einer Kerneldichteschätzung (engl. *Kernel Density Estimation*, KDE) basiert [45], nutzt *Optuna* standardmäßig einen *Tree-Structured Parzen Estimator* (TPE) [107] als Surrogatmodell in Kombination mit dem asynchronen

Hyperband-Pruning. Diese TPE-basierte Variante bietet äquivalente synergetische Eigenschaften bezüglich Ressourceneffizienz und Konvergenzgeschwindigkeit, erlaubt jedoch eine deutlich flexiblere technische Integration in die ressourcenbeschränkte Hardware-Umgebung dieser Arbeit.

Um eine möglichst belastbare Schätzung der mittleren Generalisierungsleistung zu erhalten, ist die Leistungsfähigkeit einer Hyperparameter-Konfiguration idealerweise über alle K Teilmengen der Trainingsdaten zu evaluieren. Eine vollständige K-fache Kreuzvalidierung in jedem Optimierungsschritt erhöht den Rechenaufwand jedoch erheblich und schränkt die Anzahl der untersuchbaren Konfigurationen stark ein.

Um dieser Einschränkung zu begegnen, definiere ich eine gezielte Auswahlstrategie: Anstatt in jedem Schritt eine zufällige Teilmenge des Trainingsdatensatzes zu ziehen, was eine hohe stochastische Variabilität der Zielfunktion induziert und den Konvergenzprozess des TPE-Optimierers durch „Rauschen“ in den beobachteten Metriken behindert, evaluiert das System die Leistung auf einer dedizierten Teilmenge. Diese Teilmenge wird vorab so selektiert, dass sie die geringste L_1-Differenz der Klassenverteilung im Vergleich zum Gesamtdatensatz aufweist. Dadurch lassen sich Unterschiede in der Modellleistung direkt auf die jeweiligen Hyperparameter-Konfigurationen zurückführen. Gleichzeitig sinkt der Rechenaufwand, was die Untersuchung einer größeren Anzahl von Hyperparameter-Konfigurationen innerhalb des verfügbaren Budgets ermöglicht.

Um den optimalen Hyperparameter-Satz für diese Architektur zu finden, durchläuft der Prozess insgesamt 80 Trainingsläufe. Der asynchrone Hyperband-Pruner steuert die Ressourcenallokation und bricht unzureichende Konfigurationen frühzeitig ab. Dabei kommt ein Reduktionsfaktor von 3 zum Einsatz, wodurch das System in jedem Evaluierungsschritt nur das obere Drittel der verbleibenden Trainingsläufe in die nächste Phase übernimmt. Um jedoch zu verhindern, dass vielversprechende Konfigurationen in der frühen Trainingsphase vorzeitig abbrechen, lege ich die minimale Ressourcenallokation auf 2 500 Iterationen fest, was exakt 25 % des gesamten Trainingsbudgets pro Trainingslauf entspricht. Diese gezielte Verzögerung des ersten Pruning-Schritts gewährt jedem Trainingslauf eine ausreichende Aufwärmphase, um das anfängliche Rauschen im Trainingsprozess zu überwinden, bevor die Leistungsbewertung stattfindet. Der Suchraum ist so definiert, dass die Optimierung sowohl architekturspezifische Parameter als auch Aspekte der Trainingsdynamik und Datenaugmentierung einschließt. Die Intervalle der jeweiligen Wertebereiche sind dabei an der leistungsstärksten Konfiguration der vorherigen Evaluierungsphase orientiert, um die Suche effizient einzugrenzen. Tabelle 3.4 fasst die genauen Konfigurationen zusammen.

Besonderer Fokus liegt auf der Lernrate (`lr`) und dem Multiplikator für den Decoder (`head_lr_mult`). Letzteren integriere ich, um dem Modell zu ermöglichen,

die vortrainierten Backbone-Gewichte langsamer zu aktualisieren als die zufällig initialisierten Gewichte des Segmentierungskopfes. Für beide Parameter wähle ich eine logarithmische Skala, um eine feinere Granularität in niedrigen Wertebereichen zu erreichen.

Zusätzlich umfasst der Suchraum die Regularisierung durch die Drop Path Rate sowie die Gewichtung des Weight Decay (`wd`). Die Trainingsdynamik steuert eine Aufwärmphase, welche den Anteil der Iterationen bestimmt, in denen die Lernrate linear ansteigt. Ein weiterer Bestandteil der Optimierung ist die Augmentierungswahrscheinlichkeit (`aug_prob`). Dieser Parameter fungiert als globaler Koeffizient, um die Intensität von Transformationen wie zufälliger Rotation, Gauß-Rauschen und *Coarse Dropout* (grobes Auslassen) einheitlich zu steuern.

Tabelle 3.4 – Definition des Hyperparameter-Suchraums für die *Optuna*-Optimierung.

Parameter	Wertebereich	Skalierung	Beschreibung
`lr`	$[10^{-6}, 10^{-4}]$	Log	Basislernrate
`head_lr_mult`	$[2, 10]$	Log	LR-Faktor für den Head
`drop_path_rate`	$[0.0, 0.4]$	Linear	Stochastische Tiefen-Regularisierung
`warmup_ratio`	$[0.05, 0.20]$	Linear	Anteil der Warmup-Phase
`weight_decay`	$[10^{-4}, 0.1]$	Log	L2-Regularisierung
`aug_prob`	$[0.1, 0.7]$	Linear	Augmentierungs-Wahrscheinlichkeit

3.3.4 Standardisierung und Fairness im Modellvergleich

Um belastbare Aussagen über die tatsächliche Leistungsfähigkeit der verschiedenen Architekturansätze treffen zu können, ist ein methodisch fairer Vergleich unerlässlich. Eine reine Gegenüberstellung der Metriken wäre aussagelos, wenn Leistungsunterschiede lediglich auf variierende Trainingsbedingungen oder eine unausgewogene Ressourcenverteilung zurückzuführen wären. Um diese systematischen Verzerrungen zu eliminieren, definiert der experimentelle Aufbau eine strenge Trennung zwischen konstanten (modellunabhängigen) und flexiblen (modellspezifischen) Parametern.

Konstante modellunabhängige Parameter

Alle Trainingsaspekte, die nicht inhärent an eine spezifische Netzwerkarchitektur gebunden sind, bleiben über alle Experimente hinweg strikt konstant. Dies umfasst die Wahl der Verlustfunktion, die Bildgröße beim Training (engl. *Crop Size*), die angewandte Pipeline zur Datenaugmentierung sowie die absolute Trainingsdauer, gemessen in Optimierungsschritten (Iterationen).

Eine besondere methodische und technische Herausforderung bei der Standardisierung stellt die Batch-Größe dar. Da große Transformer-Modelle (wie ViT

oder DINOv3) aufgrund ihrer Parameteranzahl und der rechenintensiven Aufmerksamkeitsmechanismen den verfügbaren Grafikspeicher der Hardware-Umgebung bei großen Batches überschreiten, wären diese im Vergleich zu speichereffizienteren Modellen (wie ResNet) benachteiligt. Um das Signal-zu-Rauschen-Verhältnis der Gradientenschätzung für alle Modelle identisch zu halten, ohne das System zu überlasten, erzwingt das Setup eine einheitliche *effektive* Batch-Größe mittels Gradientenakkumulation. Hierbei akkumuliert der Optimierer die Gradienten bei speicherintensiven Modellen über mehrere kleinere Vorwärtsdurchläufe, bevor er die Gewichte aktualisiert. Die Optimierungsdynamik bleibt somit über alle Architekturen hinweg mathematisch äquivalent.

Modellspezifische Parameter und Lernratenoptimierung

Neben den Konstanten erfordern modellspezifische Parameter eine isolierte Betrachtung. Interne Architekturdetails (wie Dropout-Raten innerhalb der Transformer-Blöcke oder spezifische Initialisierungsstrategien) orientieren sich grundsätzlich an etablierten Standards aus der Literatur sowie den Standardwerten der jeweiligen Frameworks.

Einen Sonderfall stellt jedoch die Basislernrate dar: Da sie maßgeblich die Konvergenzgeschwindigkeit und Stabilität bestimmt, führt eine statische Vorgabe für alle Architekturen zu einem unfairen Vergleich: Eine für ResNet optimale Lernrate kann bei einem Transformer-Modell zu sofortiger Divergenz führen. Um jedem Modell faire Konvergenzbedingungen zu bieten, ohne den Rechenaufwand durch eine vollständige Rastersuche über alle Teilmengen der Kreuzvalidierung zu sprengen, integriert das Design eine vorgeschaltete, isolierte Ablationsstudie.

Für jede zu evaluierende Architektur testet dieser Schritt drei repräsentative Lernraten (z. B. $10^{-4}, 5 \cdot 10^{-5}, 10^{-5}$). Um Ressourcen zu schonen, beschränkt sich dieses Vortraining ausschließlich auf die erste Datenaufteilung (Teilmenge 1). Diejenige Lernrate, die auf dem Validierungsset dieser ersten Teilmenge die höchste mIoU erzielt, gilt als modellspezifisches Optimum. Das System fixiert diesen Wert anschließend und verwendet ihn für das vollständige Training über alle weiteren Teilmengen im eigentlichen, groß angelegten Modellvergleich. Dieses Vorgehen stellt sicher, dass jede Architektur in einem für sie geeigneten Parameterbereich operiert und nicht durch unpassende Standardwerte systematisch benachteiligt wird, während der rechnerische Gesamtaufwand in einem realisierbaren Rahmen bleibt.

3.3.5 Kriterien für die Modellauswahl

Der finale Entscheidungsprozess zur Auswahl der optimalen Backbone-Decoder-Kombination nach Abschluss der Architektursuche stützt sich primär auf die quantitative Auswertung der Inferenzqualität. Als absolut ausschlaggebendes Hauptkriteri-

um dient hierbei die erreichte mIoU auf dem unabhängigen Validierungsdatensatz. Dieses Vorgehen stellt sicher, dass die Entscheidung für eine finale Zieldomänen-Architektur unmittelbar an die empirisch höchste nachweisbare Generalisierungs- und Differenzierungsleistung gekoppelt ist.

Neben diesem primären Leistungsindikator fließt als sekundäres Kriterium die Komplexität der Modelle in die Betrachtung ein. Insbesondere die Anzahl der trainierbaren Gewichte dient hierbei als Richtwert für die rechnerische Effizienz. Sofern zwei Architekturen eine nahezu identische Segmentierungsleistung (marginale mIoU-Differenzen im Nachkommastellenbereich) aufweisen, präferiert der Auswahlprozess das Modell mit der geringeren Parameteranzahl. Diese nachgelagerte Berücksichtigung der Effizienz dient dem Ziel, den Speicherbedarf und die Inferenzzeiten in der limitierten Umgebung mit einer einzelnen Grafikkarte gering zu halten, ohne Kompromisse bei der Segmentierungsqualität einzugehen. Dennoch ordnet sich die Parameteranzahl der mIoU stets unter, da die höchste räumliche Genauigkeit für die anschließende Flächen- und Solarpotenzialanalyse zwingend erforderlich ist.

3.4 Aufbau und Anwendung des Zieldatensatzes

Mit der Ermittlung einer optimalen, ressourceneffizienten Modellarchitektur ist das algorithmische Fundament gelegt. Um diese Architektur jedoch für die anvisierten stadtökologischen Analysen in der bayerischen Zieldomäne nutzbar zu machen, bedarf es einer spezifischen Datenbasis. Dieser Abschnitt beschreibt den vollständigen Prozess der Datensatzerstellung: von der systematischen Flächenauswahl über die Definition eines praxisrelevanten Klassenschemas bis hin zur Qualitätssicherung und der Festlegung der finalen Evaluierungsstrategie.

3.4.1 Akquise und Auswahl der Gebiete

Die Erstellung eines eigenen, hochqualitativen Datensatzes für semantische Segmentierung ist mit einem erheblichen zeitlichen Aufwand verbunden, wie die Voruntersuchung (vgl. Abschnitt 3.2.2) zeigte. Um eine hohe Generalisierungsfähigkeit auf diverse urbane Strukturen zu gewährleisten, konzentriert sich die Auswahlstrategie auf zwei mittelfränkische Großstädte: Nürnberg und Erlangen. Diese Kombination bietet eine repräsentative Mischung aus historischen Stadtkernen, modernen Industriearealen, universitären Campus-Strukturen und suburbanen Wohngebieten.

Die Definition der Untersuchungsgebiete erfolgt kachelbasiert. Als Kachelgröße wähle ich eine Fläche von 512×512 Metern. Bei einer GSD von 0,2 m entspricht dies einer Bilddimension von 2560×2560 Pixeln. Diese Dimensionierung bietet drei wesentliche Vorteile:

1. **Kontextuelle Integrität:** Die Fläche ist groß genug, um zusammenhängende städtebauliche Kontexte abzubilden (z. B. komplette Häuserblocks oder Industriehallen inkl. Außenanlagen), aber klein genug, um sie über das Stadtgebiet zu verteilen und so die lokale Varianz zu maximieren.

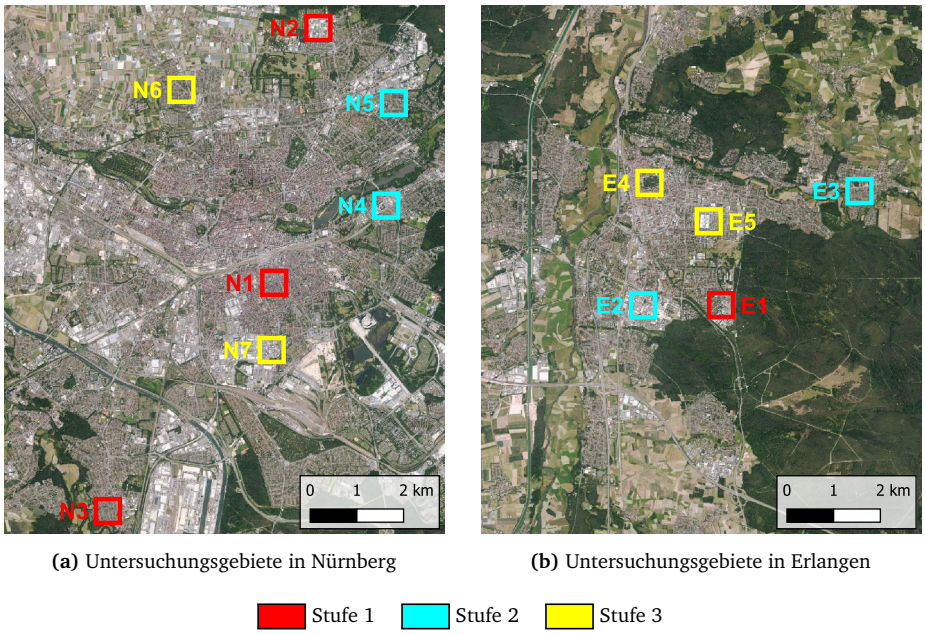
2. **Technische Kompatibilität:** Die Bilddimension von 2560 Pixeln lässt sich ohne Rest durch gängige Eingabegrößen (hier: 512 px) teilen. Dies minimiert Artefakte, die durch das Auffüllen der Ränder (engl. *Padding*) entstehen könnten.

3. **Datenvermehrung durch überlappende Bildausschnitte:** Die gewählte Kachelgröße erlaubt die effiziente Generierung einer großen Anzahl von Trainingsbildausschnitten durch einen systematischen, überlappenden Zuschnitt. Anstatt die Kacheln lediglich disjunkt zu zerlegen, ermöglicht die Wahl einer Schrittweite (engl. *Stride*), die kleiner als die Größe eines Bildausschnitts ist, eine signifikante Erhöhung der Datenmenge. Mathematisch berechnet die folgende Gleichung die Anzahl der extrahierten Bildausschnitte $N_{patches}$ pro Kachel bei einer Bildbreite W, einer Bildausschnittsgröße P und einer Schrittweite S:
$$N_{patches} = \left(\frac{W - P}{S} + 1 \right)^2 \tag{3.3}$$
Dies entspricht einer künstlichen Vervielfachung der Trainingsbasis ohne zusätzlichen Annotationsaufwand und erhöht die Generalisierungsfähigkeit des Modells [31].

Um eine objektive und diverse Auswahl der Kacheln zu gewährleisten, stützt sich die Selektion nicht rein auf visuelle Eindrücke, sondern greift auf amtliche Katasterdaten zurück. Konkret dient die „Tatsächliche Nutzung“ von Flächen als Entscheidungsgrundlage [54]. Die selektierten Kacheln decken ein möglichst breites Spektrum an relevanten Landnutzungsklassen ab (Wohnbauflächen, Industrie- und Gewerbeflächen, Flächen gemischter Nutzung, Verkehrsflächen).

Ein für die Qualitätssicherung essenzielles Kriterium ist die Verfügbarkeit von *Google Street View*-Aufnahmen. Da die Annotation (siehe Abschnitt 3.4.3) rein auf der Draufsicht (Orthophoto) basiert, dient Street View als Referenzebene, um bei mehrdeutigen Objekten (z. B. Unterscheidung zwischen Dachfenster und PV-Anlage) die korrekte Klasse zu verifizieren. Der Auswahlprozess behandelt Gebiete ohne diese Abdeckung daher nachrangig.

Angesichts des hohen manuellen Annotationsaufwands wähle ich ein dreistufiges Vorgehensmodell (siehe Abbildung 3.6 und Tabelle 3.5). Dieser Ansatz garantiert, dass zu jedem Zeitpunkt des Projekts ein in sich geschlossener, repräsentativer Datensatz vorliegt, der bei verfügbarer Restzeit modular erweitert werden kann.



(a) Untersuchungsgebiete in Nürnberg **(b)** Untersuchungsgebiete in Erlangen

Abbildung 3.6 – Räumliche Verteilung der ausgewählten Untersuchungsflächen in Nürnberg und Erlangen. Die Farbcodierung zeigt die Zuordnung zu den drei Ausbaustufen.

Tabelle 3.5 – Detaillierte Charakterisierung der Untersuchungsgebiete in Nürnberg und Erlangen.

Nürnberg

ID	Stufe	Gebietstyp	Lage
N1	**1**	Dichte Bebauung	Galgenhof
N2	**1**	Gewerbe	Mooshof
N3	**1**	Stadtrand	Eibach
N4	**2**	Gewerbe	Ostring
N5	**2**	Stadtrand	Steinplatte
N6	**3**	Stadtrand	Wetzendorf
N7	**3**	Industrie	Frankenstr.

Erlangen

ID	Stufe	Gebietstyp	Lage
E1	**1**	Campus	FAU-Südgelände
E2	**2**	Industrie	Bruck
E3	**2**	Stadtrand	Uttenreuth
E4	**3**	Innenstadt	Zentrum
E5	**3**	Gewerbe	Healthineers

- **Ausbaustufe 1 (Basisdatensatz):** Diese Ausbaustufe umfasst Kacheln mit der höchsten Priorität. Die Auswahl dieser Kacheln stellt sicher, dass sie bereits für sich genommen als minimal funktionsfähiger Datensatz eine maximale Varianz der Klassen abbilden. Das Spektrum reicht dabei von dichter urbaner Bebauung (Nürnberg-Galgenhof) und moderner Campus-Architektur (Erlangen, FAU-Südgelände) über gewerbliche Flächen (Nürnberg-Mooshof) bis hin zu suburbanen Strukturen am Stadtrand (Nürnberg-Eibach).

- **Ausbaustufe 2 (Erweiterung):** Diese Ausbaustufe erweitert den Datensatz um zusätzliche $\approx 1\,\mathrm{km}^2$ und fokussiert sich darauf, die Trainingsbeispiele für unterrepräsentierte Klassen (z. B. Wasserflächen) zu erhöhen.

- **Ausbaustufe 3 (Sättigung):** Die finale Ausbaustufe deckt Randfälle ab und steigert die Generalisierungsfähigkeit des Modells durch die Integration weiterer Randbezirke.

Die schrittweise Erweiterung sorgt in Abhängigkeit vom Bearbeitungsfortschritt für eine durchgehend ausgewogene Datenbasis, die beispielsweise Wohn- und Industriegebiete gleichermaßen abdeckt. Die räumliche Verteilung der Ausbaustufen über beide Städte hinweg verhindert zudem eine Überanpassung des Modells an lokale Eigenheiten einer einzelnen Stadt oder eines spezifischen Quartiers.

3.4.2 Klassendefinition

Die Definition des semantischen Klassenschemas bildet das inhaltliche Fundament der manuellen Datenannotation und determiniert maßgeblich die Leistungsfähigkeit der nachgelagerten Analysen. Die Auswahl der finalen 15 Klassen orientiert sich an einem praxisnahen Kompromiss zwischen zwei zentralen Anforderungen: Einerseits müssen die Klassen eine hohe Relevanz für etablierte urbane Nachhaltigkeits- und Klimaindikatoren (wie beispielsweise den DGNB-Klimafaktor oder den Abflussbeiwert) aufweisen. Andererseits begrenzen die sensorischen Limitierungen der optischen Orthophotos die Auswahl strikt. So sind beispielsweise fassadengebundene Begrünungen (Grünwände) aufgrund der reinen Nadirperspektive visuell nicht erfassbar, weshalb ich sie folglich ausschließe.

Das resultierende Schema fokussiert sich daher auf jene Bodenbedeckungen, die einen quantifizierbaren Einfluss auf die urbane Klimabilanz haben und die eine Differenzierung auf Basis der DOPs erlauben. Zudem legt das Schema einen besonderen Schwerpunkt auf eine unüblich feine Aufschlüsselung von Dachstrukturen, um die geometrische Grundlage für die spätere detaillierte Solarpotenzialanalyse zu schaffen.

Um die semantische Übersichtlichkeit zu gewährleisten, unterteile ich die 15 definierten Klassen in drei funktionale Überkategorien:

Vegetation und natürliche Oberflächen

Diese Kategorie umfasst alle unversiegelten, natürlichen Flächen, die primär für ökologische Auswertungen und die Analyse von Kaltluftentstehungsgebieten relevant sind.

- **Hohe Vegetation (*Tall Vegetation*):** Umfasst Bäume, Sträucher, Hecken und das Waldkronendach. Die Klasse schließt sichtbare Kronenränder auch bei geringer Ausdehnung ein, vernachlässigt jedoch Dachbegrünungen sowie bodennahe Gartenbepflanzungen.
- **Niedrige Vegetation (*Low Vegetation*):** Beinhaltet Rasenflächen, Grünstreifen, Wiesen und kleine krautige Pflanzen. Sie schließt spärlich bewachsene Schotterflächen aus, da die Klassifizierung primär nach dem Substrat und nicht nach vereinzelter Vegetation erfolgt.
- **Unbewachsener Boden (*Unvegetated Substrate*):** Diese Klasse umfasst offene Erdböden ohne Vegetationsdecke und schließt Sandhaufen sowie trockene Felder ein.
- **Schotter und Kies (*Crushed Stone / Gravel*):** Beschreibt Flächen, auf denen kleine Steine dominieren, wie Schotterwege, industrielle Kiesflächen oder Baustellenböden. Bei einer starken Durchmischung mit Vegetation bestimmt die visuell dominantere Klasse die Zuordnung.
- **Gewässer (*Water Bodies*):** Diese Klasse bündelt Flüsse, Seen, Teiche, Kanäle und Rückhaltebecken.

Urbane Versiegelung

Diese Klassen dienen der detaillierten Quantifizierung des Abflussbeiwerts und unterscheiden sich vor allem in ihrer Albedo (Helligkeit) und Wasserdurchlässigkeit.

- **Geringe Teilversiegelung (*Low Partial Sealing*):** Erfasst Oberflächen mit weiten, stark begrünten Fugen (z. B. Rasengittersteine, versickerungsfähige Ökopflaster).
- **Starke Teilversiegelung (*Strong Partial Sealing*):** Umfasst überwiegend mineralische Oberflächen mit schmalen Fugen, die keine vollständige Versiegelung aufweisen (z. B. Kleinsteinpflaster).
- **Helle Vollversiegelung (*Light Hard Sealing*):** Bezeichnet Betonflächen, hellen Asphalt und helle Pflastersteine mit einer hohen Albedo.

- **Dunkle Vollversiegelung (*Dark Hard Sealing*):** Integriert stark hitzeabsorbierende Flächen wie dunkle Asphaltstraßen, Parkplätze und Gehwege.

Dachstrukturen und Solarpotenzial

Um Hindernisse für PV-Anlagen zu detektieren und mikroklimatische Effekte von Dächern zu bewerten, schlüssele ich die Gebäudeinfrastruktur hochgranular auf.

- **Begrüntes Dach (*Green Roof*):** Repräsentiert intentionell bepflanzte oder vegetationsbedeckte Dachflächen.
- **Dunkles Dach (*Dark Roof*):** Umfasst typische dunkle Eindeckungen (schwarz/grau/dunkelrot), wie sie beispielsweise durch Bitumen oder dunkle Ziegel entstehen.
- **Helles Dach (*Light Roof*):** Deckt helle Schindeln, Metalldächer oder hellgraue Beton-Flachdächer ab.
- **PV-Anlage (*Photovoltaic System*):** Erfasst Solarmodule sowohl auf Dächern als auch als Freiflächenanlagen. Aufgrund der visuellen Ununterscheidbarkeit in den vorliegenden Bilddaten inkludiere ich in dieser Klasse auch Solarthermieanlagen. Für die spätere Solarpotenzialanalyse ist diese Zusammenfassung unproblematisch, da beide Anlagentypen die Dachfläche funktional blockieren und somit bereits genutztes Potenzial repräsentieren.
- **Technische Dachaufbauten (*Technical Roof Structures*):** Integriert größere funktionale Infrastruktur wie Lüfter, Klimaanlagen, Kühlaggregate, Masten oder große Schornsteine, die ein signifikantes Hindernis darstellen.
- **Sonstige Dachaufbauten (*Other Roof Structures*):** Weist kleinere Strukturen wie Dachfenster, Lichtkuppeln oder kleine Kamine aus.

3.4.3 Annotationsstrategie und -durchführung

Aufgrund der in Abschnitt 3.1.2 dargelegten Komplexität bei der Interpretation von 20 cm-Luftbildern wird explizit auf einen klassischen Crowdsourcing-Ansatz verzichtet. Stattdessen erfolgt die systematische Annotation in einem kollaborativen Prozess durch ein Team studentischer Hilfskräfte, unter meiner fachlichen Koordination. Die Bearbeitung findet auf der zentralen *CVAT*-Instanz statt. Die Bilddaten werden dabei analog zur Definition der Untersuchungsgebiete strukturiert, um eine räumlich konsistente Aufgabenverteilung zu ermöglichen.

Zur Effizienzsteigerung und Reduktion geometrischer Inkonsistenzen integriert der Ansatz die *Hausumringe* der Bayerischen Vermessungsverwaltung [52] als Vorannotationen. Ein Vorverarbeitungsschritt konvertiert die Daten dafür in ein *CVAT*-kompatibles Format. Da diese Katasterdaten jedoch primär den Gebäudeumriss

auf Bodenhöhe und nicht den tatsächlichen Dachüberstand abbilden sowie keine semantischen Dachklassen aufweisen, dienen sie lediglich als geometrische Orientierungshilfe. Die Annotatoren passen die Polygone bei Bedarf manuell an das Orthophoto an und weisen die korrekte semantische Klasse zu. Alle weiteren Objektklassen werden vollständig händisch annotiert.

Um trotz der verteilten Aufgabenbearbeitung eine hohe semantische Konsistenz zu gewährleisten, entwickle ich einen detaillierten Annotationsleitfaden (siehe Anhang C.1) als methodische Grundlage für alle Beteiligten. Ich begleite die Studierenden während der Einarbeitungsphase beratend, um ein einheitliches Verständnis der Objektklassen sicherzustellen. Für Zweifelsfälle führe ich zudem eine zentrale Dokumentation: Tritt während der Annotation eine unklare Situation auf, wird diese dort erfasst und von mir nach einheitlichen Kriterien abschließend bewertet und aufgelöst.

Ein besonderer Fokus liegt dabei auf der Reduktion von Mehrdeutigkeiten, die sich aus der reinen Betrachtung der Orthophotos ergeben (etwa bei der Unterscheidung von Versiegelungsarten). Da das Klassenschema in dieser Phase deutlich feingranularer ist als in der Voruntersuchung, definiere ich für solche schwer entscheidbaren Fälle eine externe Verifikation:

1. ***Virtuelle Begehung:*** Nutzung von *Google Street View*[2] zur Überprüfung nicht eindeutig identifizierbarer Objekte oder der Bodenbeschaffenheit sowie ergänzende Auswertung hochauflösender Luftbilder und der 3D-Visualisierung aus Google Maps, um Dachstrukturen und Objektgeometrien aus schrägen Blickwinkeln zu analysieren.

2. ***Multitemporaler Abgleich:*** Nutzung historischer Satellitenbilder in Google Maps, um temporäre Objekte oder Schattenwürfe besser zu interpretieren.

3.4.4 Qualitätskontrolle und Nachbearbeitung der Referenzmasken

Zur Sicherstellung der finalen Datenqualität kommt ein zweistufiger Überprüfungsdurchlauf zum Einsatz. Nach Abschluss der initialen Bearbeitung durch die Hilfskräfte wechselt jeder Datensatz in den Status `Validation`. Die anschließende systematische visuelle Inspektion zeigt, dass die Qualität der Annotationen zum Teil stark variiert. Häufige Fehlerquellen sind komplett übersehene Gebiete, semantische Fehlklassifikationen (z. B. Verwechslung von Versiegelungsarten) sowie unpräzise Geometrien, die zu ungewollten Lücken zwischen benachbarten Polygonen führen.

[2] Alphabet Inc. (2026): Google Maps und Google Street View. Abgerufen am 01.03.2026 unter `https://www.google.de/maps`

Aufgrund des zeitlichen Aufwands, den eine manuelle Überprüfung und Korrektur von mehreren Quadratkilometern hochauflösender Bilddaten erfordert, setze ich für die Qualitätssicherung klare Prioritäten:

1. **Semantik vor Geometrie:** Der primäre Fokus liegt auf der Korrektur semantischer Falschzuweisungen sowie dem Nachtragen größerer, komplett unannotierter Bereiche. Auf eine detaillierte Nachbesserung feiner geometrischer Ungenauigkeiten (wie beispielsweise die exakte pixelgenaue Nachführung komplexer Dachränder) verzichte ich größtenteils. Eine solche Detailkorrektur würde den zeitlichen Rahmen der Arbeit überschreiten und kommt einer vollständigen Neuannotation gleich.

2. **Automatisierte Lückenschließung:** Ein spezifisches Problem sind feine, oft nur wenige Pixel breite Lücken zwischen angrenzenden Objekten, die durch ungenaues Zeichnen der Vektorpolygone entstehen. Anstatt diese feinen Risse manuell zu schließen, folgt ein automatisierter Nachbearbeitungsschritt. Nachdem eine visuelle Inspektion sicherstellt, dass keine signifikanten Flächen unannotiert bleiben, füllt eine Nächster-Nachbar-Interpolation (engl. *Nearest-Neighbor-Interpolation*) die verbleibenden Kleinstlücken auf. Dabei weist der Algorithmus jedem unklassifizierten Pixel (Hintergrund) automatisch die Klasse des räumlich nächstgelegenen annotierten Pixels zu. Dies garantiert eine flächendeckende, lückenlose Referenzmaske ohne unverhältnismäßigen manuellen Aufwand.

Um die Fehleranfälligkeit der Annotationsaufgabe auf Basis der 20 cm-DOP-Aufnahmen später objektiv quantifizieren zu können, dokumentiert die Qualitätssicherung den Korrekturaufwand. Hierzu vergleicht die mIoU-Berechnung die rohe, initiale Annotation der Hilfskräfte mit der final korrigierten Version. Diese berechneten Metriken bilden die Grundlage für die spätere Evaluierung der Annotationsqualität in Abschnitt 4.3.1.

Trotz des zweistufigen Überprüfungsdurchlaufs und der manuellen sowie automatisierten Korrekturen gehe ich davon aus, dass der finale Datensatz weiterhin ein gewisses Maß an Annotationsrauschen aufweist. Komplexe Verdeckungen durch Vegetation, tiefe Schattenwürfe oder historisch gewachsene, unscharfe Objektgrenzen (z. B. fließende Übergänge zwischen verschiedenen Versiegelungsgraden) lassen sich auf Orthophotos oft auch bei sorgfältiger Prüfung nicht zweifelsfrei auflösen.

Das Ziel der Qualitätssicherung ist es demnach nicht, einen unrealistischen, absolut fehlerfreien Datensatz zu generieren. Vielmehr liegt der Fokus darauf, systematische semantische Fehler zu bereinigen und das grundlegende Rauschen auf ein Maß zu reduzieren, das ein stabiles Modelltraining ermöglicht. Die Auswirkungen dieses inhärenten Rauschens auf die Segmentierungsleistung diskutiere ich in der anschließenden Analyse.

3.4.5 Umfang des finalen Datensatzes

Aufgrund des zeitlichen Aufwands der manuellen, pixelgenauen Annotation auf 20 cm-Orthophotos schließt die Datenerhebung nach der Fertigstellung von sieben Untersuchungsgebieten ab. Dies umfasst sämtliche Kacheln der Ausbaustufe 1 (Basisdatensatz) sowie die ersten drei Gebiete der Ausbaustufe 2 (Erweiterung). Damit stehen insgesamt sieben hochdetaillierte Referenzkacheln zur Verfügung.

Um die Datenmenge für das Training systematisch zu vergrößern, nutzt die Pipeline für die Extraktion der Trainings-Bildausschnitte den in Abschnitt 3.4.1 beschriebenen überlappenden Zuschnitt. Hierbei definiere ich für alle sieben Kacheln konsistent eine Größe der Bildausschnitte von 512×512 Pixeln bei einer Schrittweite von 256 Pixeln. Gemäß Gleichung (3.3) vervielfacht diese 50-prozentige Überlappung die Anzahl der generierten Trainingsbeispiele pro Kachel von 25 (bei rein disjunkter Zerlegung) auf 81. Dies erhöht die Trainingsstabilität des Modells im anvisierten Low-Data-Regime signifikant.

Neben dem reinen geometrischen Umfang ist für das anschließende Modelltraining insbesondere die semantische Zusammensetzung des Datensatzes entscheidend. Tabelle 3.6 schlüsselt die annotierte Gesamtfläche auf die 15 definierten Klassen auf und verdeutlicht das natürliche Ungleichgewicht der urbanen Struktur.

Tabelle 3.6 – Statistische Verteilung der 15 semantischen Klassen im finalen Anwendungsdatensatz. Die reale Fläche in Quadratmetern ergibt sich aus der Bodenauflösung von 20 cm pro Pixel ($1\,\text{px} = 0,04\,\text{m}^2$). Die absteigende Sortierung verdeutlicht das für urbane Räume typische, starke Klassenungleichgewicht.

Klasse	Fläche (m^2)	Anteil (%)
Dark Roof	404 363	22,04
Strong Partial Sealing	319 740	17,42
Tall Vegetation	314 689	17,15
Dark Hard Sealing	287 597	15,67
Low Vegetation	214 558	11,69
Light Roof	87 708	4,78
Crushed Stone / Gravel	63 277	3,45
Other Roof Structures	44 691	2,44
Green Roof	28 725	1,57
Photovoltaic System	26 449	1,44
Technical Roof Structures	15 538	0,85
Unvegetated Substrate	11 862	0,65
Low Partial Sealing	6 580	0,36
Water Bodies	5 276	0,29
Light Hard Sealing	3 955	0,22
Gesamt	**1 835 008**	**100,00**

3.4.6 Ablationsstudie zur multimodalen Datenfusion

Neben den klassischen RGB-Kanälen der DOP20-Aufnahmen stehen mit dem NIR-Kanal sowie Höheninformationen potenziell wertvolle zusätzliche Merkmale zur Verfügung. Um dem Modell insbesondere die Unterscheidung von Gebäudehöhen und Vegetation zu erleichtern, berechnet die Pipeline aus dem DOM und dem DGM ein normalisiertes digitales Oberflächenmodell (nDOM), welches die Höhe der Objekte über dem Geländeniveau repräsentiert.

Um zu evaluieren, inwiefern das Netzwerk von dieser multimodalen Datenbasis (RGB, NIR, nDOM) profitiert, umfasst die Methodik eine gezielte Ablationsstudie. Da die vortrainierten Backbones strukturell auf exakt drei Eingangskanäle (RGB) limitiert sind, vergleicht der Versuchsaufbau drei unterschiedliche Integrationsstrategien:

1. **RGB-Basislinie:** Das Modell nutzt ausschließlich die drei Standardkanäle, die Zusatzdaten entfallen.

2. **Vorgeschaltete Kanalfusion:** Die Eingabe besteht aus einem 5-Kanal-Tensor (RGB, NIR, nDOM). Um die Kompatibilität mit dem Backbone herzustellen, projiziert eine vorgeschaltete 1×1-Faltung diesen Tensor direkt vor dem Encoder auf drei Kanäle. Das Netzwerk lernt die optimale Gewichtung und Reduktion der Kanäle somit dynamisch während des Trainings.

3. **Strukturelle Gewichtserweiterung:** Die erste Faltungsschicht des vortrainierten Backbones wird strukturell erweitert, um fünf statt drei Kanäle zu verarbeiten. Um die durch das Vortraining bereits gelernten RGB-Merkmalsrepräsentationen beim Trainingsstart nicht zu verfälschen, initialisiert das System die Gewichte für die zwei neuen Kanäle (NIR, nDOM) strikt mit null.

Die detaillierte quantitative Auswertung dieser Ansätze ist dem nachfolgenden Analysekapitel (siehe Abschnitt 4.3.2) vorbehalten. Um jedoch die nachfolgenden methodischen Schritte (Testzeit-Datenaugmentierung und Inferenz) kohärent beschreiben zu können, wird an dieser Stelle auf das Ergebnis der Evaluierung vorgegriffen: Da sich die vorgeschaltete Kanalfusion als leistungsstärkste Variante erweist, bildet sie die architektonische Grundlage für die gesamte weitere Pipeline.

3.4.7 Evaluierungsstrategie

Um die Generalisierungsleistung der Architektur, die ich zuvor anhand des stellvertretenden FLAIR-Datensatzes wähle, belastbar auf dem eigenen Datensatz zu evaluieren, wendet diese Phase eine räumliche Leave-One-Out-Kreuzvalidierung an. Da der finale Datensatz aus exakt sieben räumlich getrennten Kacheln besteht,

gliedert sich der Validierungsprozess in sieben Iterationen, wie in Abbildung 3.7 schematisch dargestellt. In jedem Iterationsschritt trainiert das Modell auf sechs Kacheln, wobei die siebte Kachel die Funktion der ungesehenen Validierungsmenge übernimmt.

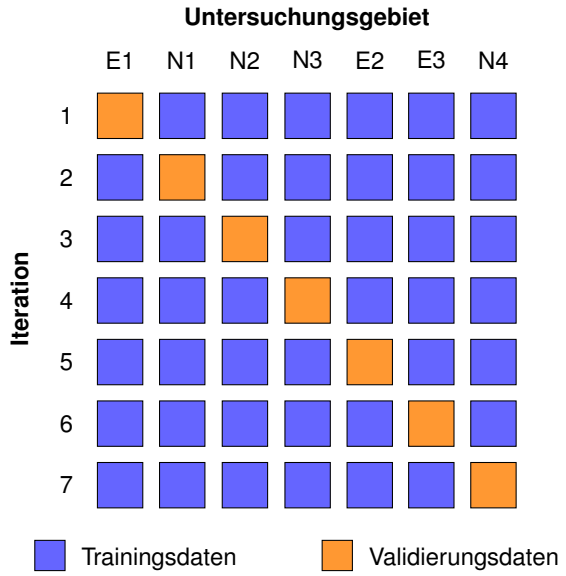


Abbildung 3.7 – Schematische Darstellung der räumlichen Leave-One-Out-Kreuzvalidierung. In jedem der sieben Iterationsschritte wird exakt ein Untersuchungsgebiet zur Validierung zurückgehalten, während die restlichen sechs Gebiete den Trainingsdatensatz bilden.

Dieses Vorgehen bietet für den vorliegenden Datensatz zwei entscheidende methodische Vorteile:

1. **Maximierung der Trainingsdaten:** In jedem Durchlauf nutzt das Verfahren ca. 85 % der annotierten Gesamtfläche für das Training. Bei der insgesamt ohnehin knappen Datenmenge ist dies essenziell, um das Modell ausreichend konvergieren zu lassen.

2. **Prüfung der räumlichen Generalisierung:** Durch die strikte Trennung kompletter Kacheln (und nicht etwa einzelner Bildausschnitte innerhalb derselben Kachel) verhindert diese Strategie ein Informationsleck durch räumliche Autokorrelation. Somit testet die Evaluierung das Modell realitätsnah darauf, wie gut es auf komplett ungesehene Stadtstrukturen und Architekturstile in anderen Stadtteilen generalisiert.

Aufgrund der limitierten Datenmenge (sieben Kacheln) verzichtet dieser experimentelle Aufbau bewusst auf die Aufteilung eines statischen, vollständig unabhängigen Testdatensatzes. Das Zurückhalten von beispielsweise zwei Kacheln für

einen statischen Testdatensatz würde die pro Durchlauf verfügbare Trainingsbasis von sechs auf vier Kacheln und damit um rund ein Drittel reduzieren, was in einem Low-Data-Regime wie diesem zu einer erheblichen Verschlechterung der Modellkonvergenz führen würde. Die gewählte Leave-One-Out-Kreuzvalidierung stellt hierfür den methodisch fundiertesten Kompromiss dar. Sie garantiert, dass für jede der sieben Kacheln exakt eine Vorhersage existiert, bei der das Modell die entsprechende Geometrie während des Trainings nicht gesehen hat (*Out-of-Fold*-Vorhersage). Da die grundlegende Architekturentscheidung und Hyperparameter-Optimierung bereits a priori auf dem unabhängigen FLAIR-Datensatz (vgl. Abschnitt 3.3.2) stattfanden, dient diese Kreuzvalidierung nicht der Modelloptimierung, sondern ausschließlich der Evaluierung. Dies minimiert den methodischen Bias, der üblicherweise entsteht, wenn Validierungsdaten für die Modellauswahl oder Hyperparameter-Optimierung verwendet werden.

3.4.8 Testzeit-Datenaugmentierung und finale Inferenz

Um die Generalisierungsfähigkeit der Vorhersagen bei der großflächigen Anwendung auf ungesehene Gebiete zu maximieren, erweitert die Pipeline den reinen Inferenzschritt um eine sogenannte Testzeit-Datenaugmentierung (engl. *Test Time Augmentation*, TTA) [31]. Anstatt das trainierte Modell nur einmal auf einen Bildausschnitt anzuwenden, generiert das System für jede Kachel mehrere augmentierte Ansichten. Konkret wendet die Pipeline hierfür eine Kombination aus geometrischen Spiegelungen (horizontal und vertikal) sowie Skalierungen auf verschiedene Bildgrößen (engl. *Multi-Scale Inference*) an.

Das Modell berechnet Vorhersagen für all diese transformierten Varianten. Anstatt jedoch finale, diskrete Klassenlabels zu mitteln, aggregiert das System die rohen Wahrscheinlichkeitsverteilungen der Vorhersagen. Nach Invertierung der räumlichen Transformationen, also der Rückspiegelung sowie der bilinearen Rückskalierung auf die Originalauflösung, berechnet die Pipeline den Mittelwert der Wahrscheinlichkeiten pro Pixel. Erst im allerletzten Schritt extrahiert der Algorithmus die finale Klassenzuweisung mittels einer Argmax-Operation. Durch diese Mittelung der Softmax-Ausgaben glättet das System die Unsicherheiten des neuronalen Netzes signifikant und reduziert typische Fragmentierungsfehler bei komplexen Objektgrenzen bereits auf algorithmischer Ebene, bevor harte Klassengrenzen gezogen werden.

Aufgrund der durch die Testzeit-Datenaugmentierung bereits erreichten hohen räumlichen und kontextuellen Konsistenz der Vorhersagen verzichtet die Pipeline im Anschluss bewusst auf jegliche formale Nachbearbeitung der Segmentierungsmasken. Der Einsatz regelbasierter topologischer Korrekturen, wie etwa das morphologische Schließen von Lücken oder das Herausfiltern vermeintlicher Rauschartefakte,

birgt bei derart hochauflösenden Daten ein unverhältnismäßiges Risiko der Überoptimierung. Eine zu aggressive Nachbearbeitung birgt stets die Gefahr, korrekte Modellvorhersagen durch harte Heuristiken künstlich zu verschlechtern. Das System greift daher nicht in die semantische Grundlogik des neuronalen Netzes ein. Alle nachfolgenden stadtökologischen Indikatoren und Solarpotenzialanalysen basieren somit direkt und unverfälscht auf den durch TTA aggregierten Inferenzmasken.

3.5 Anwendungsorientierte Verwertung und Evaluierung

Nachdem die vorangegangenen Schritte die Erzeugung räumlich konsistenter und semantisch präziser Segmentierungsmasken sichergestellt haben, fokussiert sich dieser Abschnitt auf deren anwendungsorientierte Verwertung. Das Ziel ist es, die hochdimensionalen, pixelbasierten Klassifikationen in aggregierte, quantifizierbare Kennzahlen zu überführen. Hierzu gliedert sich die nachfolgende algorithmische Auswertung in zwei inhaltliche Schwerpunkte: Zunächst stellt die Methodik die Berechnung stadtökologischer Flächenindikatoren vor, bevor sie die komplexe Pipeline zur Modellierung des lokalen Solarpotenzials aufzeigt.

3.5.1 Ableitung stadtökologischer Flächenindikatoren

Die generierten Segmentierungsmasken bilden die geometrische Grundlage zur Quantifizierung stadtklimatischer und ökologischer Parameter. Um den praktischen Mehrwert der feingranularen Klassifikation für die urbane Planungspraxis aufzuzeigen, berechnet der Algorithmus explorativ zwei Flächenindikatoren: den gebietsmittleren Abflussbeiwert und den Anteil nachhaltig genutzter Dachflächen. Die Pipeline berechnet diese Metriken kachelbasiert und demonstriert damit beispielhaft die Überführbarkeit der rein pixelbasierten Vorhersagen in aggregierte, planungsrelevante Kennzahlen.

Gebietsmittlerer Abflussbeiwert als Retentionsindikator

Zur Bewertung der urbanen Nachhaltigkeit und Klimaresilienz erfasst der Algorithmus das Niederschlags-Abfluss-Verhalten des Untersuchungsraums. Im Kontext des Schwammstadtprinzips ist ein möglichst hoher lokaler Wasserrückhalt ein zentrales Ziel, da er zur Reduktion von Abflussspitzen bei Starkregen beiträgt und über Verdunstungsprozesse das urbane Mikroklima positiv beeinflussen kann [108]. Anstatt Flächen lediglich binär in versiegelt und unversiegelt zu trennen, wähle ich einen gewichteten Ansatz, der die partielle Retentionsleistung verschiedener Oberflächen abbildet. Hierzu ordnet das Klassifikationsschema jeder Bodenbedeckungsklasse

einen normierten mittleren Abflussbeiwert $C_{\mathrm{m}} \in [0,1]$ gemäß E DIN 1986-100 [109] zu (Tabelle 3.7). Da die Fernerkundungsdaten keine detaillierten bautechnischen Parameter (wie Dachneigungen oder exakte Fugenanteile) liefern, greift das Verfahren auf repräsentative Standardwerte der Norm zurück. Für Gründächer geht die Methodik konservativ von einer Extensivbegrünung mit geringer Aufbaudicke (unter 10 cm) aus ($C_{\mathrm{m}} = 0,3$), während für sämtliche konventionellen Dach- und Asphaltflächen der Standardwert für wasserundurchlässige Flächen ($C_{\mathrm{m}} = 0,9$) angesetzt wird. Wasserflächen erhalten normgemäß den Wert 1,0, da Niederschlag auf bestehenden Gewässern nicht versickern kann und somit vollständig dem Oberflächenwasser zuzurechnen ist.

Der Algorithmus berechnet den gebietsmittleren Abflussbeiwert $\overline{C_{\mathrm{m}}}$ als arithmetisches Mittel der Faktoren aller validen Pixel:

$$\overline{C_{\mathrm{m}}} = \frac{1}{N} \sum_{i=1}^{N} C_{\mathrm{m}}(i) \tag{3.4}$$

wobei N die Anzahl der gültigen Pixel und $C_{\mathrm{m}}(i)$ den Abflussbeiwert der vorhergesagten Klasse des Pixels i darstellen. Je niedriger der aggregierte Wert, desto höher ist die natürliche Retentionsfähigkeit des untersuchten Raums.

Tabelle 3.7 – Klassenspezifische mittlere Abflussbeiwerte (C_{m}) zur Berechnung des gebietsmittleren Oberflächenabflusses, entnommen aus E DIN 1986-100 [109].

Klasse	**Faktor C_{m}**	**Klasse**	**Faktor C_{m}**
Tall Vegetation	0,1	Dark Roof	0,9
Low Vegetation	0,1	Light Roof	0,9
Unvegetated Substrate	0,1	Green Roof	0,3
Crushed Stone / Gravel	0,2	Photovoltaic System	0,9
Low Partial Sealing	0,25	Technical Roof Structures	0,9
Strong Partial Sealing	0,7	Other Roof Structures	0,9
Light Hard Sealing	0,9	Water Bodies	1,0
Dark Hard Sealing	0,9		

Anteil nachhaltig genutzter Dachflächen

Während der gebietsmittlere Abflussbeiwert die hydrologische Leistungsfähigkeit der gesamten Oberfläche betrachtet, rückt der zweite Indikator die urbane Gebäudeinfrastruktur in den Fokus. Dachlandschaften bergen ein erhebliches, oft ungenutztes Potenzial für eine nachhaltige Stadtentwicklung. Während Gründächer das Mikroklima kühlen und Niederschläge dezentral zurückhalten, leisten PV-Anlagen einen direkten Beitrag zur lokalen Energiewende. Der Indikator der nachhaltigen Dach-

nutzung quantifiziert, welcher Anteil der gesamten Dachinfrastruktur eines Gebietes bereits für diese multifunktionalen Zwecke aktiviert wurde.

Zur Berechnung aggregiert die Pipeline zunächst alle relevanten, ein Dach definierenden Klassen zu einer geometrischen Gesamtdachfläche. Der Indikator S_{roof} ergibt sich anschließend aus dem Verhältnis der kombinierten Gründach- und PV-Pixel (N_{green} und N_{pv}) zur Gesamtzahl der Dach-Pixel N_{roof}:

$$S_{\text{roof}} = \frac{N_{\text{green}} + N_{\text{pv}}}{N_{\text{roof}}} \tag{3.5}$$

Befinden sich im betrachteten Bildausschnitt keinerlei Gebäude ($N_{\text{roof}} = 0$), liefert der Algorithmus einen undefinierten Wert (`NaN`) zurück. Dies verhindert Laufzeitfehler und stellt sicher, dass unbebaute Gebiete nachgelagerte Auswertungen zum Dachbestand nicht statistisch verzerren.

3.5.2 Modellierung des lokalen Solarpotenzials

Dieser Abschnitt beschreibt detailliert die Methodik zur automatisierten Berechnung des Solarpotenzials. Abbildung 3.8 bietet vorab einen schematischen Überblick über die gesamte entwickelte Pipeline, ausgehend von den Eingangsrohdaten bis hin zur Extraktion der finalen Potenzial-Indikatoren. Die folgenden Unterkapitel erläutern die einzelnen Verarbeitungsschritte und Schnittstellen im Detail. Eine konsolidierte Übersicht aller in dieser Arbeit gewählten numerischen Schwellenwerte, Toleranzen und Systemparameter findet sich in Tabelle B.2 im Anhang. Da eine umfassende Sensitivitätsanalyse zur exakten Quantifizierung der Ergebnisrobustheit gegenüber Parameterabweichungen den Rahmen dieser Arbeit überschreiten würde, wurden die Parameter der Pipeline als empirische Konstanten festgelegt. Ihre Wahl basiert auf den technischen Spezifikationen der Eingangsdaten sowie iterativer visueller Inspektion.

Extraktion und geometrische Verfeinerung der Dachflächen

Die Basis der geometrischen Modellierung bilden öffentlich verfügbare LiDAR-Punktdaten [53] und LoD2-CityGML-Daten [51], aus denen die Pipeline initial die 3D-Koordinaten der Dachflächen extrahiert, um Dachneigung und Ausrichtung (Azimut) abzuleiten. Obwohl LoD2-Modelle eine essenzielle topologische Grundlage bieten, weisen sie aufgrund stark generalisierender photogrammetrischer Erzeugungsprozesse häufig systematische Ungenauigkeiten auf. Diese Abweichungen sind für die nachgelagerte Hinderniserkennung kritisch: Weist eine LoD2-Ebene beispielsweise einen leichten vertikalen Versatz oder eine ungenaue Neigung auf, liegt die tatsächliche LiDAR-Punktwolke des Daches schnell außerhalb der definierten Distanztoleranz. Dies führt dazu, dass der Algorithmus reguläre Dachpunkte

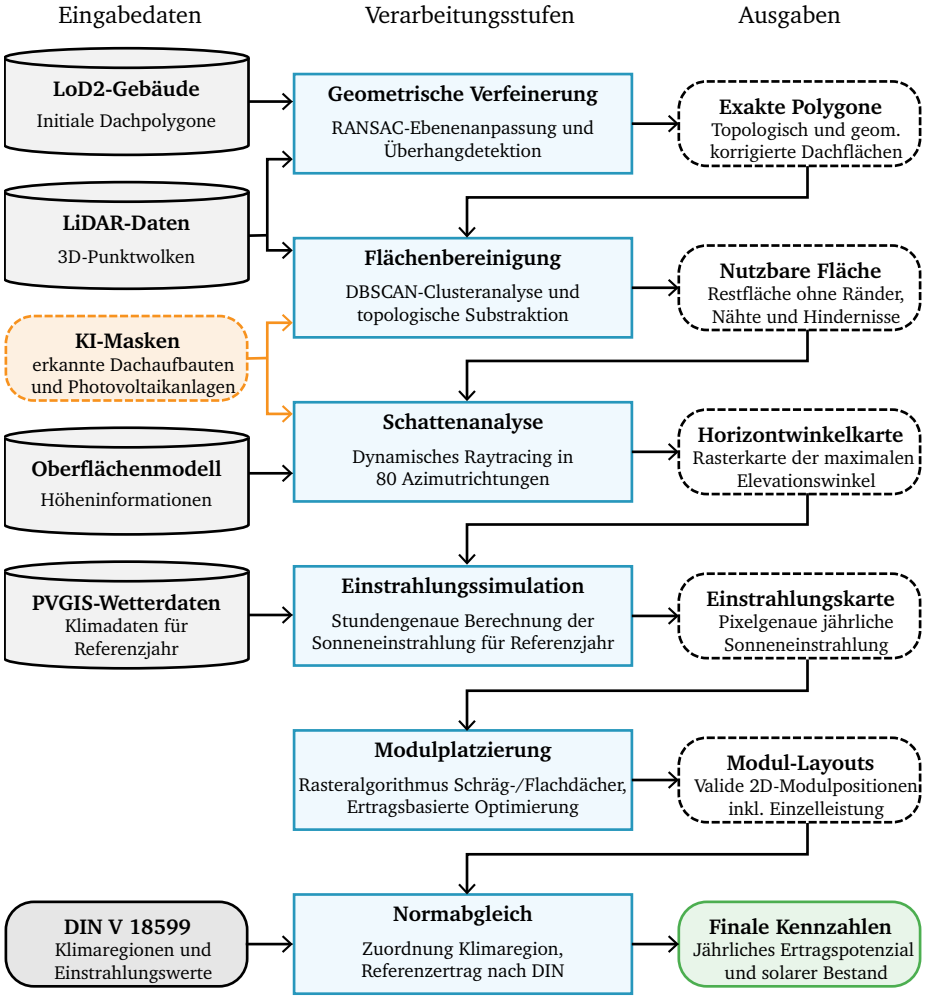


Abbildung 3.8 – Schematischer Aufbau der entwickelten Pipeline von den Eingangsdaten bis zu den finalen Indikatoren. Gestrichelte Konturen kennzeichnen temporäre Zwischenergebnisse, während durchgezogene Rahmen die finalen Endprodukte markieren. Der orange hervorgehobene Knoten kennzeichnet die Schnittstelle zu den Ergebnissen der semantischen Segmentierung (geom. = geometrisch).

fälschlicherweise als Ausreißer deklariert und als Störobjekte von der nutzbaren Solarfläche abzieht, wie Abbildung 3.9a veranschaulicht.

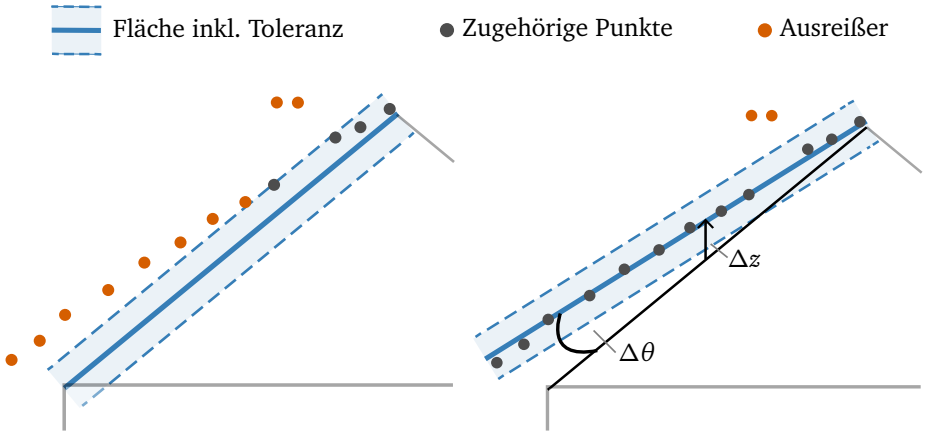
Um derartige falsch positive Klassifikationen zu vermeiden, dienen die rohen LoD2-Polygone lediglich als geometrische Initialannahme. Im ersten Schritt justiert das System diese Ebenen anhand der lokalen LiDAR-Punkte neu. Hierfür wendet die Pipeline den RANSAC-Algorithmus [58] auf die innerhalb der Gebäudegrundrisse liegenden Punktwolken an, um die Ebenengleichung lokal anzupassen. Der Algorithmus akzeptiert eine solche Anpassung der LoD2-Fläche jedoch nur dann, wenn sich der Anteil der Inlier der LiDAR-Punkte signifikant erhöht und die neue Ebene die vordefinierten Toleranzgrenzen für die Winkelabweichung ($\Delta\theta$) und den Höhenversatz (Δz) am Mittelpunkt gegenüber dem Ursprungsmodell nicht überschreitet.

Anschließend detektiert das System Dachüberhänge, die im LoD2-Standard inhärent fehlen. Über eine KD-Baum-basierte Nachbarschaftssuche identifiziert die Pipeline LiDAR-Punkte, die außerhalb der ursprünglichen LoD2-Polygone liegen, aber geometrisch zur selben Dachebene gehören. Die Berechnung von Alpha-Shapes [61] aus den projizierten 2D-Koordinaten der zugehörigen LiDAR-Punkte bestimmt die Ausdehnung der verfeinerten Dachebene. Im Gegensatz zu einer einfachen konvexen Hülle ermöglichen Alpha-Shapes dabei die detailgetreue Erfassung komplexer, nicht konvexer Dachgeometrien (wie Einschnitte oder L-Formen). Für diese Berechnung verwendet das System den empirisch bestimmten Parameter $\alpha = 1,0$. Dieser Wert orientiert sich an der spezifizierten LiDAR-Punktdichte und stellt einen ausgewogenen Kompromiss dar, um eine unerwünschte Fragmentierung der Fläche sowie künstliche Lochbildungen im Polygon zu vermeiden. Abbildung 3.9b zeigt das auf diese Weise geometrisch verfeinerte und topologisch erweiterte Dachpolygon.

Identifikation von Störobjekten und nutzbaren Restflächen

Nach der Erweiterung der Dachflächen verbleiben in den LiDAR-Daten sogenannte Ausreißerpunkte, die die Pipeline keiner der extrahierten Dachebenen mit der geforderten Distanz-Toleranz zuordnen kann. Diese Punkte entsprechen in der Regel Dachaufbauten wie Schornsteinen, Gauben oder Lüftungsanlagen.

Zur Extraktion dieser Störobjekte wendet das System DBSCAN [59] auf die verbleibenden Ausreißerpunkte an. Hierbei wähle ich die Parameter für die maximale Suchdistanz ($\epsilon = 0.4\,\text{m}$) und die minimale Punktzahl ($MinPts = 4$) empirisch auf Basis der visuellen Analyse. Diese Werte leiten sich zudem direkt aus den technischen Spezifikationen der verwendeten LiDAR-Daten ab: Bei einer nominalen Punktdichte von mindestens 4 Punkten pro m^2 (für Befliegungen ab 2012) und einer Lagegenauigkeit von ca. 0,50 m im ebenen Gelände [53] erweist sich diese Parameterkombination als Kompromiss, um das Rauschen zu ignorieren und typische Dachaufbauten zuverlässig zu isolieren.



(a) Initiale LoD2-Dachfläche **(b)** Verfeinerte und erweiterte Dachfläche

Abbildung 3.9 – Schematische Darstellung der geometrischen Dachflächenverfeinerung. (a) zeigt die initiale LoD2-Ausgangsebene, bei der reguläre LiDAR-Dachpunkte aufgrund eines Winkel- ($\Delta\theta$) und Höhenversatzes (Δz) fälschlicherweise als Ausreißer deklariert werden. (b) zeigt die verfeinerte Ebene nach der RANSAC-Optimierung, wodurch die Punkte korrekt erfasst werden. Zusätzlich ist die topologische Erweiterung zur Erfassung des Dachüberhangs dargestellt.

Um die final nutzbare Fläche für PV-Module zu berechnen, subtrahiert die Pipeline topologisch vier Restriktionen von den verfeinerten Dachpolygonen. Um bei diesem Schnittprozess die exakte räumliche Deckungsgleichheit zwischen den zweidimensionalen Inferenzmasken der Orthophotos und den dreidimensionalen LiDAR- sowie LoD2-Daten zu gewährleisten, verarbeitet die Pipeline alle Geodaten konsistent in demselben projizierten Koordinatenreferenzsystem (EPSG:25832). Dies schließt systematische Verschiebungsfehler aus. Die vier abzuziehenden Restriktionen setzen sich wie folgt zusammen:

1. **Randabstände**: Ein Puffer entlang der Dachkanten zur Gewährleistung von Wartungswegen und statischer Sicherheit.

2. **Geometrische Störobjekte**: Die durch DBSCAN identifizierten LiDAR-Ausreißer.

3. **Semantische Störobjekte**: Die vom Segmentierungsmodell erkannten Flächen der Klassen *Technical Roof Structures*, *Other Roof Structures* und *Photovoltaic Systems*.

4. **Dachnähte**: Ein berechneter Pufferbereich an den Schnittkanten (z. B. First) benachbarter, nicht koplanarer Dachflächen.

Topografische Schattenanalyse mittels dynamischen Raytracings

Da lokale Verschattungen (z. B. durch benachbarte Gebäude oder Vegetation) die verfügbare Solarenergie signifikant beeinflussen, berechnet der Algorithmus eine hochaufgelöste Horizontwinkelkarte [62] für jedes Rasterelement des DOM.

Hierbei wendet das System einen iterativen Raytracing-Ansatz an, der die Umgebung in $M = 80$ diskreten Azimutrichtungen abtastet. Um die Rechenzeit zu optimieren, ohne an Genauigkeit im kritischen Nahbereich zu verlieren, passt der Algorithmus die Schrittweite des Raytracings dynamisch an: Im Nahbereich (bis 10,0 m) entspricht die Abtastrate der Bodenauflösung des Modells (z. B. 0,2 m), während das System sie im Fernbereich (bis zu einer maximalen Suchdistanz von 500 m) auf 2,0 m erhöht. Für jeden Abtastpunkt setzt der Algorithmus den Höhenunterschied Δh zur Distanz d ins Verhältnis, um den maximalen Elevationswinkel γ entlang der jeweiligen Azimutrichtung zu bestimmen:

$$\gamma = \arctan\left(\frac{\Delta h}{d}\right) \tag{3.6}$$

Einstrahlungsmodellierung und Schattensimulation

Das entwickelte Verfahren simuliert den Solarertrag stundengenau über ein vollständiges meteorologisches Referenzjahr (8 760 Stunden). Dazu berechnet das System im ersten Schritt den genauen stündlichen Sonnenstand, definiert durch Azimut und scheinbare Sonnenhöhe, bezogen auf die geographischen Koordinaten des Grundstücks unter Verwendung der Softwarebibliothek pvlib [96]. Die klimatologischen Randbedingungen basieren auf einem typischen meteorologischen Jahr aus der PVGIS-Datenbank der Europäischen Kommission, welches stündliche Werte für die Global-, Direkt- und Diffusstrahlung liefert [110].

Um die Einstrahlung auf die geneigte Dachfläche zu quantifizieren, zieht die Pipeline das etablierte Perez-Modell [63] heran. Der Algorithmus berechnet in jedem Zeitschritt, ob ein Pixel verschattet ist, indem er die aktuelle Sonnen-Elevation mit dem vorher berechneten richtungsspezifischen Horizontwinkel der Horizontwinkelkarte vergleicht. Tritt eine Verschattung ein, blockiert das Modell die direkte Einstrahlungskomponente. Die nutzbare Einstrahlung setzt sich in diesem Fall nur noch aus der um den Sky-View-Faktor reduzierten diffusen Himmelsstrahlung und der Bodenreflexion zusammen. Das Resultat dieser Pipeline ist eine detaillierte, pixelbasierte Karte der jährlichen kumulierten Einstrahlung (in $\mathrm{W\,h/m^2}$), die als Grundlage für die finale Potenzialberechnung der identifizierten Dachflächen dient.

Algorithmische Platzierung von Solarmodulen

Der finale Schritt der Methodik umfasst die geometrische Anordnung und Ertragsoptimierung von PV-Standardmodulen auf den zuvor bereinigten, nutzbaren Dachflächen. Um die zur Verfügung stehende Fläche energetisch zu maximieren, verwendet die Methodik einen rasterbasierten Platzierungsalgorithmus, der zwischen Flach- und Schrägdächern differenziert. Ein definierter Schwellenwert der Dachneigung steuert dabei die Klassifizierung in Flach- und Schrägdächer.

Die Platzierung arbeitet nicht rein geometrisch, sondern wird direkt durch die energetische Effizienz gesteuert. Für jede iterativ getestete Modulposition (repräsentiert durch einen 2D-Begrenzungsrahmen) tastet der Algorithmus die zugrundeliegende Einstrahlungskarte (engl. *Annual Energy Map*) ab. Aus der mittleren Einstrahlung, der Nennleistung des Standardmoduls und einem angenommenen Systemnutzungsgrad (engl. *Performance Ratio*) berechnet das System den zu erwartenden Jahresertrag. Dabei greift unmittelbar ein wirtschaftliches Ausschlusskriterium: Der Algorithmus verwirft sofort Module, die aufgrund lokaler Restverschattung oder ungünstiger Randlage einen definierten Mindestertrag unterschreiten. Diese fließen nicht in die Ertragsbewertung einer Rasterkonfiguration ein, wodurch die Pipeline sicherstellt, dass die Optimierung nur wirtschaftlich sinnvolle Belegungen untersucht.

Schrägdächer Bei geneigten Dachflächen übernehmen die Module die mittlere Neigung und Ausrichtung der zugrundeliegenden Dachpolygon-Ebene. Da die Pipeline eine parallele Dachmontage simuliert, platziert sie die Module in einem starren orthogonalen Raster. Um die optimale Positionierung dieses Rasters zu ermitteln, iteriert der Algorithmus mit einem hierarchischen Ansatz über verschiedene zweidimensionale Verschiebungen des Rasters. Für jede Verschiebung wird iterativ geprüft, wie viele Module vollständig innerhalb des nutzbaren Dachpolygons platziert werden können. Zudem testet der Algorithmus sowohl eine vertikale als auch eine horizontale Modulausrichtung. Das System wählt die finale Konfiguration, welche den höchsten kumulierten Gesamtertrag für die jeweilige Dachfläche generiert.

Flachdächer Für Flachdächer simuliert die Pipeline eine Aufständerung der Module. Hierbei fixiert der Algorithmus die Neigung der Module unabhängig vom Dach auf einen festen Wert und richtet sie ideal nach Süden aus. Da aufgeständerte Module einen Schattenwurf erzeugen, berechnet der Algorithmus dynamisch einen notwendigen Reihenabstand. Dieser basiert auf dem tiefsten Sonnenstand (Wintersonnenwende) für die jeweilige geographische Breite, um eine gegenseitige Verschattung der Modulreihen zu verhindern. Im Gegensatz zum starren Raster der Schrägdächer wendet der Algorithmus hier

eine flexiblere Platzierungsstrategie an: Um asymmetrische Dachgeometrien optimal auszunutzen, iteriert er bei Flachdächern über die Längsverschiebung des Gesamtrasters, optimiert jedoch den Versatz entlang der Reihen für jede Modulreihe unabhängig voneinander. Auch hier entscheidet letztlich der maximale kumulierte Ertrag der validen Module über das finale Layout.

Normbasierte Ertragsabschätzung als Referenzwert

Neben der hochaufgelösten, pixelbasierten Einstrahlungssimulation berechnet der Algorithmus für jedes platzierte Modul zusätzlich einen normierten Jahresertrag. Dieser normative Ansatz orientiert sich an der in Abschnitt 2.5.5 vorgestellten DIN V 18599 [64]. Die parallele Erhebung dieses Wertes ist essenziell, um die simulierten Erträge mit normativen Referenzwerten vergleichen zu können.

Da die Einteilung der 15 klimatischen Referenzregionen im Originaldokument der Norm jedoch lediglich in Form einer niedrigaufgelösten Rastergrafik abgebildet ist [65], erforderte dies für die automatisierte Pipeline im Vorfeld eine räumliche Aufbereitung. Hierzu wurde die Grafik in einem manuellen Vorverarbeitungsschritt in *QGIS* importiert, über eindeutig identifizierbare Passpunkte georeferenziert und anschließend vektorisiert (siehe Abbildung C.1).

Für jedes untersuchte Grundstück bestimmt der Algorithmus anschließend über eine räumliche Abfrage die zugehörige Klimaregion. Da die geometrischen Neigungs- und Azimutwinkel der simulierten Module kontinuierliche Werte annehmen, die DIN-Norm jedoch nur diskrete tabellarische Stützstellen vorgibt, ordnet der Algorithmus diese den jeweils nächstliegenden Tabellenwerten zu. Die Pipeline weist jedem Modul dabei automatisiert den Einstrahlungswert der geometrisch passendsten Spalte zu. Basierend auf dieser regionalen und geometrischen Zuordnung berechnet das System den normativen Referenzertrag gemäß Gleichung (2.6). Hierbei wendet die Pipeline die normative altersbedingte Leistungsdegradation (Faktor 0,9) automatisiert auf die Nennleistung der simulierten Module an.

Definition der Potenzial-Indikatoren und Umgang mit Bestandsanlagen

Um das Solarpotenzial am Ende nicht nur grafisch, sondern auch quantitativ zu erfassen, extrahiert die Pipeline zwei finale Indikatoren. Auch hierbei greift das System direkt auf die Vorhersagen des Segmentierungsmodells zurück, um den energetischen Ist-Zustand des Untersuchungsgebiets abzubilden. Für Pixel, die das Modell als bereits bestehende PV-Anlage (*Photovoltaic System*) klassifiziert, berechnet der Algorithmus die einfallende simulierte Sonnenenergie. Da bei fremden Bestandsanlagen externe Leistungsparameter (wie der Systemnutzungsgrad oder die Nennleistung) unbekannt sind, ermittelt die Pipeline hierfür ausschließlich die reine, auf die Modulfläche eingestrahlte Energie über das Referenzjahr. Der Algorithmus beachtet dabei

exakt die berechnete Neigung und Ausrichtung der zugrundeliegenden Dachfläche. Als finales Resultat der Analyse berechnet die Pipeline folgende zwei Indikatoren für jedes Untersuchungsgebiet:

$P_{\text{potential}}$ Dieser Indikator quantifiziert den simulierten, potenziell erzeugbaren Strom über ein Jahr. Er summiert den elektrischen Ertrag, der entstünde, wenn Betreiber alle vom Platzierungsalgorithmus vorgeschlagenen, wirtschaftlich rentablen PV-Module auf den nutzbaren Dachflächen tatsächlich installieren.

P_{existent} Dieser Indikator beschreibt das bereits erschlossene Potenzial. Er beziffert die reine solare Einstrahlungsenergie, die im Verlauf eines Jahres auf alle durch das Modell erkannten, bestehenden PV-Anlagen fällt.

3.5.3 Evaluierung der abgeleiteten Indikatoren

Um die Zuverlässigkeit der aus den Inferenzmasken abgeleiteten Flächenindikatoren und der Solarpotenzialanalyse objektiv zu bewerten, führt das System einen systematischen Vergleich zwischen den Modellvorhersagen und den manuell annotierten Referenzdaten durch. Anstatt die Indikatoren lediglich aggregiert über die gesamte Fläche einer $512 \times 512\,\text{m}$ großen Untersuchungskachel zu vergleichen, wähle ich einen räumlich feingranularen Ansatz.

Hierzu unterteile ich jede Untersuchungskachel in ein regelmäßiges Raster aus 10×10 Zellen. Daraus ergeben sich pro Kachel 100 gleich große Referenzzellen mit einer Kantenlänge von jeweils $51,2 \times 51,2\,\text{m}$.

An dieser Stelle ist eine terminologische und methodische Abgrenzung des Begriffs „Parzellenebene“ erforderlich, der das konzeptionelle Ziel dieser Arbeit definiert. In der strikten geoinformatischen Definition entspricht eine Parzelle einem amtlichen Flurstück des Liegenschaftskatasters. Da diese amtlichen Vektordaten für den Freistaat Bayern nicht als offene Geodaten frei verfügbar sind, ist eine flächendeckende Auswertung auf Basis realer Grundstücksgrenzen im Rahmen dieser Arbeit nicht realisierbar. Um diese feingranulare, quartiersbezogene Analyse dennoch umzusetzen, nähert diese Arbeit die Parzellenebene durch das definierte $51,2 \times 51,2\,\text{m}$-Raster an. Dieses Raster fungiert im weiteren Verlauf als räumliche Aggregationsebene und wird terminologisch synonym zur stadtplanerischen Parzellenebene verwendet, da es die typische Größenordnung urbaner Bebauungsstrukturen adäquat abbildet und eine lückenlose, blockweise Analyse der Fehlerverteilung ermöglicht.

Die Evaluierungs-Pipeline, deren schematischer Ablauf in Abbildung 3.10 dargestellt ist, wendet auf Basis der in Abschnitt 3.4.7 beschriebenen Kreuzvalidierung für jede der 700 resultierenden Rasterzellen (7 Kacheln $\times$ 100 Zellen) folgende Schritte an:

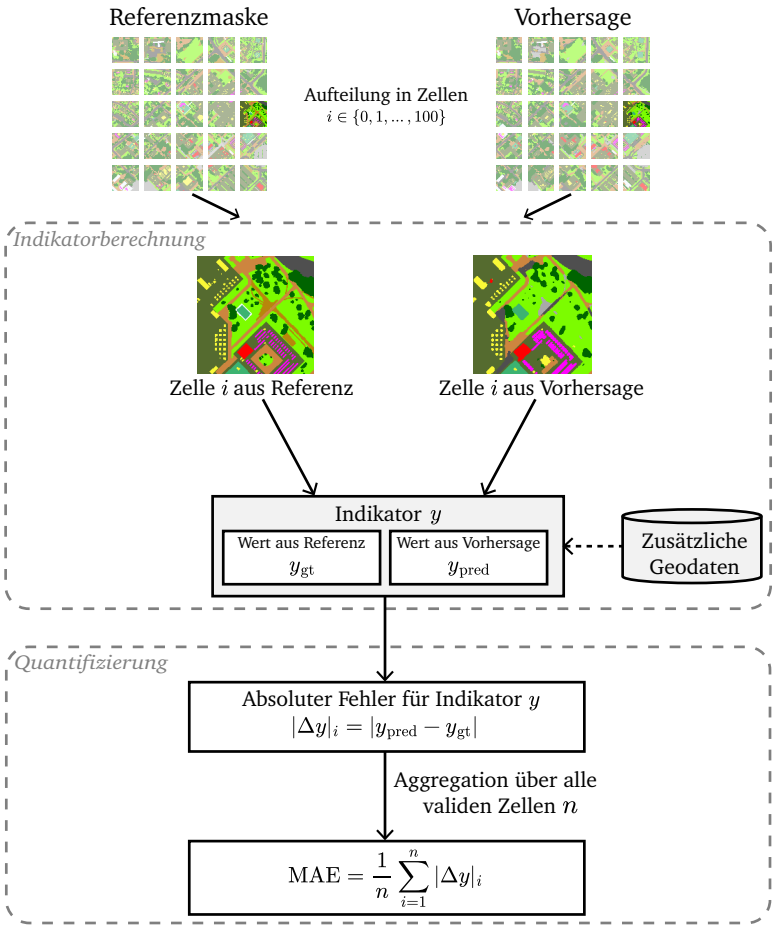


Abbildung 3.10 – Schematische Darstellung der rasterbasierten Evaluierungs-Pipeline zur Validierung der Indikatoren für ein beispielhaftes Untersuchungsgebiet. Der Prozess veranschaulicht die Extraktion einer Einzelzelle i, die separate Berechnung des absoluten Fehlers für jeden Indikator y und die anschließende Aggregation zum mittleren absoluten Fehler (MAE). Der gestrichelte Pfeil verdeutlicht hierbei, dass zusätzliche 3D-Geodaten indikatorspezifisch ausschließlich in die Berechnung des Solarpotenzials einfließen.

1. Isolieren des jeweiligen Bildausschnitts aus der vorhergesagten Inferenzmaske (welche strikt auf den ungesehenen *Out-of-Fold*-Vorhersagen der Kreuzvalidierung basiert) sowie aus der Referenzmaske.
2. Voneinander unabhängiges Berechnen der Indikatoren (Abflussbeiwert, Anteil nachhaltig genutzter Dachflächen, PV-Potenzial) für beide Masken.
3. Quantifizieren der absoluten Abweichung (engl. *Absolute Error*) zwischen Vorhersage und Referenz pro Zelle.

Als primäre Metrik zur Bewertung der Gesamtleistung definiere ich den mittleren absoluten Fehler über alle validen Rasterzellen:

$$\mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_{pred,i} - y_{gt,i}| \tag{3.7}$$

wobei n die Anzahl der auswertbaren Rasterzellen, $y_{pred,i}$ den berechneten Indikatorwert auf Basis der Segmentierungsvorhersage und $y_{gt,i}$ den berechneten Indikatorwert auf Basis der Referenzdaten für die Zelle i beschreibt.

Um statistische Verzerrungen bei der Fehlerkalkulation zu vermeiden, schließt die Evaluierungs-Pipeline mathematisch undefinierte Randfälle explizit aus. Fehlt in einer Rasterzelle beispielsweise die bauliche Grundlage für einen spezifischen Indikator (wie Gebäude für den Anteil nachhaltig genutzter Dachflächen), weist der Algorithmus der Zelle einen `NaN`-Wert zu. Bei der finalen Berechnung des MAE ignoriert das System diese Zellen dynamisch. Dies garantiert, dass künstliche Nullfehler-Werte aus irrelevanten Gebieten den Fehlerdurchschnitt nicht verfälschen.

Die hier definierten, präzisen Evaluierungsmetriken garantieren die wissenschaftliche Belastbarkeit der generierten Indikatoren. Um den praktischen Nutzen dieser automatisierten Analysepipeline jedoch auch über den akademischen Kontext hinaus greifbar zu machen, überführt der nachfolgende Abschnitt die Methodik abschließend in einen anwendungsorientierten Prototypen.

3.5.4 Interaktiver Demonstrator

Zur Demonstration der entwickelten Pipeline und für den Transfer der Methodik in ein praktisches Anwendungsszenario implementiere ich eine prototypische Web-Anwendung. Diese ermöglicht es Nutzerinnen und Nutzern, ein beliebiges Untersuchungsgebiet auszuwählen und die gesamte Verarbeitungs-Pipeline über eine API automatisiert auszuführen.

Für den produktiven Einsatz innerhalb dieses Demonstrators kommt ein im Vorfeld separat trainiertes, finales Segmentierungsmodell zum Einsatz. Im Gegensatz zur methodischen Evaluierung, die strikt auf den *Out-of-Fold*-Vorhersagen der Kreuzvalidierung basiert, nutzt dieser finale Trainingslauf den vollständigen Datensatz.

Durch die Einbeziehung aller sieben Kacheln ohne Zurückhaltung eines Validierungsdatensatzes wird sichergestellt, dass das Modell für die praktische Anwendung auf ungesehene DOP20-Aufnahmen die höchstmögliche semantische Varianz urbaner Strukturen gelernt hat. Obwohl für dieses finale Modell naturgemäß keine unabhängigen Validierungsmetriken mehr berechnet werden können, entspricht dieses Vorgehen der etablierten Praxis im maschinellen Lernen. Die in der Kreuzvalidierung ermittelten Metriken (siehe Abschnitt 4.3.4) dienen als verlässlicher Schätzwert für die Generalisierungsfähigkeit der gewählten Architektur.

Während der Berechnung in der Weboberfläche zeigt die Anwendung Statusmeldungen an und visualisiert die Ergebnisse schrittweise. Hierzu gehören die semantische Segmentierung der Luftbilder, die Darstellung der solaren Einstrahlungskarte sowie die algorithmische Platzierung der PV-Module. Die Benutzeroberfläche dient somit nicht der wissenschaftlichen Fehlerkalkulation, sondern der anschaulichen Überprüfung der Berechnungsergebnisse sowie der visuellen Validierung der Methodik auf neuen Flächen. Der Anhang liefert eine detaillierte Übersicht der Nutzeroberfläche (Abbildungen D.1 bis D.4).

Kapitel 4

Ergebnisse und Diskussion

Dieses Kapitel behandelt die systematische Auswertung und Interpretation der erzielten Ergebnisse. Um die Leistungsfähigkeit der entwickelten Segmentierungs-Pipeline umfassend und methodisch fundiert zu bewerten, folgt die Analyse einer strengen, mehrstufigen Evaluierungslogik, von der isolierten Untersuchung grundlegender Daten- und Architekturparameter bis hin zur operationalen Anwendung in der urbanen Planungspraxis.

Zunächst präsentiert Abschnitt 4.1 die Ergebnisse einer Voruntersuchung, welche die wesentlichen Erfolgsfaktoren für die Generalisierungsfähigkeit auf hochauflösenden 20 cm-Luftbildern ermittelt. Auf diesen Erkenntnissen aufbauend dokumentiert der darauffolgende Abschnitt die systematische Architektursuche. Unter der Prämisse streng limitierter Annotationsressourcen wird hierbei die parametereffizienteste und leistungsstabilste Kombination aus Basismodell und Klassifikationskopf identifiziert sowie hinsichtlich ihrer Dateneffizienz validiert.

Nach der Fixierung der Methodik erfolgt der Transfer auf die finale Zieldomäne. In diesem Schritt wird zunächst die Qualität und Variabilität der manuellen Referenzdaten untersucht, bevor die reine Segmentierungsgüte sowie die räumliche Generalisierungsfähigkeit des Modells mittels Kreuzvalidierung detailliert quantifiziert werden.

Anschließend überführt Abschnitt 4.4 die pixelbasierten Vorhersagen in die Anwendungsebene. Hierbei wird evaluiert, wie zuverlässig sich stadtökologische Flächenindikatoren und solare Potenziale aus den automatisierten Segmentierungsmasken ableiten lassen. Den Abschluss des Kapitels bildet eine kritische Diskussion, welche die erzielten Resultate in den Kontext methodischer Kompromisse und der physikalischen Limitationen optischer Luftbilder einordnet.

4.1 Ergebnisse der Voruntersuchung

Ziel der Voruntersuchung ist es zu evaluieren, welche Eigenschaften der Trainingsdaten die Generalisierungsfähigkeit auf hochauflösende 20 cm-Luftbilder am stärksten begünstigen. Hierbei standen sich Modelle gegenüber, die entweder auf extrem hoher räumlicher Auflösung, aber geringer geographischer Varianz (ISPRS Potsdam und Vaihingen) oder auf moderater Auflösung bei gleichzeitig sehr hoher Szenenvarianz (OEM) trainiert wurden.

Die quantitative Auswertung auf dem unabhängigen bayerischen Validierungsdatensatz (DOP20) zeigt ein eindeutiges Leistungsgefälle zugunsten der Domänenvarianz. Wie in Tabelle 4.1 dargestellt, erzielt das auf OEM trainierte Modell mit einer mIoU von 74,74 % und einer PA von 86,19 % die mit Abstand besten Ergebnisse.

Tabelle 4.1 – Quantitative Evaluierung der Voruntersuchung auf dem unabhängigen DOP20-Testgebiet (Werte in %). Die *Clutter*-Klasse sowie die *Car*-Klasse wurden gemäß der in der Methodik definierten Evaluierungsstrategie von der Berechnung der Metriken ausgeschlossen, da sie im Zielgebiet in dieser Form nicht existieren oder re-klassifiziert wurden.

			Klassenspezifische IoU			
Datensatz	**GSD**	**mIoU**	**ImpSurf**	**Building**	**LowVeg**	**Tree**
ISPRS Potsdam	5 cm	42,55	38,44	51,06	10,31	70,38
ISPRS Vaihingen	9 cm	65,17	61,08	70,63	48,87	80,10
OpenEarthMap	25–50 cm	**74,74**	**74,51**	**82,50**	**59,05**	**82,91**

Die teils deutliche Leistungsdiskrepanz, insbesondere der Einbruch des Potsdam-Modells (mIoU: 42,55 %), lässt sich qualitativ auf einen starken Maßstabsunterschied zwischen Trainings- und Zieldomäne zurückführen. Obwohl das Potsdam-Modell in seiner nativen Domäne mit 5 cm Bodenauflösung hochpräzise segmentiert, scheitert es beim Transfer auf die 20 cm-Zielauflösung.

Die visuelle Analyse der Inferenzmasken (siehe Abbildung 4.1) zeigt, dass das Potsdam-Modell weite Teile des Bildes fälschlicherweise als *Clutter* (Hintergrund) klassifiziert. Der Grund hierfür liegt in den fehlenden Frequenzinformationen: Das auf 5 cm trainierte Netz hat gelernt, sich auf extrem feine Texturen, wie einzelne Dachziegel, scharfe Bodenstrukturen oder Fahrbahnmarkierungen, zu verlassen. Da diese Hochfrequenzdetails in den glatteren 20 cm-DOP-Aufnahmen verschwimmen oder gänzlich fehlen, verliert das Modell sein Unterscheidungsvermögen. Es stuft die ihm unbekannten, texturärmeren Flächen als fremd ein und weicht auf die Hintergrundklasse aus, was folglich die Sensitivität der regulären urbanen Klassen stark senkt.

Das Vaihingen-Modell (9 cm) zeigt dieses Verhalten in abgemilderter Form und positioniert sich im Mittelfeld. Das OEM-Modell hingegen profitiert von zwei sich

ergänzenden Effekten: Erstens liegt seine native Trainingsauflösung (25–50 cm) deutlich näher an der Zieldomäne (20 cm), wodurch der Maßstabsunterschied minimiert wird. Zweitens zwingt die geographische Variabilität der OEM-Daten das Netzwerk dazu, übertragbarere, globalere Objektmerkmale, wie Geometrie und räumlichen Kontext, zu erlernen, anstatt eine Überanpassung an auflösungsspezifische Mikrotexturen vorzunehmen.

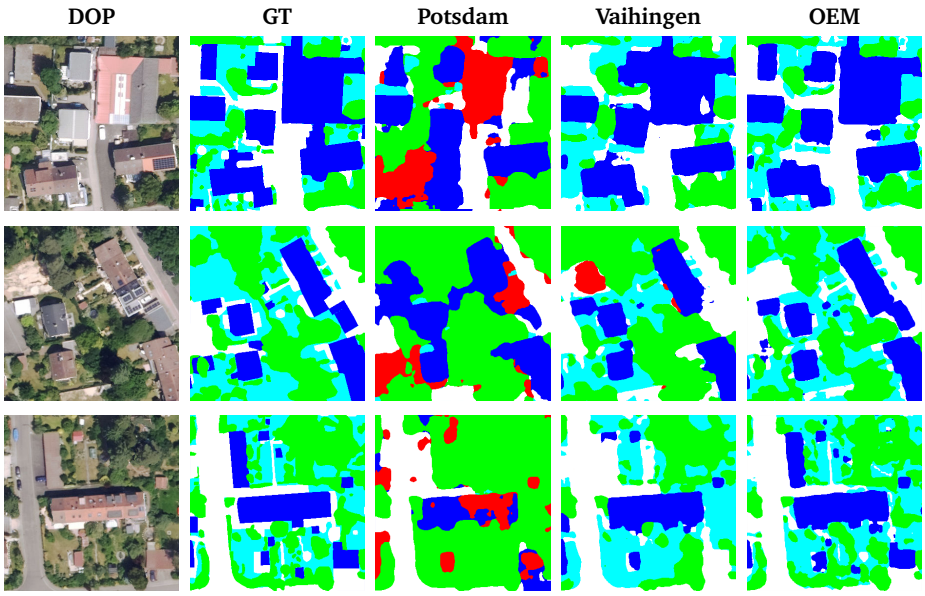


Abbildung 4.1 – Qualitativer Vergleich der Segmentierungsgüte anhand verschiedener Detailausschnitte. Die Spalten zeigen jeweils von links nach rechts: Luftbild (DOP), Referenzmaske (GT) sowie die Vorhersagen der Modelle, die auf ISPRS Potsdam (5 cm), ISPRS Vaihingen (9 cm) und OpenEarthMap (25–50 cm) trainiert wurden. Über die verschiedenen Beispiele hinweg wird deutlich sichtbar, dass das Potsdam-Modell an den texturärmeren Zieldaten scheitert und weite Teile fälschlicherweise der Hintergrundklasse (■) zuordnet. Das Vaihingen-Modell weist diesen Fehler zwar seltener auf, erfasst jedoch die Gebäudegeometrien (■) im Vergleich zum OEM-Modell sichtbar unpräziser. Letzteres segmentiert die urbanen Strukturen durchgehend am genauesten.

Dieses Vorexperiment belegt empirisch, dass für die Segmentierung hochauflösender, aber vergleichsweise texturarmer Orthophotos eine hohe räumliche und bauliche Varianz im Trainingsdatensatz wesentlich kritischer für die Generalisierungsleistung ist als die reine Maximierung der Bildauflösung. Diese Erkenntnis legitimiert den methodischen Schritt, für die finale Architektursuche den FLAIR-Datensatz heranzuziehen. FLAIR vereint die hier identifizierten Erfolgsfaktoren: Er eliminiert den Maßstabsfehler durch eine exakt übereinstimmende 20 cm-Auflösung und maximiert gleichzeitig die Domänenvarianz durch Daten aus 50 verschiedenen Regionen.

4.2 Ergebnisse der Architekturoptimierung

Nachdem die Voruntersuchung die Bedeutung einer hohen Domänenvarianz und der Bodenauflösung für die Generalisierungsfähigkeit auf Luftbilddaten belegt hat, behandelt dieser Abschnitt die systematische Auswahl der finalen Modellarchitektur. Wie in der Methodik dargelegt, dient hierfür eine repräsentative Teilmenge des FLAIR-Datensatzes mit der Bodenauflösung von 20 cm als Evaluierungsumgebung.

Um ein realistisches Kaltstart-Szenario mit streng limitierten Annotationsressourcen zu simulieren, operieren alle in diesem Abschnitt durchgeführten Experimente in einem strikten Low-Data-Regime. Dieses beschränkt die verfügbare Anzahl der annotierten Trainingsdaten auf lediglich 75 Bildausschnitte. Zur Gewährleistung einer hohen statistischen Belastbarkeit und zur Minimierung zufälliger Ausreißer basiert die Evaluierung jedes Modells zudem auf fünf unabhängigen Trainingsläufen, die jeweils auf einer eigenständig gezogenen Trainingsteilmenge ($N = 75$) bei gemeinsamem Validierungsset trainiert werden. Alle nachfolgend berichteten Mittelwerte und Standardabweichungen beziehen sich auf diese fünf Läufe.

Da der FLAIR-Datensatz in dieser Phase der Arbeit ausschließlich als methodischer Stellvertreter zur Ermittlung der leistungsstärksten Architektur dient, verzichte ich an dieser Stelle auf eine detaillierte qualitative visuelle Auswertung. Der Fokus liegt stattdessen auf dem rein quantitativen Vergleich der Generalisierungsfähigkeit unter starker Datenknappheit. Ziel ist es, objektiv diejenige Architektur zu identifizieren, die aus einem Minimum an Daten die aussagekräftigsten Merkmale extrahiert und folglich das höchste Potenzial für den späteren Transfer auf die eigentliche, bayerische Zieldomäne aufweist.

Um die Einflüsse einzelner Netzwerkkomponenten isoliert bewerten zu können, erfolgt die Auswertung nach einer konsequenten Bottom-up-Strategie: Der erste Schritt ermittelt die stärkste Basisrepräsentation durch den Vergleich verschiedener Encoder-Paradigmen. Darauf aufbauend evaluiert der zweite Schritt die parametrische Effizienz unterschiedlicher Decoder-Architekturen auf diesen Merkmalen. Den Abschluss bildet die zielgerichtete Hyperparameter-Optimierung der leistungsstärksten Gesamtkombination.

4.2.1 Encoder-Auswahl

Die quantitativen Ergebnisse der Encoder-Evaluierung (siehe Tabelle 4.2) zeigen im definierten Low-Data-Regime ein signifikantes Leistungsgefälle zwischen den untersuchten Architekturparadigmen. Die klassische, rein überwacht trainierte Basislinie in Form von ResNet-101 erzielt mit einer mittleren mIoU von 34,40 % den niedrigsten Wert unter den auf externen Daten vortrainierten Standard-Backbones. Die durch das überwachte ImageNet-Vortraining gelernten, stark texturbezogenen Merk-

male erweisen sich als unzureichend, um bei Datenknappheit aussagekräftige und separierbare Repräsentationen für hochauflösende Luftbilder zu abstrahieren. Einen ersten Leistungssprung verzeichnen moderne Architekturen: Sowohl der überwacht trainierte hierarchische Swin-Transformer (41,68 %) als auch das auf deformierbaren Faltungen basierende InternImage-Modell (42,80 %) aggregieren strukturelle Bildinformationen durch ihren architektonischen induktiven Bias weitaus effizienter.

Tabelle 4.2 – Ergebnisse der Backbone-Evaluierung im Low-Data-Regime ($N = 75$). Zur Wahrung der Vergleichbarkeit wurden alle Architekturen an einen einheitlichen UperNet-Decoder gekoppelt. Die angegebene Parameteranzahl (Param.) bezieht sich isoliert auf den jeweiligen Backbone (exklusive Decoder). Angegeben ist zudem die mittlere mIoU sowie die Standardabweichung über die fünf unabhängigen Trainingsläufe. Die leistungsstärkste Architektur ist fett hervorgehoben.

Backbone	Vortraining	Datenbasis	Param. (M)	mIoU (%)
CNN Basislinien				
ResNet-101	Überwacht	ImageNet-1K	42,52	34,40 ± 1,04
InternImage	Überwacht	ImageNet-22K	218,95	42,80 ± 0,64
Hierarchischer ViT				
Swin-Base	Überwacht	ImageNet-22K	86,75	41,68 ± 1,32
Basismodelle				
DINOv3-Large	SSL (DINOv3)	LVD-1689M	326,39	**43,70 ± 0,78**
DINOv3-Large	SSL (DINOv3)	SAT-493M	326,39	42,96 ± 1,25
Domänenspezifisch				
Eigener ViT-Base	SSL (Masked AE)	DOP20 (Bayern)	100,80	32,48 ± 1,06

Eine wesentliche methodische Weiterentwicklung zeigen jedoch die selbstüberwacht trainierten Basismodelle. Das allgemeine DINOv3-Modell (LVD) erzielt im Vergleich mit einer mittleren mIoU von 43,70 % die besten Ergebnisse. Bemerkenswert ist hierbei die Kombination aus dieser hohen Segmentierungsgüte und einer gleichzeitig starken Konvergenz, die sich in einer geringen Standardabweichung von lediglich ± 0,78 % äußert. Dieser Wert belegt empirisch die Dateneffizienz dieses stark skalierten Basismodells: Obwohl das überwacht trainierte InternImage-Modell eine leicht geringere Varianz (± 0,64 %) aufweist, liefert DINOv3-LVD ein höheres Leistungsniveau. Die auf 1,69 Milliarden Bildern gelernten Merkmale sind derart ausdrucksstark, dass das Netzwerk nahezu unabhängig von der zufälligen Zusammensetzung der wenigen Trainingsbilder stets stabil konvergiert.

Die weitere Auswertung zeigt, dass diese Stabilität nicht pauschal aus dem unüberwachten Lernparadigma oder der Transformer-Architektur resultiert. So erreicht zwar auch das überwacht trainierte InternImage-Modell durch seine deformierbaren Faltungen eine ähnlich hohe Stabilität (±0,64 %), bleibt in der absoluten Segmentierungsgüte jedoch hinter dem Basismodell zurück. Gleichzeitig belegt die höhere Varianz des DINOv3-SAT-Modells (±1,25 %), dass SSL allein keine Stabilitätsgarantie

bietet. Vielmehr ist es erst die Kombination aus dem selbstüberwachten Lernziel und der großen visuellen Diversität des LVD-Datensatzes, die den entscheidenden Vorteil im Low-Data-Regime liefert.

Das wichtigste Resultat dieser Evaluierungsphase zeigt sich jedoch im direkten Vergleich der Vortrainingsdomänen. Unter der klassischen Prämisse, dass domänenspezifische Daten den Modelltransfer erleichtern, wäre zu erwarten, dass das auf Satellitendaten trainierte DINOv3-SAT-Modell oder der eigens auf bayerischen Daten trainierte Masked-AE-Backbone das allgemeine DINOv3-LVD-Modell übertreffen. Die Ergebnisse widerlegen diese Intuition deutlich: Das LVD-Modell schlägt sowohl das SAT-Modell (42,96 %) als auch den eigenen Masked-AE-Ansatz (32,48 %).

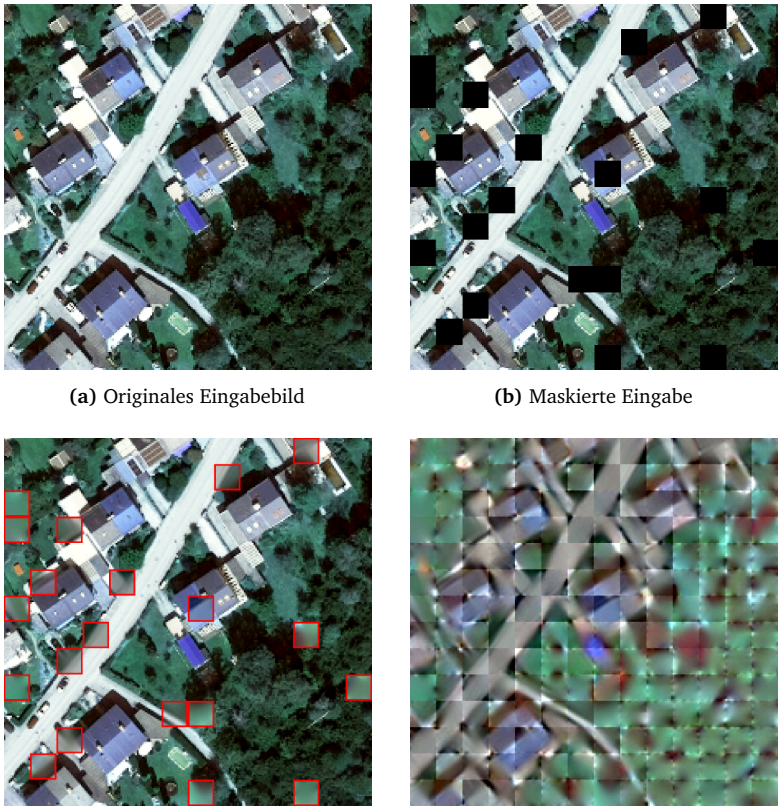
Dieser Befund lässt sich auf das komplexe Zusammenwirken von Datenmasse und semantischer Struktur zurückführen. Obwohl Satellitendaten (wie SAT-493M) die Nadirperspektive der Zieldomäne exakt abbilden, weisen sie auf Objektebene eine stark reduzierte visuelle Varianz auf; urbane Strukturen erscheinen primär als flache Texturen. Der LVD-Datensatz enthält hingegen 1,69 Milliarden Fotografien mit großer 3D-Varianz, tiefen Schattenwürfen und komplexen Verdeckungen. Die Ergebnisse belegen, dass das LVD-Modell durch diese starke Skalierung universellere und übertragbarere Kanten- sowie Formdetektoren erlernt hat. Dieser gewonnene Abstraktionsgrad überkompensiert den theoretischen Nachteil der Domänenlücke zur Vogelperspektive vollständig. Folglich wähle ich die Kombination aus DINOv3-LVD und dem modifizierten ViT-Adapter als primären und leistungsstärksten Merkmalsextraktor für die nachfolgende Evaluierung der Decoder aus. Um darüber hinaus das Zusammenspiel zwischen Decoder-Kapazität und Encoder-Größe zu analysieren, ergänze ich die nachfolgende Decoder-Evaluierung um den Swin-Transformer als leichtgewichtige Kontrollgruppe.

Evaluierung des Masked-AE-Vortrainings

Bevor die Pipeline im nächsten Schritt um den Decoder erweitert wird, wird im Folgenden der auffällig große Leistungsabstand des eigens vortrainierten Masked-AE-Modells methodisch diskutiert. Die verhältnismäßig schwache Leistung (32,48 %) erfordert eine differenzierte Betrachtung der zugrundeliegenden Trainingsbedingungen. Während Basismodelle wie DINOv3 auf Industrie-Rechenclustern über Wochen hinweg trainiert werden, unterlag mein Masked-AE-Vortraining den strengen Hardware-Restriktionen dieser Arbeit sowie einem eng begrenzten zeitlichen Budget. Unter diesen Bedingungen ist eine Skalierung der Epochenzahlen und effektiven Batch-Größen, die für das Erlernen feingranularer, hochkomplexer Repräsentationen zwingend erforderlich ist, praktisch ausgeschlossen.

Dass das Modell trotz dieser Limitierungen fundamentale visuelle Konzepte der Zieldomäne erlernt hat, belegt eine qualitative Analyse der Rekonstruktionsfähigkeit

(siehe Abbildung 4.2). Die Visualisierung demonstriert das Verhalten des Modells bei der Wiederherstellung maskierter Bildbereiche. Obwohl die interpolierten Areale sowie die vollständige Bildrekonstruktion erwartungsgemäß verwaschen wirken und feine Hochfrequenzdetails (wie scharfe Dachkanten) fehlen, stellt das Netzwerk die makroskopische Geometrie, wie Gebäudegrundrisse und Vegetationsgrenzen, farblich und strukturell logisch wieder her.



(a) Originales Eingabebild

(b) Maskierte Eingabe

(c) Teilrekonstruktion der Areale

(d) Vollständige Bildrekonstruktion

Abbildung 4.2 – Qualitative Evaluierung der Rekonstruktionsfähigkeit des Masked-AE-Modells anhand von DOP20-Aufnahmen. Während hochfrequente Details stark verwaschen erscheinen, gelingt dem Modell eine farblich und strukturell logische Wiederherstellung der makroskopischen Geometrie (z. B. Vegetationsgrenzen und Gebäudegrundrisse).

Diese qualitative Betrachtung liefert einen entscheidenden methodischen Beweis: Das Masked-AE-Modell hat die semantischen Grundstrukturen der bayerischen DOP20-Aufnahmen erfolgreich verinnerlicht und ist nicht an der Algorithmus-Logik gescheitert. Gleichzeitig untermauert dieses Ergebnis jedoch eine zentrale Erkenntnis für die angewandte Forschungspraxis: Selbst bei Vorliegen idealer, domänenspezifischer unannotierter Daten ist das Training wettbewerbsfähiger Basismodelle auf Systemen mit nur einem Grafikprozessor ressourcentechnisch nicht realisierbar.

Der Rückgriff auf universelle, stark skalierte Modelle (wie DINOv3-LVD) stellt unter diesen Rahmenbedingungen somit den einzig praktikablen Lösungsweg dar.

4.2.2 Decoder-Auswahl

Nachdem ich in der ersten Evaluierungsphase DINOv3 als leistungsstärksten Merkmalsextraktor im Low-Data-Regime identifiziert habe, fokussiere ich mich im zweiten Schritt auf die Effizienz der räumlichen Rekonstruktion. Um das Zusammenwirken zwischen Encoder-Repräsentationen und Decoder-Architekturen umfassend zu quantifizieren, erweitere ich den Decoder-Vergleich bewusst um den hierarchischen Swin-Transformer. Mit seinen 86,75 M Parametern bildet er den idealen Kontrast zu den 326,39 M des DINOv3-Modells. Dies ermöglicht die Evaluierung der methodischen Hypothese, ob ein kapazitätsstarker Decoder die geringere Repräsentationskraft eines leichten Encoders kompensieren und so einen hocheffizienten Kompromiss für ressourcenlimitierte Anwendungsszenarien darstellen kann. Tabelle 4.3 fasst die Ergebnisse der fünf Trainingsläufe zusammen. Zur Wahrung der Fairness operieren alle Kombinationen mit der im Vorfeld (gemäß Abschnitt 3.3.4) modellspezifisch ermittelten optimalen Basislernrate. Als methodische Basislinie dient hierbei der UperNet-Decoder, welcher in der vorangegangenen Phase als konstanter Klassifikationskopf fungierte.

Tabelle 4.3 – Ergebnisse der Decoder-Evaluierung im Low-Data-Regime ($N = 75$). Verglichen werden die beiden leistungsstärksten Backbones mit vier grundlegend verschiedenen Decoder-Paradigmen. Angegeben ist die mittlere mIoU sowie die Standardabweichung über die fünf unabhängigen Trainingsläufe. Die angegebene Parameteranzahl (Param.) bezieht sich isoliert auf den jeweiligen Klassifikationskopf (exklusive Backbone). Die leistungsstärkste Gesamtkombination aus Backbone und Decoder ist fett hervorgehoben.

Decoder (Head)	Backbone	Param. (M)	mIoU (%)
Pyramid Pooling Basislinie			
UperNet	DINOv3-Large (LVD)	32,26	43,70 ± 0,78
UperNet	Swin-Base (ImageNet)	32,26	41,68 ± 1,32
Linear / MLP			
SegFormer	**DINOv3-Large (LVD)**	**3,16**	**44,60 ± 0,71**
SegFormer	Swin-Base (ImageNet)	3,16	41,91 ± 1,17
Faltungsbasiert (ASPP)			
DeepLabV3+	DINOv3-Large (LVD)	15,10	43,82 ± 1,11
DeepLabV3+	Swin-Base (ImageNet)	15,10	39,53 ± 1,58
Maskenklassifikation			
Mask2Former	DINOv3-Large (LVD)	59,00	40,67 ± 2,33
Mask2Former	Swin-Base (ImageNet)	59,00	42,84 ± 1,10

Die Ergebnisse der Decoder-Evaluierung zeigen ein auf den ersten Blick überraschendes Resultat: Der architektonisch simpelste Klassifikationskopf, der All-MLP-Decoder der SegFormer-Architektur, erzielt in Kombination mit dem DINOv3-Backbone die höchste Segmentierungsgüte (44,60 %) und übertrifft damit nicht nur die UperNet-Basislinie, sondern auch komplexe Ansätze nach dem aktuellen Stand der Technik wie Mask2Former oder DeepLabV3+.

Bei einer detaillierten Betrachtung der Trainingsdynamik unter starker Datenknappheit erweist sich dieses Resultat jedoch als methodisch schlüssig. Die Überlegenheit der SegFormer-DINOv3-Kombination lässt sich auf zwei sich ergänzende Effekte zurückführen:

1. **Vermeidung von Overfitting durch parametrische Reduktion:** In einem Low-Data-Regime mit einer effektiv annotierten Fläche von deutlich unter $1\,\mathrm{km}^2$ werden kapazitätsstarke Decoder zum limitierenden Faktor. Ansätze wie Mask2Former erfordern aufgrund ihres komplexen Bipartite-Matching-Verfahrens und der maskierten Attention-Schichten typischerweise zehntausende Trainingsbeispiele. Auf dem vorliegenden Datensatz neigen diese hochkomplexen Köpfe zu starker Überanpassung: Sie memorisieren das statistische Rauschen, anstatt generische Konzepte zu verinnerlichen. Auch die mittelkomplexe Feature-Pyramide des UperNet bindet hier noch zu viele Ressourcen. Der SegFormer-Decoder umgeht dieses Problem durch seine architektonische Einfachheit. Da er ausschließlich aus linearen Projektionsschichten besteht, besitzt er mit lediglich 3,16 M Parametern nur einen Bruchteil der Kapazität eines UperNet- (32,26 M) oder Mask2Former-Kopfes (59,00 M). Er fungiert primär als leichtgewichtiger Flaschenhals, der einer Überanpassung aktiv entgegenwirkt.

2. **Vollständige Ausnutzung der Merkmale des Basismodells:** Ein solch simpler Decoder ist jedoch nur dann funktionsfähig, wenn die vorgeschalteten Merkmalskarten bereits extrem dicht und linear separierbar sind. Das selbstüberwacht vortrainierte DINOv3-Modell erfüllt exakt diese Anforderung. Da der Transformer-Backbone durch seine tiefen Self-Attention-Mechanismen den wesentlichen Teil der räumlichen und semantischen Kontextualisierung bereits geleistet hat, reicht eine lineare Hochskalierung durch den SegFormer-Kopf aus, um hochpräzise Masken zu generieren.

Diese empirischen Resultate belegen, dass für hochauflösende stadtökologische Analysen unter strenger Datenknappheit nicht die Maximierung der architektonischen Komplexität im Decoder zielführend ist. Vielmehr liegt das Optimum in der Paarung eines starken, repräsentationsstarken Basismodells mit einem parametereffizienten Klassifikationskopf. Folglich fixiere ich die Kombination aus DINOv3-Large,

dem modifizierten ViT-Adapter und dem All-MLP SegFormer-Decoder als finale Zielarchitektur für die nachfolgende Hyperparameter-Optimierung.

4.2.3 Hyperparameter-Optimierung

Basierend auf der in Abschnitt 3.3.3 definierten Suchstrategie mittels *Optuna* und asynchronem Hyperband-Pruning evaluiert dieser Abschnitt die Ergebnisse der 80 durchgeführten Trainingsläufe. Ziel war es, das volle architektonische Potenzial der zuvor identifizierten Kombination aus DINOv3 und SegFormer auszuschöpfen und an die spezifischen Herausforderungen des Low-Data-Regimes anzupassen.

Die Analyse der Parameter-Wichtigkeit liefert hierbei zentrale Erkenntnisse über die Trainingsdynamik von Basismodellen unter starker Datenknappheit. Tabelle 4.4 fasst die relative Bedeutung der einzelnen Hyperparameter sowie die final ermittelte, leistungsstärkste Konfiguration (Iteration 68) zusammen, während der vollständige Suchverlauf inklusive Pruning-Historie in Abbildung B.1 visualisiert ist.

Tabelle 4.4 – Die durch *Optuna* ermittelte, leistungsstärkste Hyperparameter-Konfiguration für die DINOv3-SegFormer-Architektur. Die Wichtigkeit beziffert den prozentualen Einfluss des jeweiligen Parameters auf die Zielfunktion (mIoU).

Parameter	Beschreibung	Wichtigkeit	Optimaler Wert
`drop_path_rate`	Stochastische Tiefen-Regularisierung	37 %	0,0356
`head_lr_mult`	LR-Faktor für den Decoder	30 %	6,4013
`lr`	Basislernrate (Encoder)	18 %	$4{,}36 \times 10^{-5}$
`warmup_ratio`	Anteil der Aufwärm-Phase	12 %	0,1420
`aug_prob`	Augmentierungs-Wahrscheinlichkeit	1 %	0,1097
`weight_decay`	L2-Regularisierung	1 %	0,0545

Entgegen der klassischen Annahme, dass die Lernrate den singulär wichtigsten Parameter darstellt, dominiert die stochastische Tiefen-Regularisierung (`drop_path_rate`) mit einem Einfluss von 37 % den Optimierungsprozess. Die Auswertung der Trainingsläufe zeigt dabei ein eindeutiges Muster: Hohe Segmentierungsgüten wurden ausschließlich dann erreicht, wenn dieser Parameter extrem gering ($< 0{,}05$) gewählt wurde. Dieses Verhalten ist für den Einsatz vortrainierter Basismodelle im Low-Data-Regime schlüssig. Die durch DINOv3 extrahierten Merkmalsrepräsentationen sind bereits derart informationstief und semantisch dicht, dass das zufällige Deaktivieren von Netzwerkschichten (Stochastic Depth) den Wissenstransfer stört. Anstatt einer Überanpassung entgegenzuwirken, zerstört eine hohe Drop-Path-Rate hier den präzisen Informationsfluss zum Decoder.

Den zweitgrößten Einflussfaktor bilden die Lernraten (`head_lr_mult` und `lr`) mit kumuliert 48 % Wichtigkeit. Die Ergebnisse bestätigen die in der Methodik

formulierte Notwendigkeit einer differenziellen Lernrate: Der *TPE*-Algorithmus konvergierte bei einem Multiplikator von ca. 6,4. Der SegFormer-Kopf muss demnach mehr als sechsmal schneller lernen als der DINOv3-Backbone. Diese Entkopplung ist essenziell, um den zufällig initialisierten Klassifikationskopf schnell an die Zieldomäne anzupassen, ohne gleichzeitig durch zu große Gradientenanpassungen ein katastrophales Vergessen (engl. *Catastrophic Forgetting*) der wertvollen Encoder-Gewichte zu provozieren.

Klassische Regularisierungsansätze wie L2-Regularisierung (`weight_decay`) und Bildaugmentierung (`aug_prob`) spielten für die Variation der Segmentierungsgüte in diesem spezifischen Suchraum eine vernachlässigbare Rolle (jeweils 1 %).

Durch die Fixierung dieser optimal kalibrierten Konfiguration erreichte die Architektur über die fünf Trainingsläufe (Iteration 68) eine mIoU von 48,15 ± 0,71 % (im Vergleich zu 44,60 ± 0,71 % mit Standard-Parametern). Besonders hervorzuheben ist hierbei, dass die Optimierung die absolute Segmentierungsgüte um etwa dreieinhalb Prozentpunkte steigerte, während die Stabilität des Modells über wechselnde Trainingsteilmengen hinweg konstant auf dem Niveau der Baseline erhalten blieb. Dieser kalibrierte Zustand bildet somit die leistungsmaximierte methodische Grundlage für die nachfolgende Validierung der Dateneffizienz.

4.2.4 Analyse der Dateneffizienz

Um die Leistungsstabilität der gewählten DINOv3-SegFormer-Architektur und der Hyperparameter unter extremen Kaltstart-Bedingungen endgültig zu quantifizieren und die methodische Definition des Low-Data-Regimes ($N = 75$) nachträglich zu legitimieren, untersucht dieser finale Schritt der Architektursuche die Dateneffizienz. Hierzu skaliere ich die Trainingsmenge schrittweise von $N = 25$ auf den gesamten Datensatz ($N \approx 14\,000$). Die hierfür benötigten Teilmengen der Größen $N \in \{25, 50, 100, 200, 500, 1\,000\}$ werden mit dem in Abschnitt 3.3.2 beschriebenen Selektionsverfahren aus dem FLAIR-Trainingsdatensatz gezogen. Hinzu kommen der bereits aus der Architektursuche vorliegende Stützpunkt bei $N = 75$ sowie das Training auf dem vollständigen Datensatz. Der Kurvenverlauf in Abbildung 4.3 visualisiert die korrespondierende Segmentierungsgüte (mIoU) in Abhängigkeit der verfügbaren Trainingsbeispiele und zeigt ein stark ausgeprägtes, logarithmisches Sättigungsverhalten.

Die Daten belegen deutlich die Stichprobeneffizienz der selbstüberwacht gelernten DINOv3-Merkmale. Bereits bei minimaler Datenzugabe zeigt das Modell eine steile Lernkurve: Eine Verdopplung der initialen Trainingsmenge von 25 auf 50 Bilder führt zu einem signifikanten Leistungssprung von $+5,31$ Prozentpunkten. Im zuvor für die Architektursuche definierten Low-Data-Regime ($N = 75$) erreicht die Pipeline eine mIoU von 48,15 %.

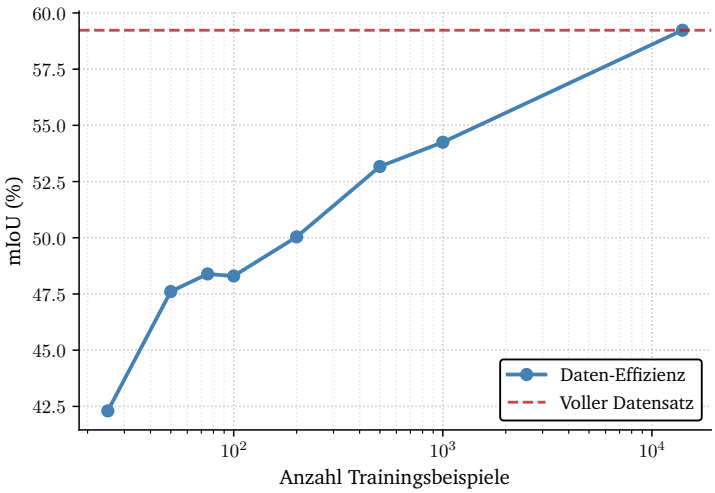


Abbildung 4.3 – Dateneffizienz des Modells. Dargestellt ist die Segmentierungsgüte (mIoU) in Abhängigkeit der Anzahl verwendeter Trainingsbeispiele (logarithmische Skalierung). Die horizontale gestrichelte Linie markiert die Referenzleistung (59,23 mIoU) des Modells, welches auf dem vollständigen Datensatz mit etwa 14 000 Beispielen trainiert wurde.

Setzt man diesen Wert in Relation zur Referenzleistung des Modells, das auf dem vollständigen Datensatz trainiert wurde (59,23 %), zeigt sich das zentrale methodische Ergebnis dieser Evaluierung: Mit weniger als 0,6 % der gesamten Trainingsdaten erreicht die gewählte Architektur bereits 81,3 % ihrer maximalen Leistungsfähigkeit.

Mit zunehmender Datenmenge unterliegt der Leistungszuwachs einem starken Sättigungseffekt. Eine Verzehnfachung der Daten von 100 auf 1 000 Beispiele ($N = 1\,000$) steigert die Leistung lediglich noch um knapp 6 Prozentpunkte (auf 54,25 %). Die Hochskalierung auf den gesamten Datensatz (eine weitere Vervierzehnfachung der Daten) liefert schließlich nur noch die verbleibenden 5 Prozentpunkte zur Sättigungsgrenze.

Aus diesen Ergebnissen leiten sich zwei methodische Konsequenzen ab:

1. **Legitimation des Low-Data-Regimes:** Die Wahl von $N = 75$ für die Architektursuche und Hyperparameter-Optimierung erweist sich rückwirkend als optimaler methodischer Kompromiss. Das Modell hat hier die grundlegende Semantik der Domäne bereits verinnerlicht (hohes Basisniveau), befindet sich aber noch im steilen Bereich der Lernkurve, in dem architektonische Verbesserungen und Hyperparameter-Anpassungen noch deutlich messbare Ausschläge in den Metriken verursachen, ohne durch die Datenmasse geglättet zu werden.

2. **Rechtfertigung für den eigenen Anwendungsdatensatz:** Die Analyse beweist, dass das Trainieren performanter Segmentierungsmodelle für 20 cm-Orthophotos durch den Einsatz starker Basismodelle keine großen, zehntausen-

de Bilder umfassenden Annotationskampagnen mehr erfordert. Die Erkenntnis, dass sich die Leistungskurve bereits bei sehr geringen Datenmengen abflacht, liefert die wissenschaftliche Legitimation, den Transfer auf die finale, manuelle bayerische Zieldomäne (vgl. Abschnitt 3.4) mit einer stark begrenzten, aber qualitativ hochwertigen Anzahl an Kacheln durchzuführen.

4.3 Ergebnisse auf dem eigenen Datensatz

Nachdem die vorangegangene Architektursuche auf dem Stellvertreterdatensatz (FLAIR) die Modellkonfiguration ermittelt hat, vollzieht dieser Abschnitt den Transfer auf die eigentliche Zieldomäne. Die empirischen Erkenntnisse des Low-Data-Regimes, konkret die Kombination des repräsentationsstarken, selbstüberwachten DINOv3-Backbones mit einem parametereffizienten SegFormer-Decoder sowie die durch Optuna optimierten Hyperparameter, bilden die methodische Grundlage für die nun folgende Praxisanwendung.

Um eine unvoreingenommene Evaluierung zu gewährleisten und eine methodische Überanpassung auf die Zieldaten auszuschließen, fixiere ich sowohl die Architektur als auch die Trainingsdynamik für diesen Schritt strikt. Ich wende das Modell nun ausschließlich auf den selbst erstellten, feingranularen 15-Klassen-Datensatz der bayerischen Untersuchungsgebiete (Nürnberg und Erlangen) an.

Dieser finale Analyseschritt gliedert sich in drei zentrale Ebenen, um die praktische Einsatzfähigkeit der Pipeline ganzheitlich zu bewerten: Zunächst wird die Qualität und Variabilität der manuell erstellten Referenzdaten analysiert, da diese das absolute Leistungslimit für jedes überwachte Lernverfahren definieren. Darauf aufbauend evaluiert dieser Abschnitt die reine Segmentierungsgüte und Kalibrierung des Netzwerks im Rahmen der räumlichen Kreuzvalidierung. Den Abschluss bildet die Bewertung der aus den Masken abgeleiteten Flächen- und Solarindikatoren, um aufzuzeigen, wie präzise sich die pixelbasierten Vorhersagen in belastbare Kennzahlen für die urbane Planungspraxis übersetzen lassen.

4.3.1 Analyse der Annotationsqualität und -variabilität

Wie im Methodikkapitel (vgl. Abschnitt 3.1.2) dargelegt, stellt die manuelle semantische Segmentierung von hochauflösenden DOP20-Aufnahmen eine erhebliche methodische Herausforderung dar. Um das Ausmaß dieser Herausforderung sowie den notwendigen Qualitätssicherungsaufwand quantitativ zu erfassen, vergleicht die Auswertung die initiale Annotation der Hilfskräfte mit der final korrigierten Version anhand der mIoU. Die aggregierten Ergebnisse nach Annotatoren sind in Tabelle 4.5 zusammengefasst.

Tabelle 4.5 – Zusammenfassung der Annotationsqualität. Die mIoU wurde durch den Vergleich der initialen Hilfskraft-Annotation mit der finalen, korrigierten Version berechnet. Die bearbeitete Fläche leitet sich aus der Kachelgröße von $512 \times 512\,\text{m}$ ($\approx 0.26\,\text{km}^2$ pro Kachel) ab. Die detaillierte Charakterisierung der referenzierten Gebiets-IDs findet sich in Tabelle 3.5.

Annotator	Fläche	Gebiete (IDs)	mIoU
Annotator 1	$0.79\,\text{km}^2$	E1, E3, N3	0,7189
Annotator 2	$0.26\,\text{km}^2$	N1	0,5286
Annotator 3	$0.52\,\text{km}^2$	N2, N4	0,8459
Autor (Referenz) *	$0.26\,\text{km}^2$	E2	1,0000
Gesamt (Hilfskräfte)	$\mathbf{1.57\,\text{km}^2}$		**0,7295**

* Da die Korrektur durch den Autor erfolgte, dient dieses Gebiet als interne Referenz (mIoU = 1,0) und fließt nicht in den aggregierten Gesamtwert ein.

Die Auswertung zeigt mit einer durchschnittlichen mIoU von 0,7295 eine deutliche Varianz in der Annotationsqualität. Dieser Wert verdeutlicht, wie anspruchsvoll die manuelle Interpretation der vorliegenden Fernerkundungsdaten ist. Selbst unter Zuhilfenahme eines definierten Regelwerks und externer Referenzen (*Google Street View*) weichen die Einschätzungen der Hilfskräfte im Schnitt um über 27 % von der qualitätsgesicherten Referenz ab.

Besonders auffällig ist die Diskrepanz zwischen den einzelnen Bearbeitern und Gebieten. Während Annotator 3 (zuständig für Nbg-Mooshof und Nbg-Ostring) mit einer mIoU von 0,8459 eine vergleichsweise hohe Übereinstimmung erzielt, fällt der Wert bei Annotator 2 (Nbg-Galgenhof) auf 0,5286 ab. Diese starke Abweichung lässt sich auf eine Kombination aus zwei Faktoren zurückführen: Einerseits erfordert das Gebiet N1 (Galgenhof) aufgrund seiner extrem dichten Altstadtbebauung, der verschachtelten Hinterhöfe und tiefer Schattenwürfe ein hohes Maß an Konzentration und Detailgenauigkeit. Im Gegensatz dazu umfassen die von Annotator 3 bearbeiteten Gebiete vorwiegend großflächige Gewerbestrukturen, die klarere geometrische Abgrenzungen zulassen. Andererseits spiegeln sich in derart niedrigen Werten auch die generellen methodischen Grenzen manueller Datenakquise wider. Die pixelgenaue Segmentierung ist ein hochgradig repetitiver Prozess, der inhärent anfällig für menschliche Ermüdungseffekte und subjektive Interpretationsspielräume ist. Dies führt unweigerlich zu einer messbaren Inter-Annotator-Variabilität und inkonsistenten Auslegungen des Annotationsleitfadens, was den nachgelagerten, standardisierten Überprüfungsprozess zwingend erforderlich macht.

Die detaillierte Betrachtung der Einzelergebnisse (siehe Tabelle A.1) zeigt zudem, dass insbesondere die Abgrenzung semantisch ähnlicher Klassen (z. B. *Crushed Stone* vs. *Unvegetated Substrate* oder verschiedene Grade der Versiegelung) stark fehleranfällig ist.

Die Ergebnisse dieser Analyse rechtfertigen meinen methodischen Entschluss, den Datensatz einem systematischen Überprüfungsprozess zu unterziehen. Ohne diese Korrekturphase hätte das Modell während des Trainings ein erhebliches Maß an Annotationsrauschen gelernt, was die Generalisierungsleistung auf ungesehene Gebiete maßgeblich beeinträchtigt hätte. Gleichzeitig unterstreichen diese Werte die Limitierung manuell erstellter Referenzdaten: Absolute Fehlerfreiheit ist in komplexen urbanen Räumen auf Basis von 20 cm-Orthophotos durch menschliche Annotation kaum zu gewährleisten.

4.3.2 Ablationsstudie zur multimodalen Datenfusion

Nachdem die fundamentale Qualität und die methodischen Grenzen der Referenzdaten geklärt sind, gilt es vor der finalen Kreuzvalidierung zunächst, die optimalen Eingangsmodalitäten des Modells zu fixieren. Wie im Methodikkapitel (vgl. Abschnitt 3.4.6) dargelegt, integriere ich zur Steigerung der semantischen Trennschärfe neben den optischen RGB-Kanälen auch den NIR-Kanal sowie das normalisierte Oberflächenmodell in das Netz. Um die effektivste Strategie zur Verknüpfung dieser fünf Kanäle mit dem auf drei Kanäle limitierten, vortrainierten DINOv3-Backbone zu ermitteln, vergleicht dieser Abschnitt die reine RGB-Basislinie mit den Fusionsstrategien *Vorgeschaltete Kanalfusion* und *Strukturelle Gewichtserweiterung*.

Tabelle 4.6 fasst die quantitativen Ergebnisse dieser Evaluierung zusammen. Die Auswertung bestätigt die Hypothese, dass die Integration von Höhen- und Infrarotdaten einen messbaren Mehrwert liefert, zeigt jedoch starke Abhängigkeiten von der gewählten Integrationsmethode.

Tabelle 4.6 – Ergebnisse der Ablationsstudie zur Integration zusätzlicher Datenkanäle (NIR und nDOM). Verglichen werden die RGB-Basislinie mit zwei unterschiedlichen Fusionsstrategien. Angegeben ist die mittlere mIoU auf dem Validierungsdatensatz. Die leistungsstärkste Methode ist fett hervorgehoben.

Eingabe	Kanäle	Integrationsstrategie	mIoU (%)
RGB	3	Basislinie (Standard-ViT)	45,73
RGB + NIR + nDOM	5	Strukturelle Gewichtserweiterung	40,69
RGB + NIR + nDOM	**5**	**Vorgeschaltete Kanalfusion**	**47,26**

Die strukturelle Gewichtserweiterung, bei der ich das Modell um zwei nullinitialisierte Eingangskanäle erweiterte, erzielt eine mIoU von 40,69 % und fällt damit sogar deutlich hinter das Niveau der reinen RGB-Basislinie (45,73 %) zurück. Dieses Verhalten lässt sich auf die Störung der stark optimierten, vortrainierten Aufmerksamkeitsmechanismen des Basismodells zurückführen. Da die neuen Kanäle initial kein nützliches Signal liefern, benötigt das Netzwerk im streng limitierten Low-

Data-Regime zu viele Iterationen, um die Repräsentationen der neuen Modalitäten effektiv mit den vortrainierten RGB-Gewichten in Einklang zu bringen.

Die vorgeschaltete Kanalfusion umgeht dieses Problem und erzielt mit 47,26 % die höchste Segmentierungsgüte. Die vorgeschaltete 1×1-Faltung fungiert hierbei als adaptiver, lernbarer Flaschenhals: Sie projiziert den multimodalen Fünf-Kanal-Tensor weich auf die vom Encoder erwarteten drei Dimensionen. Das Modell lernt somit dynamisch, die für die Klassenunterscheidung (z. B. Gebäudehöhe vs. Bodenversiegelung) relevantesten Informationen aus dem nDOM und NIR-Kanal zu extrahieren und in eine für den Backbone vertraute Repräsentation zu übersetzen, ohne die wertvollen Gewichte des Basismodells aus der ersten Schicht zu destabilisieren.

Aufgrund dieser klaren Leistungssteigerung bildet die vorgeschaltete Kanalfusion die architektonische Basis für alle nachfolgenden Analysen, einschließlich der Testzeit-Datenaugmentierung und der Inferenz zur Indikatorberechnung.

4.3.3 Ablationsstudie zur Inferenzskalierung und TTA

Um den Einfluss der räumlichen Bildskalierung auf die Segmentierungsgüte, insbesondere bei kleinteiligen und geometrisch anspruchsvollen Klassen wie PV-Anlagen, zu quantifizieren, evaluiere ich die finalen Modelle in einer Ablationsstudie. Dabei zeigt sich ein klassischer Zielkonflikt zwischen der exakten Erfassung lokaler Details und dem Verständnis des globalen räumlichen Kontexts.

Die Standard-Inferenz auf der nativen Bildauflösung (1x-Skalierung) erfasst großflächige Strukturen zuverlässig, weist jedoch Schwächen an feinen Objektkanten sowie bei der Detektion von Kleinstobjekten auf. Der Grund hierfür liegt in der Architektur des verwendeten Basismodells: Der DINOv3-Backbone (ViT-L) zerlegt das Eingangsbild in sogenannte Patches einer festen Größe von 16×16 Pixeln. Bei einer Bodenauflösung von 20 cm repräsentiert ein einzelner Patch somit eine reale Fläche von $3,2 \times 3,2$ Metern. Objekte, die kleiner als dieser Bereich sind oder ungünstig über Patch-Grenzen hinweg verlaufen, wie etwa kompakte Dachaufbauten oder schmale PV-Module, lassen sich durch diese grobe Tokenisierung nur schwer mit exakten Kanten rekonstruieren.

Eine künstliche Hochskalierung der Eingangsdaten (3x-Skalierung) während des Trainings und der Inferenz behebt dieses Problem lokal: Ein 16×16-Patch deckt nun nur noch eine reale Fläche von ca. $1,07 \times 1,07$ Metern ab. Dadurch bildet das Modell die Kanten von PV-Anlagen deutlich schärfer ab und die Detektionsrate kleiner Dachaufbauten steigt spürbar. Allerdings profitiert die globale Gesamt-mIoU nur marginal (siehe Tabelle 4.7). Dieser ausbleibende globale Leistungssprung lässt sich auf das geschrumpfte rezeptive Feld des Netzwerks zurückführen: Durch den faktisch verengten Bildausschnitt fehlt dem Modell der globale Kontext, um weitreichende Strukturen fehlerfrei in ihren urbanen Zusammenhang einzuordnen.

Tabelle 4.7 – Ergebnisse der Ablationsstudie zur Inferenzskalierung. Verglichen werden die Standard-Inferenz auf nativer Bildauflösung (1x), eine künstliche 3x-Hochskalierung der Eingangsdaten sowie die Testzeit-Datenaugmentierung (TTA). Angegeben sind die globale mIoU, die klassenspezifische IoU für PV-Anlagen sowie die jeweilige Inferenzzeit pro Untersuchungsgebiet (512×512 m).

Ansatz	mIoU (%)	PV-IoU (%)	Inferenzzeit
1x (Nativ)	47,26	69,04	0,48 min
3x (Skaliert)	47,76	76,81	3,23 min
TTA	49,89	75,10	9,98 min

Die Erweiterung der Pipeline um eine Testzeit-Datenaugmentierung (TTA), konkret eine Kombination aus Multiskalen-Inferenz und Spiegelungen, löst diesen Zielkonflikt algorithmisch. Die Konsensbildung über verschiedene Transformationen hinweg glättet Vorhersagefehler und vereint die scharfen Objektgrenzen der hochskalierten Ansichten mit dem globalen Kontextverständnis der nativen Auflösung. Dies resultiert in der höchsten erreichten Segmentierungsgüte. Der qualitative Vergleich in Abbildung 4.4 macht diese Unterschiede zwischen den Ansätzen anschaulich sichtbar.

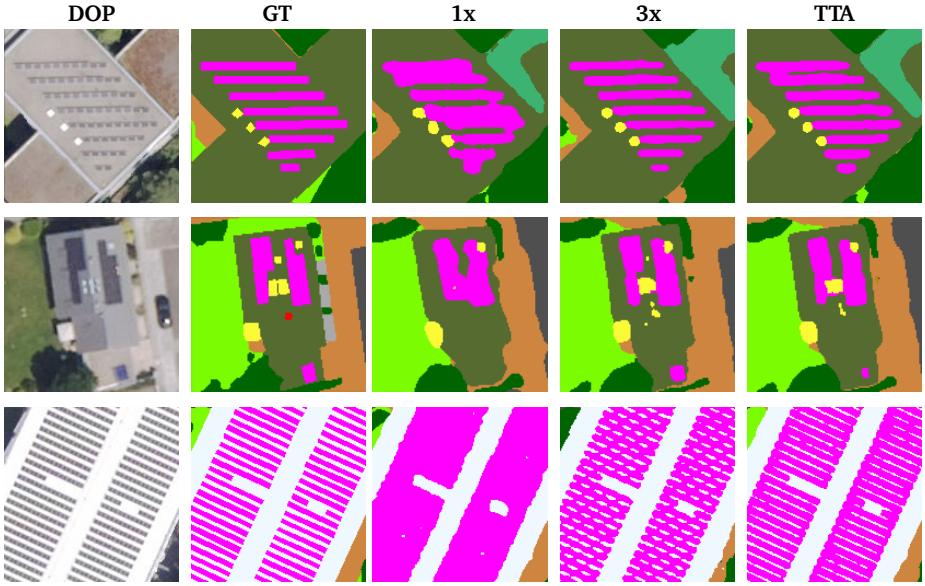


Abbildung 4.4 – Qualitativer Vergleich der Segmentierungsansätze an Dachstrukturen mit PV-Anlagen (■). Die Spalten zeigen von links nach rechts: Luftbild (DOP), Referenzmaske (GT), Standard-Inferenz (1x), Inferenz mit 3x-Hochskalierung sowie die Testzeit-Datenaugmentierung (TTA). Die 3x-Skalierung rekonstruiert die Objektkanten der PV-Module schärfer als die native 1x-Auflösung, während die TTA Detailtreue und globalen Kontext vereint.

Wie die Betrachtung der Inferenzzeiten (siehe Tabelle 4.7) jedoch verdeutlicht, vervielfacht sich der Rechenaufwand durch die mehrfache Vorhersage pro Kachel. Während die TTA somit das theoretische Leistungsmaximum der Architektur aufzeigt, schließt sie sich für eine produktive, flächendeckende Datenerfassung (z. B. auf Bundeslandebene) aus Latenzgründen aus. Um die stadtökologischen Indikatoren und das Solarpotenzial im Folgenden dennoch auf ihre maximal erreichbare geometrische Präzision hin zu evaluieren, basieren alle nachfolgenden Analysen (Klassenmetriken, Konfusionsmatrizen und Downstream-Evaluierung) auf den Vorhersagen des leistungsstärksten Modells inkl. TTA.

4.3.4 Modellleistung in der räumlichen Kreuzvalidierung

Um die Generalisierungsfähigkeit des Modells auf ungesehene Stadtstrukturen zu quantifizieren, erfolgt eine räumliche Leave-One-Out-Kreuzvalidierung über die definierten Untersuchungsgebiete. Die aggregierten Ergebnisse der Vorhersagen (Inferenz mit TTA und 3x-Skalierung) sind in Tabelle 4.8 detailliert aufgeschlüsselt. Mit einer PA von 80,10 % zeigt das Modell eine solide Basisleistung, offenbart jedoch durch die deutlich niedrigere mIoU (49,89 %) und mittlere Sensitivität (61,50 %) ein stark klassenspezifisches und räumlich variierendes Fehlerprofil.

Tabelle 4.8 – Aggregierte klassenspezifische Metriken der räumlichen Leave-One-Out-Kreuzvalidierung. Angegeben sind Präzision (*Precision*), Sensitivität (*Recall*), F1-Score und die Intersection over Union (IoU) in Prozent. Die Klassen sind absteigend nach ihrer IoU sortiert. Die Zeile *Gesamt* bezeichnet das ungewichtete arithmetische Mittel über alle 15 Klassen.

Klasse	Präzision	Sensitivität	F1-Score	IoU
Dark Roof	87,57	87,62	87,59	77,93
Tall Vegetation	86,03	86,81	86,42	76,08
Photovoltaic System	78,11	95,12	85,78	75,10
Dark Hard Sealing	82,03	81,37	81,69	69,05
Low Vegetation	75,31	80,41	77,78	63,63
Light Roof	76,47	77,96	77,21	62,87
Strong Partial Sealing	74,29	75,14	74,71	59,63
Green Roof	78,08	71,11	74,44	59,28
Technical Roof Structures	69,50	73,54	71,47	55,60
Crushed Stone / Gravel	72,05	57,17	63,75	46,79
Other Roof Structures	56,64	62,56	59,46	42,31
Unvegetated Substrate	54,62	31,12	39,65	24,73
Low Partial Sealing	55,10	28,98	37,99	23,45
Light Hard Sealing	48,75	13,53	21,18	11,84
Water Bodies	0,00	0,00	0,00	0,00
Gesamt	**66,30**	**61,50**	**62,61**	**49,89**

Einfluss der Klassenhäufigkeit und räumliche Artefakte

Ein Abgleich der segmentspezifischen Evaluierungsmetriken (siehe Tabelle 4.8) mit der statistischen Klassenverteilung des Datensatzes (vgl. Tabelle 3.6) zeigt eine nahezu lineare Korrelation zwischen der absoluten Pixelanzahl einer Klasse und ihrer Vorhersagegüte. Das Modell weist eine deutliche Präferenz für die dominanten Klassen auf: Die fünf flächenmäßig größten Klassen (u. a. *Dark Roof*, *Strong Partial Sealing*, *Tall Vegetation*) erzielen überwiegend die höchsten IoU-Werte. Im Gegensatz dazu scheitert das Modell fast vollständig an den Randklassen der Verteilung, wie etwa *Light Hard Sealing* (0,22 % Flächenanteil, 11,84 % IoU).

Dies bestätigt die systembedingte Tendenz tiefer neuronaler Netze, sich auf die Mehrheitsklassen zu fokussieren, um den globalen Verlust während des Trainings zu minimieren. Eine bemerkenswerte Ausnahme von dieser Regel bildet jedoch die Klasse *Photovoltaic System*. Obwohl sie mit nur 1,44 % Flächenanteil stark unterrepräsentiert ist, erreicht sie den dritthöchsten IoU-Wert (75,10 %). Dies lässt sich auf die starke visuelle und geometrische Eindeutigkeit der PV-Module (scharfe rechteckige Kanten, homogene dunkle Textur) zurückführen. Stark ausgeprägte Merkmale können ein quantitatives Klassenungleichgewicht demnach teilweise kompensieren.

Das *Water-Bodies*-Artefakt

Während die Klasse der PV-Anlagen belegt, dass visuelle Eindeutigkeit eine geringe Flächenrepräsentanz ausgleichen kann, zeigt die Klasse *Water Bodies* ein gegenteiliges Extrem: das methodische Scheitern an einer ungünstigen räumlichen Verteilung. Diese Klasse erfordert besondere Aufmerksamkeit, da sie über alle Validierungsdurchläufe hinweg eine mIoU von 0,00 % aufweist. Ein isolierter Blick auf diese Metrik könnte die Fehlinterpretation zulassen, das Modell sei grundsätzlich unfähig, Wasserflächen zu detektieren. Tatsächlich handelt es sich hierbei jedoch um ein methodisches Artefakt, das durch die räumliche Clusterbildung der Klasse in Kombination mit der Leave-One-Out-Kreuzvalidierung entsteht.

Die Pixel der Klasse *Water Bodies* sind im vorliegenden Datensatz sehr ungleichmäßig verteilt, sie konzentrieren sich nahezu vollkommen auf ein einziges Untersuchungsgebiet. Dies führt in der Kreuzvalidierung zu folgendem methodischen Konflikt:

- Fungiert das Gebiet mit der großen Wasserfläche als Validierungsdatensatz (Hold-out), enthält das verbleibende Trainingsset nahezu keine Wasserpixel. Das Modell kann die Klasse folglich nicht erlernen und ignoriert sie bei der Inferenz.

- Dient das wasserreiche Gebiet hingegen als Trainingsdaten, enthält das Validierungsgebiet entweder gar kein Wasser oder lediglich morphologisch völlig

abweichende Kleinstinstanzen (z. B. kleine, trübe Gartenteiche). Das auf große, offene Wasserflächen trainierte Modell transferiert dieses Wissen nicht auf diese minimalen, optisch abweichenden Instanzen.

Dieses Ergebnis zeigt eine fundamentale methodische Grenze der räumlichen Kreuzvalidierung auf kleinen, geographisch stark geclusterten Datensätzen: Klassen, die nicht über mehrere Gebiete hinweg repräsentativ gestreut sind, können durch diesen Evaluierungsansatz nicht valide bewertet werden. Für eine flächendeckende Anwendung des Modells ist jedoch stark davon auszugehen, dass die Architektur bei entsprechender Datenpräsenz problemlos in der Lage ist, Wasserflächen zuverlässig zu segmentieren.

Ein theoretischer Lösungsansatz für dieses Problem wäre ein sogenannter *Stratified Spatial Split*, bei dem die Daten so in Teilmengen gruppiert oder zugeschnitten werden, dass jede Klasse in jedem Validierungsdurchlauf repräsentativ vertreten ist. In der vorliegenden Arbeit wurde sich jedoch bewusst gegen einen solchen kleinteiligeren Zuschnitt und für die strikte räumliche Trennung anhand der vorgegebenen, großen Untersuchungskacheln entschieden. Der Grund hierfür liegt in der Vermeidung von räumlicher Autokorrelation. Somit stellt das *Water-Bodies*-Artefakt keinen Fehler des Modells dar, sondern ist der methodische Preis eines kompromisslos strikten Validierungsdesigns, welches die korrekte Messung der echten Generalisierungsleistung über die isolierte Auswertbarkeit stark geclusterter Randklassen stellt.

Räumliche Varianz und strukturelle Komplexität

Die Auswertung der einzelnen Validierungsdurchläufe zeigt eine signifikante räumliche Varianz der Modellgüte (siehe Abbildung 4.5). Es lässt sich ein klares Leistungsgefälle entlang der strukturellen Komplexität der Gebiete erkennen (vgl. Tabelle 3.5): Während Areale, die durch großflächige Industrie- und Gewerbestrukturen, weitläufige Campusanlagen sowie aufgelockerte Stadtrandlagen geprägt sind (wie E2, E3, N2 und E1), durchweg mIoU-Werte von über 50 % erreichen, ordnen sich heterogenere Rand- und Gewerbegebiete (wie N3 und N4) mit Werten zwischen 41 % und gut 45 % im Mittelfeld ein.

Der stärkste Leistungseinbruch zeigt sich jedoch in dem als dichte Bebauung klassifizierten Gebiet Nbg-Galgenhof (N1) mit lediglich 37,95 %. Dieses Resultat ist schlüssig und korreliert direkt mit den Ergebnissen der initialen Annotatoren-Analyse (vgl. Abschnitt 4.3.1). Das Gebiet N1 zeichnet sich durch eine extrem dichte Blockrandbebauung aus. Verschachtelte Hinterhöfe, kleinräumige Versiegelungswechsel und vor allem tiefe, durch hohe Gebäude geworfene Schlagschatten zerstören die visuelle Kohärenz der optischen Merkmale. Es zeigt sich eine klare methodische Parallele: Wenn Formen und Verdeckungen in den DOP20-Daten so komplex sind,

dass sie für Menschen nicht mehr eindeutig erkennbar sind, tut sich auch das Modell schwer, daraus verlässliche Merkmale abzuleiten.

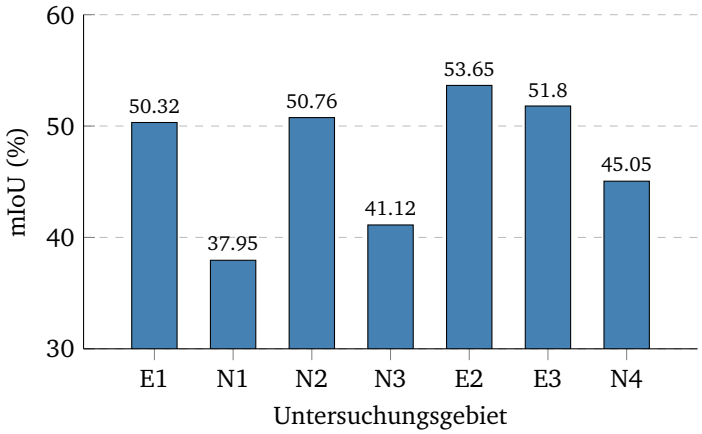


Abbildung 4.5 – Übersicht der erreichten Segmentierungsgüte (mIoU) pro Untersuchungsgebiet in der Leave-One-Out-Kreuzvalidierung. Der signifikante Leistungseinbruch bei Gebiet N1 (Nbg-Galgenhof) ist deutlich zu erkennen und verdeutlicht den Einfluss der urbanen Morphologie auf die Ergebnisqualität.

Präzision vs. Sensitivität am Beispiel der PV

Eine genauere Analyse von Präzision und Sensitivität zeigt spezifische Stärken und Schwächen des Modells auf, die bei einer reinen mIoU-Auswertung nicht sichtbar werden. Besonders für die Solarpotenzialanalyse ist die Klasse *Photovoltaic System* von zentraler Bedeutung. Hier erzielt das Modell eine Sensitivität von 95,12 %, bei einer etwas niedrigeren Präzision von 78,11 %.

Das bedeutet praktisch: Das Modell übersieht fast keine real existierenden Solaranlagen (hohe Sensitivität, wenig FN). Die Diskrepanz zur Präzision zeigt jedoch, dass das Modell zu einer leichten Übersegmentierung neigt (FP). Es detektiert fälschlicherweise gelegentlich PV-Anlagen, typischerweise indem es visuell ähnliche Strukturen wie dunkle Dachfenster oder Objekte auf einem Dach als PV klassifiziert.

Darüber hinaus zeigt sich bei der geometrischen Kantendetektion eine weitere Ursache für die reduzierte Präzision: Wie bereits im qualitativen Vergleich der Inferenzskalierungen (siehe Abbildung 4.4) ersichtlich wurde, kann das Modell die feinen Zwischenräume einzelner Modulreihen auf Flachdächern oftmals nicht trennscharf auflösen. Anstatt einzelne PV-Reihen trennscharf zu segmentieren, aggregiert das Netzwerk diese häufig zu einer durchgehenden, zusammenhängenden Fläche. Da das Modell somit auch die schmalen Dachflächen zwischen den Modulen als PV klassifiziert, entstehen zwangsläufig geometrische FP, welche die Präzisionsmetrik senken.

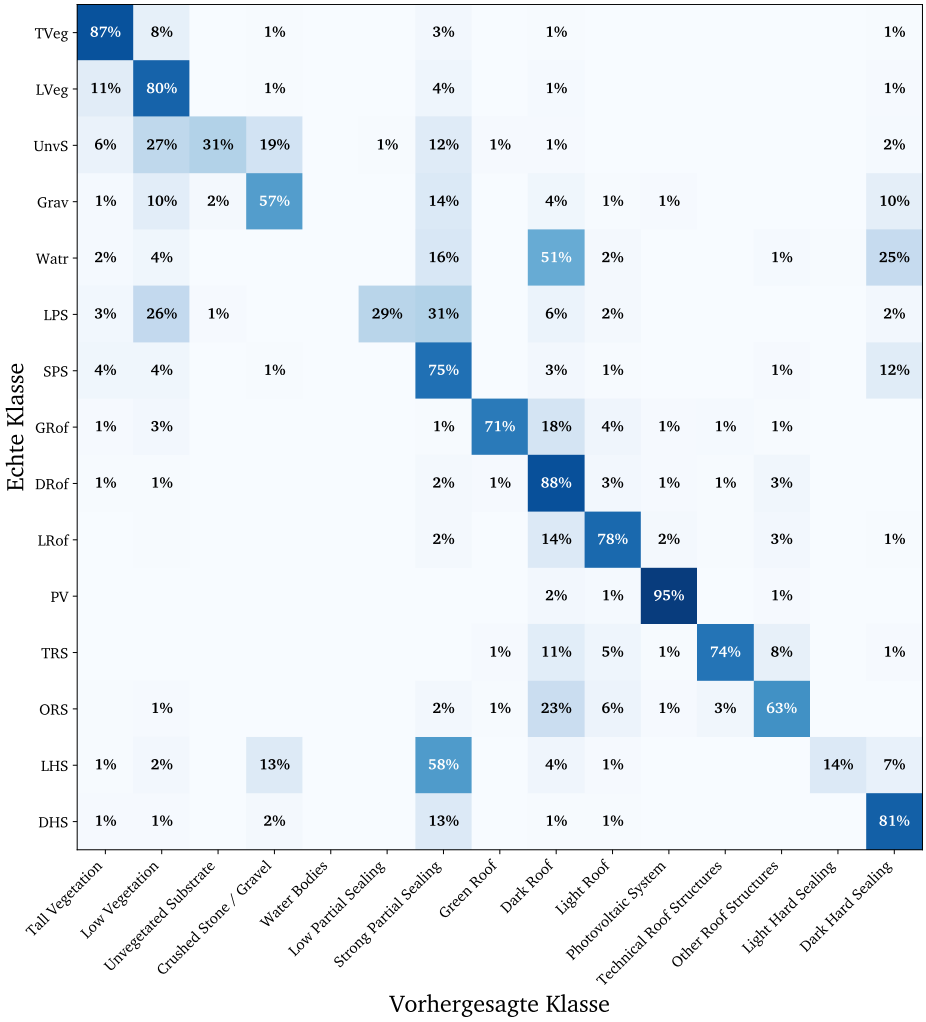
Für die automatisierte Potenzialanalyse ist dieses grundlegende Verhalten (hohe Sensitivität auf Kosten der Präzision) jedoch strategisch vorteilhaft: Ein konservatives Modell, das Anlagen tendenziell eher detektiert und leicht überschätzt (sodass diese in einem nachgelagerten Schritt gefiltert oder geometrisch bereinigt werden können), ist für die Praxisanwendung wertvoller als ein Modell, das kleine Dachflächenanlagen systematisch übersieht.

Weiterhin zeigt die Konfusionsmatrix in Abbildung 4.6 erhebliche systematische Verwechslungen, die sich bei genauerer Betrachtung jedoch als physikalisch und semantisch äußerst schlüssig erweisen. Es lassen sich primär vier Fehlercluster identifizieren, die auf die fließenden Übergänge in der urbanen Morphologie zurückzuführen sind:

- **Das physikalische Kontinuum und Mischpixel:** Die größten Unsicherheiten weist das Modell bei Bodenklassen auf, die in der Realität keine harten Grenzen, sondern stufenlose materielle Übergänge bilden. Dies betrifft vor allem das unbewachsene Substrat (*Unvegetated Substrate*), welches das Modell zu 27,28 % mit niedriger Vegetation und zu 18,62 % mit Schotter verwechselt. Das Modell scheitert hier nicht algorithmisch, sondern an der künstlichen Aufgabe, physikalisch fließende Übergänge (wie verwitternde Schotterflächen oder spärlich bewachsenen Boden) in ein diskretes Klassenschema pressen zu müssen.

- **Visuelle Textur-Äquivalenz harter Oberflächen:** Eine eng verwandte Fehlerquelle resultiert aus der fehlenden visuellen Trennschärfe unterschiedlicher Materialien in 20 cm-Luftbildern. Dies manifestiert sich exemplarisch bei der Klasse *Crushed Stone / Gravel*. Neben der korrekten Zuordnung (57,17 %) kommt es zu Verwechslungen mit starker Teilversiegelung (13,72 %) und dunkler Vollversiegelung (9,67 %). Obwohl diese Klassen unterschiedliche Versiegelungsgrade und bauliche Eigenschaften repräsentieren, erzeugen lose geschütteter Schotter, grobes Kopfsteinpflaster und stark verwitterter Asphalt ein nahezu identisches graues, wenig texturiertes Pixelmuster. Da das Modell ausschließlich auf diese zweidimensionale Textursignatur angewiesen ist, kommt es zwangsläufig zu Fehlallokationen.

- **Materialdominanz und Token-Glättung bei Dachstrukturen:** Auf der Ebene der Gebäude zeigt sich eine starke Tendenz, funktionale Unterklassen dem dominierenden Hauptmaterial zuzuordnen. So klassifiziert das Modell sonstige Dachstrukturen (*Other Roof Structures*) wie Schornsteine, Gauben oder Erker zu knapp 23 % als reguläres dunkles Dach (*Dark Roof*). Auch kleine begrünte Dachflächen (*Green Roof*) weisen eine knapp 18-prozentige Verwechslung mit dem dunklen Dach auf. Die Ursache liegt im Zusammenspiel aus Materialgleichheit und Architektur: Da Dachaufbauten oft identisch zum Hauptdach



TVeg = Tall Vegetation LVeg = Low Vegetation UnvS = Unvegetated Substrate
Grav = Crushed Stone / Gravel Watr = Water Bodies LPS = Low Partial Sealing
SPS = Strong Partial Sealing GRof = Green Roof DRof = Dark Roof
LRof = Light Roof PV = Photovoltaic System TRS = Technical Roof Structures
ORS = Other Roof Structures LHS = Light Hard Sealing DHS = Dark Hard Sealing

Abbildung 4.6 – Aggregierte Konfusionsmatrix der Leave-One-Out-Kreuzvalidierung, berechnet über alle Teilmengen auf Pixelebene. Die Zeilen entsprechen den echten, die Spalten den vorhergesagten Klassen; jede Zeile ist auf die Gesamtpixelzahl der jeweiligen Klasse normiert, sodass die Diagonalwerte den klassenweisen Recall wiedergeben. Die Farbintensität kodiert den jeweiligen Anteil zwischen 0 % (weiß) und 100 % (dunkelblau). Werte unter 1 % sind aus Lesbarkeitsgründen nicht beschriftet, in der Farbgebung jedoch weiterhin sichtbar. Die Kurzcodes der Zeilenbeschriftung sind unterhalb der Matrix aufgeschlüsselt.

eingedeckt sind und der DINOv3-Backbone mit festen *Patch-Tokens* operiert, glättet das Modell kleine, geometrisch nicht eindeutige Kanten und aggregiert sie in den semantischen Hintergrundkontext der großen Dachfläche.

- **Unterrepräsentation und das *Water-Bodies*-Artefakt:** Den vierten Cluster bildet der komplette Ausfall bei der Detektion von Wasserflächen. Wie bereits zuvor methodisch dargelegt, handelt es sich hierbei primär um ein Resultat der räumlichen Kreuzvalidierung. In Ermangelung verlässlicher Trainingsbeispiele greift das Modell bei der Inferenz zwangsläufig auf diejenigen Klassen zurück, die der optischen Signatur von Wasser (starke Lichtabsorption, homogene dunkle Flächen) am nächsten kommen. Dies erklärt die Fehlklassifikationen als dunkles Dach (51,08 %) oder dunkle Vollversiegelung (24,71 %).

Neben den semantischen und geometrischen Ursachen dieser Fehlklassifikationen bleibt die Frage nach der numerischen Konfidenz dieser inkorrekten Vorhersagen. Ob das Netzwerk bei diesen Verwechslungen eine hohe Unsicherheit aufweist oder die Fehler mit starker Überzeugung produziert, quantifiziere ich im Rahmen der Modellkalibrierung im folgenden Abschnitt.

4.3.5 Quantifizierung der Modellunsicherheit und Kalibrierung

Für den verlässlichen Praxiseinsatz automatisierter Segmentierungsmodelle ist die Zuverlässigkeit der internen Modellkonfidenz essenziell. Ein perfekt kalibriertes System gibt Vorhersagewahrscheinlichkeiten aus, die exakt der empirischen Trefferquote entsprechen. Um dieses Verhalten für das entwickelte Architekturmodell zu evaluieren, aggregierte ich die Vorhersagen aller Pixel der ungesehenen Untersuchungsgebiete über sämtliche Iterationen der Kreuzvalidierung. Die resultierende Gegenüberstellung von prognostizierter Wahrscheinlichkeit und tatsächlicher Genauigkeit ist in Form eines globalen Zuverlässigkeitsdiagramms über zehn äquidistante Intervalle in Abbildung 4.7 dargestellt.

Die quantitative Auswertung liefert einen globalen erwarteten Kalibrierungsfehler (engl. *Expected Calibration Error*) von 7,11 % und zeigt ein zweigeteiltes Profil: Im niedrigen Konfidenzspektrum (10–40 %) zeigt sich eine leichte Selbstunterschätzung, bei der die tatsächliche Genauigkeit oberhalb der prognostizierten Wahrscheinlichkeit liegt. Ab etwa 60 % kippt dieses Verhalten in eine moderate Selbstüberschätzung, die zwischen 70 % und 90 % am stärksten ausgeprägt ist. Sagt das Netzwerk hier eine Klasse mit einer Sicherheit von rund 80 % voraus, liegt die reale Treffersicherheit leicht darunter. Im untersten Intervall (0–10 %) treten faktisch keine Werte auf. Da die Softmax-Funktion eine Wahrscheinlichkeitsverteilung über die 15 Klassen erzeugt, deren Werte sich zu 100 % summieren, kann die jeweils höchste Klassenwahrscheinlichkeit den Wert $1/15 \approx 6{,}7\,\%$ nicht unterschreiten. In

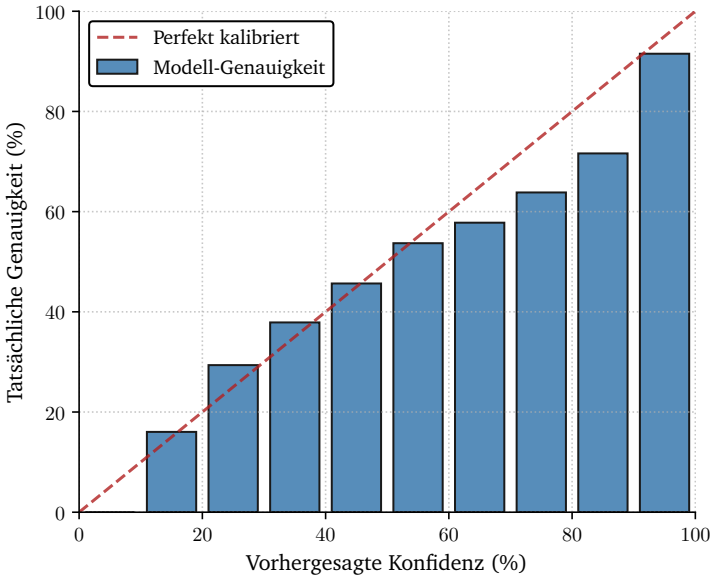


Abbildung 4.7 – Globales Zuverlässigkeitsdiagramm der aggregierten Kreuzvalidierung. Die gestrichelte Diagonale repräsentiert die perfekte Kalibrierung. Das Modell zeigt ein zweigeteiltes Verhalten: Eine leichte Unterkonfidenz im niedrigen Wahrscheinlichkeitsbereich (10–40 %) sowie eine systematische Überkonfidenz in den oberen Segmenten (60–100 %). Im Intervall 0–10 % treten faktisch keine Pixel auf, da die höchste Softmax-Wahrscheinlichkeit bei 15 Klassen mindestens $1/15 \approx 6{,}7\,\%$ beträgt und die Modellkonfidenz durchweg darüber liegt.

Kombination mit der durchweg hohen Vorhersagekonfidenz des Modells bleibt dieses Intervall daher leer.

Diese Charakteristik resultiert aus der Wechselwirkung zwischen der Transformer-Architektur und der Testzeit-Datenaugmentierung. Letztere erzeugt durch die Mittelwertbildung der Wahrscheinlichkeiten in komplexen urbanen Übergangszonen einen Glättungseffekt, der die finalen Wahrscheinlichkeiten senkt. Da dieser Konsens jedoch ähnlich fehlerresistent wie ein Ensemble-Ansatz wirkt, bleibt die gewählte Klasse überdurchschnittlich oft korrekt, was die Selbstunterschätzung im unteren Bereich erklärt [111]. In homogenen Flächen treibt der Konsens die Konfidenzwerte hingegen künstlich nahe an 100 %, sodass vereinzelte Fehlklassifikationen als deutliche Selbstüberschätzung sichtbar werden. Für ein unkalibriertes Gesamtsystem im Umfeld knapper Trainingsdaten stellt ein Fehler von 7,11 % dennoch ein sehr stabiles Ergebnis dar, welches deutlich unter den in der Literatur häufig berichteten Fehlkalibrierungen moderner tiefer neuronaler Netze liegt [112].

Selektives Inferenz-Thresholding und operationale Anwendung

Da das Modell im hohen Wahrscheinlichkeitssegment eine valide Verlässlichkeit aufweist, lässt sich die Segmentierungsgüte über eine gezielte Schwellenwertbildung steuern. Pixel unterhalb eines definierten Konfidenzwerts τ gelten dabei als unsicher und werden aus der automatisierten Auswertung ausgeschlossen, um sie optional einer manuellen Nachdigitalisierung zuzuführen. Der Kurvenverlauf in Abbildung 4.8 veranschaulicht diesen Zielkonflikt zwischen erreichbarer Segmentierungsgüte (mIoU) und der prozentualen Datenabweisungsrate für $\tau \in [0{,}00; 0{,}95]$.

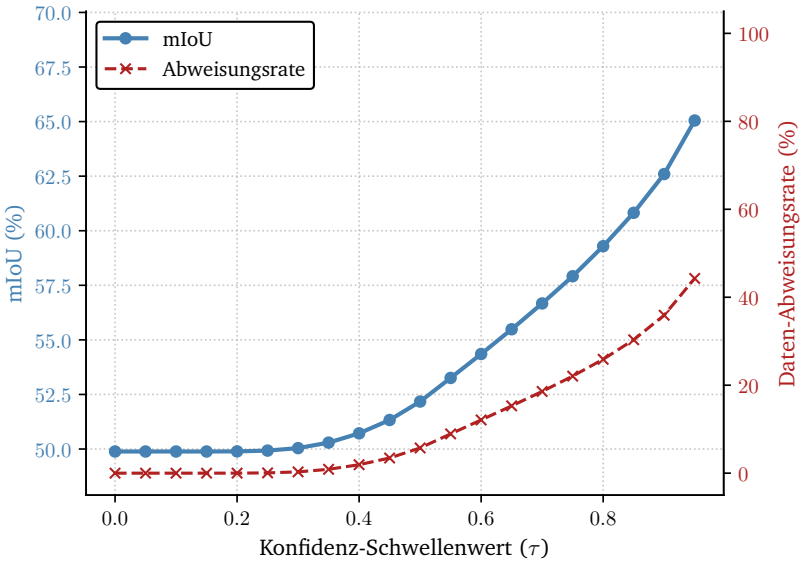


Abbildung 4.8 – Analyse des Zielkonflikts zwischen Vorhersagegüte und Datenabweisung. Dargestellt ist die globale Segmentierungsgüte (mIoU, linke Ordinate) und die prozentuale Datenabweisungsrate (rechte Ordinate) in Abhängigkeit des angesetzten Konfidenz-Schwellenwerts τ. Der Kurvenverlauf belegt das Potenzial einer kontrollierten Qualitätssteigerung auf Kosten einer steigenden Abweisungsrate.

Die Ergebnisse bestätigen, dass die systeminterne Unsicherheit als effektives Kontrollinstrument dienen kann. Ohne Filterung ($\tau = 0{,}00$) liegt die Basis-mIoU bei 49,89 %. Mit steigendem Schwellenwert zeigt die Vorhersagegüte einen streng monotonen Anstieg. Ein anwendungsorientierter Wert von $\tau = 0{,}70$ schließt 18,58 % der Gesamtpixel aus und steigert die Genauigkeit der verbleibenden Areale sprunghaft auf 56,67 % (+6,78 Prozentpunkte). Besonders bemerkenswert ist die Anhebung auf $\tau = 0{,}80$: Bei einer moderaten Abweisungsrate von 25,89 % steigt die mIoU auf 59,29 %. Allein durch das Aussortieren unsicherer Kanten- und Schattenbereiche gewinnt die verbleibende Vorhersage somit rund 9,4 Prozentpunkte gegenüber der ungefilterten Ausgangsgüte. Für stadtökologische Analysen ermöglicht dies im Vor-

feld eine fundierte Ressourcenplanung, bei der sich der manuelle Korrekturaufwand präzise gegen die angestrebte Automatisierungsgüte abwägen lässt.

Statistische Trennschärfe der Modellkonfidenz

Um die konfidenzbasierte Filterung statistisch zu fundieren, untersucht dieser Schritt die Trennschärfe des Netzwerks zwischen korrekten und fehlerhaften Klassifikationen. Die pixelweisen Unsicherheitswerte aus allen Validierungsdurchläufen wurden hierzu in zwei separate Gruppen unterteilt und als Boxplot in Abbildung 4.9 gegenübergestellt.

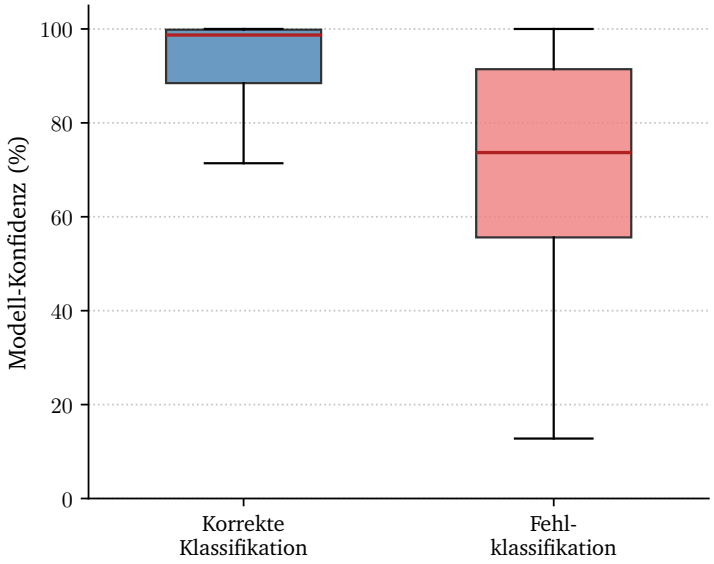


Abbildung 4.9 – Vergleich der pipeline-internen Konfidenzverteilungen für korrekte Vorhersagen (blau) und Fehlklassifikationen (rot) über alle aggregierten Kreuzvalidierungs-Durchläufe. Die roten Linien markieren die jeweiligen Mediane. Die vertikale Verschiebung der Boxen belegt die Trennschärfe des Systems, welche durch einen ROC-AUC-Score von 0,8017 für die Fehlerdetektion untermauert wird.

Die Auswertung zeigt eine Diskrepanz zwischen beiden Verteilungsräumen: Bei korrekten Vorhersagen konvergiert die Modellkonfidenz stark gegen das Maximum (Median: 98,71 %). Geringere Werte in dieser Gruppe spiegeln primär den Glättungseffekt an anspruchsvollen Mischpixeln wider, bei denen sich trotz reduzierter Wahrscheinlichkeit die korrekte Klasse durchsetzt. Bei Fehlklassifikationen bricht die Konfidenz im Median hingegen sprunghaft um 25,03 Prozentpunkte auf 73,68 % ein. Der weite Streubereich der Fehlerverteilung verdeutlicht dabei zwei Randphänomene: Sehr niedrige Wahrscheinlichkeiten repräsentieren harte Kantenentscheidungen, während Extremwerte nahe 100 % auf eine ausgeprägte Selbstüberschätzung bei

systematischen Fehlern (wie tiefen Gebäudeschatten) oder auf unpräzises Annotationsrauschen in den Referenzdaten hindeuten.

Die globale Fähigkeit des Systems, eigene Fehler verlässlich zu identifizieren, lässt sich über die *Receiver Operating Characteristic* (ROC) quantifizieren [113]. Die zugehörige Fläche unter der ROC-Kurve (*Area Under the Curve*, AUC) beträgt 0,8017 und belegt die Eignung der internen Unsicherheit als automatisierter Indikator für lokale Segmentierungsfehler in ungesehenen Zielgebieten.

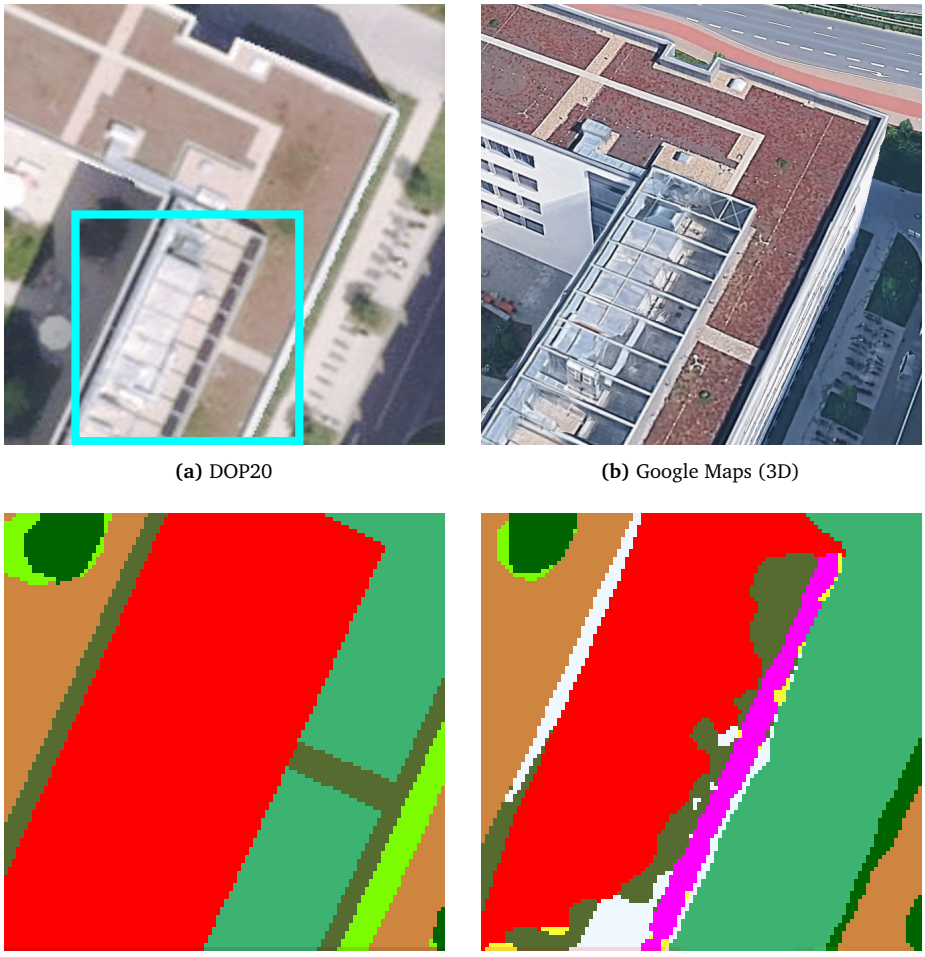
4.3.6 Qualitative Fehleranalyse und Randfälle

Zur Einordnung der quantitativen Evaluierungsmetriken erfolgt eine qualitative Betrachtung spezifischer Randfälle. Diese visuelle Fehleranalyse veranschaulicht, an welchen Stellen das Modell nicht an fehlender Repräsentationskraft scheitert, sondern systematisch an die physikalischen und optischen Grenzen der zweidimensionalen Nadirperspektive stößt.

Ein anschauliches Beispiel für eine solche Limitierung ist die Fehlklassifikation von PV-Anlagen durch Schattenwürfe (siehe Abbildung 4.10). Auf hochauflösenden DOP20-Aufnahmen weisen tiefe, scharfkantige Schatten oder dunkle Dachaufbauten innerhalb komplexer Dachstrukturen eine nahezu identische optische Signatur auf wie moderne Solarmodule. Beide zeichnen sich im RGB-Spektrum durch extrem niedrige Albedo-Werte, homogene dunkle Texturen und harte, oft rechtwinklige Kanten aus. Wie die nachträgliche Gegenüberstellung mit der 3D-Ansicht belegt, handelt es sich bei den in Abbildung 4.10 dargestellten Bereichen in der Realität um einen technischen Dachaufbau. Dieser visuelle Randfall demonstriert anschaulich, dass derartige geometrische Fehlklassifikationen eine methodische Grenze der rein optischen Segmentierung darstellen.

Ein weiterer systematischer Randfall, der die operationale Ableitung stadtökologischer Indikatoren erschwert, resultiert aus der physikalisch begrenzten räumlichen Auflösung der Luftbilder. Abbildung 4.11 zeigt, dass die optische Unterscheidbarkeit zwischen harter Vollversiegelung (wie durchgehendem Asphalt) und starker Teilversiegelung (wie dichtem Pflaster) in den DOP20-Daten oftmals kaum bis gar nicht gegeben ist.

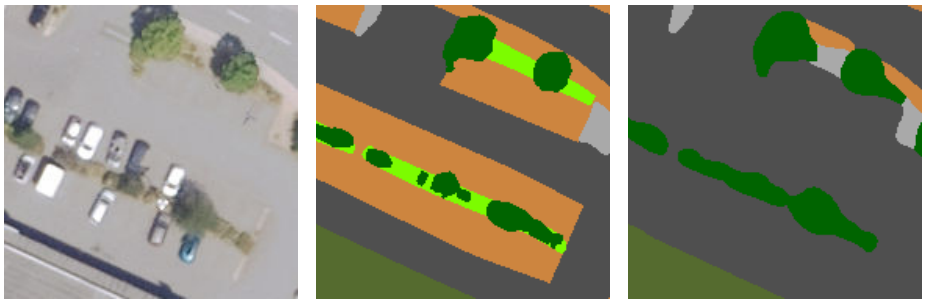
Die Ursache hierfür liegt in der Frequenz der Bildinformation: Bei einer Bodenauflösung von 20 cm pro Pixel werden feine strukturelle Merkmale, wie die Fugen eines Pflasterbelags im Bildraum schlichtweg nicht mehr aufgelöst. Da Pflastersteine und durchgehende Betondecken zudem häufig identische spektrale Eigenschaften (Grauwerte) im sichtbaren RGB-Bereich aufweisen, entfällt auch das farbliche Unterscheidungsmerkmal. Das Modell verliert an diesen Flächen die Möglichkeit, die Versiegelungsart anhand lokaler Texturen zu klassifizieren. Es ist somit gezwungen, die Zuweisung auf Basis des makroskopischen räumlichen Kontexts abzuleiten.



(a) DOP20 **(b)** Google Maps (3D)

(c) Referenzmaske **(d)** Inferenzmaske

Abbildung 4.10 – Detailanalyse eines visuellen Randfalls bei der Detektion von PV-Anlagen. Die obere Reihe zeigt den baulichen Kontext im Luftbild (a) und in der 3D-Perspektive (b). Der cyanfarbene Rahmen in (a) definiert den Ausschnitt für die vergrößerten Detailansichten der Referenzmaske (c) und der Vorhersage (d). Im DOP20 weisen tiefe Schatten innerhalb eines Dachaufbaus starke farbliche und geometrische Ähnlichkeiten zu Solarmodulen auf. Während die 3D-Verifikation (b) diesen Bereich zweifelsfrei als Teil der technischen Dachaufbauten (■) identifiziert, klassifiziert das Modell diese Pixel fälschlicherweise als PV-Anlage (■). Hinweis: Weitere im Detailausschnitt sichtbare Abweichungen zwischen Referenz und Vorhersage werden zugunsten der Fokussierung auf diesen spezifischen Randfall nicht weiter thematisiert.



(a) DOP20 **(b)** Referenzmaske **(c)** Inferenzmaske

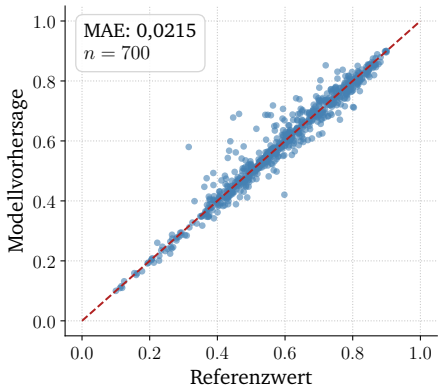
Abbildung 4.11 – Detailanalyse eines Auflösungsartefakts bei Bodenversiegelungen auf einem Parkplatz. Die Referenzmaske (b) wurde manuell unter Einbeziehung von Bodenperspektiven erstellt. Während das 20 cm-Orthophoto eine homogene vollversiegelte Fläche suggeriert, liegt real eine Materialtrennung vor: Die Fahrwege sind asphaltiert (Vollversiegelung ■), die Parkbuchten hingegen gepflastert (starke Teilversiegelung ■). Da das Modell die schmalen Fugen aufgrund der begrenzten Bodenauflösung und minimaler farblicher Kontraste nicht erfassen kann, klassifiziert es die Parkbuchten fälschlicherweise ebenfalls als Vollversiegelung.

Dies erklärt die bereits in der Konfusionsmatrix beobachtete hohe Verwechslungsrate dieser Klassen und belegt, dass selbst ein perfekt trainiertes Modell auf Nadiraufnahmen dieser Auflösungsstufe einer inhärenten Unschärfe bei der exakten Bestimmung des Versiegelungsgrades unterliegt.

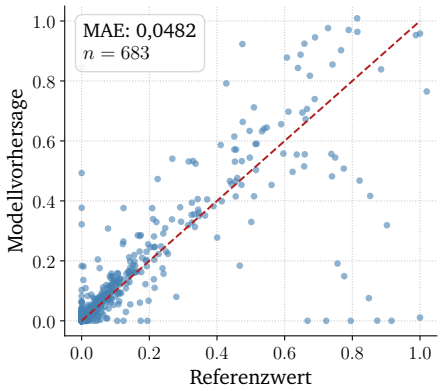
4.4 Evaluierung der abgeleiteten Flächen- und Solarindikatoren

Um zu beurteilen, inwiefern sich die auf Pixelebene identifizierten Segmentierungsfehler auf die operationale Nutzbarkeit für die Stadtplanung und Energieanalyse auswirken, wertet dieser Abschnitt die aggregierten Kennzahlen aus. Wie in Abschnitt 3.5 definiert, unterteilt die Evaluierungs-Pipeline hierzu die sieben Untersuchungsgebiete in insgesamt 700 Rasterzellen der Größe $51,2 \times 51,2$ m. Für jede Zelle stellt das System die abgeleiteten Indikatoren der Modellvorhersage den manuell erfassten Referenzwerten gegenüber.

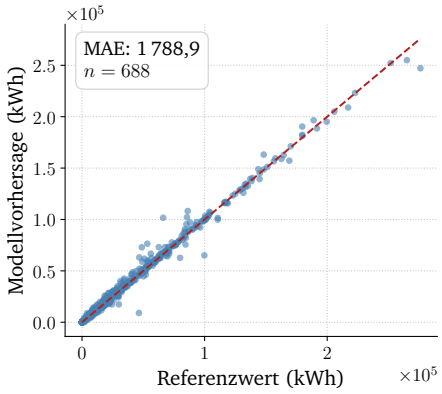
Abbildung 4.12 visualisiert diese Gegenüberstellung für alle vier untersuchten stadtökologischen und energetischen Parameter. Sie belegt einen starken, aber je nach Metrik unterschiedlich ausgeprägten Aggregationseffekt: Fehler an Objektgrenzen oder unscharfe Konturen, die die pixelbasierte mIoU stark abstrafen, mitteln sich auf der Makroebene einer urbanen Parzelle teils heraus, während spezifische geometrische Fehlklassifikationen punktuell starke Auswirkungen auf Quotienten und Absolutwerte haben.



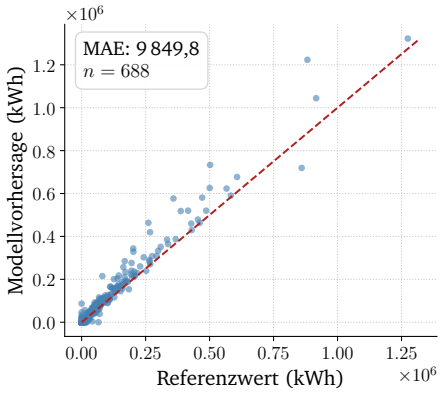
(a) Gebietsmittlerer Abflussbeiwert



(b) Anteil nachhaltiger Dachflächen



(c) Erschließbares Solarpotenzial



(d) Solare Einstrahlung der Bestandsanlagen

Abbildung 4.12 – Korrelation der abgeleiteten Indikatoren pro Rasterzelle ($51,2 \times 51,2$ m). Die gestrichelte Diagonale repräsentiert die fehlerfreie Vorhersage. Während flächige Indikatoren wie der Abflussbeiwert (a) eine hohe Robustheit durch den Aggregationseffekt aufweisen, zeigen stark nicht-lineare Metriken wie der Dachindikator (b) und binär geprägte Zielgrößen wie der PV-Bestand (d) eine naturgemäß höhere Varianz.

Gebietsmittlerer Abflussbeiwert

Für den gebietsmittleren Abflussbeiwert ($\overline{C_m}$) ergibt sich über alle 700 Rasterzellen ein MAE von lediglich 0,0215. Wie das Streudiagramm (Abbildung 4.12a) verdeutlicht, gruppiert sich die überwiegende Mehrheit der Vorhersagen als dichte Punktwolke um die ideale Diagonale. Dies bestätigt, dass das Modell das globale Verhältnis aus versiegelten und unversiegelten Flächen nahezu exakt reproduziert.

Dieser geringe Fehler resultiert dabei nicht allein aus der räumlichen Mittelung. Die dominanten Verwechslungen verlaufen überwiegend innerhalb grober Versiegelungsgruppen: Die vollversiegelten Klassen teilen sich nahezu durchgängig einen Faktor von $C_m = 0{,}9$, während Vegetations- und Substratklassen am unteren Ende (0,1–0,2) clustern (vgl. Tabelle 3.7). Verwechslungen innerhalb dieser Gruppen verschieben den gewichteten Gebietsmittelwert daher nur marginal. Die wenigen verbleibenden Ausreißer entstehen folglich gerade dort, wo das Modell gruppenübergreifend wechselt, etwa bei der Verwechslung starker Teilversiegelung ($C_m = 0{,}7$) mit dunkler Vollversiegelung (0,9), und konzentrieren sich auf stark verschattete Bereiche der dichten Blockrandbebauung, in denen versiegelte Hinterhöfe fehlerhaften Kategorien zugeordnet werden. Insgesamt ist ein Fehler von rund 2 % für stadtplanerische Zwecke, wie etwa die Starkregenvorsorge, methodisch hochgradig belastbar.

Anteil nachhaltiger Dachflächen

Der Anteil nachhaltig genutzter Dachflächen weist bei $n = 683$ gültigen Rasterzellen einen MAE von 0,0482 auf (17 unbebaute Zellen wurden systemseitig ignoriert). Obwohl ein Fehler von knapp 4,8 % quantitativ ein sehr gutes Ergebnis darstellt, zeigt das Streudiagramm (Abbildung 4.12b) als einziges eine visuell deutlich höhere Streuung. Dieses Verhalten ist mathematisch in der Quotientenbildung begründet: Befindet sich in einer Zelle nur ein kleines Gebäude, führt bereits eine geringfügige Übersegmentierung von PV, also die Fehlklassifikation von PV auf Dachaufbauten (vgl. Abschnitt 4.3.4), zu einem starken Ausschlag der prozentualen Metrik für diese spezifische Zelle.

Simuliertes Solarpotenzial auf ungenutzten Dächern

Für das auf aktuell ungenutzten Dachflächen simulierte, erschließbare Solarpotenzial ($P_{\text{potential}}$) erzielt das Modell einen sehr soliden MAE von 1 788,9 kWh pro Rasterzelle. Setzt man diesen absoluten Fehler in Relation zum durchschnittlichen Referenz-Solarpotenzial der untersuchten Rasterzellen, ergibt sich ein normalisierter Fehler (nMAE) von lediglich 5,12 %. Das entsprechende Streudiagramm (Abbildung 4.12c) zeigt eine unerwartet starke lineare Korrelation mit minimaler Streuung. Eine leich-

te, punktuelle Unterschätzung des Potenzials unterhalb der Ideallinie resultiert aus der modellspezifischen Erkennung bestehender Anlagen: Neigt das Modell dazu, Dachaufbauten fälschlicherweise als PV zu klassifizieren, wertet der Platzierungsalgorithmus diese Fläche als „bereits belegt“ und zieht sie vom Potenzialdach ab, was in den betroffenen Zellen zu einer systematischen, aber geringfügigen Unterschätzung führt.

Solare Einstrahlung auf Bestandsanlagen

Für die bereits bestehenden PV-Anlagen (P_{existent}) liegt der MAE bei 9 849,8 kWh pro Rasterzelle (Abbildung 4.12d). Dieser absolute Wert fällt zwar numerisch rund fünfeinhalbmal höher aus als beim erschließbaren Potenzial, ist mit diesem jedoch nicht unmittelbar vergleichbar: Während das erschließbare Potenzial den finalen *elektrischen Stromertrag* (unter Annahme eines Systemnutzungsgrades) abbildet, quantifiziert der Bestandsindikator aufgrund fehlender externer Leistungsparameter lediglich die reine *solare Einstrahlungsenergie* auf die Modulfläche (vgl. Abschnitt 3.5). Beide Größen sind somit in unterschiedlichen physikalischen Einheiten und über unterschiedliche Bezugsflächen definiert, weshalb erst eine metrikinterne Relativierung eine belastbare Einordnung erlaubt.

Bezogen auf das durchschnittliche Referenz-Einstrahlungsvolumen der Rasterzellen von 44 311,1 kWh ergibt sich ein nMAE von 22,23 %. Dieser liegt erwartungsgemäß über dem Wert des Potenzialindikators (5,12 %), was jedoch nicht auf eine geringere Segmentierungsleistung, sondern auf die quasi-binäre Charakteristik der Zielgröße zurückzuführen ist: Bestandsanlagen treten als diskrete, räumlich konzentrierte Objekte auf, sodass in Zellen mit nur geringem Anlagenbestand bereits die Fehlklassifikation einer einzelnen Anlage einen starken prozentualen Ausschlag verursacht. Da der nMAE hier als normierter Fehler (MAE bezogen auf den Mittelwert der Referenz) und nicht als zellweise gemittelte prozentuale Abweichung berechnet wird, ist er gegenüber kleinen Nennern robust und bildet die aggregierte Vorhersagegüte unverzerrt ab.

Qualitative Analyse der automatisierten PV-Platzierung

Um die quantitativen Ergebnisse der Potenzialanalyse in einen räumlichen Bezug zu setzen und die Funktionsweise des nachgelagerten Platzierungsalgorithmus zu validieren, werden im Folgenden drei exemplarische Ausschnitte aus der Ende-zu-Ende-Validierung im Detail betrachtet. Diese Gegenüberstellung eines Positiv- und zweier Negativbeispiele (siehe Abbildung 4.13) veranschaulicht, wie die auf Pixelebene generierten Segmentierungsmasken in die geometrische Anordnung der Solarmodule überführt werden und an welchen Stellen die algorithmischen Grenzen dieses Verfahrens liegen.

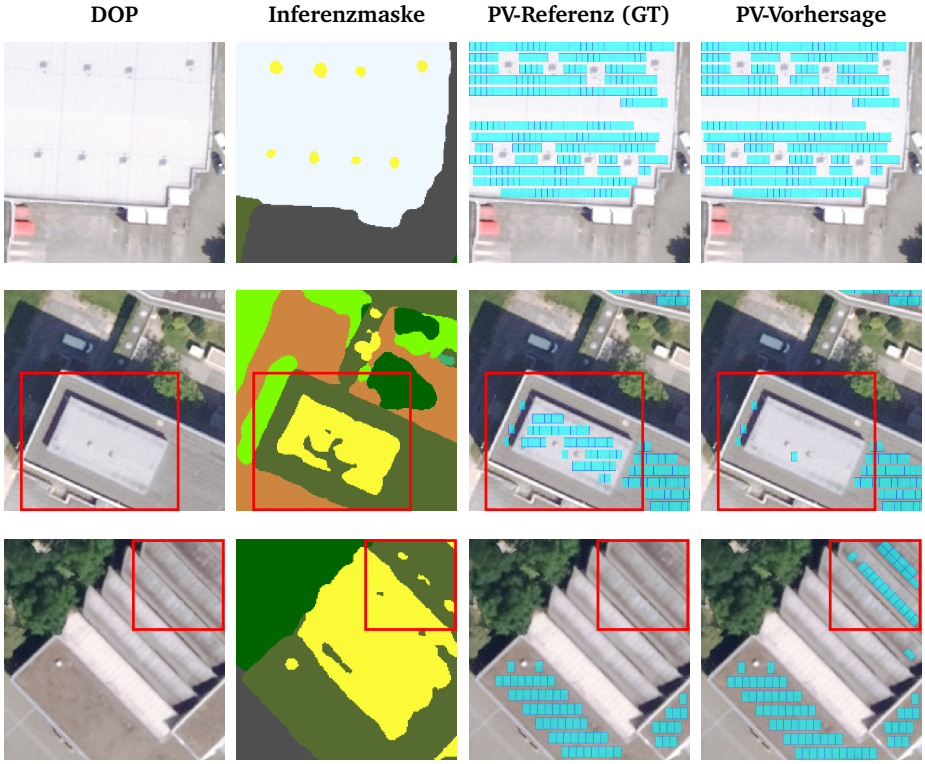


Abbildung 4.13 – Analyse der Fehlerfortpflanzung von der semantischen Segmentierungsebene auf die PV-Platzierung. **Obere Reihe (Positiv-Beispiel):** Das Modell segmentiert die Dachfläche und Störstrukturen fehlerfrei, wodurch die simulierte Modulbelegung exakt der Referenz entspricht. **Mittlere Reihe (Negativ-Beispiel – Unterschätzung):** Ein Teil der freien Dachfläche wird fälschlicherweise als Störstruktur (Dachaufbau ■) klassifiziert. Der Platzierungsalgorithmus spart diese Flächen konsequent aus, was zu einer Unterschätzung des realen Solarpotenzials führt. **Untere Reihe (Negativ-Beispiel – Überschätzung):** Eine real existierende Störstruktur wird vom Modell übersehen. Infolgedessen simuliert der Algorithmus Solarmodule an physisch unmöglichen Positionen, was das berechnete Ertragspotenzial künstlich überschätzt. Die simulierten PV-Module sind jeweils als türkisfarbene Rechtecke dargestellt; rote Rahmen markieren in den Negativ-Beispielen den jeweils fehlerhaften Bereich.

Das erste Beispiel (obere Reihe) demonstriert eine erfolgreiche Potenzialermittlung auf einer klaren, homogenen Dachstruktur. Das System identifiziert die nutzbare, unbeschattete Dachfläche präzise und maximiert die Modulbelegung unter Berücksichtigung der optimalen Ausrichtung. Bereits im Vorfeld vom Segmentierungsmodell korrekt detektierte Störstrukturen, wie Dachfenster oder andere Dachaufbauten, erkennt der Platzierungsalgorithmus und umgeht sie durch entsprechende Aussparungsradien fehlerfrei. In solchen Fällen spiegelt das simulierte Solarpotenzial die reale Erschließbarkeit genau wider.

Demgegenüber zeigen die beiden Negativbeispiele die Fehlerfortpflanzung bei fehlerhaft segmentierten Ausgangsdaten in zwei entgegengesetzte Richtungen. Führt eine initiale Fehlklassifikation durch das Modell zu einer Halluzination von Störstrukturen, beispielsweise durch die Fehlinterpretation einer versetzten, höhergelegenen Dachebene als störender Dachaufbau, fragmentiert der Platzierungsalgorithmus eigentlich nutzbare Dachflächen (mittlere Reihe). Dies resultiert in einer künstlichen Unterschätzung des Ertragspotenzials. Im umgekehrten Fall (untere Reihe) führt das unerkannte Verschmelzen real existierender Dachaufbauten mit der restlichen Dachfläche in der Inferenzmaske dazu, dass Solarmodule an physikalisch unmöglichen Positionen simuliert werden, was folglich eine Überschätzung des Potenzials nach sich zieht.

Diese visuellen Stichproben untermauern die zuvor im Streudiagramm beobachteten Ausreißer: Während die automatisierte Potenzialabschätzung auf großflächigen, klaren Dachgeometrien präzise und belastbare Werte liefert, pflanzen sich Segmentierungsunschärfen bei kleinteilig gegliederten Dachlandschaften direkt in die simulierte Modulbelegung fort und führen zu lokalen Abweichungen des finalen Ertragsindikators.

4.5 Kritische Diskussion und methodische Limitationen

Um die zuvor quantifizierten Leistungsmetriken, von der isolierten Pixelgenauigkeit bis hin zu den aggregierten Nachhaltigkeitsindikatoren, abschließend umfassend einzuordnen, diskutiert der folgende Abschnitt die methodischen Limitationen.

Bei der Interpretation der Evaluierungsmetriken auf dem eigenen Datensatz ist das Fehlen eines strikt isolierten, finalen Testdatensatzes als methodischer Kompromiss des Low-Data-Regimes zu berücksichtigen. Da die Hyperparameter-Optimierung und die Architekturauswahl bereits vollständig a priori auf dem unabhängigen FLAIR-Datensatz erfolgten, schließe ich damit eine Verzerrung der Modellarchitektur durch eine manuelle Feinabstimmung auf den eigenen Daten aus.

Eine leicht optimistische Verzerrung entsteht jedoch durch die Nutzung des jeweiligen Validierungs-Teildatensatzes für die finale Modellauswahl während der Kreuzvalidierung. Da das System für eine feste Anzahl von Iterationen trainiert und nachträglich denjenigen Modellzwischenstand auswählt, der die höchste mittlere Validierungs-mIoU auf genau diesen Validierungsdaten aufweist, sind die *Out-of-Fold*-Vorhersagen nicht vollkommen unabhängig vom Trainingsprozess. Die ermittelten Fehlerwerte der stadtökologischen Indikatoren und des Solarpotenzials stellen somit ein leicht optimistisches Szenario dar. Für die großflächige Anwendung des Modells auf gänzlich neue bayerische Stadtstrukturen ist daher potenziell mit einem geringfügigen Abfall der Segmentierungsgüte zu rechnen.

Ein weiterer essenzieller Aspekt betrifft die natürlichen physikalischen und perspektivischen Limitationen der senkrechten Vogelperspektive. Obwohl das Modell neben den optischen 20 cm-Luftbildern (DOP20) durch die vorgeschaltete Kanalfusion auch auf topografische Informationen (nDOM) zurückgreift, unterliegt es unweigerlich dem Problem der Verdeckung: Dichte Baumkronen verdecken Teile des Straßenraums, während überhängende Dächer die darunterliegende Bodenversiegelung visuell verbergen. Da ein Oberflächenmodell prinzipbedingt keine Informationen über die verdeckten Bereiche unterhalb solcher Strukturen liefert, lässt sich diese Unschärfe der 2D-Projektion auch durch die multimodale Datenfusion im Bildraum nicht vollständig auflösen. Unabhängig von dieser geometrischen Limitierung zeigen vereinzelte Fehler zudem, dass die 1×1-Projektion auf drei Kanäle die physikalische Eindeutigkeit der Höhendaten nicht in allen Fällen perfekt konserviert. Folglich können abgeleitete stadtökologische Kennzahlen, wie der Abflussbeiwert oder das exakte Solarpotenzial ungenutzter Flächen, selbst durch ein ideal trainiertes Modell immer nur mit einer gewissen Unschärfe berechnet werden. Die tatsächliche dreidimensionale Komplexität und nutzbare Fläche der urbanen Morphologie wird durch dieses Verfahren zwangsläufig leicht unterschätzt.

Neben diesen räumlichen und perspektivischen Einschränkungen stellt auch die zeitliche Dimension eine nicht zu vernachlässigende Limitierung für die Übertragbarkeit der Ergebnisse dar. Jede optische Fernerkundungsaufnahme ist systembedingt stets nur eine Momentaufnahme. Die Modellvorhersagen sind stark an die spezifischen Aufnahmebedingungen (Belaubungszustand der Vegetation, Sonnenstand und resultierende Schattenwürfe) zum Zeitpunkt der Befliegung gekoppelt. Für einen flächendeckenden, ganzjährigen operationalen Einsatz müsste die zeitliche und atmosphärische Varianz durch ein noch breiteres Spektrum an Zieldaten weiter abgefedert werden.

Kapitel 5

Zusammenfassung und Ausblick

Das Ziel dieser Arbeit war ein KI-gestütztes Framework zur Ableitung stadtökologischer Indikatoren und Solarpotenziale aus 20 cm-Luftbildern (DOP20) bei stark limitierten Trainingsdaten. Dafür erstellte ich einen feingranularen 15-Klassen-Datensatz für Mittelfranken.

Um die optimale Segmentierungsarchitektur für dieses komplexe Zielschema zu ermitteln, führte ich zunächst eine ressourceneffiziente Architektursuche auf dem Stellvertreterdatensatz FLAIR durch. Die Ergebnisse zeigten, dass das auf Alltagsbildern selbstüberwacht vortrainierte *DINOv3-Large (LVD)* nicht nur überwachten CNN- und Transformer-Basislinien deutlich überlegen ist, sondern auch domänenspezifisch vortrainierte Varianten, etwa ein auf Satellitenbildern trainiertes Modell sowie einen eigens auf bayerischen DOP20-Daten trainierten Masked Autoencoder, übertrifft. In Kombination mit dem parametereffizienten *SegFormer*-Decoder erreichte das Modell im Low-Data-Regime die höchste Segmentierungsgüte. Die Analyse der Dateneffizienz belegte zudem, dass diese Architekturkombination bereits mit Bruchteilen eines regulären Datensatzes über 80 % seiner potenziellen Sättigungsleistung abrufen kann, was den Einsatz im datenarmen urbanen Raum methodisch legitimiert.

Beim Transfer auf den eigens erstellten bayerischen Anwendungsdatensatz erreichte das Modell in der räumlichen Kreuzvalidierung eine PA von 80,10 % bei einer aggregierten mIoU von 49,89 %. Während dominante Klassen (*Dark Roof*: 77,93 % IoU) sowie geometrisch prägnante, unterrepräsentierte Klassen wie *Photovoltaic System* (75,10 % IoU, bei einer Sensitivität von 95,12 %) sehr zuverlässig erfasst wurden, zeigten sich erwartbare Schwächen bei visuell schwer trennbaren Bodenbedeckungsklassen (z. B. *Strong Partial Sealing* vs. *Dark Hard Sealing*).

Ergänzend belegte die Kalibrierungsanalyse, dass sich die netzinterne Vorhersagekonfidenz als verlässlicher Indikator für lokale Fehlklassifikationen eignet (ROC-AUC: 0,80). Über ein gezieltes Konfidenz-Thresholding, das die unsichersten rund 26 %

der Pixel von der automatisierten Auswertung ausschließt, ließ sich die mIoU der verbleibenden Flächen von 49,89 % auf 59,29 % steigern.

Die nachgelagerte Anwendungsstufe belegte, dass sich pixelbasierte Segmentierungsfehler durch räumliche Aggregation auf Parzellenebene stark relativieren. Im Vergleich zu den aus den manuellen Referenzmasken berechneten Werten ließen sich der gebietsmittlere Abflussbeiwert (MAE: 0,0215) und der Anteil nachhaltiger Dachflächen (MAE: 0,0482) mit hoher Präzision ableiten. Für die energetischen Zielgrößen ergab sich beim ungenutzten Solarpotenzial ein mittlerer Fehler von 1 788,9 kW h pro Rasterzelle (nMAE: 5,12 %) und bei der solaren Einstrahlung der Bestandsanlagen ein Fehler von 9 849,8 kW h (nMAE: 22,23 %). Der höhere Relativfehler des Bestands ist strukturell durch die quasi-binäre Natur dieser Zielgröße bedingt. Es ist jedoch zu betonen, dass diese Metriken primär die methodische Konsistenz der Pipeline sowie eine geringe Fehlerfortpflanzung belegen; eine Validierung der absoluten physikalischen Genauigkeit gegen reale Messdaten stellt dies noch nicht dar.

Aus diesen Befunden ergeben sich drei zentrale Schlussfolgerungen für die angewandte Geoinformatik:

Erstens reduzieren leistungsstarke Basismodelle (wie DINOv3) durch ihre unüberwacht gelernten Repräsentationen die Abhängigkeit von massiven Annotationskampagnen drastisch. Dies ermöglicht es einer Vielzahl von Akteuren, wie etwa Kommunen, Quartiersplanern oder Liegenschaftsverwaltern, domänenspezifische Modelle ressourceneffizient auf lokale Gegebenheiten anzupassen.

Zweitens erweist sich bei extremer Datenknappheit eine Limitierung der architektonischen Komplexität als entscheidender Vorteil. Die Kopplung eines kapazitätsstarken Merkmalsextraktors mit einem leichtgewichtigen Decoder fungiert als effektive Regularisierung gegen Überanpassung und ist aufwendigeren Modellen im Kaltstart-Szenario überlegen.

Drittens darf die operationale Nutzbarkeit von KI-Vorhersagen nicht allein auf der Pixelebene beurteilt werden. Die rein pixelbasierte mIoU unterschätzt den praktischen Mehrwert eines Modells für flächen- und verhältnisbasierte Indikatoren, da sich lokale Segmentierungsunschärfen bei der raumbezogenen Aggregation auf Parzellen- oder Rasterzellenebene weitgehend kompensieren. Stärker punktuell oder quasi-binär geprägte Zielgrößen, wie der Bestand einzelner PV-Anlagen, bleiben hingegen deutlich empfindlicher gegenüber einzelnen Fehlklassifikationen. Maßgeblich für die Praxistauglichkeit ist somit nicht die isolierte Pixelmetrik, sondern die Genauigkeit der jeweils abgeleiteten Zielindikatoren; vergleichbare Pipelines in der angewandten Geoinformatik sollten ihren Erfolg daher konsequent auf der Aggregationsebene der konkreten Anwendung messen.

Aus den identifizierten Fehlerclustern und Limitierungen der rein optischen Fernerkundung ergeben sich drei primäre Anknüpfungspunkte für zukünftige Arbeiten.

Ein zentraler Ansatz ist die multimodale Datenfusion: Um visuelle Ambiguitäten besser aufzulösen, sollten zusätzliche Geodaten direkt in den Lernprozess integriert werden. Vielversprechend ist hierbei die Anreicherung der Bilddaten mit 3D-Oberflächenmerkmalen (LoD2) und semantischen Katasterinformationen. Es bleibt zu evaluieren, ob der resultierende Genauigkeitsgewinn den erhöhten Aufwand der Datenvorbereitung rechtfertigt.

Da sich komplexe Vorhersageartefakte durch statische Regeln oft nur bedingt korrigieren lassen, empfehle ich darüber hinaus die Erforschung generativer und adaptiver ML-Paradigmen. Ein nachgelagertes Diffusionsmodell könnte mittels *Testzeit-Diffusion* als datengetriebenes Korrektiv fungieren, um topologische Inkonsistenzen aufzulösen. Parallel dazu bieten *Meta-Lern-Verfahren* das Potenzial, Modelle hochdynamisch und mit nur wenigen lokalen Beispielen an völlig neue städtebauliche Typologien anzupassen.

Abschließend empfehle ich eine Machbarkeitsstudie auf Basis höheraufgelöster Bilddaten (z. B. 5 cm GSD), um zu prüfen, ob eine gesteigerte Pixeldichte die schwer trennbaren, feingranularen Klassen besser auflöst. Kritisch abzuwägen ist dabei jedoch das inhärente Skalierungsproblem: Der Gewinn an Detailtiefe darf nicht zu einer verstärkten Überanpassung an irrelevante Mikrotexturen führen oder den Verlust des globalen räumlichen Kontexts pro Bild-Token bedingen.

Insgesamt zeigt diese Arbeit, dass sich stadtökologische Kennzahlen und Solarpotenziale auch unter knappen Annotationsressourcen automatisiert und in sich konsistent aus 20 cm-Luftbildern (DOP20) in Kombination mit ergänzenden Geobasisdaten annähern lassen. Die erreichte Segmentierungsgüte und die systematischen Verwechslungen visuell ähnlicher Klassen verdeutlichen jedoch zugleich, dass das entwickelte Framework die fachliche Erhebung nicht ersetzt, sondern als ressourceneffizientes Werkzeug zur Vorqualifizierung dient, das den verbleibenden manuellen Aufwand gezielt auf die unsicheren Bereiche der Vorhersage lenken kann.

Anhang A

Datenbasis und Vorverarbeitung

A.1 Geographische Verteilung der SSL-Trainingsdaten

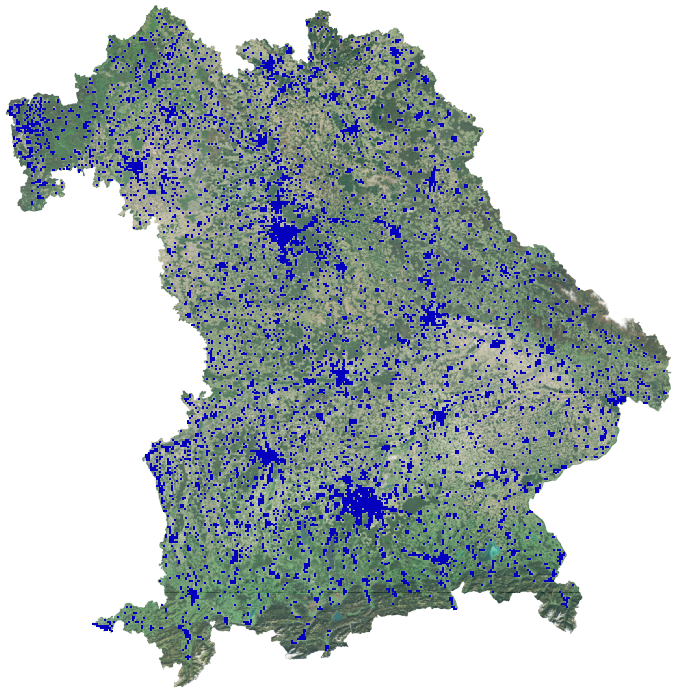


Abbildung A.1 – Geographische Verteilung der ausgewählten DOP20-Kacheln in Bayern. Die dunkelblauen Rechtecke markieren die für das SSL-Training genutzten Kacheln. Der Datensatz wurde auf urbane Räume gefiltert, definiert durch eine Mindestanzahl an Gebäuden pro Kachel auf Basis von Hausumringen.

A.2 Detaillierte Ergebnisse der Annotationsqualität

Tabelle A.1 – Detaillierte IoU-Werte nach Gebieten und Objektklassen im Vergleich zur qualitätsgesicherten Referenz. Leere Zellen bedeuten, dass die Klasse im jeweiligen Gebiet nicht vorkam. Werte von 0,00 weisen auf eine vollständige Fehlklassifikation oder das Übersehen der Klasse hin.

Gebiet	mIoU	TVeg	LVeg	UnvS	Grav	Watr	LPS	SPS	GRof	DRof	LRof	PV	TRS	ORS	LHS	DHS
N1	0,53	0,92	0,65	0,75	0,00	0,00	0,01	0,59	0,30	0,92	0,63	0,89	0,83	0,72	0,00	0,73
N2	0,82	0,98	0,97	0,36	0,97	–	1,00	0,97	0,92	0,97	0,98	0,97	0,63	0,57	0,26	0,93
N3	0,72	0,89	0,89	0,99	0,86	0,01	0,98	0,98	0,56	0,97	0,43	0,88	1,00	0,32	0,11	1,00
N4	0,87	0,98	0,97	0,74	0,78	0,99	0,98	0,95	0,73	0,87	0,87	0,99	1,00	0,38	0,98	0,87
E1	0,72	0,93	0,94	0,61	0,55	–	0,16	0,89	0,41	0,89	0,68	0,98	0,78	0,78	0,58	0,93
E2	1,00	1,00	1,00	1,00	1,00	–	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
E3	0,71	0,98	0,98	0,73	0,05	0,00	1,00	0,98	0,93	0,99	0,65	0,97	0,95	0,45	0,08	0,93

Abkürzungen der Klassen: TVeg: Tall Vegetation, **LVeg:** Low Vegetation, **UnvS:** Unvegetated Substrate, **Grav:** Crushed Stone / Gravel, **Watr:** Water Bodies, **LPS:** Low Partial Sealing, **SPS:** Strong Partial Sealing, **GRof:** Green Roof, **DRof:** Dark Roof, **LRof:** Light Roof, **PV:** Photovoltaic System, **TRS:** Technical Roof Structures, **ORS:** Other Roof Structures, **LHS:** Light Hard Sealing, **DHS:** Dark Hard Sealing.

Anhang B

Konfigurationen der Algorithmen

B.1 Hyperparameter des Vorexperiments

Tabelle B.1 – Wesentliche Hyperparameter und Konfiguration des FT-UNetFormer- Referenzmodells.

Parameter	Einstellung
Architektur	FT-UNetFormer (Swin-Base Backbone) [5]
Input-Größe	512 × 512 Pixel
Epochen	45
Batch Size	8
Optimierer	Lookahead (AdamW)
Verlustfunktion	Joint Loss (CE + Dice)

B.2 Verlauf der Hyperparameter-Optimierung

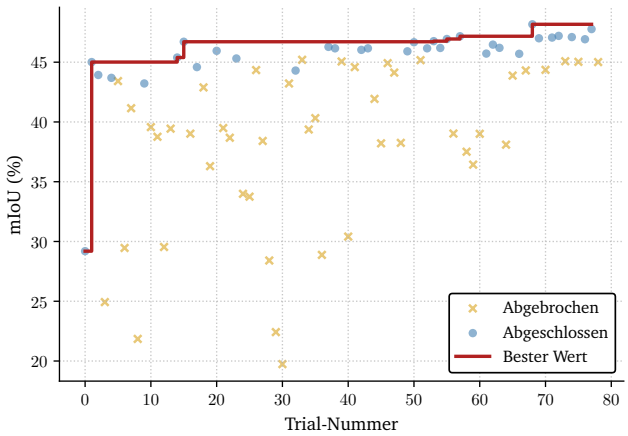


Abbildung B.1 – Verlauf der 80 Trainingsläufe während der asynchronen Hyperparameter-Optimierung mit *Optuna*.

B.3 Parameter der Solaranalyse

Tabelle B.2 – Übersicht aller empirisch ermittelten und statischen Parameter der Solarpotenzial-Pipeline, unterteilt nach den jeweiligen Verarbeitungsschritten.

Kategorie / Parameter	Wert	Einheit	Beschreibung
Geometrische Modellierung & Dachflächen			
Bodenauflösung (GSD)	0,2	m	Basisauflösung des Rastermodells (DOM)
RANSAC Distanztoleranz	0,1	m	Maximale Distanz für Inlier bei Ebenenanpassung
Max. Winkelabweichung ($\Delta\theta$)	5	°	Toleranzgrenze für RANSAC-Ebenenkorrektur
Max. Höhenversatz (Δz)	0,30	m	Toleranzgrenze am Mittelpunkt für RANSAC
Alpha (α)	1,0	-	Parameter zur Berechnung der Alpha-Shapes
Dachüberhang-Toleranz	0,4	m	Puffer für KDTree-basierte Nachbarschaftssuche
Outlier-Distanztoleranz	0,25	m	Toleranz zur Zuweisung von Punkten zur Dachebene
Hinderniserkennung (DBSCAN)			
Max. Suchradius (ϵ)	0,4	m	Radius für die Clusteranalyse
Min. Punktzahl (*MinPts*)	4	-	Mindestanzahl an Punkten pro Cluster
Schattenanalyse & Raytracing			
Azimutrichtungen (M)	80	-	Anzahl der diskreten Abtastrichtungen (360°)
Max. Suchdistanz	500	m	Maximale Reichweite des Raytracings
Nahbereichsgrenze	10,0	m	Distanz für den Wechsel der Abtastrate
Solarmodul-Eigenschaften (Standardmodul)			
Modulbreite	1,13	m	Physische Breite des Referenzmoduls
Modulhöhe	1,76	m	Physische Höhe des Referenzmoduls
Nennleistung	430	Wp	Angenommene Modulleistung unter STC
Systemnutzungsgrad (PR)	0,85	-	Systemnutzungsgrad der Gesamtanlage
Platzierungs- und Optimierungslogik			
Randabstand (Dachkante)	1,0	m	Topologischer Puffer entlang der äußeren Dachkanten
Störobjekt-Abstand	1,0	m	Pufferbereich um detektierte LiDAR-Outlier
Dachnaht-Abstand	0,6	m	Pufferbereich an Schnittkanten nichtkoplanarer Flächen
Reihenabstand (Flachdach)	0,5	m	Basis-Zusatzabstand plus dynamischer Schattenabstand
Mindestertrag	600	kWh/kWp	Ausschlusskriterium je Modul
Spezifische Parameter für Flachdächer			
Grenzwert Dachneigung	5	°	Schwelle zur Klassifizierung als Flachdach
Aufständerungswinkel	15	°	Neigung bei aufgeständerten Modulen

Anhang C

Methodische Richtlinien und Referenzdaten

C.1 Annotationsrichtlinien für die semantische Segmentierung

Der folgende Leitfaden stellt eine gekürzte und leicht angepasste Version der originalen, englischsprachigen Annotationsrichtlinien dar, die im Rahmen dieser Arbeit erstellt und den annotierenden Hilfskräften zur Verfügung gestellt wurden. Der Auszug umfasst ausschließlich die für die Auswertung relevanten semantischen Klassen sowie die zugehörigen Annotationsregeln. Bedienungsanleitungen und Workflow-Beschreibungen wurden bewusst ausgelassen. Die Einhaltung dieser Richtlinien ist entscheidend für die Konsistenz der Trainingsdaten und beeinflusst direkt die Qualität der späteren Modellierungsergebnisse. Der Leitfaden wird im Original in englischer Sprache wiedergegeben, da er in dieser Form den Annotatoren bereitgestellt wurde.

Annotationsleitfaden (Auszug aus externen Instruktionen)

1 General Annotation Rules

1.1 Polygon Tightness

- Annotate objects as tightly as possible.
- Do not include unnecessary background pixels or unrelated objects.

1.2 Overlap Rules

- **Avoid overlap** between classes (except classes where overlap is explicitly allowed, see Section 4 (within this guideline).
- There should not be a gap between the annotation of two touching objects.

1.3 Occlusion Rules

- If only part of an object is visible, annotate only the visible portion.
- Occluded parts must not be inferred based on prior knowledge.
- *Examples:* A building partly hidden behind a tree crown → annotate visible building only. A roof partly out of frame → annotate the visible portion.

1.4 Handling of Shadows

- Shadows are not separate objects.
- Always annotate **the underlying surface or object**, even when in deep shadow.
- Do not draw boundaries around shadows.
- If the shadow makes the underlying class ambiguous, choose the most likely visible material.
- *Examples:* A tree shadow on pavement → label as pavement. A roof section in shadow → still part of the roof class.

2 Verification & Handling Ambiguities

Sometimes the satellite imagery may be blurry, shadowed, or ambiguous. If a class cannot be recognized with certainty, annotators must cross-reference with external tools:

- **Satellite View Verification:** Using Google Maps Satellite view to analyze imagery that may differ in quality or lighting, offering a clearer perspective.

- **Street View Verification:** Using Google Street View to observe the object from ground level. This is particularly useful for distinguishing between sub-classes like *Low Partial Sealing* vs. *Hard Sealing*.
- If classification remains impossible after external verification, the most probable class based on spatial context is selected.

3 Class Definitions

1. Tall Vegetation: Includes trees, shrubs, hedges, forest canopy. Includes visible crown edges even if small. Does not include green roofs, short garden plants or grass.

2. Low Vegetation: Includes lawns, grassy strips, meadows, small herbaceous plants. Excludes gravel with sparse plants (classify by substrate, not by vegetation).

3. Unvegetated Substrate: Bare ground without vegetation. Includes construction soil, sand piles, dry fields.

4. Crushed Stone / Gravel: Surfaces dominated by small stones, gravel paths, or industrial gravel areas. If heavily mixed with vegetation, choose the class that is more visually prevalent.

5. Water Bodies: Includes rivers, lakes, ponds, canals, retention basins.

6. Low Partial Sealing: Surfaces with wide vegetated gaps/joints (e.g., grass pavers, green permeable blocks).

7. Strong Partial Sealing: Surfaces with narrow joints, mostly mineral surfaces but not fully sealed (e.g., small-joint paving stones).

8. Green Roof (Building): Planted or vegetation-covered roof.

9. Dark Roof (Building): Typically black/grey/red roofing materials.

10. Light Roof (Building): Bright shingles, metal, light-colored concrete roofing materials.

11. Photovoltaic System: Solar panels on rooftops or ground-mounted arrays.

12. Technical Roof Structures: Fans, AC units, cooling aggregates, masts, large chimneys. Larger functional infrastructure.

13. Other Roof Structures: Small chimneys, skylights, roof windows.

14. Light Hard Sealing: Concrete, light-colored asphalt, light pavers.

15. Dark Hard Sealing: Dark asphalt roads, parking lots, sidewalks.

4 Specific Overlap Rules & Unclassifiable Objects

Most classes must not overlap. Each pixel should belong to exactly one class. However, the following exceptions apply:

- Roof Structures (Classes 12 & 13) **must** overlap fully with the underlying roof class.
- Photovoltaic Systems (Class 11) **must** overlap with the underlying roof class if they are mounted on top of roofs.

Unclassifiable Objects: Some objects do not belong to any predefined class (e.g., cars, buses, bicycles, furniture, people, manholes). In these cases, the underlying surface is annotated as if the object were not there. For example, a car parked on a driveway would not be annotated as a separate object; instead, the driveway surface (e.g., *Dark Hard Sealing*) is annotated as if the car were not present.

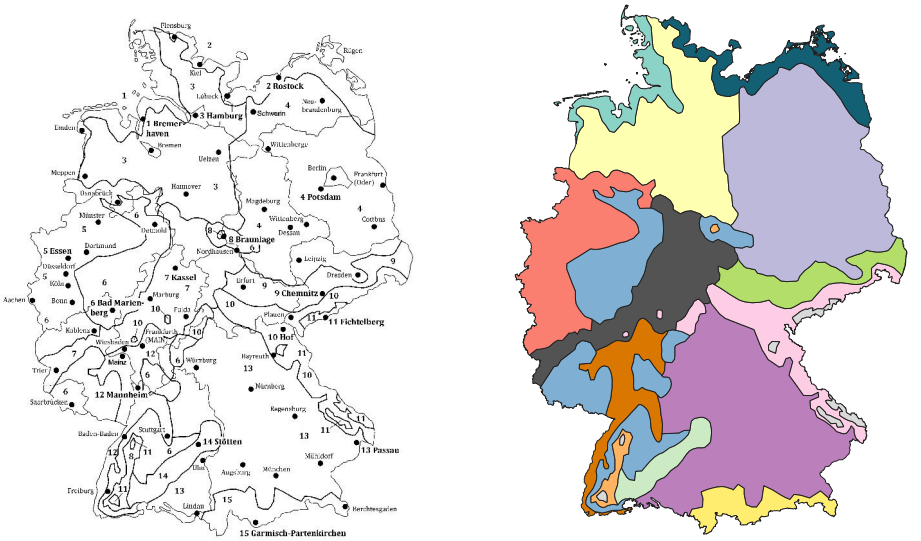
5 Quality Control

Before finalizing an annotation, the following aspects must be verified:

- **Tightness:** All polygons are tightly fitted to the objects.
- **Class Assignment:** Correct labels are assigned.
- **Consistency:** Similar objects are annotated consistently.
- **Completeness:** The entire image is fully annotated.

C.2 Referenzdaten zur Solaranalyse

C.2.1 Vektorisierung der Klimazonenkarte



(a) Original-Karte der Klimazonen aus DIN V 18599-10 [65].

(b) Georeferenzierte und vektorisierte Klimazonenkarte.

Abbildung C.1 – Vergleich der originalen Klimazonenkarte (a) mit der von der Pipeline georeferenzierten und vektorisierten Version (b).

C.2.2 Tabellarische Stützstellen der DIN V 18599

Die folgende Tabelle C.1 zeigt exemplarisch für die Klimaregion 1 (Bremerhaven) die tabellierten Einstrahlungswerte in kW h/m^2. Sie verdeutlicht die im Methodikteil beschriebenen diskreten Stützstellen für Modulneigung und -ausrichtung, auf welche die kontinuierlichen geometrischen Werte der PV-Module abgebildet wurden.

Tabelle C.1 – Referenzbestrahlungsstärken der Klimaregion 1 (Bremerhaven) nach DIN V 18599-10 [65].

Region 1 – Bremerhaven		**Mittlere monatliche Strahlungsintensität** I_S W/m^2												**Jahreswert** kW h/(m^2 a)
Orientierung	**Neigung**	**Jan**	**Feb**	**Mär**	**Apr**	**Mai**	**Jun**	**Jul**	**Aug**	**Sep**	**Okt**	**Nov**	**Dez**	**Jan bis Dez**
horizontal	0°	19	48	83	173	187	196	203	176	115	71	31	16	**964**
Süd	30°	27	62	97	197	191	194	204	189	128	91	43	25	**1059**
	45°	29	65	98	196	180	181	191	183	127	95	47	27	**1038**
	60°	31	65	94	185	163	161	172	169	120	95	48	29	**974**
	90°	28	57	76	140	114	110	118	123	94	81	44	27	**741**
Süd-Ost	30°	26	59	96	194	192	192	207	189	128	87	40	23	**1050**
	45°	28	62	96	192	182	180	197	184	127	90	43	25	**1029**
	60°	28	61	93	183	167	161	180	172	121	88	44	26	**969**
	90°	26	52	75	143	122	115	130	130	96	75	39	24	**751**
Süd-West	30°	23	54	88	182	182	190	193	176	117	80	37	20	**982**
	45°	23	55	85	176	170	178	179	167	112	80	38	21	**939**
	60°	23	53	79	164	153	160	160	153	103	77	37	21	**866**
	90°	19	43	61	124	110	113	114	112	77	63	31	18	**649**
Ost	30°	20	48	84	172	182	185	200	174	115	71	31	17	**952**
	45°	19	47	80	165	171	171	188	166	110	68	30	17	**902**
	60°	19	44	75	154	156	154	172	154	103	64	28	16	**833**
	90°	15	35	59	122	118	115	130	119	80	51	23	13	**644**
West	30°	17	43	74	155	168	180	179	156	101	64	27	14	**863**
	45°	16	40	67	143	153	166	162	143	92	59	25	13	**790**
	60°	14	36	60	130	136	148	143	128	82	53	23	12	**707**
	90°	10	27	45	98	99	108	104	95	61	40	17	9	**523**
Nord-West	30°	14	35	62	129	153	169	167	137	87	50	21	11	**757**
	45°	13	30	53	109	132	148	143	117	76	43	19	10	**654**
	60°	11	27	46	94	114	127	123	101	66	38	17	9	**566**
	90°	8	21	35	71	83	92	89	75	49	29	13	7	**419**
Nord-Ost	30°	14	37	68	142	165	173	183	151	97	53	22	12	**818**
	45°	13	33	60	124	147	154	164	134	86	47	20	11	**727**
	60°	11	29	53	110	129	135	145	118	76	42	18	9	**642**
	90°	9	23	41	86	96	100	108	90	58	33	13	7	**486**
Nord	30°	14	32	57	121	153	167	169	135	84	44	20	11	**738**
	45°	12	29	48	92	128	143	144	108	69	40	19	10	**616**
	60°	11	26	43	78	104	116	116	89	61	36	17	9	**516**
	90°	8	20	33	63	77	84	85	68	47	28	13	7	**390**

Anhang D

Systemimplementierung und ergänzende Visualisierungen

D.1 Interaktiver Demonstrator

Der interaktive Demonstrator visualisiert sämtliche Ergebnisse der entwickelten Pipeline. Dazu gehören unter anderem die Semantische Segmentierung, Platzierung von PV-Modulen sowie die Darstellung der Einstrahlungskarten und weiterer Umweltindikatoren. Die Benutzeroberfläche dient sowohl der Überprüfung der Berechnungsergebnisse als auch der anschaulichen Darstellung der Methodik.

Die nachfolgend gezeigten Screenshots dienen der Dokumentation und sind exemplarisch für die verschiedenen Funktionalitäten des Demonstrators. Alle Interaktionen und Ergebnisse können direkt über die Benutzeroberfläche nachvollzogen werden.

Abbildung D.1 – Auswahl eines Untersuchungsgebiets und Trigger der Pipeline, inklusive Logausgabe.

Abbildung D.2 – Visualisierung einer inferierten Segmentierung des Modells. Auf der rechten Seite sind genauere Informationen zu den Indikatoren und dem Inferenzergebnis zu sehen.

Abbildung D.3 – Visualisierung von platzierten PV-Modulen auf einem Schrägdach mit einer farblichen Darstellung der simulierten jährlichen Einstrahlung.

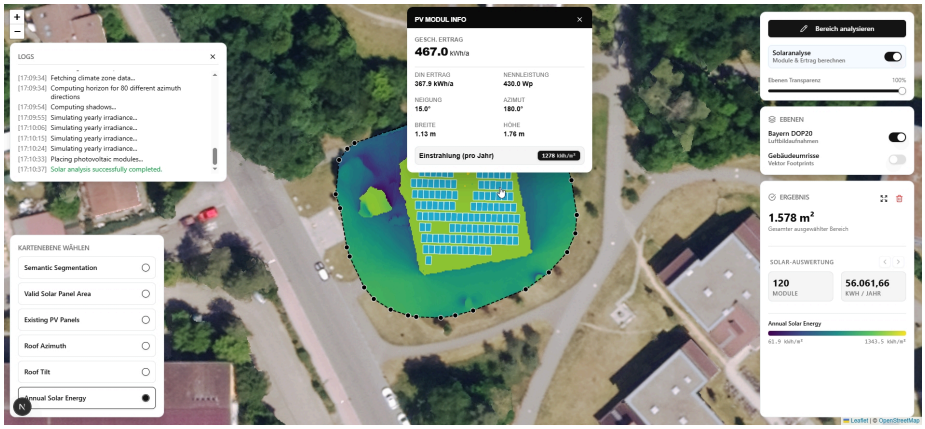


Abbildung D.4 – Visualisierung von platzierten PV-Modulen auf einem Flachdach inklusive genauerer Informationen zu einem ausgewählten Modul.

Abkürzungsverzeichnis

AUC	Area Under the Curve
BOHB	Bayesian Optimization and Hyperband
CNN	Convolutional Neural Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DGM	Digitales Geländemodell
DGNB	Deutsche Gesellschaft für Nachhaltiges Bauen
DOM	Digitales Oberflächenmodell
DOP	Digitales Orthophoto
FLAIR	French Land cover from Aerospace ImageRy
FN	False Negative
FP	False Positive
GSD	Ground Sampling Distance
IoU	Intersection over Union
ISPRS	International Society for Photogrammetry and Remote Sensing
KI	Künstliche Intelligenz
LiDAR	Light Detection and Ranging
LoD	Level of Detail
MAE	Mean Absolute Error
Masked AE	Masked Autoencoders
mIoU	Mean Intersection over Union
nDOM	Normalisiertes Digitales Oberflächenmodell
NIR	Nahinfrarot
nMAE	normalized Mean Absolute Error
OEM	Open Earth Map
PA	Pixel Accuracy
PV	Photovoltaik
RANSAC	Random Sample Consensus
RGB	Rot-Grün-Blau

ROC	Receiver Operating Characteristic
SSL	Self-Supervised Learning
TN	True Negative
TP	True Positive
TPE	Tree-Structured Parzen Estimator
TTA	Test Time Augmentation
ViT	Vision Transformer

Symbolverzeichnis

λ_1	Gewichtungsfaktor für den Diversitätsterm
λ_2	Gewichtungsfaktor für den Vollständigkeitsterm
$\mathbf{h}_{\text{global}}$	Klassenverteilung (normierter Histogramm-Vektor) des globalen Gesamtdatensatzes
$\mathbf{h}_{\text{subset}}$	Klassenverteilung (normierter Histogramm-Vektor) der vorgeschlagenen Teilmenge
$\mathbf{z}$	Vektor der Logits (Rohausgaben des Netzwerks)
$\overline{C_{\text{m}}}$	Gebietsmittlerer Abflussbeiwert
F1	F1-Score (harmonisches Mittel aus Präzision und Sensitivität)
FN	Anzahl fälschlich als negativ klassifizierter Pixel
FP	Anzahl fälschlich als positiv klassifizierter Pixel
IoU	Intersection over Union (Jaccard-Index)
mIoU	Mean Intersection over Union (Mittelwert über alle Klassen)
PA	Pixelgenauigkeit
Precision	Präzision: Anteil tatsächlich positiver unter den positiv vorhergesagten Pixeln
Recall	Sensitivität: Anteil korrekt erkannter unter den tatsächlich positiven Pixeln
$\text{Softmax}(\mathbf{z})_i$	Softmax-Wahrscheinlichkeit für die Klasse i
TN	Anzahl korrekt als negativ klassifizierter Pixel
TP	Anzahl korrekt als positiv klassifizierter Pixel

C	Anzahl der Klassen
C_{cov}	Metrik für die Diversität (Abdeckungsgrad der k-Means-Cluster)
C_{m}	Spezifischer Abflussbeiwert einer Bodenbedeckungsklasse
E_{sol}	Monatliche solare Bestrahlungsenergie auf das PV-System
f_{perf}	Systemleistungsfaktor
I_{ref}	Referenzsolarbestrahlungsstärke
N_{green}	Anzahl der als Gründach klassifizierten Pixel
N_{pv}	Anzahl der als Photovoltaikanlage klassifizierten Pixel
N_{roof}	Gesamtzahl der einem Dach zugeordneten Pixel
P_{existent}	Solare Einstrahlungswerte auf bestehende PV-Anlagen über ein Jahr
$P_{\text{pk,m}}$	Mittlere Peakleistung über 25 Jahre unter Norm-Prüfbedingungen inkl. Degradation
$P_{\text{potential}}$	Akkumuliertes Solarpotenzial über ein Jahr
$Q_{\text{f,prod,PV,i}}$	Monatliche Netto-Stromproduktion aus dem PV-System
R_{miss}	Metrik für die Unvollständigkeit (Strafterm für das vollständige Fehlen einzelner Klassen)
S_{roof}	Anteil nachhaltig genutzter Dachflächen
S_{score}	Gesamter Score-Wert zur iterativen Bewertung einer vorgeschlagenen Teilmenge
z_i	Logit-Wert der Klasse i

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Deutscher Bundestag, *Gebäudeenergiegesetz (GEG)*, In der Fassung vom 8. September 2024, 2020.

[2] Deutsche Gesellschaft für Nachhaltiges Bauen e. V. (DGNB), *DGNB System – Kriterienkatalog Quartiere*, Stuttgart, 2020. Adresse: https://www.dgnb.de/de/zertifizierung/quartiere (besucht am 01. 06. 2026).

[3] Europäisches Parlament und Europäischer Rat, *Richtlinie 2007/2/EG des Europäischen Parlaments und des Rates vom 14. März 2007 zur Schaffung einer Geodateninfrastruktur in der Europäischen Gemeinschaft (INSPIRE)*, 2007. Adresse: http://data.europa.eu/eli/dir/2007/2/oj.

[4] K. Dabrock, N. Pflugradt, J. M. Weinand und D. Stolten, *ETHOS.BUILDA: Building footprint and height dataset Germany*, Juni 2024. DOI: 10.5281/ZENODO.11845992.

[5] L. Wang, R. Li, C. Zhang u. a., „UNetFormer: A UNet-like transformer for efficient semantic segmentation of remote sensing urban scene imagery“, *ISPRS Journal of Photogrammetry and Remote Sensing*, Jg. 190, S. 196–214, Aug. 2022. DOI: 10.1016/j.isprsjprs.2022.06.008.

[6] A. Garioud, S. Peillet, E. Bookjans, S. Giordano und B. Wattrelos, *FLAIR #1: Semantic segmentation and domain adaptation dataset*, Apr. 2023. DOI: 10.13140/RG.2.2.30183.73128/1.

[7] J. Yu, Z. Wang, A. Majumdar und R. Rajagopal, „DeepSolar: A machine learning framework to efficiently construct a solar deployment database in the United States“, *Joule*, Jg. 2, Nr. 12, S. 2605–2617, Dez. 2018. DOI: 10.1016/j.joule.2018.11.021.

[8] K. Mayer, B. Rausch, M.-L. Arlt u. a., „3D-PV-locator: Large-scale detection of rooftop-mounted photovoltaic systems in 3D“, *Applied Energy*, Jg. 310, S. 118 469, März 2022. DOI: 10.1016/j.apenergy.2021.118469.

[9] S. Dutta und M. Das, „Remote sensing scene classification under scarcity of labelled samples: A survey of the state-of-the-arts“, *Computers & Geosciences*, Jg. 171, S. 105 295, Feb. 2023. DOI: 10.1016/j.cageo.2022.105295.

[10] Y. Roh, G. Heo und S. E. Whang, „A survey on data collection for machine learning: A big data – AI integration perspective“, *IEEE Transactions on Knowledge and Data Engineering*, Jg. 33, Nr. 4, S. 1328–1347, Apr. 2021. DOI: 10.1109/TKDE.2019.2946162.

[11] I. Goodfellow, Y. Bengio und A. Courville, *Deep learning*. MIT Press, 2016.

[12] I. Loshchilov und F. Hutter, „Decoupled weight decay regularization“, in *International Conference on Learning Representations*, Sep. 2018. Adresse: https://openreview.net/forum?id=Bkg6RiCqY7.

[13] F. Chollet, *Deep learning with Python*, 1. Aufl. USA: Manning Publications Co., 2017.

[14] K. He, X. Zhang, S. Ren und J. Sun, „Deep residual learning for image recognition“, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, Juni 2016, S. 770–778. DOI: 10.1109/CVPR.2016.90.

[15] J. Redmon, S. Divvala, R. Girshick und A. Farhadi, „You only look once: Unified, real-time object detection“, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Juni 2016, S. 779–788. DOI: 10.1109/CVPR.2016.91.

[16] E. Shelhamer, J. Long und T. Darrell, „Fully convolutional networks for semantic segmentation“, *IEEE Trans. Pattern Anal. Mach. Intell.*, Jg. 39, Nr. 4, S. 640–651, Apr. 2017. DOI: 10.1109/TPAMI.2016.2572683.

[17] K. He, G. Gkioxari, P. Dollár und R. B. Girshick, „Mask R-CNN“, *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017. DOI: 10.1109/TPAMI.2018.2844175.

[18] International Society for Photogrammetry and Remote Sensing, *2D Semantic Labeling Contest – Potsdam*, 2015. Adresse: https://www.isprs.org/resources/datasets/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx (besucht am 01. 06. 2026).

[19] O. Ronneberger, P. Fischer und T. Brox, „U-Net: Convolutional networks for biomedical image segmentation“, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells und A. F. Frangi, Hrsg., Cham: Springer International Publishing, 2015, S. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[20] V. Badrinarayanan, A. Kendall und R. Cipolla, „SegNet: A deep convolutional encoder-decoder architecture for image segmentation“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 39, Nr. 12, S. 2481–2495, Dez. 2017. DOI: 10.1109/TPAMI.2016.2644615.

[21] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell und S. Xie, „A ConvNet for the 2020s“, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Juni 2022, S. 11 966–11 976. DOI: 10.1109/CVPR52688.2022.01167.

[22] A. Dosovitskiy, L. Beyer, A. Kolesnikov u. a., „An image is worth 16x16 words: Transformers for image recognition at scale“, in *International Conference on Learning Representations*, Okt. 2020. Adresse: https://openreview.net/forum?id=YicbFdNTTy.

[23] Z. Chen, Y. Duan, W. Wang u. a., „Vision transformer adapter for dense predictions“, in *The Eleventh International Conference on Learning Representations*, Sep. 2022. Adresse: https://openreview.net/forum?id=plKu2GByCNW.

[24] Z. Liu, Y. Lin, Y. Cao u. a., „Swin Transformer: Hierarchical vision transformer using shifted windows“, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Okt. 2021, S. 9992–10 002. DOI: 10.1109/ICCV48922.2021.00986.

[25] W. Wang, J. Dai, Z. Chen u. a., „InternImage: Exploring large-scale vision foundation models with deformable convolutions“, in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Juni 2023, S. 14 408–14 419. DOI: 10.1109/CVPR52729.2023.01385.

[26] T. Xiao, Y. Liu, B. Zhou, Y. Jiang und J. Sun, „Unified perceptual parsing for scene understanding“, in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu und Y. Weiss, Hrsg., Cham: Springer International Publishing, 2018, S. 432–448. DOI: 10.1007/978-3-030-01228-1_26.

[27] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff und H. Adam, „Encoder-decoder with atrous separable convolution for semantic image segmentation“, in *Computer Vision – ECCV 2018*, V. Ferrari, M. Hebert, C. Sminchisescu und Y. Weiss, Hrsg., Cham: Springer International Publishing, 2018, S. 833–851. DOI: 10.1007/978-3-030-01234-2_49.

[28] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez und P. Luo, „SegFormer: Simple and efficient design for semantic segmentation with transformers“, in *Advances in Neural Information Processing Systems*, Bd. 34, Curran Associates, Inc., 2021, S. 12 077–12 090. Adresse: https://proceedings.neurips.cc/paper_files/paper/2021/hash/64f1f27bf1b4ec22924fd0acb550c235-Abstract.html.

[29] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov und R. Girdhar, „Masked-attention mask transformer for universal image segmentation“, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Juni 2022, S. 1280–1289. DOI: 10.1109/CVPR52688.2022.00135.

[30] F. Milletari, N. Navab und S.-A. Ahmadi, „V-Net: Fully convolutional neural networks for volumetric medical image segmentation“, in *2016 Fourth International Conference on 3D Vision (3DV)*, Okt. 2016, S. 565–571. DOI: 10.1109/3DV.2016.79.

[31] C. Shorten und T. M. Khoshgoftaar, „A survey on image data augmentation for deep learning“, *Journal of Big Data*, Jg. 6, Nr. 1, S. 60, Juli 2019. DOI: 10.1186/s40537-019-0197-0.

[32] T. DeVries und G. W. Taylor, *Improved regularization of convolutional neural networks with cutout*, 2017. DOI: 10.48550/arxiv.1708.04552.

[33] S. J. Pan und Q. Yang, „A survey on transfer learning“, *IEEE Transactions on Knowledge and Data Engineering*, Jg. 22, Nr. 10, S. 1345–1359, Okt. 2010. DOI: 10.1109/TKDE.2009.191.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li und L. Fei-Fei, „ImageNet: A large-scale hierarchical image database“, in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, Juni 2009, S. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[35] J. Yosinski, J. Clune, Y. Bengio und H. Lipson, „How transferable are features in deep neural networks?“, in *Advances in Neural Information Processing Systems*, Bd. 27, Curran Associates, Inc., 2014. Adresse: https://proceedings.neurips.cc/paper_files/paper/2014/hash/532a2f85b6977104bc93f8580abbb330-Abstract.html.

[36] A. A. Mukhlif, B. Al-Khateeb und M. A. Mohammed, „Incorporating a novel dual transfer learning approach for medical images“, *Sensors*, Jg. 23, Nr. 2, S. 570, Jan. 2023. DOI: 10.3390/s23020570.

[37] K. He, X. Chen, S. Xie, Y. Li, P. Dollár und R. Girshick, „Masked autoencoders are scalable vision learners“, in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Juni 2022, S. 15 979–15 988. DOI: 10.1109/CVPR52688.2022.01553.

[38] M. Caron, H. Touvron, I. Misra u. a., „Emerging properties in self-supervised vision transformers“, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Okt. 2021, S. 9630–9640. DOI: 10.1109/ICCV48922.2021.00951.

[39] O. Siméoni, H. V. Vo, M. Seitzer u. a., *DINOv3*, Aug. 2025. DOI: 10.48550/arXiv.2508.10104.

[40] J. Lv, Q. Shen, M. Lv, Y. Li, L. Shi und P. Zhang, „Deep learning-based semantic segmentation of remote sensing images: A review“, *Frontiers in Ecology and Evolution*, Jg. 11, Juli 2023. DOI: 10.3389/fevo.2023.1201125.

[41] J. Bergstra und Y. Bengio, „Random search for hyper-parameter optimization“, *Journal of machine learning research*, Jg. 13, Nr. 2, S. 281–305, Feb. 2012. DOI: 10.5555/2188385.2188395.

[42] J. Snoek, H. Larochelle und R. P. Adams, „Practical Bayesian optimization of machine learning algorithms“, in *Advances in Neural Information Processing Systems*, Bd. 25, Curran Associates, Inc., 2012. Adresse: https://papers.nips.cc/paper_files/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html.

[43] D. Passos und P. Mishra, „A tutorial on automatic hyperparameter tuning of deep spectral modelling for regression and classification tasks“, *Chemometrics and Intelligent Laboratory Systems*, Jg. 223, S. 104 520, Apr. 2022. DOI: 10.1016/j.chemolab.2022.104520.

[44] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh und A. Talwalkar, „Hyperband: A novel bandit-based approach to hyperparameter optimization“, *J. Mach. Learn. Res.*, Jg. 18, Nr. 1, S. 6765–6816, Jan. 2017. Adresse: https://jmlr.org/papers/v18/16-558.html.

[45] S. Falkner, A. Klein und F. Hutter, „BOHB: Robust and efficient hyperparameter optimization at scale“, in *Proceedings of the 35th International Conference on Machine Learning*, PMLR, Juli 2018, S. 1437–1446. Adresse: https://proceedings.mlr.press/v80/falkner18a.html.

[46] K. Janowicz, S. Gao, G. McKenzie, Y. Hu und B. Bhaduri, „GeoAI: Spatially explicit artificial intelligence techniques for geographic knowledge discovery and beyond“, *International Journal of Geographical Information Science*, Jg. 34, Nr. 4, S. 625–636, Apr. 2020. DOI: 10.1080/13658816.2019.1684500.

[47] X. X. Zhu, D. Tuia, L. Mou u. a., „Deep learning in remote sensing: A comprehensive review and list of resources“, *IEEE Geoscience and Remote Sensing Magazine*, Jg. 5, Nr. 4, S. 8–36, Dez. 2017. DOI: 10.1109/MGRS.2017.2762307.

[48] Bayerische Vermessungsverwaltung, *Orthophotos*, Amtliche Geobasisdaten des Bayerischen Landesamtes für Digitalisierung, Breitband und Vermessung (LDBV), 2025. Adresse: https://www.ldbv.bayern.de/produkte/luftbilder/orthophotos.html (besucht am 01. 06. 2026).

[49] Bayerische Vermessungsverwaltung, *Digitales Oberflächenmodell*, Amtliche Geobasisdaten des Bayerischen Landesamtes für Digitalisierung, Breitband und Vermessung (LDBV), 2025. Adresse: https://www.ldbv.bayern.de/produkte/landschaftsinformationen/dom.html (besucht am 01. 06. 2026).

[50] Bayerische Vermessungsverwaltung, *Geländemodell*, Amtliche Geobasisdaten des Bayerischen Landesamtes für Digitalisierung, Breitband und Vermessung (LDBV), 2025. Adresse: https://www.ldbv.bayern.de/produkte/landschaftsinformationen/gelaende.html (besucht am 01. 06. 2026).

[51] Bayerische Vermessungsverwaltung, *3D-Gebäudemodell*, Amtliche Geobasisdaten des Bayerischen Landesamtes für Digitalisierung, Breitband und Vermessung (LDBV), 2025. Adresse: https://www.ldbv.bayern.de/produkte/liegenschaftsinformationen/gebaeudemodell.html (besucht am 01. 06. 2026).

[52] Bayerische Vermessungsverwaltung, *Hausumringe*, Amtliche Geobasisdaten des Bayerischen Landesamtes für Digitalisierung, Breitband und Vermessung (LDBV), 2025. Adresse: https://www.ldbv.bayern.de/produkte/liegenschaftsinformationen/hausumringe.html (besucht am 01. 06. 2026).

[53] Bayerische Vermessungsverwaltung, *Laserpunkte*, Amtliche Geobasisdaten des Bayerischen Landesamtes für Digitalisierung, Breitband und Vermessung (LDBV), 2025. Adresse: https://www.ldbv.bayern.de/produkte/landschaftsinformationen/laser.html (besucht am 01. 06. 2026).

[54] Bayerische Vermessungsverwaltung, *ALKIS® – Tatsächliche Nutzung (TN)*, Amtliche Geobasisdaten des Bayerischen Landesamtes für Digitalisierung, Breitband und Vermessung (LDBV), 2025. Adresse: https://www.ldbv.bayern.de/produkte/liegenschaftsinformationen/tat_nutzung.html (besucht am 01. 06. 2026).

[55] K. Šušteršič, A. Šašić Kežul und M. Kosmatin Fras, „Automatically produced true orthophotos for large urban areas“, *GIM International*, Nr. 4, 2021. Adresse: https://www.gim-international.com/content/article/automatically-produced-true-orthophotos-for-large-urban-areas (besucht am 01. 06. 2026).

[56] International Society for Photogrammetry and Remote Sensing, *2D Semantic Labeling Contest – Vaihingen*, 2014. Adresse: https://www.isprs.org/resources/datasets/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx (besucht am 01. 06. 2026).

[57] J. Xia, N. Yokoya, B. Adriano und C. Broni-Bediako, „OpenEarthMap: A benchmark dataset for global high-resolution land cover mapping“, in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2023, S. 6243–6253. DOI: 10.1109/WACV56688.2023.00619.

[58] M. A. Fischler und R. C. Bolles, „Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography“, *Communications of the ACM*, Jg. 24, Nr. 6, S. 381–395, Juni 1981. DOI: 10.1145/358669.358692.

[59] M. Ester, H.-P. Kriegel, J. Sander und X. Xu, „A density-based algorithm for discovering clusters in large spatial databases with noise“, in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Ser. KDD'96, Portland, Oregon: AAAI Press, 1996, S. 226–231. Adresse: http://cdn.aaai.org/KDD/1996/KDD96-037.pdf.

[60] J. L. Bentley, „Multidimensional binary search trees used for associative searching“, *Commun. ACM*, Jg. 18, Nr. 9, S. 509–517, Sep. 1975. DOI: 10.1145/361002.361007.

[61] H. Edelsbrunner, D. Kirkpatrick und R. Seidel, „On the shape of a set of points in the plane“, *IEEE Transactions on Information Theory*, Jg. 29, Nr. 4, S. 551–559, Juli 1983. DOI: 10.1109/TIT.1983.1056714.

[62] J. Dozier und J. Frew, „Rapid calculation of terrain parameters for radiation modeling from digital elevation data“, *IEEE Transactions on Geoscience and Remote Sensing*, Jg. 28, Nr. 5, S. 963–969, Sep. 1990. DOI: 10.1109/36.58986.

[63] R. Perez, P. Ineichen, R. Seals, J. Michalsky und R. Stewart, „Modeling daylight availability and irradiance components from direct and global irradiance“, *Solar Energy*, Jg. 44, Nr. 5, S. 271–289, Jan. 1990. DOI: 10.1016/0038-092X(90)90055-H.

[64] DIN Deutsches Institut für Normung e. V., *DIN V 18599-9, Energetische Bewertung von Gebäuden – Teil 9: End- und Primärenergiebedarf von stromproduzierenden Anlagen*, Berlin, Sep. 2018.

[65] DIN Deutsches Institut für Normung e. V., *DIN V 18599-10, Energetische Bewertung von Gebäuden – Teil 10: Nutzungsrandbedingungen und Klimadaten*, Berlin, Sep. 2018.

[66] C. Northcutt, L. Jiang und I. Chuang, „Confident learning: Estimating uncertainty in dataset labels“, *J. Artif. Int. Res.*, Jg. 70, S. 1373–1411, Mai 2021. DOI: 10.1613/jair.1.12125.

[67] M. Cordts, M. Omran, S. Ramos u. a., „The Cityscapes dataset for semantic urban scene understanding“, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Juni 2016, S. 3213–3223. DOI: 10.1109/CVPR.2016.350.

[68] J. Lu, W. Li, Q. Wang und Y. Zhang, „Research on data quality control of crowdsourcing annotation: A survey“, in *2020 IEEE International Conference on DASC/PiCom/CBDCom/CyberSciTech*, Aug. 2020, S. 201–208. DOI: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00044.

[69] G. M. Foody, L. See, S. Fritz u. a., „Assessing the accuracy of volunteered geographic information arising from multiple contributors to an Internet based collaborative project“, *Transactions in GIS*, Jg. 17, Nr. 6, S. 847–860, 2013. DOI: 10.1111/tgis.12033.

[70] S. Budd, E. C. Robinson und B. Kainz, „A survey on active learning and human-in-the-loop deep learning for medical image analysis“, *Medical Image Analysis*, Jg. 71, S. 102 062, Juli 2021. DOI: 10.1016/j.media.2021.102062.

[71] J. Beck, S. Eckman, C. Kern und F. Kreuter, „Bias in the loop: How humans evaluate AI-generated suggestions“, *Harvard Data Science Review*, März 2026. DOI: 10.1162/99608f92.0e98898d.

[72] B. C. Russell, A. Torralba, K. P. Murphy und W. T. Freeman, „LabelMe: A database and web-based tool for image annotation“, *International Journal of Computer Vision*, Jg. 77, Nr. 1, S. 157–173, Mai 2008. DOI: 10.1007/s11263-007-0090-8.

[73] B. Sekachev, N. Manovich, M. Zhiltsov u. a., *OpenCV/CVAT: v1.1.0*, Zenodo, Aug. 2020. DOI: 10.5281/zenodo.4009388.

[74] T. Hanyu, K. Yamazaki, M. Tran u. a., „AerialFormer: Multi-resolution transformer for aerial image segmentation“, *Remote Sensing*, Jg. 16, Nr. 16, S. 2930, Jan. 2024. DOI: 10.3390/rs16162930.

[75] M. B. A. Gibril, R. Al-Ruzouq, A. Shanableh u. a., „Transformer-based semantic segmentation for large-scale building footprint extraction from very-high resolution satellite images“, *Advances in Space Research*, Jg. 73, Nr. 10, S. 4937–4954, Mai 2024. DOI: 10.1016/j.asr.2024.03.002.

[76] Y. Cong, S. Khanna, C. Meng u. a., „SatMAE: Pre-training transformers for temporal and multi-spectral satellite imagery“, *Advances in Neural Information Processing Systems*, Jg. 35, S. 197–211, Dez. 2022. Adresse: https://papers.nips.cc/paper_files/paper/2022/hash/01c561df365429f33fcd7a7faa44c985-Abstract-Conference.html.

[77] D. Szwarcman, S. Roy, P. Fraccaro u. a., „Prithvi-EO-2.0: A versatile multitemporal foundation model for earth observation applications“, *IEEE Transactions on Geoscience and Remote Sensing*, Jg. 64, S. 1–20, 2026. DOI: 10.1109/TGRS.2025.3642610.

[78] C. Finn, P. Abbeel und S. Levine, „Model-agnostic meta-learning for fast adaptation of deep networks“, in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Juli 2017, S. 1126–1135. Adresse: https://proceedings.mlr.press/v70/finn17a.html.

[79] T. Hospedales, A. Antoniou, P. Micaelli und A. Storkey, „Meta-learning in neural networks: A survey“, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 44, Nr. 9, S. 5149–5169, Sep. 2022. DOI: 10.1109/TPAMI.2021.3079209.

[80] D. Baranchuk, A. Voynov, I. Rubachev, V. Khrulkov und A. Babenko, „Label-efficient semantic segmentation with diffusion models“, in *International Conference on Learning Representations*, Okt. 2021. Adresse: https://openreview.net/forum?id=SlxSY2UZQT.

[81] B. Kawar, M. Elad, S. Ermon und J. Song, „Denoising diffusion restoration models“, in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, Ser. NIPS '22, Red Hook, NY, USA: Curran Associates Inc., Nov. 2022, S. 23 593–23 606. Adresse: https://proceedings.neurips.cc/paper_files/paper/2022/hash/95504595b6169131b6ed6cd72eb05616-Abstract-Conference.html.

[82] A. Lefebvre, C. Sannier und T. Corpetti, „Monitoring urban areas with Sentinel-2A data: Application to the update of the Copernicus high resolution layer imperviousness degree“, *Remote Sensing*, Jg. 8, Nr. 7, S. 606, Juli 2016. DOI: 10.3390/rs8070606.

[83] G.-H. Strand, „Accuracy of the Copernicus high-resolution layer imperviousness density (HRL IMD) assessed by point sampling within pixels“, *Remote Sensing*, Jg. 14, Nr. 15, S. 3589, Juli 2022. DOI: 10.3390/rs14153589.

[84] S. Mahyoub, H. Rhinane, M. Mansour, A. Fadil, Y. Akensous und A. Al Sabri, „The use of deep learning in remote sensing for mapping impervious surface: A review paper“, *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Jg. XLVI-4-W3-2021, S. 199–203, Jan. 2022. DOI: 10.5194/isprs-archives-XLVI-4-W3-2021-199-2022.

[85] Z. Li, A. Zhang, G. Sun, Z. Han und X. Jia, „Automatic impervious surface mapping in subtropical China via a terrain-guided gated fusion network“, *International Journal of Applied Earth Observation and Geoinformation*, Jg. 127, S. 103 608, März 2024. DOI: 10.1016/j.jag.2023.103608.

[86] R. Yin, G. He, G. Wang u. a., „Enhanced 30 m impervious surfaces for China (2020, 2022) via 2 m/30 m data fusion“, *Scientific Data*, Jg. 13, Nr. 1, S. 297, Jan. 2026. DOI: 10.1038/s41597-026-06619-3.

[87] T. Zhong, Z. Zhang, M. Chen u. a., „A city-scale estimation of rooftop solar photovoltaic potential based on deep learning“, *Applied Energy*, Jg. 298, S. 117 132, Sep. 2021. DOI: 10.1016/j.apenergy.2021.117132.

[88] G. Li, Z. Wang und C. Xu, „A novel rooftop solar energy potential estimation method based on deep learning at district level“, Sep. 2023. DOI: 10.26868/25222708.2023.1450.

[89] Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), „Konzeptentwicklung für die Informationsgewinnung zum Gebäudebestand in Deutschland aus Fernerkundungsdaten (G-DAT DE)“, Deutsches Zentrum für Luft- und Raumfahrt e.V., Oberpfaffenhofen, Endbericht SWD – 10.08.17.7-18.13, 2021.

[90] A. Paszke, S. Gross, F. Massa u. a., „PyTorch: An imperative style, high-performance deep learning library“, in *Advances in Neural Information Processing Systems*, Bd. 32, Curran Associates, Inc., 2019. Adresse: https://proceedings.neurips.cc/paper_files/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html.

[91] MMSegmentation Contributors, *OpenMMLab's semantic segmentation toolbox and benchmark*, Juli 2020. Adresse: https://github.com/open-mmlab/mmsegmentation (besucht am 01. 06. 2026).

[92] MMPreTrain Contributors, *OpenMMLab's pre-training toolbox and benchmark*, Apr. 2023. Adresse: https://github.com/open-mmlab/mmpretrain (besucht am 01. 06. 2026).

[93] T. Akiba, S. Sano, T. Yanase, T. Ohta und M. Koyama, „Optuna: A next-generation hyperparameter optimization framework“, in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Ser. KDD '19, New York, NY, USA: Association for Computing Machinery, Juli 2019, S. 2623–2631. DOI: 10.1145/3292500.3330701.

[94] P. Virtanen, R. Gommers, T. E. Oliphant u. a., „SciPy 1.0: Fundamental algorithms for scientific computing in Python“, *Nature Methods*, Jg. 17, Nr. 3, S. 261–272, März 2020. DOI: 10.1038/s41592-019-0686-2.

[95] F. Pedregosa, G. Varoquaux, A. Gramfort u. a., „Scikit-learn: Machine learning in Python“, *J. Mach. Learn. Res.*, Jg. 12, S. 2825–2830, Nov. 2011. Adresse: https://www.jmlr.org/papers/v12/pedregosa11a.html.

[96] K. S. Anderson, C. W. Hansen, W. F. Holmgren, A. R. Jensen, M. A. Mikofski und A. Driesse, „Pvlib Python: 2023 project update“, *Journal of Open Source Software*, Jg. 8, Nr. 92, S. 5994, Dez. 2023. DOI: 10.21105/joss.05994.

[97] QGIS Development Team, *QGIS geographic information system*, QGIS Association, 2025. Adresse: https://www.qgis.org (besucht am 01. 06. 2026).

[98] F. Isensee, P. F. Jaeger, S. A. A. Kohl, J. Petersen und K. H. Maier-Hein, „nnU-Net: A self-configuring method for deep learning-based biomedical image segmentation“, *Nature Methods*, Jg. 18, Nr. 2, S. 203–211, Feb. 2021. DOI: 10.1038/s41592-020-01008-z.

[99] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso und A. Torralba, „Scene parsing through ADE20K dataset“, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Juli 2017, S. 5122–5130. DOI: 10.1109/CVPR.2017.544.

[100] L. Ericsson, H. Gouk und T. M. Hospedales, „How well do self-supervised models transfer?“, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Juni 2021, S. 5410–5419. DOI: 10.1109/CVPR46437.2021.00537.

[101] A. Oliver, A. Odena, C. Raffel, E. D. Cubuk und I. Goodfellow, „Realistic evaluation of deep semi-supervised learning algorithms“, in *Advances in Neural Information Processing Systems*, Bd. 31, Curran Associates, Inc., 2018. Adresse: https://proceedings.neurips.cc/paper/2018/hash/c1fea270c48e8079d8ddf7d06d26ab52-Abstract.html.

[102] T. Elsken, J. H. Metzen und F. Hutter, „Neural architecture search: A survey“, *J. Mach. Learn. Res.*, Jg. 20, Nr. 1, S. 1997–2017, Jan. 2019. Adresse: https://jmlr.org/papers/v20/18-598.html.

[103] S. Yu, L. Chen, S. Ahmadian und M. Fadaee, „Diversify and conquer: Diversity-centric data selection with iterative refinement“, Okt. 2024. Adresse: https://openreview.net/forum?id=V0GlKhMLFl.

[104] L. van der Maaten, G. Hinton und Y. Rachmad, „Visualizing data using t-SNE“, *Journal of Machine Learning Research*, Jg. 9, S. 2579–2605, Nov. 2008. Adresse: https://www.jmlr.org/papers/v9/vandermaaten08a.html.

[105] T. Darcet, M. Oquab, J. Mairal und P. Bojanowski, „Vision transformers need registers“, in *International Conference on Learning Representations (ICLR)*, 2024. Adresse: https://openreview.net/forum?id=2dnO3LLiJ1.

[106] J. Su, M. Ahmed, Y. Lu, S. Pan, W. Bo und Y. Liu, „RoFormer: Enhanced transformer with rotary position embedding“, *Neurocomput.*, Jg. 568, Nr. C, Feb. 2024. DOI: 10.1016/j.neucom.2023.127063.

[107] J. Bergstra, R. Bardenet, Y. Bengio und B. Kégl, „Algorithms for hyper-parameter optimization“, in *Advances in Neural Information Processing Systems*, Bd. 24, Curran Associates, Inc., 2011. Adresse: https://papers.nips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html.

[108] H. Li, L. Ding, M. Ren, C. Li und H. Wang, „Sponge city construction in China: A survey of the challenges and opportunities“, *Water*, Jg. 9, Nr. 9, S. 594, Aug. 2017. DOI: 10.3390/w9090594.

[109] DIN Deutsches Institut für Normung e. V., *E DIN 1986-100, Entwässerungsanlagen für Gebäude und Grundstücke - Teil 100: Bestimmungen in Verbindung mit DIN EN 752 und DIN EN 12056*, Norm-Entwurf, Berlin, Juni 2025.

[110] T. Huld, R. Müller und A. Gambardella, „A new solar radiation database for estimating PV performance in Europe and Africa“, *Solar Energy*, Jg. 86, Nr. 6, S. 1803–1815, Juni 2012. DOI: 10.1016/j.solener.2012.03.006.

[111] A. Ashukha, A. Lyzhov, D. Molchanov und D. Vetrov, „Pitfalls of in-domain uncertainty estimation and ensembling in deep learning“, in *Eighth International Conference on Learning Representations*, Apr. 2020. Adresse: https://iclr.cc/virtual_2020/poster_BJxI5gHKDr.html.

[112] C. Guo, G. Pleiss, Y. Sun und K. Q. Weinberger, „On calibration of modern neural networks“, in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Juli 2017, S. 1321–1330. Adresse: https://proceedings.mlr.press/v70/guo17a.html.

[113] D. Hendrycks und K. Gimpel, „A baseline for detecting misclassified and out-of-distribution examples in neural networks“, in *International Conference on Learning Representations*, Feb. 2017. Adresse: https://openreview.net/forum?id=Hkg4TI9xl.