

# Bachelorarbeit

vorgelegt an der Technischen Hochschule Würzburg-Schweinfurt  
an der Fakultät Kunststofftechnik und Vermessung

Thema:

## **Evaluierung vortrainierter KI-Algorithmen zur automatisierten Erkennung von Siedlungen in historischen Topographischen Karten**

Betreuer/in: Gerald Riedel  
Prüfer/in: Prof. Dr. Melanie Brandmeier  
Abgabetermin: 4. Mai 2023

Eingereicht von  
Emilie Lüdicke  
geb. in Berlin Pankow  
Matrikelnummer: 6019005  
Würzburg, den 04.05.2023

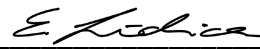
## Erklärung zur Bachelorarbeit

Hiermit versichere ich, dass die vorgelegte Bachelorarbeit selbstständig verfasst und noch nicht anderweitig zu Prüfungszwecken vorgelegt wurde.

Alle verwendeten Quellen und Hilfsmittel sind angegeben.

Wörtliche und sinngemäße Zitate wurden als solche gekennzeichnet.

Würzburg, den 4. Mai 2023



Unterschrift des Studierenden

## Einwilligung zur Überprüfung einer Arbeit mit der Plagiatserkennungssoftware PlagScan

Name: Lüdicke

Vorname: Emilie

Matr.-Nr.: 6019005

Adresse: Bahnhofstr. 31, 97215 Uffenheim

E-Mail: emilie.luedicke@study.thws.de

Studiengang: Vermessung und Geoinformatik

Titel der Arbeit: Evaluierung vortrainierter KI-Algorithmen zur automatisierten Erkennung von Siedlungen in historischen Topographischen Karten

Betreuer: Prof. Dr. Melanie Brandmeier, Gerald Riedel

Auf Grund der Zielvereinbarung der FHWS mit dem Bayerischen Staatsministerium für Bildung und Kultus, Wissenschaft und Kunst von 19. März 2014, Ziffer 2.3, hat die Hochschule entschieden, Studien- und Abschlussarbeiten künftig durch die Plagiatserkennungssoftware PlagScan elektronisch auf Plagiate hin zu überprüfen.

Die zu überprüfenden Arbeiten werden an den Dienst Plagscan übermittelt, dort auf Übereinstimmung mit externen Quellen untersucht und zum Zweck des Abgleichs mit zukünftig zu überprüfenden Studien- und Prüfungsarbeiten gespeichert. **Die befristete Speicherung Ihrer Arbeit in der Datenbank sowie die Weitergabe Ihrer persönlichen Daten im Rahmen der Plagiatsprüfung ist nur mit Ihrer Einwilligung zulässig.**

Mit meiner Unterschrift erkläre ich meine **Einwilligung**, dass

- die von mir vorgelegte und verfasste Arbeit zum Zweck der Überprüfung auf Plagiate hin an PlagScan übermittelt und vorübergehend (5 Jahre) in der von PlagScan geführten Datenbank gespeichert wird;
- meine persönlichen Daten (Vorname, Name, studentische E-Mail-Adresse) zusammen mit dem Text digital gespeichert und verwendet werden. Diese Daten sind nur meiner Prüferin oder meinem Prüfer/meinen Prüferinnen oder Prüfern zugänglich.

### Hinweis:

*Diese Einwilligungserklärung ist freiwillig. Sie haben die Möglichkeit die Erklärung abzulehnen. Durch die Verweigerung der Einwilligung kann bei Entfernung der persönlichen Angaben und Wahrung der urheberrechtlichen Vorgaben die Plagiatsprüfung nicht verhindert werden. Die Einwilligung zur Speicherung und Verwendung der persönlichen Daten kann jederzeit durch Erklärung gegenüber der Fakultät mit Wirkung für die Zukunft widerrufen werden.*

Ort: Würzburg

Datum: 4. Mai 2023

Unterschrift: 

## **Kurzfassung der Aufgabenstellung**

Historische Karten enthalten eine Vielzahl an Informationen, die für Entwicklungsanalysen herangezogen werden können. Die handgezeichneten Karten liegen im Staatsarchiv aber nur in Papierform vor oder sind als Bild digital referenziert. Eine Digitalisierung der einzelnen Objekte in den Karten hat aufgrund des unverhältnismäßig hohen Aufwands einer manuellen Vorgehensweise bisher nicht stattgefunden. Diese Arbeit widmet sich der Aufgabe, Siedlungen in historischen Topographischen Karten automatisiert erkennen zu lassen. Dafür sollen passende in ArcGIS Pro implementierte KI-Algorithmen verwendet und bezüglich ihrer Leistung untersucht werden. Die Auswirkungen der unterschiedlichen Kombinationen von Modell und Parametern ist darzustellen. Schließlich soll die Verwendbarkeit der vortrainierten Modelle beurteilt und soweit möglich optimiert werden.

## **Kurzfassung der Arbeit**

In dieser Arbeit werden in ArcGIS Pro implementierte vortrainierte Deep-Learning-Modelle für die automatische Segmentierung von Siedlungen getestet. Es erfolgt eine kurze Einführung zur Funktionsweise Neuronaler Netze und zu den Hintergründen der verwendeten Modelle. Die Datengrundlage sowie die aktuelle Datenmodellierung der Siedlungsflächen werden vorgestellt. Anhand eines Topographischen Kartenblatts im Maßstab 1:25 000 werden Siedlungsgebiete manuell digitalisiert und bekannte Open-Source-Modelle mittels dieses Trainingsdatensatzes unter Verwendung unterschiedlicher Parameter trainiert. Die Performance der Modelle wird auf einem Testgebiet untersucht und bewertet. Anschließend werden Verbesserungsmöglichkeiten mittels Nachverarbeitung vorgestellt. Die Ergebnisse werden diskutiert und das Modell mit der besten Leistung in ArcGIS Online veröffentlicht.

# Inhaltsverzeichnis

Abbildungsverzeichnis.....	VI
Tabellenverzeichnis.....	VI
1 Einleitung.....	1
2 Theoretische Grundlagen.....	2
2.1 Entwicklung der Topographischen Karten Bayerns.....	2
2.2 KI und Neuronale Netzwerke.....	3
2.3 Convolutional Neural Networks.....	5
3 Vorbereitung.....	6
3.1 Software und Hardware.....	6
3.2 Datengrundlage.....	7
4 Erstellung der Trainingsdaten.....	8
4.1 Datenmodellierung im ATKIS-Basis-DLM.....	8
4.2 Analyse der Kartengrundlage 1983.....	9
4.3 Zusammenstellung der Grundsätze für die Digitalisierung.....	10
4.4 Vorgehensweise in ArcGIS Pro.....	12
5 Methodik.....	14
5.1 Transfer Learning.....	14
5.2 Modelle.....	15
5.2.1 Vortrainierte Open-Source-Modelle.....	16
5.2.2 Modelle zur Objektsegmentierung.....	18
5.3 Genauigkeitsbewertung von Deep-Learning-Modellen.....	19
5.4 Einflussgrößen auf die Genauigkeit.....	20
5.5 Trainingswerkzeug in ArcGIS Pro.....	22
5.6 Methoden zur Genauigkeitsbeurteilung.....	24

7	Ergebnisse .....	25
7.1	Performance auf bekannten Rasterdaten .....	26
7.2	Performance auf ungesehenen Rasterdaten .....	29
7.3	Nachverarbeitung .....	35
7.4	Transferstudie.....	36
8	Diskussion.....	37
9	Fazit .....	40
	Literaturverzeichnis.....	VII
	Anlagen.....	X

## Abbildungsverzeichnis

Abb. 1 Ausschnitt der TK25 von links: 1941, 1983, 2001 im Maßstab 1:25000.....	2
Abb. 2 Architektur eines Künstlichen Neuronales Netzes.....	3
Abb. 3 Vereinfachte Darstellung der Funktionsweise eines Backpropagation-Netzwerks.....	4
Abb. 4 Funktionsweise der Convolution Operation mit einem 3x3 Kernel .....	5
Abb. 5 Ausschnitt aus dem Kartenblatt 7146 der Topographischen Karte von 1983.....	9
Abb. 6 Auszug aus der Legende der TK von 1983 .....	10
Abb. 7 Auszug aufs dem ATKIS-Objektartenkatalog für die Objektart Ortslage.....	10
Abb. 8 Digitalisierte Trainingsgebiete (ROT) im Maßstab 1:15 000.....	12
Abb. 9 Beispiel Funktionsweise Transfer Learning.....	14
Abb. 10 Berechnungsformel Panoptic Quality .....	19
Abb. 11 Kombinationsmöglichkeiten der verwendeten Datenformate, Modelltypen und Backbones in ArcGIS Pro .....	23
Abb. 12 Beispielaufbau einer Konfusionsmatrix für 2 Klassen.....	25
Abb. 13 Erkannte Siedlungsflächen im Westen der Ortschaft Grafenau.....	27
Abb. 14 Ausschnitt Digitalisierter Siedlungsflächen im Testgebiet.....	29
Abb. 15 „False Positives“ von links nach rechts: DarkNet-53, ResNet-18 .....	31
Abb. 16 Erkannte Siedlungsgebiete im Testgebiet im Maßstab 1:20 000 .....	33
Abb. 17 Erkannte Siedlungsgebiete auf dem Testgebiet mit Detailansicht.....	34
Abb. 18 Detailansichten 1:10 000 von links nach rechts: DarkNet-53, ResNet-34, VGG-16 ..	34
Abb. 19 Erkannte Siedlungsflächen nach Nachverarbeitung.....	35
Abb. 20 Erkannte Siedlungsflächen von links nach rechts: 1972, 1983, 1992.....	36
Abb. 21 Differenzflächen: SOLL – Segmentierte Siedlungsflächen mit ResNet-34, U-Net .....	39

## Tabellenverzeichnis

Tabelle 1 Verfügbare Backbone-Modelle in ArcGIS Pro .....	15
Tabelle 2 Begriffsübersicht möglicher Ereignisse .....	19
Tabelle 3 Parameter des Werkzeugs "Deep-Learning-Modell trainieren" .....	22
Tabelle 4 Trainierte Modelle und zugehörige Genauigkeitsangaben für das Testgebiet .....	30
Tabelle 5 Die 7 "besten" Modelle, nach OA absteigend sortiert .....	32
Tabelle 6 Erkannte Siedlungsflächen auf dem Testgebiet im Jahresvergleich .....	37

## 1 Einleitung

Gesichtserkennung, Smart-Home, ChatGPT - Künstliche Intelligenzen werden in vielen Lebensbereichen immer präsenter. Die teilweise kontrovers diskutierten Innovationen erleichtern aber nicht nur den Alltag, sondern können enorm arbeitsaufwendige Aufgaben für den Menschen übernehmen und damit vorher undenkbar Projekte ermöglichen. Ein großes Projekt begannen die Landesämter für Vermessung Anfang der 90er Jahre mit der schrittweisen Umstellung auf digitale Datenbestände. Allein für die ATKIS-Datenbank wurden von der Erstellung und kontinuierlichen Anpassung des Landschaftsmodells bis zur abgeschlossenen Digitalisierung der Topographischen Karten im Maßstab 1:25 000 etwa 5 Jahre in Anspruch genommen (AdV, 2019). Der hohe Arbeits- und Zeitaufwand sorgte dafür, dass eine Digitalisierung früherer Zeitbestände bisher ausblieb. Historische Karten beinhalten jedoch eine Fülle an Informationen, die nicht nur zum Verständnis vergangener Entwicklungen herangezogen werden können, sondern in Zusammenhang mit aktuellen Daten auch Prognosen ermöglichen. Eine automatisierte Extraktion von Gebäuden in mittelmaßstäbigen Topographischen Karten haben Hecht et al. im städtischen Gebiet erfolgreich mit dem Programm SEMENTA umsetzen können (Hecht et al., 2008; Meinel et al., 2014). Das allein lässt allerdings kaum weitere Aussagen als über den veränderten Gebäudebestand zu. Insbesondere für nachhaltige Ziele wie die Erhaltung der Ökosysteme ist eine Überwachung des Siedlungs- und Infrastrukturausbaus interessant. Die beanspruchte Fläche neuer Siedlungsgebiete in der Landschaft geht weit über die reine Grundfläche der Gebäude hinaus. Eine Abgrenzung der gesamten Siedlungsfläche wie in aktuellen Datenbeständen ist in den historischen Karten kaum vorhanden. Im Vergleich zu den markant symbolisierten Gebäuden sind sie daher schwierig zu extrahieren. Aktuelle GIS-Software ermöglicht es, auch ohne Programmierkenntnisse, das Potenzial Künstlicher Intelligenz abzugreifen. Diese Möglichkeit soll im Zuge dieser Arbeit genutzt werden. Um Siedlungsgebiete automatisiert erkennen und segmentieren zu lassen, werden vortrainierte Künstliche Neuronale Netze auf einem historischen Kartenblatt trainiert. Es wird untersucht, inwieweit ein funktionstüchtiges Deep-Learning-Modell auf Grundlage anderer Netze und minimaler Trainingsdaten zu erreichen ist und sich für Entwicklungsanalysen eignen könnte.



## 2 Theoretische Grundlagen

### 2.1 Entwicklung der Topographischen Karten Bayerns

Topographische Karten bilden die Erdoberfläche und Geländeform in mittlerem bis großem Maßstab ab. Sie enthalten neben Höhenlinien und markanten Höhenpunkten je nach Maßstab Straßen, Bahnstrecken, Gewässer, Gebäude, Ortschaften und andere sichtbare Details der Erdoberfläche. Zahlreiche historische Kartenblätter, die die räumliche Entwicklung Bayerns dokumentieren, werden in der Bayerischen Staatsbibliothek aufbewahrt (LDBV, 2022).

Mit Gründung des „Statistisch-Topographischen Bureaus“ im Jahr 1801 begann die systematische Vermessung Bayerns. Die Landaufnahmen im Maßstab 1:2 500 bzw. 1:5 000 bildeten die Grundlage für den sogenannten Topographischen Atlas, der eine erste flächendeckende Topographische Karte Bayerns in einem Maßstab von 1:50 000 darstellte (Bavarikon, o. J.). In den Jahren 1817 bis 1841 entstanden die ersten handgezeichneten Positionsblätter im Maßstab 1:25 000, die ab 1872 vervielfältigt wurden (LDBV, 2017). Die Topographischen Karten Bayerns entwickelten sich bezüglich ihrer Symbolisierung ständig weiter. Die anfangs einfarbig gedruckten Karten wurden 1882 um eine zweite, braune Farbe für die Höhenlinien ergänzt und erhielten ab 1901 schließlich eine dritte, blaue Farbe für das Gewässernetz. Die Erstellung der Topographischen Karten zog sich über Jahrzehnte hinweg. Erst ab 1960 ist die dreifarbige Ausgabe flächendeckend verfügbar, nur für wenige Regionen Bayerns bereits für Anfang des 20. Jahrhunderts (LDBV, 2022). Die moderne TK25 wird im amtlichen Geoinformationssystem ATKIS fortlaufend aktualisiert, ist digital verfügbar und zeichnet sich durch eine stark mehrfarbige Symbolisierung aus. Die Entwicklung der Kartendarstellungen wird beispielhaft in Abb. 1 veranschaulicht.

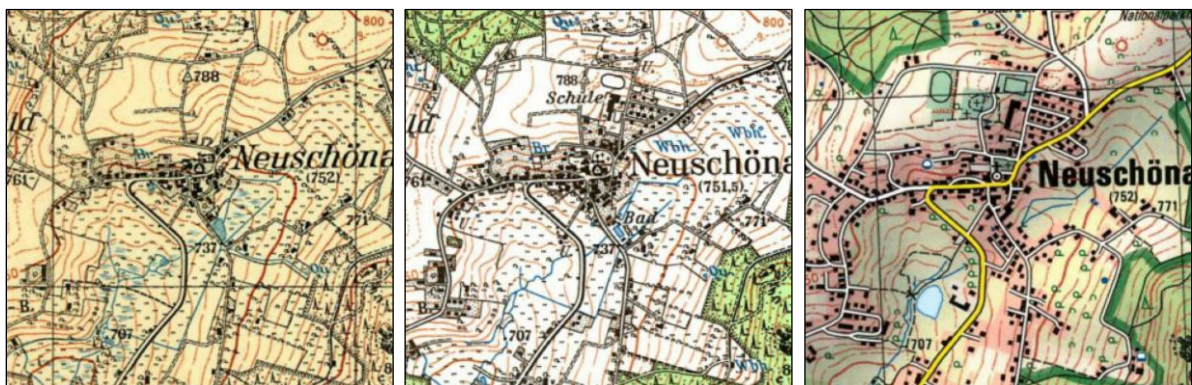


Abb. 1 Ausschnitt der TK25 von links: 1941, 1983, 2001 im Maßstab 1:25000  
(Quelle: Bayern Atlas, Zeitreihe TK, <https://v.bayern.de/sjqqb>)

## 2.2 KI und Neuronale Netzwerke

Die Künstliche Intelligenz (KI) ist ein Teilgebiet der Informatik, das sich damit beschäftigt, menschliche Intelligenzprozesse mithilfe von Maschinen zu simulieren und zu automatisieren. Der Begriff Neuronales Netz kommt ursprünglich aus den Neurowissenschaften und beschreibt dort die Vernetzung beliebig vieler biologischer Neuronen im menschlichen Nervensystem. Ein künstliches neuronales Netzwerk (KNN) nutzt künstliche Neuronen, um die Nervenzellen im Gehirn mithilfe von Algorithmen nachzuahmen. Es besteht grundsätzlich aus drei Schichten: der Eingabeschicht (*Input Layer*), den Versteckten Schichten (*Hidden Layers*) und einer Ausgabeschicht (*Output Layer*). Grundlage der KNN sind Künstliche Neuronen, die die Eingabedaten (z.B. ein Bild) verarbeiten, gewichten und summieren. Im Input Layer werden den Neuronen die Werte zugeordnet, die das Netz erhalten hat. Jedes Neuron gibt einen Wert zurück, der durch gewichtete Verbindungen mit einem weiteren Layer Neuronen verknüpft wird.

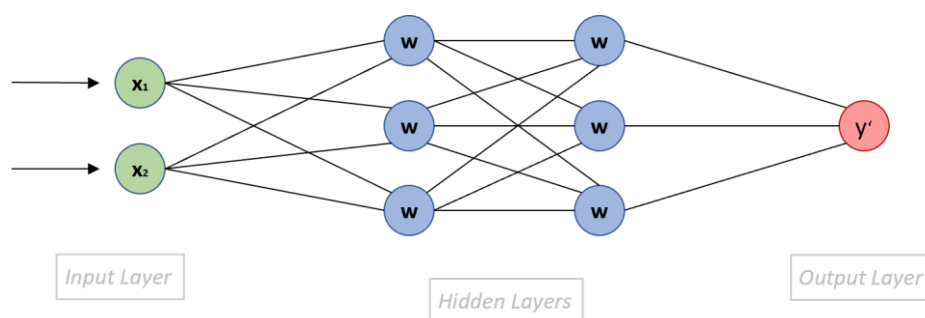


Abb. 2 Architektur eines Künstlichen Neuronales Netzes  
(Quelle: angelehnt an Kalirane, 2023)

Die Schichten von Neuronen, die die Ausgabe ihrer vorhergehenden Layer verarbeiten, werden „Hidden Layers“ genannt. Die genauen Entscheidungsvorgänge darin sind nicht nachvollziehbar. Im Regelfall ergibt sich der Wert eines Neurons aus der gewichteten Summe der Inputwerte (seiner Vorgängerschicht), einer „Activation Function“ und einem addierten Bias. Die Aktivierungsfunktion dient dazu, den Ausgabewert des Neurons in einen bestimmten Wertebereich einzugrenzen oder anzupassen. Eine erprobte Aktivierungsfunktion ist *Rectified Linear Units* (ReLU), die alle negativen Werte aus der Eingabe auf NULL setzt und alle positiven Werte unverändert beibehält (Bhattacharyya et al., 2017). Die Funktion hat nur einen geringen Rechenaufwand und beschleunigt den Lernprozess.

Die Gewichte müssen während des Trainings erlernt werden. Im Deep Learning kommt hierfür die sogenannte *Backpropagation* zum Einsatz. Die Werte der Gewichte und Bias werden zu Beginn völlig zufällig gewählt (*Random Initialization*), das bedeutet das Netzwerk führt nahezu garantiert zu einem falschen Ergebnis. Dieses „falsche“ Ergebnis wird mit einer Kostenfunktion (*Cost Function*) beschrieben, die z.B. die Summe der quadrierten Differenzen (*Mean Squared Error*) zwischen dem vorhergesagten Ergebnis und dem real richtigen Ergebnis bildet. Für die Suche nach dem globalen Minimum der Kostenfunktion wird das Gradientenabstiegsverfahren (*Stochastic Gradient Descent*) verwendet.

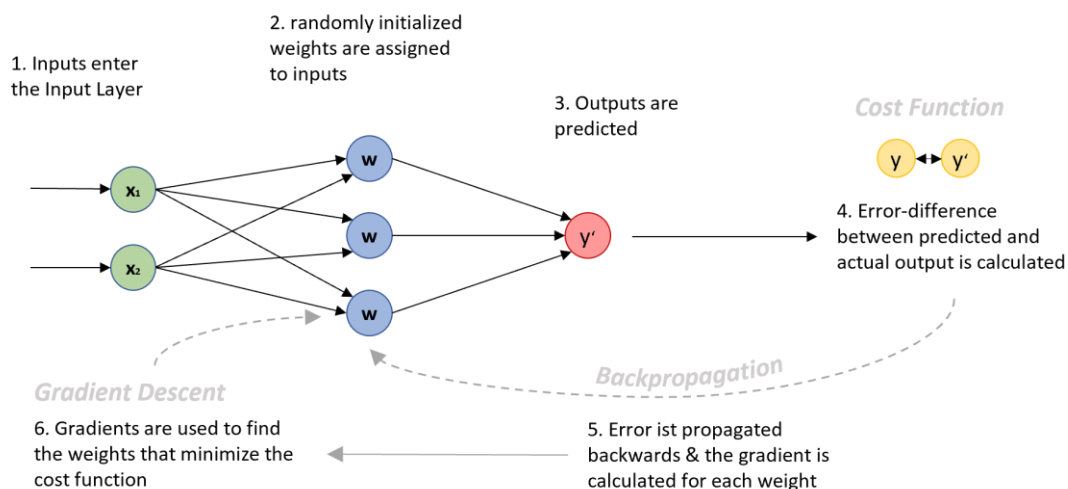


Abb. 3 Vereinfachte Darstellung der Funktionsweise eines Backpropagation-Netzwerks  
(Quelle: angelehnt an Kalirane, 2023)

Im Output Layer sind genau so viele Neuronen enthalten, wie das Netzwerk verschiedene Kategorien erkennen kann (z.B. Anzahl unterschiedlicher Objektklassen in einem Bild). Die Kategorie mit dem höchsten Wert wird dem Input zugeordnet (Voithofer, 2019).

Netze mit mehr als einer verborgenen Schicht (Hidden Layer) werden als tiefe neuronale Netzwerke bezeichnet (Deep Neural Networks). Die Methode, tiefe neuronale Netze zu verwenden, um eine Aufgabenstellung anzulernen, wird als Deep Learning bezeichnet.

Die Anwendungen von Deep-Learning-Algorithmen reichen von der Sprach- und Gesichtserkennung, über Chatbots im Kundenservice und die medizinische Forschung bis hin zur Entwicklung automatisierten Fahrens (Thompson et al., 2022; Bhatt et al., 2021; Ronneberger et al., 2015; Cui et al., 2017; Grigorescu et al., 2020).

## 2.3 Convolutional Neural Networks

Convolutional Neural Networks (kurz CNN, oder ConvNets) sind eine Art von KNN, die auf die Analyse von Bildern spezialisiert ist. Als Ursprung gilt eine bereits 1989 von Le Cun et al. vorgestellte Netzwerkarchitektur, die zur Erkennung von handgeschriebenen Ziffern entwickelt wurde. „Convolution“, zu deutsch Faltung, wird auf Bilder angewandt, um Features zu erkennen. Die Convolution-Operation verwendet einen Filter, den sogenannten „Kernel“, dessen Werte er systematisch mit allen Stellen des Inputrasters abgleicht. Die Werte des Kernels werden für jede Position mit denen des Ausschnitts aus dem Bild multipliziert und anschließend summiert.

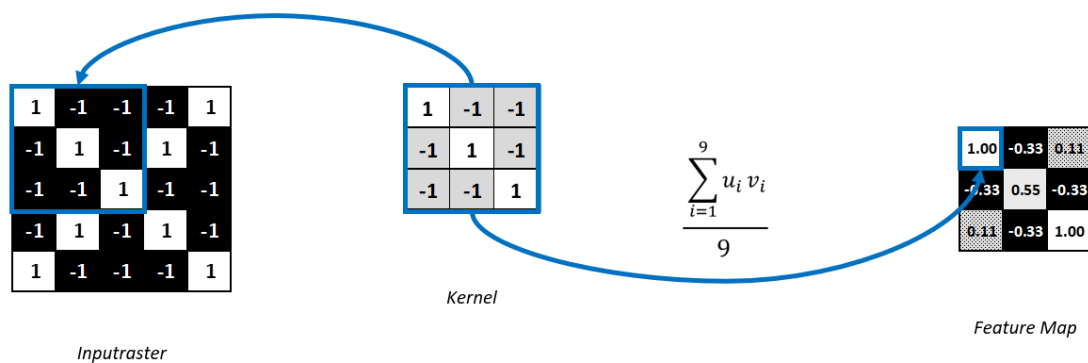


Abb. 4 Funktionsweise der Convolution Operation mit einem 3x3 Kernel  
(Quelle: eigene Darstellung, angelehnt an: <https://serokell.io/blog/introduction-to-convolutional-neural-networks>)

Ergebnis der Convolution-Operation auf einem Imagechip ist die „Feature Map“, ein speziell gemustertes Raster, das auf geometrische Formen und Merkmale hinweist (Aloysius & Madathilkulangara, 2017). Der Wert von 1 im linken oberen Eckpunkt der Feature Map in Abb. 4 bedeutet eine hundertprozentige Übereinstimmung des Filters mit dem Eingabebild an dieser Stelle. Der Kernel ist damit ein Detektor für ein oder mehrere bestimmte Features in einem Bild. Das Verfahren wird mit verschiedenen Kernels auf dem Inputraster angewandt, sodass unterschiedlichste Merkmale erkannt werden können. Die benötigten Werte des Kernels werden im Trainingsvorgang durch das Netzwerk erlernt.

Ein CNN besteht in der Regel nicht nur aus mehreren Convolution Layern, sondern auch aus sogenannten Pooling Layern. Ein Pooling Layer dient der Verringerung der Bildgröße und damit dem Beschleunigen des Trainingsprozesses. Das Eingabebild wird bspw. mit einer 2x2 Matrix systematisch abgeschritten und alle Werte darin zu Einem zusammengefasst. Typische Pooling-Rechenoperationen sind *Mean Pooling*, wobei der Mittelwert aller enthaltenen Werte berechnet wird, oder *Max Pooling*, die den höchsten Wert aus der Matrix übernimmt (Voithofer, 2019).

### **3 Vorbereitung**

#### **3.1 Software und Hardware**

In dieser Arbeit wird die Software ArcGIS Pro 3.0 des Herstellers Esri verwendet. Die Desktop-Anwendung bietet umfangreiche Funktionen zum Arbeiten mit Geoinformationen, unterstützt das Analysieren, Verwalten und Visualisieren von Geodaten. Über die Hochschule können die Software und mehrere Lizenzerweiterungen von den Studierenden bezogen werden. Mit der Image-Analyst-Lizenz steht das Toolset „Deep Learning“ zur Verfügung, mit dem sowohl eigene Modelle trainiert als auch vortrainierte Modelle für die eigene Problemstellung erweitert werden können.

Die verwendete Grafikkarte ist eine NVIDIA GeForce RTX 3060 mit 3584 CUDA Cores und 12 GB VRAM (Typ GDDR6) Standard-Speicherkonfiguration.

Für eine volle Funktionsfähigkeit der Deep-Learning-Werkzeuge müssen die Deep Learning Bibliotheken für ArcGIS (Deep Learning Frameworks) installiert sein. Das Paket enthält Komponenten wie PyTorch, Tensorflow und Fast.ai. Für die Kommunikation zwischen den Deep-Learning-Werkzeugen und der Grafikkarte wird das Nvidia CUDA Toolkit benötigt, das auf der Webseite des Herstellers kostenlos zum Download bereitsteht. Zur Überwachung der GPU-Auslastung wurde zusätzlich eine einfache Open-Source-Software installiert.

## 3.2 Datengrundlage

Grundlage für die Untersuchung der Deep Learning Algorithmen bilden ausgewählte Topographische Karten aus dem Jahr 1983 des Landkreises Freyung-Grafenau in Bayern. Unmittelbar angrenzend an den Nationalpark Bayerischer Wald handelt es sich um ein für eine Entwicklungsanalyse interessantes Gebiet.

Die Topographische Karte im Maßstab 1:25 000 ist die im größten Maßstab durchgängig verfügbare Topographische Karte und bildet als einzige einzelne Gebäude und kleinere Wege ab. Durch ihr Detailreichtum und die jahrzehntelange Fortführung eignet sie sich gut für Entwicklungsanalysen.

Das Landesamt für Digitalisierung, Breitband und Vermessung (LDBV) stellt für die Arbeit die zwei Kartenblätter 7046 und 7146 zur Verfügung. Sie wurden über einen Link zum Download im TIF-Format bereitgestellt und können mit einem beigelegten „World-File“ georeferenziert werden. Bei einem World-File handelt es sich um eine Textdatei mit der Dateierweiterung .tfw, die Angaben zur Pixelgröße sowie die x- und y-Koordinate des obersten linken Bildpunkts enthält. Gleichzeitig mit der TIF-Datei in ArcGIS Pro geladen, wird das World-File dem Raster automatisch zugeordnet und das Bild im Koordinatensystem UTM 32 (EPSG 25832) referenziert.

## 4 Erstellung der Trainingsdaten

Die Erstellung der Trainingsdaten wird an einem der zur Verfügung gestellten Kartenblätter (7146) vorgenommen und erstreckt sich über ein Gebiet von etwa 136 km<sup>2</sup>.

Es wurden Grundsätze aufgestellt, an die sich während der manuellen Digitalisierung gehalten werden musste. Die Erstellung dieser Regeln basiert auf der aktuellen Datenmodellierung im ATKIS-Basis-DLM und dem in der historischen Karte zur Verfügung stehenden Informationsgehalt.

### 4.1 Datenmodellierung im ATKIS-Basis-DLM

Vor dem Hintergrund die im Ergebnis detektierten Flächen für eine Siedlungsentwicklungsanalyse nutzbar zu machen, wird bei der Erstellung berücksichtigt, wie mögliche Vergleichsdaten aus anderen Epochen aussehen. Eine Siedlungsfläche als eigene Objektart findet sich im Bayernatlas erstmals im Jahr 2001.

Die modernen Topographischen Karten werden aus dem bundesweit einheitlichem ATKIS-Basis-DLM abgeleitet. Diese Daten können für den aktuellen Zeitstand in Form von Vektordaten (shp) käuflich erworben werden. Ein Beispieldatensatz steht zum Download frei zur Verfügung. Im ATKIS-Objektartenkatalog für das Digitale Basis Landschaftsmodell (bereitgestellt von der Arbeitsgemeinschaft der Vermessungsverwaltungen der Länder der Bundesrepublik Deutschland (Adv)) sind alle Objektarten inklusive ihrer Bildungsregeln, Erfassungskriterien, Attributart und weiterer Festlegungen enthalten. Fast alle Objektarten der Gruppe Siedlung wie beispielsweise Wohngebäude sind „vollzählig“ zu erfassen. Das heißt sie sind unabhängig ihrer Größe als solche Objektart zu modellieren und zu kennzeichnen.

Das bedeutet zwar Vollständigkeit, aber auch eine kleinteilige und aufwendige Digitalisierung. Abhilfe schafft die weitere im ATKIS enthaltene Objektart ‚Ortslage‘ mit der Kennnummer 52001, die für Karten im größeren Maßstab gedacht ist.

*„Die Ortslage enthält neben 'Wohnbaufläche', 'Industrie- und Gewerbefläche', 'Fläche gemischter Nutzung', 'Fläche besonderer funktionaler Prägung' auch die dazu in einem engen räumlichen und funktionalen Zusammenhang stehenden Flächen des Verkehrs, von Gewässern, von Flächen, die von 'Bauwerke und sonstige Einrichtungen' für Erholung, Sport und Freizeit belegt sind, sowie von 'Vegetationsflächen'.“*

– Adv, 2008

Bei der Digitalisierung des Datenbestandes für das ATKIS wurden bundesweit topographische Karten unterschiedlicher Maßstäbe herangezogen. Das Ausmaß der Siedlungs- und Ortslagenflächen ist daher dennoch nicht einheitlich. Die Lagegenauigkeit im ATKIS-Basis-DLM beträgt laut AdV  $\pm 3$  m für wesentliche lineare Objekte wie zu modellierende Straßen, schienengebundene Verkehrswege, oberflächliche Gewässer und topologische Knoten im Straßen- und Verkehrsnetz. Die übrigen Objekte sind mit einer Lagegenauigkeit von  $\pm 15$  m angegeben (AdV, 2019).

#### 4.2 Analyse der Kartengrundlage 1983

Bei einer Pixelbreite von 2,12 m in der Topographischen Karte von 1983 könnte eine Genauigkeit  $\pm 15$  m für die Siedlungsflächen theoretisch einhaltbar sein. Problematisch ist allerdings die Zuweisung der Grenzen, da die Symbolisierung in der 1983er Karte noch deutlich weniger umfangreich ausfällt. Abbildung 4 zeigt den Ort Harschetsreuth in der Topographischen Karte von 1983.



Abb. 5 Ausschnitt aus dem Kartenblatt 7146 der Topographischen Karte von 1983  
(Quelle: eigene Darstellung, ArcGIS Pro)

Eine Abgrenzung des Siedlungsgebiets fehlt zu diesem Zeitpunkt noch und muss daher manuell nach eigens festgelegten Kriterien erfolgen. Ortschaften sind hier durch eng beieinander liegende Gebäude und das Flächenobjekt Garten geprägt. Ein Ausschnitt aus der zugehörigen Legende zeigt einen Teil vorhandener Flächen- und Linienobjekte (Abb. 6).



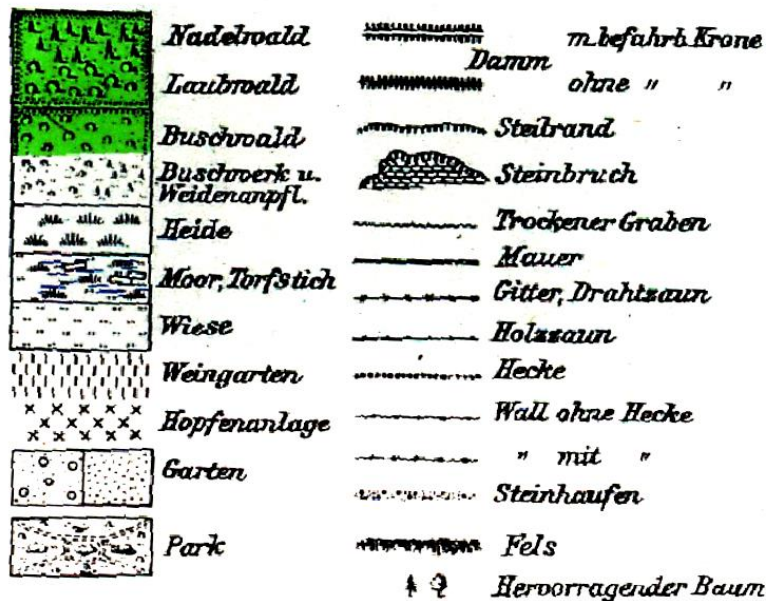


Abb. 6 Auszug aus der Legende der TK von 1983  
(Quelle: Kartenrahmen TK25 1983, LDBV)

Die Festlegung der Außengrenzen für die Siedlungsgebiete kann aufgrund der fehlenden Detailschärfe nur angelehnt an die im ATKIS Objektartenkatalog enthaltenen Definitionen erfolgen.

#### 4.3 Zusammenstellung der Grundsätze für die Digitalisierung

Alle in der TK von 1983 enthaltenen, den Objektarten der Gruppe Siedlung (ATKIS) zuordenbaren Flächen werden bei der Digitalisierung als Siedlungsfläche aufgenommen. Die Erfassung der Vollzähligkeit gestaltet sich schwierig, da aus der Karte nicht ersichtlich ist, ob es sich um Wohngebäude handelt oder nicht. Einzelgebäude oder kleinere Gebäudegruppen im Offenland könnten sowohl kleinere Siedlungen darstellen als auch Scheunen oder Lagerräume. Eine Hinzunahme der georeferenzierten Gebäudeadresse (Hauskoordinate), wie bei der aktuellen Erfassung von Wohngebäuden im ATKIS, ist für den damaligen Zeitstand nicht möglich. Um die Vergleichbarkeit unterschiedlicher Zeitstände zu gewährleisten, sollte die Beurteilung nicht nur nach rein eigenem Ermessen des Digitalisierenden erfolgen. Die Objektart ‚Ortstlage‘ bietet mit ihren Erfassungskriterien einen guten Richtwert.

##### Objektartengruppe: Besondere Anlagen auf Siedlungsflächen

Kennung	Name	Erfassungskriterien
52001	Ortstlage	Fläche >= 10 ha oder ab 10 Anwesen

Abb. 7 Auszug aus dem ATKIS-Objektartenkatalog für die Objektart Ortstlage  
(Quelle: AdV, 2008)

Dennoch sollte das Ziel der Vollzähligkeit angestrebt werden. Daher werden auch Gebäudegruppen kleiner 10 Anwesen digitalisiert, wenn sie sich in einen engeren räumlichen Zusammenhang zu anderen Ortschaften bringen lassen. Andere alleinstehende Gebäude, insbesondere jene, die sich nur an Feld- und Waldwegen finden, werden von der Digitalisierung ausgeschlossen.

In Anlehnung an die Modellierung im ATKIS-Beispieldatensatz ergeben sich folgende Grundsätze für die Digitalisierung:

- ✓ Gebäude müssen vollständig innerhalb der Siedlungsflächen liegen
- ✓ Innerörtliche Straßen werden überdeckt, angrenzende Straßen zur Hälfte ihrer Breite ausgefüllt
- ✓ Gärten, Freizeitanlagen, Sportanlagen, Gewerbegebiete, Ortskerne sind Teil der Siedlungsfläche
- ✓ Kleinere, vollständig von Gebäuden umschlossene Flächen zählen zur Siedlungsfläche
- ✓ Kleinere, außerörtliche Gebäudegruppen werden nur erfasst, wenn sie mindestens 10 Gebäude umfassen, oder
  - ✓ sich weniger als 300 m von einer eindeutigen Siedlung entfernt befinden und an diese durch eine Straße angebunden sind
  - ✓ sie durch ihre Größe, Straßenanbindung und umliegende Gartenfläche eine Nutzung als Wohngebäude vermuten lassen

Eine uneingeschränkte Vergleichbarkeit zum aktuellen ATKIS-Datenbestand kann auf der Datengrundlage nicht hergestellt werden. Einzelne Gebäude, die im aktuellen Bestand unter Hinzunahme der Hauskoordinate womöglich als Wohngebäude erfasst sind, können aus der Karte von 1983 nicht als solche digitalisiert werden, auch wenn sie bereits existierten. Sollten die Ergebnisdaten für eine Entwicklungsanalyse einsetzbar sein, muss diese Tatsache berücksichtigt werden und kleine Gebäudegruppen sollten vom Vergleich ausgeschlossen werden. Eine Siedlungsentwicklungsanalyse befasst sich in der Regel jedoch mit der Entwicklung neuer Wohngebiete, der Entstehung von Gewerbegebieten und der Verdichtung der Wohnanlagen. Diese Kriterien sind mit den festgelegten Grundsätzen erfüllbar.

#### 4.4 Vorgehensweise in ArcGIS Pro

In den Klassifizierungswerkzeugen von ArcGIS Pro kann der Trainingsgebiet-Manager aufgerufen werden, in dem Trainingsgebiete manuell erstellt werden. Dafür wurde ein neues Schema mit einer Klasse Ortslage angelegt und die Siedlungsflächen als Polygone digitalisiert.

An dieser Stelle sei erwähnt, dass sich die in Kapitel 4.3 zusammengestellten Grundsätze vor allem auf die Vergleichbarkeit zu anderen Daten konzentrierten und für das entfernte Ziel einer Siedlungsentwicklungsanalyse gedacht sind. Im Laufe des Modelltrainings (Kapitel 5.5) stellte sich heraus, dass eine derart differenzierte Klassifizierung der Objekte nicht möglich ist. Da diese Arbeit primär zum Ziel hat, Siedlungsgebiete zu erkennen, wurde ein zweiter Trainingsdatensatz erstellt, der zusätzlich alle Gebäudegruppen ab 2 Gebäuden unabhängig ihrer Lage involviert. Die Vergleichbarkeit der Daten für eine Entwicklungsanalyse kann auch nach der Objekterkennung hergestellt werden - etwa durch eine Bereinigung von Objekten bis zu einer bestimmten Flächengröße. Für die Beschriftung der Topographischen Karte ergeben sich daher zwei Polygonlayer: der eine enthält nur Objekte nach den in Kapitel 4.3 festgelegten Grundsätzen, der andere enthält alle Gebäudegruppen ab einer Anzahl von 2 Gebäuden.

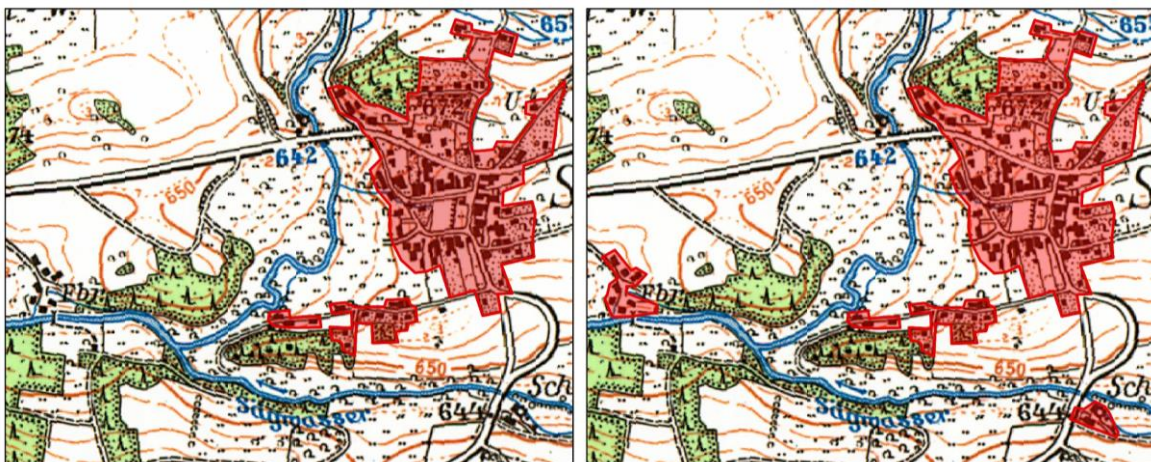


Abb. 8 Digitalisierte Trainingsgebiete (ROT) im Maßstab 1:15 000  
links: nach Grundsätzen gemäß Seite 11; rechts: alle Gebäudegruppen außer Einzelgebäuden  
(Quelle: eigene Darstellung, ArcGIS Pro)

Insgesamt wurden 187 bzw. 256 Objekte erstellt. Mit der Image Analyst- und Spatial Analyst-Lizenz können die Trainingsdaten mit einem Werkzeug konkret für Deep-Learning-Aufgaben exportiert werden.

Die Kachelgröße in den Exporteinstellungen ist standardmäßig auf 256 x 256 Pixel festgelegt, die Schrittweite auf 128, womit eine Überlappung von genau 50% gewährleistet wird. Das Metadatenformat gibt an, wie die ausgegebenen Bildschnipsel aussehen sollen bzw. beschriftet werden. Hier muss ein Format gewählt werden, das Segmentierungsmasken für jede Instanz der Klasse erstellt. Für diesen Zweck stehen RCNN-Masken, Klassifizierte Kacheln und die Panoptische Segmentierung zur Auswahl. Der Drehwinkel für die Bildschnipsel beträgt standardmäßig 0 Grad. Wird ein anderer Winkel in diesem Feld angegeben, dann werden nach der Erstellung der normal ausgerichteten Bildschnipsel zusätzliche Daten generiert, indem die Bildschnipsel um den angegebenen Winkel verdreht werden. Dieselben Trainingsgebiete können damit mehrfach in unterschiedlich verdrehten Bildschnipseln gespeichert werden.

Für den ersten Export wurden alle Einstellungen bis auf den Drehwinkel auf Standard belassen. Bei einer wie in diesem Fall so geringen Menge an beschrifteten Daten, wird eine Vervielfältigung in jedem Fall als sinnvoll angesehen. Für das Training auf Grundlage mancher Backbone-Modelle ist ein bestimmtes Metadatenformat nötig. Im Verlauf des Modelltrainings wurden daher mehrere Trainingsdatensätze exportiert, deren Einstellungen und Ergebnisse in Anlage 2 dokumentiert sind.

## 5 Methodik

### 5.1 Transfer Learning

Der Begriff Transfer Learning beschreibt das Konzept ein bereits existierendes, gut funktionierendes Modell auf die eigene Aufgabenstellung anzupassen. Während das Trainieren eines Neuronalen Netzes von Grund auf sehr aufwendig ist, da viele eigens beschriftete Daten benötigt werden, kann diese Herausforderung beim Transfer Learning umgangen werden. Es existieren einige Open-Source-Modelle, die auf einem sehr großen Datensatz trainiert worden sind und viele Millionen Parameter enthalten. Der Ansatz besteht darin, das Wissen, das erlernt wurde, wie Gewichtungen und Merkmale, auf eine neue Aufgabenstellung zu übertragen.

Die ersten Convolution Layer in einem CNN erkennen in der Regel einfache Geometrien, mit zunehmender Tiefe des Netzes steigt die Komplexität der Merkmale. „Einfache“ Merkmale in einem Bild sind für verschiedenste Aufgaben der Bildklassifizierung und Objekterkennung gleichermaßen nutzbares Wissen. Beim Transfer Learning werden daher die Gewichte und Feature Maps aus den ersten Layern unverändert beibehalten und nur die Werte in den letzten Layern (oft für die komplexeren Merkmalerkennungen) aktualisiert. Damit erspart das Transfer Learning nicht nur Zeit bei der Trainingsdatenerstellung, sondern auch bezüglich des Rechenaufwands (Hussain et al., 2019).

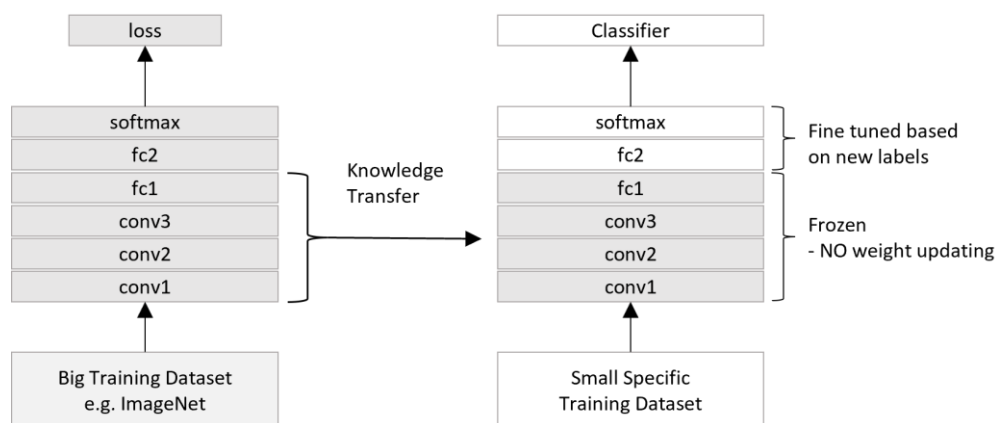


Abb. 9 Beispiel Funktionsweise Transfer Learning  
(Quelle: angelehnt an Sarkar, 2018)

## 5.2 Modelle

Im Werkzeug „Deep Learning Modell trainieren“ stehen 2 Möglichkeiten zur Verfügung, ein anderes Modell auf die eigene Aufgabenstellung anzupassen. Die Option „Vortrainiertes Modell“ ermöglicht die Verwendung einer Esri-Modelldefinitionsdatei (.emd) oder eines Deep-Learning-Pakets (.dlpk), das vom Hersteller oder anderen Nutzern in ArcGIS Online zur Verfügung gestellt wurde. Diese Modelle sind auf eine bestimmte Aufgabe spezialisiert, der zugrundeliegende Trainingsdatensatz ist teilweise nicht transparent dargestellt. Ein vortrainiertes Modell kann bei gleicher Aufgabenstellung ohne weiteres Training auf gleiche wie im Modell verwendete Rasterdaten angewendet werden.

Unter dem Auswahlfeld „Backbone-Modell“ kann aus insgesamt 21 bekannten, frei zugänglichen Modellen ausgewählt werden. Sie können mithilfe von neuen beschrifteten Daten für eine andere Aufgabenstellung nachtrainiert werden. Alle VGG-Modelle sind auch mit Batch-Normalisierung verfügbar - einer Methode, die über alle Neuronen eines Mini-Batches berechnet wird, zu stabileren Gradienten führen und ein schnelleres Training ermöglichen soll (Oppermann, 2021).

DenseNet-121	ResNet-18	VGG-11	Reid_v1
DenseNet-161	ResNet-34	VGG-13	Reid_v2
DenseNet-169	ResNet-50	VGG-16	ResNet31
DenseNet-201	ResNet-101	VGG-19	InceptionV4
MobileNet-Version 2	ResNet-152	DarkNet-53	EfficientNetB3

*Tabelle 1 Verfügbare Backbone-Modelle in ArcGIS Pro  
(Quelle: eigene Darstellung, ArcGIS Pro)*

Die wichtigsten Modelle und Begriffe werden in den folgenden Unterkapiteln kurz vorgestellt.

### 5.2.1 Vortrainierte Open-Source-Modelle

ImageNet ist eine fortlaufend erweiterte Open Source Datenbank mit mehreren Millionen beschrifteten Bildern. Das 2009 erstmals veröffentlichte Projekt ist nach der hierarchischen Struktur von WordNet organisiert, einer Datenbank, die englische Wörter in Beziehungen zueinander bringt, gruppiert und Überbegriffen zuordnet (Miller, 1995). Jedem Begriff sind mehrere Bilder zugehörig und umgekehrt – die Daten wurden durch Menschen gesammelt, beschriftet und kontrolliert. Ein Bild eines Hundes ist beispielsweise u.a. den Begriffen „mammal“ (Säugetier), „carnivore“ (Fleischfresser) und „dog“ (Hund) zugeordnet (Deng et al., 2009). ImageNet wird weitverbreitet als Quelle für eine große Menge an sorgfältig beschrifteten Bilddaten für das Training tiefer neuronaler Netzwerke verwendet.

VGG (Visual Geometry Group) ist ein auf ImageNet-Bilddaten trainiertes tiefes CNN, erstmals vorgestellt von Simonyan und Zisserman im Jahr 2014. Die bekanntesten Ausführungen sind VGG-16 und VGG-19, wobei die Ziffer für die Anzahl an Gewichteten Layern im Netzwerk steht. Die VGG-Modelle basieren auf einer Inputrastergröße von 224x224 Pixeln und nutzen kleine Kernels der Größe 3x3 für die Convolution Operation. Auf mehrere Convolution Layer folgt ein Max-Pooling-Layer. Alle Hidden Layer nutzen RELUs als Aktivierungsfunktion (Simonyan & Zisserman, 2014). Mit 138 Millionen Parametern und sehr vielen Filtern ist das VGG-16-Modell ein vergleichsweise großes und damit langsames Modell.

Ebenfalls auf dem ImageNet-Datensatz trainiert, wurde das erste Modell von ResNet 2015 durch He et al. veröffentlicht und verfolgte das Ziel, das Problem des verschwindenden Gradienten (*Vanishing Gradient Issue*) zu lösen. Das weit verbreitete Problem tritt beim Training Künstlicher Neuronaler Netze mit Backpropagation auf, wenn die Anzahl der Layer erhöht wird. Beim schrittweisen Zurückführen des Fehlers über alle vorhergehenden Layer wird der Gradient schwindend gering und damit auch sein Einfluss auf das Modell. In ResNet-Modellen werden mehrere Layer dafür in Blocks eingeteilt, die bei der Backpropagation übersprungen werden. Die Gradienten nutzen die Abkürzungsverbindungen (*Skip connections*), durchlaufen damit weniger Gewichtungsschichten und verschwinden auch über viele Layer hinweg nicht. Diese Methode sorgt auch dafür, dass die Rechenzeit für das Training solcher Modelle trotz ihrer Tiefe nicht übermäßig ansteigt (He et al., 2015).

DarkNet-53 wurde mit einer Bildgröße von 256x256 Pixeln auf ImageNet-Daten trainiert und wird häufig für Aufgabenstellungen der Objekterkennung eingesetzt. Die Architektur basiert auf dem Vorgängermodell DarkNet-19, das von Redmon & Farhadi 2016 als Grundlage für das Objekterkennungsmodell YOLOv2 entwickelt wurde. Ähnlich den VGG-Modellen werden 3x3- und 1x1-Filter verwendet (Redmon & Farhadi, 2016). Der 2018 veröffentlichte Nachfolger DarkNet-53 hebt sich durch 53 statt 19 Convolution Layer und einer Implementierung von Residualen Verbindungen von seinem Vorgänger ab (Redmon & Farhadi, 2018).

MobileNet-V2 ist eine durch Google Inc. veröffentlichte Netzwerkarchitektur, die speziell für mobile Anwendungen und Systeme eingeschränkter Leistung entwickelt wurde. Die Konstruktion des Netzes baut auf dem 2017 durch Howard et al. vorgestellten MobileNetV1 auf, das sich durch die Verwendung einer sogenannten *Depthwise Separable Convolution* (erstmalig vorgestellt durch Sifre & Mallat, 2014) auszeichnet. Dieser Ansatz unterteilt die normale Convolution-Operation in zwei Schritte, wobei zuerst ein einziger Convolution-Filter für jeden Bildkanal angewendet wird (*Depthwise Convolution*) und das zusammengesetzte Ergebnis anschließend eine festlegbare Anzahl an 1x1-Filtern durchläuft (*Pixelwise Convolution*). Das Netz erfordert dadurch deutlich weniger Rechenoperationen als andere (Howard et al., 2017). Im Gegensatz zur Version 1 nutzt MobileNetV2 einen dritten vorgeschalteten „Expansion-Layer“ und residuale Verbindungen für die Fehlerrückführung. Trotz gleichbleibend geringer Komplexität ist MobileNetV2 damit nicht nur genauer, sondern auch schneller als sein Vorgänger (Sandler et al., 2019).

Ein weiteres im Verlauf dieser Arbeit getestetes CNN ist das von Huang et al. entwickelte DenseNet, das einen ähnlichen blockbasierten Ansatz wie ResNet verfolgt, dabei aber alle Layer direkt miteinander verbindet (Huang et al., 2018).



### 5.2.2 Modelle zur Objektsegmentierung

Max-DeepLab ist ein von Wang et al. in Zusammenarbeit mit der John-Hopkins-Universität und Google Research entwickeltes Modell zur Erstellung von Segmentierungsmasken. Resultat des Modells ist die sogenannte Panoptische Segmentierung, die die Instanz-Segmentierung und die semantische Segmentierung vereint. Das bedeutet, das Modell weist jedem Pixel im Inputraster eine Klasse zu und unterscheidet zudem einzelne Instanzen einer Klasse. Während frühere Modelle die Panoptische Segmentierung aus den Ergebnissen mehrerer Zwischenschritte zusammensetzten, bezeichnen Wang et al. Max-DeepLab als erste „End-to-End“-Lösung (Wang et al., 2021).

Mask R-CNN ist ein tiefes Neuronales Netz, basierend auf Faster R-CNN, das neben Umgebungsrahmen für die Objekte auch pixelbasierte Segmentierungsmasken für die Instanzen einer Klasse erstellt. Das 2018 von He et al. vorgestellte Modell dient damit der Instanzsegmentierung (He et al., 2018).

U-Net ist ein von Ronneberger et al. ursprünglich für biomedizinische Zwecke entwickeltes Neuronales Netz zur semantischen Segmentierung. Die Architektur besteht aus einem „Contracting Path“, einer Abfolge von Convolution und Pooling Layern, die die Eingabebildgröße verringern (Downsampling), und einem „Expansive Path“, über den die Informationen schrittweise wieder auf das Originalformat vergrößert werden (Upsampling). Statt den im ersten Teil angewendeten und für CNN üblichen Pooling-Operationen werden an dieser Stelle dementsprechend umgekehrt wirkende, datenerweiternde Rechenoperationen verwendet. Besonders an U-Net sind eingebaute „Skip Connections“, die das Ergebnis der Convolution Operationen im Contracting Path dem Ergebnis im Expansive Path zufügen (Ronneberger et al., 2015).

### 5.3 Genauigkeitsbewertung von Deep-Learning-Modellen

Für das Verständnis der im Folgenden vorgestellten Genauigkeitsangaben werden vorerst die im Ergebnis möglich auftretenden Fälle schematisch dargestellt:

		Vorhergesagte Klasse durch Modell	
		Siedlung	Keine Siedlung
Ground Truth	Siedlung	<b>True positive</b>	<b>False negative</b>
	Keine Siedlung	<b>False positive</b>	<b>True negative</b>

Tabelle 2 Begriffsübersicht möglicher Ereignisse  
(Quelle: eigene Darstellung)

*Overall Accuracy* stellt das Verhältnis der korrekten Vorhersagen zu allen Vorhersagen des Modells insgesamt dar. In der Regel wird die Anzahl der *True positive* und *True Negative* Vorhersagen als Anteil an der Summe der Anzahl aller Vorhersagen ausgegeben.

*Panoptic Quality* ist eine von Kirillov et al. entwickelte Metrik, die vor allem bei der Segmentierung verwendet wird. Sie setzt sich aus der Segmentierungsqualität (*Segmentation Quality SQ*) und der Erkennungsqualität (*Recognition Quality RQ*) zusammen (Kirillov et al., 2019).

$$PQ = \frac{\sum_{(p,g) \in TP} \text{IoU}(p, g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}$$

Abb. 10 Berechnungsformel Panoptic Quality  
(Quelle: (Kirillov et al., 2019))

IoU steht für *Intersection over Union*, ein verbreitet eingesetzter Wert, der die Überlappung zwischen der Vorhersagung und der tatsächlichen Wahrheit (*Ground Truth*) beschreibt. In der Objekterkennung wird diese Überlappung aus der übereinstimmenden Fläche der Objekte umgebenden Rechtecke im Verhältnis zur Gesamtfläche beider umgebenden Rechtecke gewonnen. Bei Segmentierungsaufgaben erfolgt die Bestimmung der *Intersection over Union* anhand einer pixelweisen Berechnung. Der IoU-Wert kann für die Generierung von Genauigkeitsaussagen als Schwellwert dienen, so wird eine Vorhersage in der Objekterkennung z.B. erst ab einer Überlappung von 0.5 als *True positive* bewertet.

*Training Loss* und *Validation Loss* werden während des Lernprozesses immer wieder neu ermittelt und sind Anhaltspunkte, wie gut das Netz bis dahin auf den Trainingsdaten (*Training Loss*) beziehungsweise auf den Validierungsdaten (*Validation Loss*) funktioniert. Beide ergeben sich aus den aufgetretenen Fehlern und schwinden im Verlaufe des Trainingsvorgangs idealerweise auf ein Minimum.

Sind die Werte für *Validation Loss* auffallend größer als die des *Training Loss*, weist das auf eine sogenannte Überanpassung (Overfitting) hin. Overfitting ist das beim Trainieren Neuronaler Netze nicht selten auftretende Problem, dass das Modell zu stark auf die Trainingsdaten angepasst ist. Damit liefert es für die Trainingsdaten zwar genaue Vorhersagen, ist aber nicht in der Lage zu generalisieren und funktioniert auf neuen Daten nur schlecht. Ein Lösungsansatz könnte darin bestehen, ein weniger komplexes Modell zu wählen.

Bei etwa gleichen Werten für *Training* und *Validation Loss* ist eher von einer Unteranpassung (Underfitting) zu sprechen. Das bedeutet, das Modell kann aus den Trainingsdaten nicht hinreichend Merkmale für die Aufgabe extrahieren. Das kann beispielsweise daran liegen, dass zu wenig Trainingsgebiete vorhanden sind oder sich diese in ihren Merkmalen zu stark unterscheiden.

#### 5.4 Einflussgrößen auf die Genauigkeit

Die Netzwerkstruktur sowie das Modelltraining können durch Anpassen vieler verschiedener Parameter beeinflusst werden. Es folgt eine kurze Vorstellung der Wichtigsten, die mittels der Trainingsumgebung in ArcGIS Pro angepasst werden können.

Die Genauigkeit von Deep Learning Modellen steigt mit zunehmender **Größe des Trainingsdatensatzes**, d.h. mit einer steigenden Anzahl an beschrifteten Daten (Lei et al., 2022). Wie viele Trainingsdaten beim Transfer Learning benötigt werden, hängt von der Ähnlichkeit der eigenen Aufgabenstellung mit der des verwendeten Modells ab. Die Siedlungsgebiete in der Topographischen Karte von 1983 sind nur durch eine schwarz-weiße Farbgebung gekennzeichnet. Die zu detektierenden Merkmale wie Gebäude, Straßen und Gartenflächen zeichnen sich durch einfache Symbolik wie Punkte, Linien und Ecken aus.

Die Anzahl der Trainingsgebiete lässt sich im Umfang dieser Arbeit nicht erhöhen, es können nur Ansätze zur *Data Augmentation* getestet werden. Das bedeutet die Erstellung von zusätzlichen synthetischen Daten wie durch die Angabe eines Drehwinkels beim Export der Trainingsdaten (Kapitel 4.4).

Die **Batch-Größe**, engl. *Batch Size*, definiert die Anzahl an Stichproben, in diesem Fall beschriftete Kacheln, die gleichzeitig durch das Netzwerk geleitet werden (bevor eine Aktualisierung der Gewichte stattfindet). Bei einer Batch-Größe von 4 werden dem Netz 4 Elemente gleichzeitig für das Training zugeführt. Je größer die Batch Size, desto schneller kann eine Epoche im Regelfall durchlaufen werden, da mehrere Samples gleichzeitig verarbeitet werden (Radiuk, 2017).

Eine **Epoche** beschreibt einen vollständigen Durchlauf aller Trainingsdaten durch das Neuronale Netz (Mohd Aszemi & Panneer Selvam, 2019). Bei einem Trainingsdatensatz von 1000 Bildern und einer Batch-Größe von 4, bedeutet das 250 Iterationen, bis eine Epoche abgeschlossen ist. Mit steigender Anzahl an Epochen steigt auch die Anzahl an Wertanpassungen für die Gewichte. Das kann die Genauigkeit des Modells positiv beeinflussen, bei zu vielen Epochen aber auch zu einer Überanpassung führen (Ying, 2019).

Der Exportparameter **Kachelgröße** (*image size*) legt fest wie groß die beschrifteten Bildschnipsel sein werden, die dem Modell als Trainingsdatensatz zugeführt werden. Eine kleinere Kachelgröße erlaubt bei begrenzter Speicherkapazität der Grafikkarte die Verwendung einer höheren Batch-Größe. Eine zu starke Verkleinerung kann aber dafür sorgen, dass für die Objekterkennung relevante Merkmale durch die Reduzierung der Bildgröße verloren gehen.

Die **Lernrate** ist ein Hyperparameter, der vorgibt um welche Schrittweite die Werte bei Verwendung des Gradientenabstiegs in jedem Iterationsschritt verändert werden können. Eine sehr kleine Lernrate sorgt für lange Berechnungszeiten, bis das globale Minimum der Kostenfunktion erreicht ist. Eine zu große Lernrate kann dazu führen, dass der optimale Wert überschritten wird (Erb, 1993). Die Lernrate ist ein optionaler Parameter in den Einstellungen des Trainingswerkzeugs und wird laut Hersteller während des Trainingsprozesses ermittelt.

## 5.5 Trainingswerkzeug in ArcGIS Pro

Wie in Kapitel 5.2 erwähnt, unterstützt das Geoverarbeitungswerkzeug „Deep-Learning-Modell trainieren“ in ArcGIS Pro das Nachtrainieren von „Backbone-Modellen“. Alle weiteren zur Verfügung stehenden Parameter und Optionen sind in Tabelle 3 beschrieben.

<b>Parameter</b>	<b>Funktion</b>
<b>Eingabe-Trainingsdaten</b>	Ordner mit beschrifteten Bildschnipseln
<b>Ausgabemodell</b>	Speicherort des Ausgabeordners für das trainierte Modell
<b>Max. Epochen (optional)</b>	maximale Anzahl von Epochen, für die das Modell trainiert wird (einmal vorwärts und einmal rückwärts durch das neuronale Netz geleitet wird) Default: 20
<b>Modelltyp (optional)</b>	Modelltyp, der zum Trainieren des Deep-Learning-Modells verwendet wird
<b>Batch-Größe (optional)</b>	Anzahl der Trainingsstichproben, die gleichzeitig bearbeitet werden
<b>Modellargumente (optional)</b>	zusätzliche Deep-Learning-, Funktionsargumente wie "chip_size", "monitor"
<b>Lernrate (optional)</b>	Rate, mit der vorhandene Informationen während des gesamten Trainingsprozesses durch neu erworbene Informationen überschrieben werden. Ist kein Wert angegeben, wird die optimale Lernrate während des Trainingsprozesses aus der Lernkurve extrahiert.
<b>Backbone-Modell (optional)</b>	Vorkonfiguriertes neuronales Netzwerk, das als Architektur für das Training des neuen Modells verwendet werden soll
<b>Prozentsatz für Validierung (optional)</b>	Prozentsatz der Trainingsgebiete, der für die Überprüfung des Modells verwendet werden soll Default: 10
<b>Beenden, wenn sich das Modell nicht mehr verbessert (optional)</b>	<input checked="" type="checkbox"/> : Modelltraining wird vorzeitig beendet, wenn sich das Modell nicht mehr verbessert <input type="checkbox"/> : Modelltraining wird fortgesetzt bis Anzahl Max. Epochen erreicht wurde
<b>Modell fixieren (optional)</b>	<input checked="" type="checkbox"/> : Layer des Backbone-Modells werden fixiert - die vordefinierten Gewichtungen und Verzerrungen werden im Parameter Backbone-Modell nicht geändert <input type="checkbox"/> : Backbone-Layer werden nicht fixiert - die Gewichtungen und Verzerrungen des Parameters Backbone-Modell können geändert werden, um sie an die Trainingsgebiete anzupassen

Tabelle 3 Parameter des Werkzeugs "Deep-Learning-Modell trainieren"  
(Quelle: angelehnt an Esri, 2022; ArcGIS Pro Dokumentation)

Vom Werkzeug berechnete Werte für *Training* und *Validation Loss* sowie die Lernrate können in den Details für jede Epoche eingesehen werden. Nach Beendigung des Trainings wird je nach Modelltyp ein Wert für Panoptic Quality (Max-DeepLab) oder Overall Accuracy (Mask RCNN, U-Net) ausgegeben. Der Ausgabeordner des Modells enthält neben der Esri-Modelldefinitionsdatei und dem Deep-Learning-Paket einen Ordner „Model Characteristics“. Darin finden sich ein Graph der Verlustfunktion sowie eine PNG-Datei, die fünf verschiedene Kacheln mit den gelabelten Trainingsgebieten gegenüber den Vorhersagen des Modells darstellt.

Die zur Verfügung stehenden Modelltypen hängen von dem exportierten Metadatenformat der Trainingsdaten ab. Der verwendete Modelltyp beeinflusst wiederum die zur Auswahl stehenden Backbone-Modelle. Die Kombinationsmöglichkeiten der in dieser Arbeit verwendeten Formate und Modelle sind in Abb. 11 dargestellt.

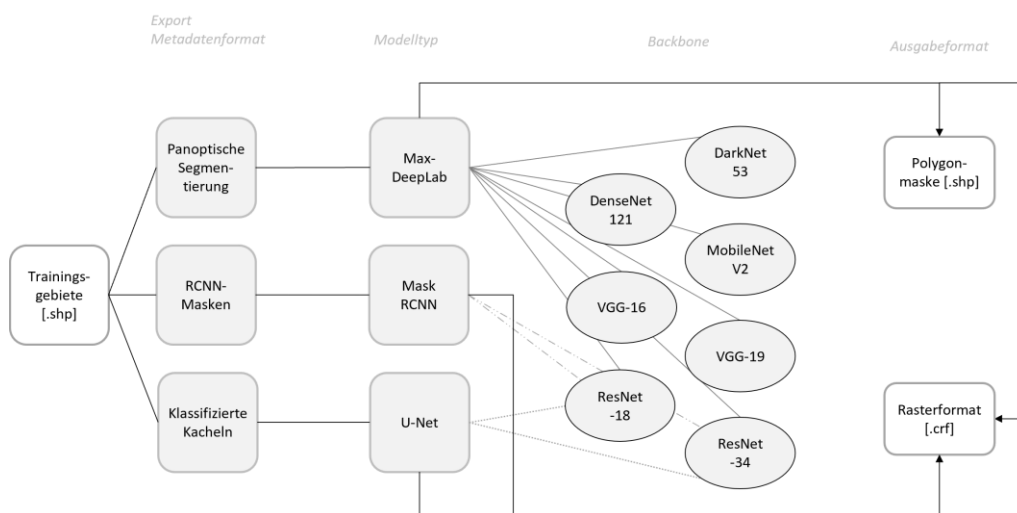


Abb. 11 Kombinationsmöglichkeiten der verwendeten Datenformate, Modelltypen und Backbones in ArcGIS Pro (Quelle: eigene Darstellung)

Insgesamt wurden 24 verschiedene Modelle trainiert. Bei der testweisen Objekterkennung mittels der ersten Modelle auf dem Kartenblatt 7146 und einer Genauigkeitsberechnung (Kapitel 5.6) stellte sich heraus, dass viele kleine Gebäudegruppen als Siedlung erkannt wurden, die im Trainingsdatensatz ausgerichtet an den Grundsätzen in Kapitel 4.3 nicht als solche enthalten sind. Das sorgte für eine hohe Anzahl an *False Positives* (siehe Kapitel 6.1), die faktisch keine sind. Da diese vermeintliche Fehldetektion wiederholt auftrat, wurde ein zweiter Trainingsdatensatz erstellt, der alle Gebäudegruppen ab 2 Gebäuden enthalten sollte.

Im Auswertungsprozess fiel auf, dass wenige zu diesem Kriterium zählende Instanzen trotz systematischer Vorgehensweise bei der Digitalisierung fehlen. Das Problem wurde damit zwar deutlich minimiert, jedoch nicht vollständig behoben. In der Dokumentation finden sich dementsprechend Modelle, die auf dem ursprünglichen Datensatz trainiert worden sind, sowie solche, deren verwendete Trainingsgebiete deutlich mehr kleinflächige Gebäudegruppen umfassen. Zur Unterscheidung wird letzterer Datensatz im Laufe der Arbeit als „standardisiert“ bezeichnet.

## 5.6 Methoden zur Genauigkeitsbeurteilung

In der Beurteilung von Klassifizierungsergebnissen ist es üblich, für die bestimmten Klassen stichprobenartig Punkte zu generieren und mit einem Attribut der zugewiesenen Klasse zu versehen. In Verbindung mit Referenzdaten kann jedem Punkt zusätzlich der wahre Wert (Ground Truth) zugewiesen und eine Fehlermatrix abgeleitet werden.

Entsprechend dieser Methode wurden für die erkannten Siedlungsgebiete zufällige Punkte erstellt, deren Anzahl proportional zur Fläche der Objekte steigt. Die Punktzahl ergibt sich aus der Objektfläche in Quadratmetern dividiert durch 1000 - jedoch mit einem Mindestabstand der Punkte untereinander von 40 Metern. Die Punkte erhielten die Attributfelder *Classified* und *GrndTruth*, wobei den Siedlungen im *Classified*-Feld ein Wert von 1 verliehen wird. Analog wurden um die erkannten Objekte Puffer in einer Distanz von 100 Metern erstellt und diese ebenfalls stichprobenartig mit Punkten versehen. Ihnen wurde der Wert 0 für das Feld *Classified* zugewiesen. Falls vorhanden wurden dem Pufferlayer vorher alle diejenigen Features der digitalisierten Trainingsgebiete zugefügt, die sich weder mit den als Siedlung klassifizierten Flächen noch mit den generierten Pufferflächen überschneiden. Zusammengeführt zu einer Punkdatei, wurde das Attributfeld *GrndTruth* mithilfe des Werkzeugs „Punkte für Genauigkeitsbewertung aktualisieren“ anhand der digitalisierten Trainingsgebiete mit einem „wahren“ Wert angepasst. Das Klassifizierungswerkzeug „Konfusionsmatrix berechnen“ errechnet aufgrund dieses Punktformats eine Tabelle, die einen Überblick über die Performance der Modelle verschafft (Abb. 12).

		Ground Truth				
		c0	c1	Gesamt	U_Accuracy	Kappa
Classified	c0	9056	4314	13370	0.68	0
	c1	778	4551	5329	0.85	0
	Gesamt	9834	8865	18699	0	0
	P_Accuracy	0.92	0.51	0	0.73	0
	Kappa	0	0	0	0	0.44

„85% der als Siedlung klassifizierten Punkte sind auch Siedlung“

„51 % der Punkte, die wahrhaft zur Klasse Siedlung gehören, wurden als solche erkannt“

Overall Accuracy  

$$= \frac{\text{Anzahl TP} + \text{TN}}{\text{Gesamt}}$$

Abb. 12 Beispielaufbau einer Konfusionsmatrix für 2 Klassen  
 (Quelle: eigene Darstellung)

Die Konfusionsmatrix zeigt die Anzahl der für jede Klasse erkannten Punkte und ordnet sie zudem ihrem wahren Wert zu. Daraus werden die Nutzer- und die Produzentengenauigkeit berechnet. Welche Aussagen aus diesen Angaben getroffen werden können, ist in Abb. 12 beispielhaft für die Klasse c1 (Siedlung) dargestellt. Aus der Punktverteilung abgeleitet ist in der Konfusionsmatrix außerdem eine Gesamtgenauigkeit (Overall Accuracy) enthalten.

Parallel zur Beurteilung über die kalkulierten Genauigkeitsangaben erfolgt eine visuelle Analyse. Dafür wurden die Modelle mittels des Werkzeugs „Objekte mit Deep Learning erkennen“ einerseits auf bekannten Bilddaten (Kartenblatt 7146) angewandt und zur Überprüfung der Funktionsfähigkeit auf ungesehenen Daten, einem kleinen Testgebiet aus dem Kartenblatt 7046, eingesetzt. Für mit Mask RCNN und U-Net trainierte Modelle musste das Werkzeug „Pixel mit Deep Learning klassifizieren“ verwendet werden. Die im Rasterformat ausgegebenen erkannten Objekte wurden in Polygone umgewandelt. Für das Testgebiet wurden ebenfalls Polygone digitalisiert, die bei der Konfusionsmatrixberechnung als Referenzdatensatz dienten.

## 7 Ergebnisse

Die Mehrheit der Modelle leistete bezüglich der *Panoptic Quality* Angaben ähnlich gute Ergebnisse und bewegte sich laut den Angaben des Trainingswerkzeuges zwischen 0.65 und 0.79. Lediglich das Training mit einem ResNet-50-Modell ergab eine OA von 0.0 und erkannte damit kein Objekt. Das VGG-11-Modell ergab zwar eine PQ von 0.6579, zeigte aber keine erkannten Objekte. Beide werden daher im Folgenden nicht weiter beschrieben.

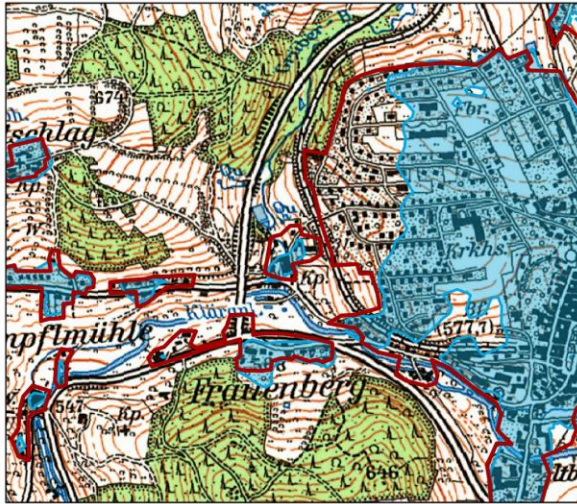


## 7.1 Performance auf bekannten Rasterdaten

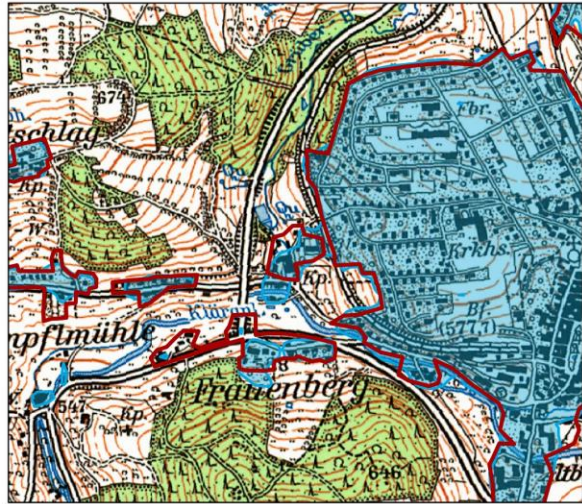
Angewandt auf das Kartenblatt 7146 zeigt das mit DenseNet-121 auf den standardisierten Trainingsdaten trainierte Modell viele, auch kleine, erkannte Siedlungsflächen. Dabei handelt es sich um zum Teil einzelne Gebäude und damit mehr als in den Trainingsgebieten definiert wurden, was sich auch in der Konfusionsmatrix durch viele *False Positives* widerspiegelt. Der visuellen Analyse nach handelt es sich bei fast allen *False Positives* um tatsächliche Gebäude, nur bei einzelnen wurde ein Busch- oder Baumsymbol detektiert. Auffällig ist, dass schon kleinste Freiflächen zwischen Siedlungselementen nicht als solche klassifiziert wurden und somit auch größere Siedlungsflächen kleinteilig detektiert sind und Lücken aufweisen. Besonders markant ist außerdem, dass im größten Ort Grafenau ein großes Wohngebiet im Westen nicht als solches erkannt wurde, obwohl es sich in der Struktur nicht zu der richtig klassifizierten Fläche des Ortes unterscheidet. Das gleiche Phänomen entstand beim Training des MobileNet-V2-Modells auf dem standardisierten Trainingsdatensatz.

Bei allen anderen auf MobileNet trainierten Modellen ist das nicht der Fall, es werden viele Siedlungen richtig erkannt und alle generierten *False Positives* sind tatsächliche Gebäudegruppen. Das mit einer maximalen Anzahl an Epochen von 150 trainierte Modell – das bereits nach 16 Epochen stoppte - schneidet laut Genauigkeitsberechnung knapp am besten ab. Bei einem Verhältnis von 1:4 der letzten *Training* und *Validation Loss* Werte ist womöglich eine leichte Überanpassung zu erwarten. Das schlechteste MobileNet-V2 ist das auf einer Batch-Größe von 8 trainierte Modell, das einige kleine Ortschaften nicht erkennt, was sich durch eine geringere Produzentengenauigkeit für die Klasse Siedlung zeigt. Das Modell, bei dem die Fixierung der Backbone-Layer aufgehoben wurde, hebt sich nicht signifikant von den anderen ab. In allen MobileNet-Modellen werden die Grenzen der Siedlungen abgerundet gezogen und fassen am Ortsrand häufiger kleine Busch- oder Baumsymbole mit ein.

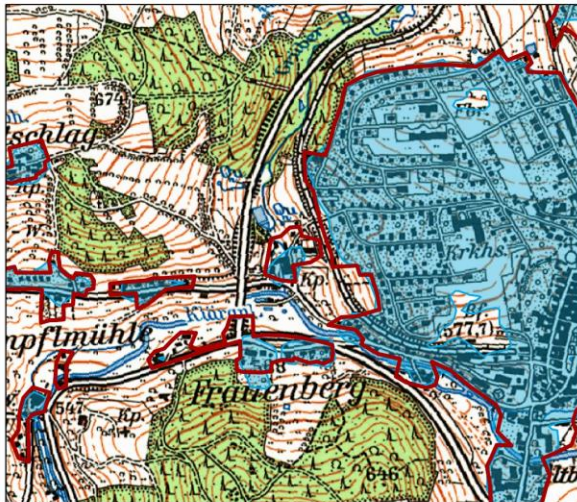
Die PQ- und OA-Angaben auf der folgenden Übersicht entsprechen den vom Trainingswerkzeug in ArcGIS Pro ausgegebenen Werten und decken sich nicht zwangsläufig mit der wahren Performance. Die genaue Berechnungsweise der Angaben ist seitens des Herstellers nicht veröffentlicht.



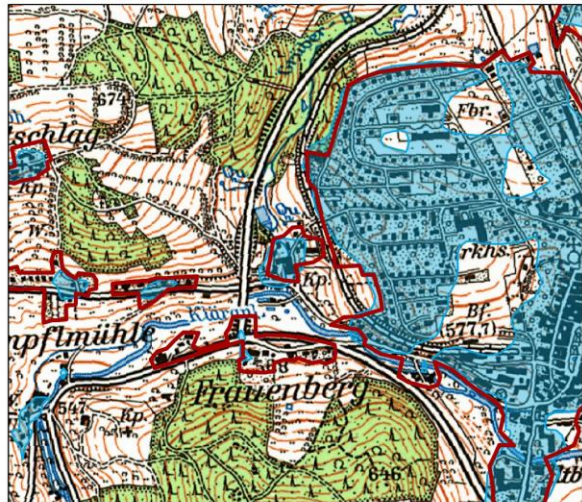
DenseNet-121: Standardisierter Trainingsdatensatz, Kachelgröße: 256x256, Batch-Size: 16, Modelltyp: Max-DeepLab, Max. 30 Epochen - Panoptic Quality: 0.7623



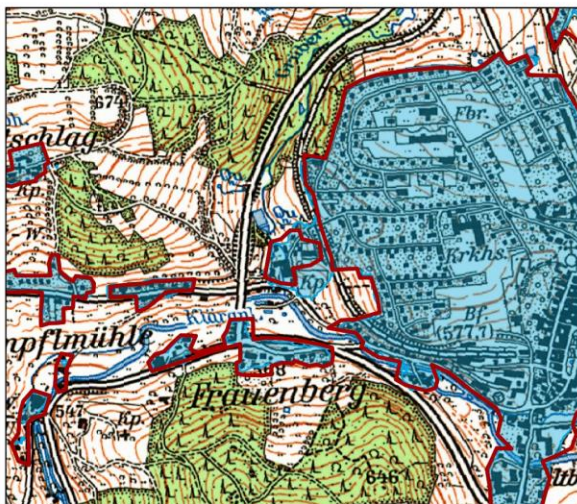
MobileNet-V2: Original Trainingsdaten, Kachelgröße 256x256, Batch-Size: 16, Modelltyp: Max-DeepLab, Max. 50 Epochen, Backbone-Layer nicht fixiert - Panoptic Quality: 0.7494



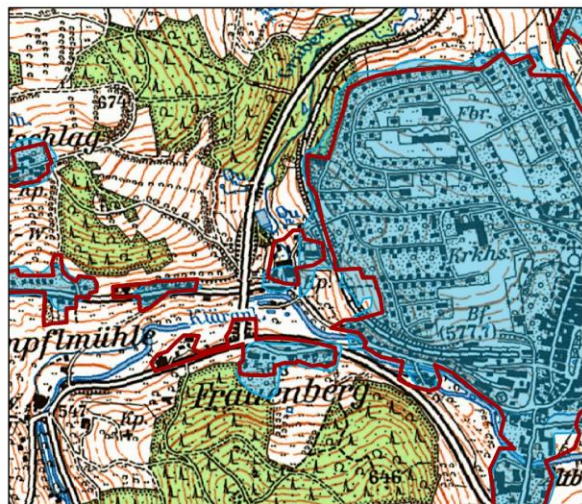
VGG-16: mit Batch-Normalisierung, Standardisierter Trainingsdatensatz, Kachelgröße 256x256, Batch-Size: 16, Modelltyp: Max-DeepLab, Max. 20 Epochen - Panoptic Quality: 0.7265



ResNet-18: Originale Trainingsdaten, Kachelgröße 256x256, Batch-Size: 4, Modelltyp: Max-DeepLab, Max. 20 Epochen - Panoptic Quality: 0.7507



ResNet-34: Standardisierter Trainingsdatensatz, Kachelgröße 256x256, Batch-Size: 16, Modelltyp: U-Net, Max. 20 Epochen - Overall Accuracy: 0.98



DarkNet-53: Originale Trainingsdaten, Kachelgröße 256x256, Batch-Size: 16, Modelltyp: Max-DeepLab, Max. 100 Epochen - Panoptic Quality: 0.7055

Umgrenzung Trainingsgebiet  Erkannte Siedlungsfläche

Abb. 13 Erkannte Siedlungsflächen im Westen der Ortschaft Grafenau  
Kartenansichten im Maßstab 1:20000; TK25 Kartenblatt 7146 von 1983  
(Quelle: eigene Darstellung, ArcGIS Pro)

Das auf dem ersten Trainingsdatensatz trainierte VGG-19-Modell erkennt nur wenige Siedlungsflächen und das sehr kleinteilig. Zwar sind die erkannten Siedlungsflächen richtig klassifiziert, es wurden aber nicht hinreichend der für die Klasse real existierenden Objekte erkannt. Beim VGG-16-Modell verhält es sich umgekehrt, trotz ebenso oft kleinteiliger Segmentierung wurde deutlich mehr Fläche segmentiert und viele der realen Objekte wurden richtig klassifiziert. Allerdings wurden im Verhältnis dazu auch viele vermeintliche *False Positives* generiert, bei denen es sich zumeist um Einzelgebäude handelt.

Die mit Max-DeepLab trainierten ResNet-Modelle funktionieren weniger gut. Bei allen wurde nicht ausreichend Siedlungsfläche erkannt. Das zeigt sich insbesondere für das testweise mit einer Batch-Größe von 4 trainierte Modell, das auch innerhalb der Siedlungsflächen große Lücken aufweist. Interessant ist, dass sich die vom Trainingswerkzeug ausgegebene Panoptic Quality mit 0.7507 kaum von den anderen unterscheidet. In der berechneten Konfusionsmatrix zeigt sich jedoch, dass nur etwa 56 Prozent der Ground Truth Siedlungspunkte als solche klassifiziert wurden. Das entspricht mehr als 10 Prozent weniger als bei den anderen mit Max-DeepLab trainierten ResNet-Modellen. Noch weniger Siedlungsgebiete hat nur das Modell erkannt, bei dem die Backbone-Layer nicht fixiert wurden. Die beiden mit U-Net trainierten Modelle heben sich mit einer aus der zufälligen Punktverteilung berechneten OA von über 90% deutlich von den anderen ResNet-Modellen ab. Die visuelle Analyse zeigt auch, dass dieser Modelltyp als einziger vielfach klare Grenzen entlang der Straßen oder Gartenumgrenzungen zieht und sich damit dem Digitalisierungsvorgehen am meisten annähert.

Das ResNet-34-Modell, das mit Mask-RCNN trainiert wurde, zeigt die wenigsten Objekte. Es wird deutlich, dass nur größere Ortschaften registriert wurden - viele kleine Dörfer fehlen. Es ist jedoch das einzige Modell, bei dem es so gut wie keine detektierten Kleinstflächen in der Ausgabe gibt. In allen anderen Modellen wurden gleichermaßen winzigste Kleinstflächen von bis zu nur drei Quadratmetern Fläche als Siedlungsfläche segmentiert. Da die mit U-Net trainierten Modelle Pixel klassifizieren und bei der Umwandlung in Polygone teilweise einzelne Pixel konvertiert werden, ist das Ausmaß an Kleinstflächen hier besonders hoch. Unabhängig davon, ob jene Flächen in ihrer Lage korrekt einer Siedlungsfläche zugewiesen wurden oder nicht, haben sie durch ihre Größe keinen Mehrwert. Alle Flächen kleiner 1000 Quadratmetern wurden daher von der Punkterstellung für die Genauigkeitsbeurteilung ausgeschlossen.

## 7.2 Performance auf ungesehenen Rasterdaten

Entscheidend für die Einsatzfähigkeit eines Modells ist seine Arbeitsweise auf unbekanntem Daten. Dafür wurde ein kleiner Bereich im Süden des Kartenblatts 7046 definiert, der Siedlungen enthält, die nicht für den Trainingsdatensatz herangezogen worden sind. Die für den Vergleich manuell definierten Siedlungsgebiete für Spiegelau sind in Abb. 14 dargestellt.

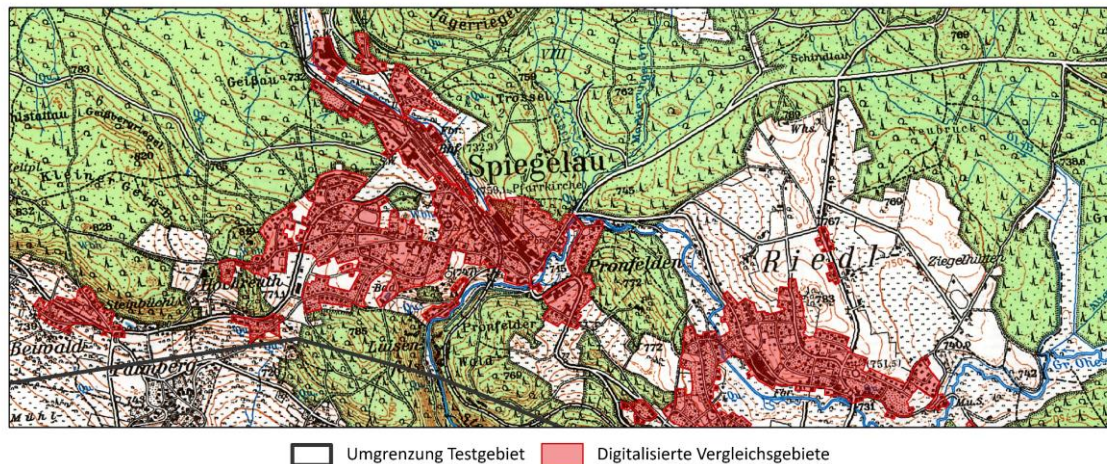


Abb. 14 Ausschnitt Digitalisierter Siedlungsflächen im Testgebiet  
Karte im Maßstab 1:35 000 (Quelle: eigene Darstellung, ArcGIS Pro)

Für eine möglichst zuverlässige Genauigkeitsaussage wurden die zufälligen Punkte auf dem Testgebiet mit einem Mindestabstand von 10 Metern und bis zu 100 Punkten pro Hektar dicht verteilt. Da das Testgebiet mit einer Fläche von ca. 35 km<sup>2</sup> deutlich kleiner ausfällt, hält sich der damit einhergehend gestiegene Rechenaufwand in Grenzen. Erkannte Siedlungsflächen kleiner 1000 Quadratmetern wurden dennoch aus dem als Siedlung klassifizierten Polygonlayer ausgeschlossen. Die Leistung aller Modelle ist auf den ungesehenen Daten zwangsläufig schlechter als auf dem für das Training verwendeten Rasterbild. Dennoch ist das Ausmaß dieser Differenz bei vielen Modellen nur geringfügig und die Leistung der Modelle im Verhältnis zueinander verbleibt ähnlich. Die Einstellungen aller trainierten Modelle sowie deren wichtigste Genauigkeitsangaben sind Tabelle 4 zu entnehmen.

Lfd. Nr.	Backbone-Modell	Trainingsparameter							Angaben aus Konfusionsmatrix [%]					
		Batch Size	Kachelgröße	Max. Epochen	Trainingsdaten nach Grundsätzen	Standardisierte Trainingsdaten	Modelltyp	Backbone Layer Fixed	PQ/OA laut Trainingswerkzeug in %	Training:Validation (gerundet)	User Accuracy Klasse Siedlung	Producer Accuracy Klasse Siedlung	Overall Accuracy	
1	DarkNet -	53	16	256	100	✓		M	✓	70.6	1:3	70.7	89.2	85.1
2			16	256	30	✓		M		78.1	1:1	76.4	84.5	86.4
3			16	256	20	✓		M	✓	79.2	1:1	75.6	87.1	86.4
4			16	256	50		✓	M	✓	65.6	1:1	58.0	91.7	80.6
5			16	256	20		✓	M	✓	73.9	1:1	80.5	81.3	87.8
6	DenseNet-	121	16	256	30		✓	M	✓	76.2	1:1	77.3	80.1	86.8
7	MobileNet-	V2	16	256	150	✓		M	✓	72.2	1:4	75.2	84.0	85.5
8			16	256	50	✓		M		77.8	1:1	78.6	82.8	86.6
9			16	256	20	✓		M	✓	74.9	1:1	75.2	87.5	86.7
10			8	256	20	✓		M	✓	78.1	1:1	78.9	78.9	85.4
11			16	256	30		✓	M	✓	76.8	1:2	77.1	76.9	85.4
12	ResNet -	18	32	128	20	✓		M	✓	48.6	1:1	41.7	10.5	61.9
13			16	256	50	✓		M	✓	77.6	1:1	71.8	74.7	86.2
14			16	256	50	✓		M		73.6	1:1	85.4	51.3	72.8
15			16	256	50	✓		M	✓	69.4	1:1	69.7	87.0	85.7
16			16	256	20	✓		M	✓	75.1	1:1	80.7	77.7	86.1
17			4	256	20	✓		M	✓	98.0	1:1	61.2	74.2	80.9
18			16	256	20		✓	U	✓	68.5	1:1	74.1	80.8	87.9
19		34	16	256	20	✓		R	✓	39.1	3:1	78.0	57.3	60.7
20			16	256	50		✓	U	✓	98.0	1:1	76.7	84.4	89.6
21		50	32	224	30		✓	R	✓	0.0	1:1	-	-	-
22	VGG -	11	16	256	20	✓		M	✓	65.8	1:1	-	-	-
23		16	16	256	20		✓	M	✓	72.7	1:1	77.4	82.0	87.2
24		19	16	256	20	✓		M	✓	70.6	1:1	78.5	71.2	83.1

Tabelle 4 Trainierte Modelle und zugehörige Genauigkeitsangaben für das Testgebiet  
 Modelltypen: M:Max-DeepLab, R: Mask RCNN, U:U-Net; Für mit Max-DeepLab trainierte Modelle entsprechen die Angaben in der Spalte "PQ/OA[...]" der Panoptic Quality, für alle anderen der Overall Accuracy. Das Verhältnis von Training und Validation wurden aus den Werten der jeweils letzten durchgeführten Epoche berechnet. (Quelle: eigene Darstellung)

Mit einer Gesamtgenauigkeit von 89.6 % liefert das mit U-Net trainierte ResNet-34-Modell die beste Segmentierung. Dem folgen das mit U-Net trainierte ResNet-18-Modell, sowie das laut OA ebenso leistungsfähige auf dem standardisierten Trainingsdatensatz mit einem Validierungsanteil von 10 Prozent trainierte DarkNet-53-Modell.

Dieses weist allerdings im Vergleich zum ResNet-18 eine um 6.4 Prozentpunkte höhere Nutzergenauigkeit sowie eine minimal höhere Produzentengenauigkeit auf. Jene spiegelt sich durch eine richtige Segmentierung eines kleinen Wohngebiets nördlich Spiegelau wider, das in den mit U-Net trainierten Modellen nicht erkannt wird. Die deutlich höhere Nutzergenauigkeit ist hauptsächlich auf den Umstand zurückzuführen, dass die U-Net-Modelle im Gegensatz zum DarkNet-53 viele kleine Einzelgebäude klassifizieren, die im Referenzdatensatz nicht mit digitalisiert worden sind. Die visuelle Beurteilung stellt klar, dass die meisten dieser segmentierten Flächen tatsächliche Gebäude sind (Abb. 15).

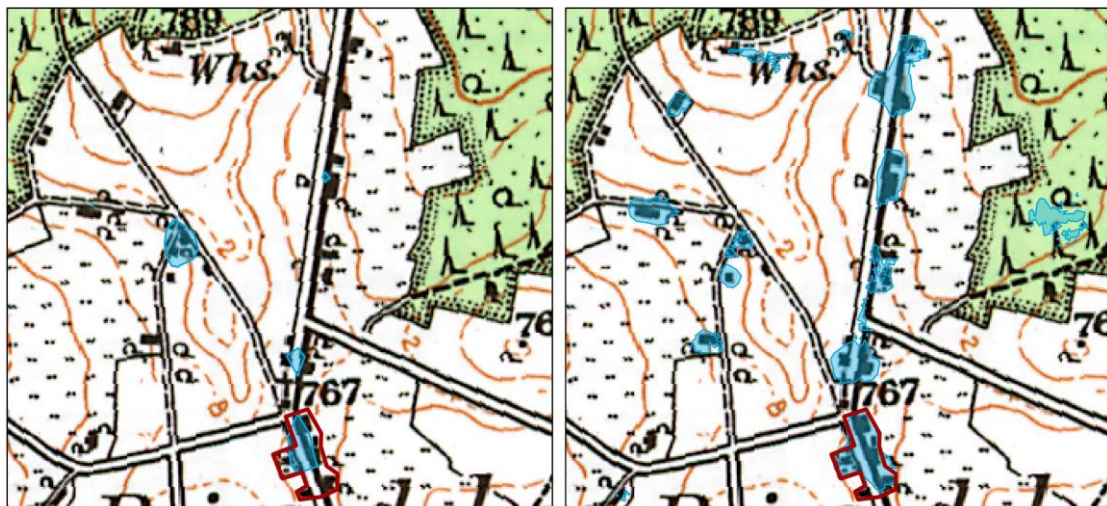


Abb. 15 „False Positives“ von links nach rechts: DarkNet-53, ResNet-18  
Maßstab 1:10 000 (Quelle: eigene Darstellung, ArcGIS Pro)

In ihrer Leistung folgen das VGG-16 und das auf DenseNet-121 trainierte Modell, beide zeigen die Schwäche, an den Außengrenzen der Siedlungsflächen befindliche Gebäude nicht mit einzubeziehen. Beim DenseNet-Modell fehlt ein kleinerer Teil inmitten des Ortes Spiegelau, der sich nicht nur als Lücke in einem geschlossenen Polygon äußert und damit für eine Bildsegmentierung nicht verwertbar wäre (Abb. 16).

Die besten 5 Modelle auf den ungesehenen Daten sind alle auf den standardisierten Daten trainiert worden. Für die darauffolgenden auf MobileNet-V2 trainierten Modelle hat der veränderte Trainingsdatensatz keine Verbesserung gebracht, die besten auf diesem Backbone trainierten Modelle sind das auf den Standardeinstellungen trainierte und jenes, bei dem die Fixierung der Backbone-Layer aufgehoben wurde.

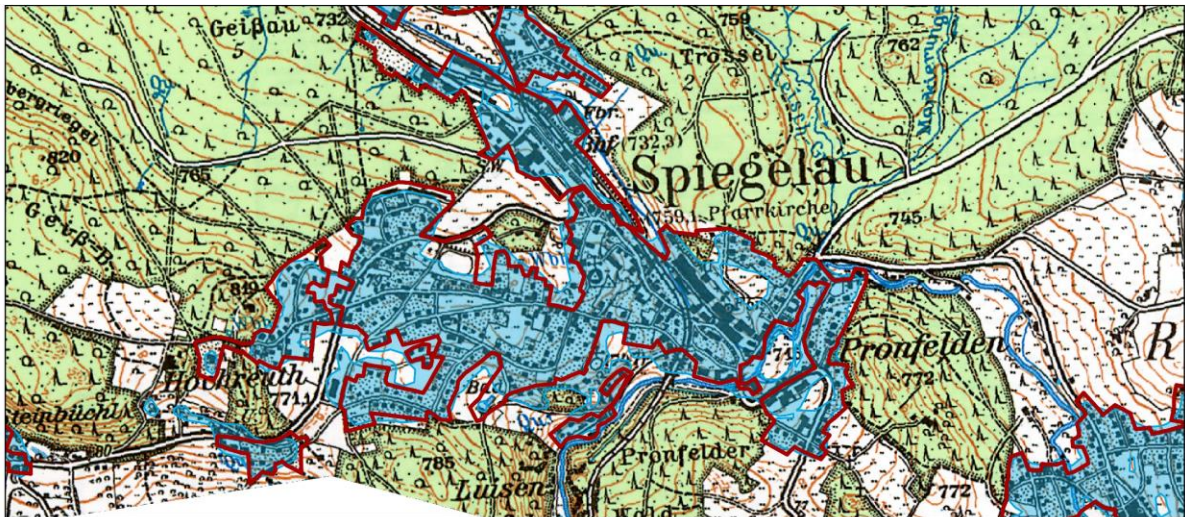
Lfd. Nr.	Backbone-Modell	Batch Size	Trainingsparameter						Angaben aus Konfusionsmatrix [%]					
			Kachelgröße	Max. Epochen	Trainingsdaten n. Grundsätzen	Standardisierte Trainingsdaten	Modelltyp	Backbone Layer Fixed	User Accuracy Klasse Siedlung	User Accuracy Klasse None	Producer Accuracy Klasse Siedlung	Producer Accuracy Klasse None	Overall Accuracy	
1	ResNet	34	16	256	50		✓	U	✓	76.7	94.6	84.4	91.3	<b>89.6</b>
2	ResNet	18	16	256	20		✓	U	✓	74.1	93.2	80.8	90.3	<b>87.9</b>
3	DarkNet	53	16	256	20		✓	M	✓	80.5	91.3	81.3	90.9	<b>87.8</b>
4	VGG	16	16	256	20		✓	M	✓	77.4	91.8	82.0	89.5	<b>87.2</b>
5	DenseNet	121	16	256	30		✓	M	✓	77.3	91.2	80.1	89.8	<b>86.8</b>
6	MobileNet	V2	16	256	20	✓		M	✓	75.2	93.6	87.5	86.4	<b>86.7</b>
7	MobileNet	V2	16	256	50	✓		M		78.6	91.1	82.8	88.6	<b>86.6</b>

*Tabelle 5 Die 7 "besten" Modelle, nach OA absteigend sortiert  
(Quelle: eigene Darstellung)*

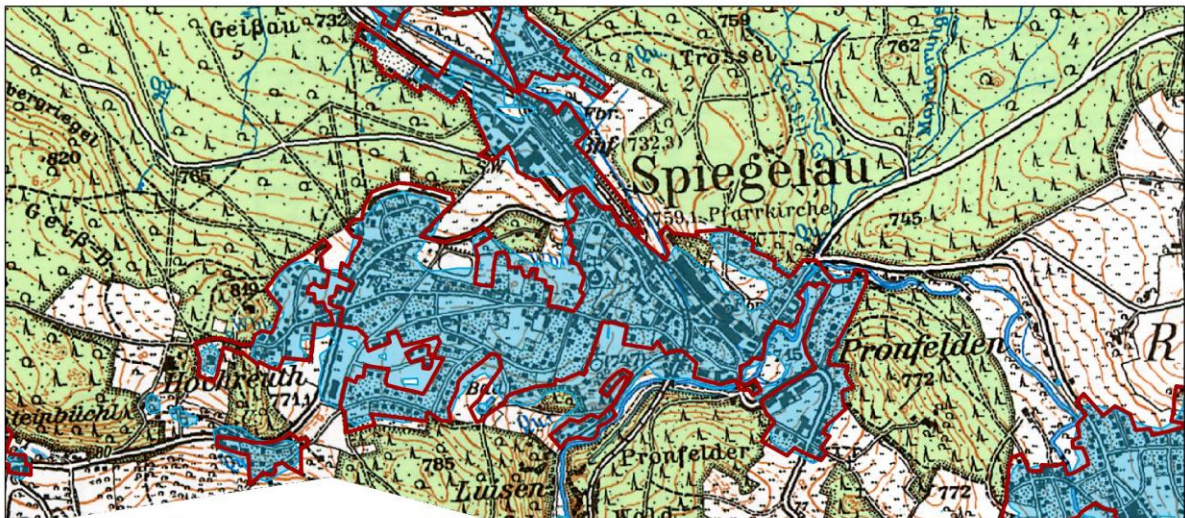
Modell 7 (Tab. 5) zeigt mehr innerörtliche Lücken auf als das, bei welchem die Backbone-Layer fixiert sind. Die Grenzen werden nach visueller Abschätzung präziser gezogen, hier neigt Modell 6 im Vergleich dazu, angrenzenden Waldrand in die Segmentierung mit einzubeziehen.

Aufgrund des Informationsumfangs ist in den Tabellen nicht ersichtlich, dass bei den Modellen 3 - 5 ein größerer Trainingsdatensatz verwendet worden ist. Während viele Modelle zuerst mit den unter Standardeinstellungen exportierten Trainingsdaten trainiert worden sind, wurde testweise eine geringere Schrittweite für die Kachelgenerierung von 0.4 statt 0.5 gewählt. Der Bilddatensatz ist damit bereits 1.5 GB größer als der Ursprüngliche. Außer für die MobileNet-V2-Modelle sind die auf diesem Datensatz basierten Modelle gemessen an der OA leistungsfähiger als die auf dem gleichen Backbone mit einem kleineren Trainingsdatensatz trainierten Modelle.

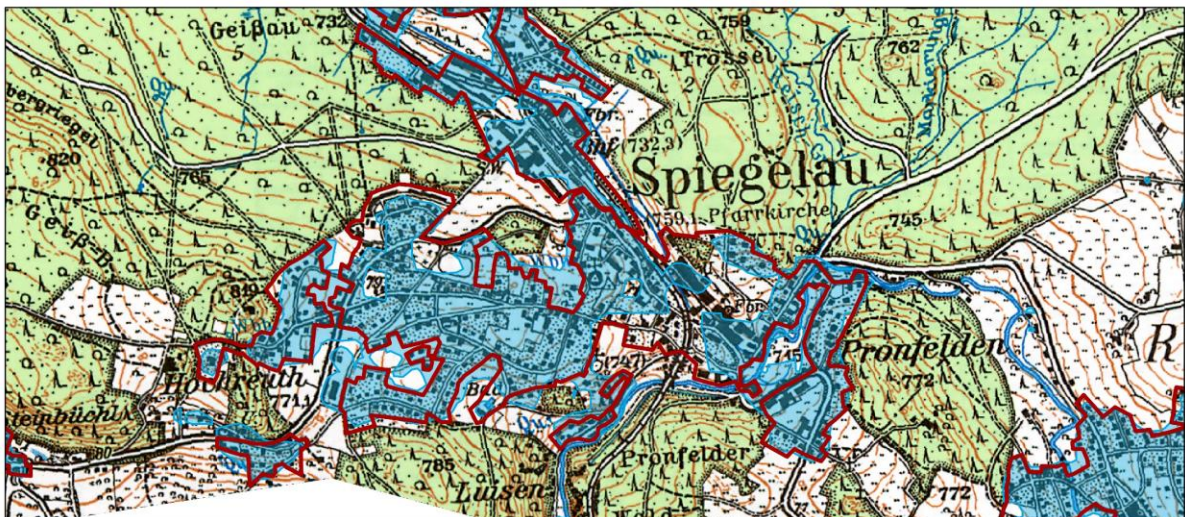
Die detektierten Objekte von 3 ausgewählten Modellen für die Ortschaft Spiegelau folgen in Abb. 16.



DarkNet-53: Standardisierter Trainingsdatensatz, Modelltyp: Max-DeepLab, OA: 87.8 %



VGG-16: mit Batch-Normalisierung, Standardisierter Trainingsdatensatz, Modelltyp: Max-DeepLab, OA: 87.2 %



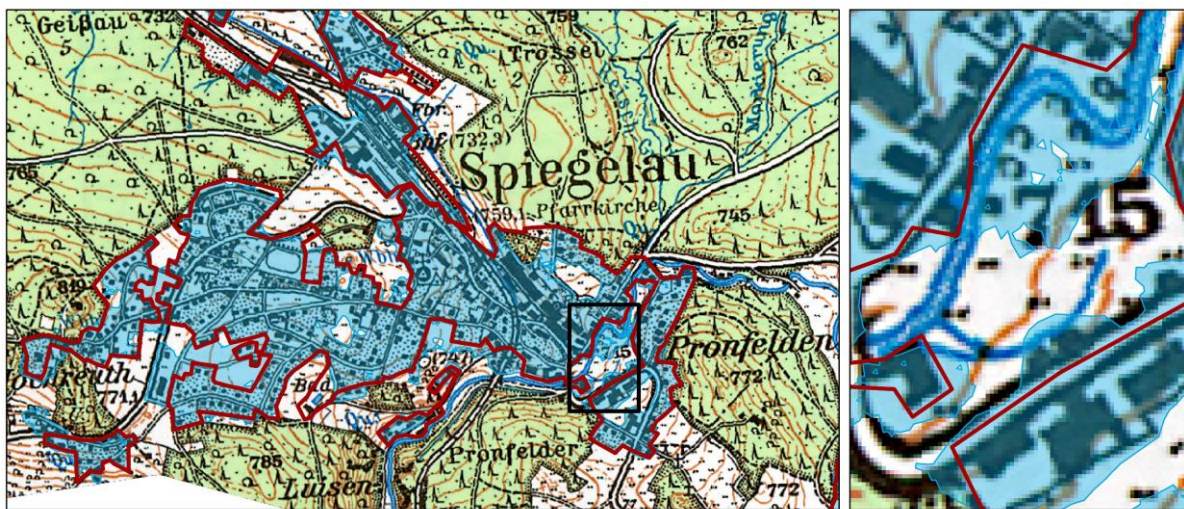
DenseNet-121: Standardisierter Trainingsdatensatz, Modelltyp: Max-DeepLab, OA: 86.8 %

Digitalisierte Grenzen
  Erkannte Siedlungsfläche

Abb. 16 Erkannte Siedlungsgebiete im Testgebiet im Maßstab 1:20 000  
(Quelle: eigene Darstellung, ArcGIS Pro)



Zwar wird für die mit U-Net trainierten Modelle die höchste Overall Accuracy zurückgegeben, dennoch wird ein kleiner Teil eines Wohngebiets im Norden von Spiegelau von beiden nicht richtig erkannt. Der nicht detektierte Bereich umfasst 10 Gebäude, die von anderen Modellen richtig klassifiziert wurden. Besonders auffällig für diese beiden Modelle ist außerdem, dass die Grenzlinien aufgrund der pixelweisen Segmentierung teilweise durch sehr kleinteilig verstreute Polygone und feine Zick-Zack-Linien gekennzeichnet sind (Abb. 17). Innerhalb des Ortes tauchen für alle Modelle wiederholt kleine Freiflächen auf.



ResNet-34: Standardisierter Trainingsdatensatz, Modelltyp: U-Net, OA: 89.61 %, Maßstab 1:20 000 links; Detailausschnitt kleinteiliger Grenzfestlegungen im Maßstab 1:4000 rechts

Abb. 17 Erkannte Siedlungsgebiete auf dem Testgebiet mit Detailsicht  
(Quelle: eigene Darstellung, ArcGIS Pro)

Im Großen und Ganzen sind die Umgrenzungen besser an Straßenverläufen und Gartenumgrenzungen orientiert als bei anderen Modellen (Abb. 18). Das ResNet-18-Modell schneidet geringfügig schlechter ab - klassifiziert angrenzende Freiflächen teils als Siedlung und weist wenige Fehlklassifizierungen von Schrift oder eines kleinen Waldstückes auf.

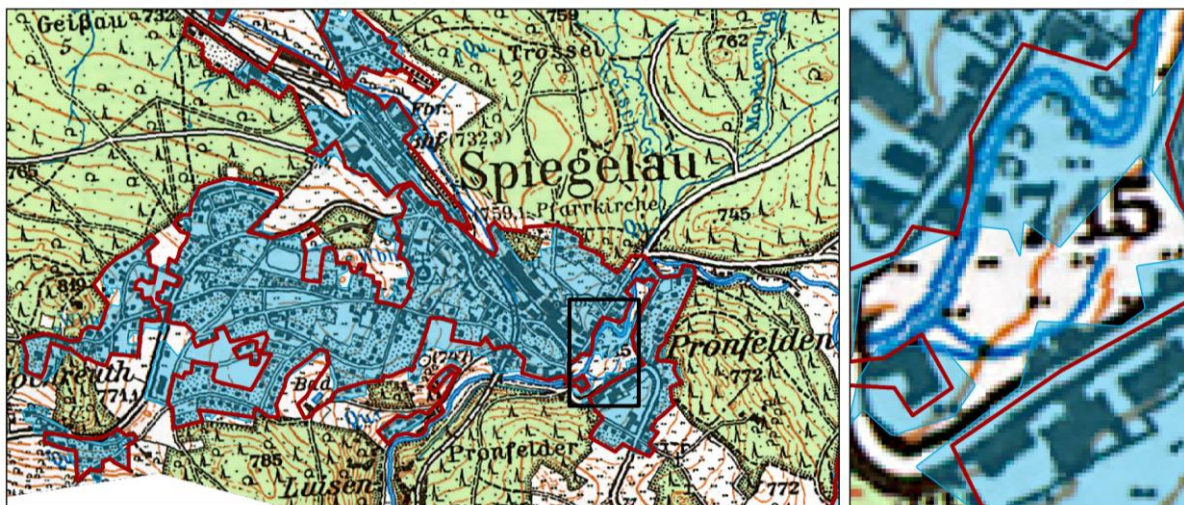


Abb. 18 Detailsichten 1:10 000 von links nach rechts: DarkNet-53, ResNet-34, VGG-16  
(Quelle: eigene Darstellung, ArcGIS Pro)

### 7.3 Nachverarbeitung

Die Grenzlinien der Siedlungsobjekte aller Modelle sind ungleichmäßig und enthalten viele unnötige Stützpunkte. Um Unregelmäßigkeiten wie in Abb. 17 zu entfernen, die visuelle Darstellung zu verbessern und einer manuellen Digitalisierung anzunähern, wurde ein kurzer Ablauf von Werkzeugen entwickelt, der die vom Modell erkannten Siedlungspolygone generalisiert.

Im ersten Schritt wurde der Layer von Flächenobjekten kleiner 2000 m<sup>2</sup> entfernt. Dieser Grenzwert ergab sich aus eigenem Ermessen und orientierte sich an der minimalen Flächengröße der manuell digitalisierten Siedlungsobjekte. Im Folgenden wurden Lücken kleiner 15 000 m<sup>2</sup>, die vollständig innerhalb der detektierten Siedlungsflächen liegen, entfernt. Anschließend wurden die Umgrenzungslinien der Polygone generalisiert und geglättet. Insbesondere für das ResNet-34-Modell sorgte dieser Schritt für eine deutlich verbesserte Darstellung (Abb. 19).



ResNet-34: Standardisierter Trainingsdatensatz, Modelltyp: U-Net, Nachverarbeitet: Flächen < 2000m<sup>2</sup> entfernt, vollumschlossene Lücken geschlossen, Grenzlinien generalisiert, geglättet - OA: 89.86 %, Maßstab 1:20 000 links; Detailausschnitt Maßstab 1:4000 rechts

Abb. 19 Erkannte Siedlungsflächen nach Nachverarbeitung  
(Quelle: eigene Darstellung, ArcGIS Pro)

Kleine spitzwinkelige Unregelmäßigkeiten existieren ersichtlich immer noch, eine stärkere Generalisierung wurde jedoch aufgrund der schwindenden Linientreue als nicht sinnvoll erachtet. Die Modelle mit der Nummer 1, 3, 4 und 5 (Tabelle 4) wurden entsprechend der erarbeiteten Werkzeuge generalisiert und neue Fehlermatrizen berechnet. Die Gesamtgenauigkeit verbleibt für alle Modelle annähernd gleich, aufgrund der geschlossenen Lücken ist bei den meisten eine geringfügig gesteigerte Produzentengenauigkeit für die Klasse Siedlung zu erkennen.

## 7.4 Transferstudie

Da diese Arbeit aus der ursprünglichen Aufgabe entstand, eine Siedlungsentwicklungsanalyse zu verwirklichen, wurden testweise noch 2 andere Zeitstände der TK25 vom LDBV angefordert. Über einen längeren Zeitraum hinweg änderte sich die Symbolisierung in den Karten zwar stark, die Ausgaben der Jahre 1972 und 1992 sind von der Ausgestaltung jedoch sehr ähnlich gehalten. Die Hintergrundfarbgebung und Intensität der Farben unterscheiden sich leicht. Die Anwendung des besten mit U-Net trainierten Modells zusammen mit den zuvor erarbeiteten Nachverarbeitungsschritten auf dem Testgebiet zeigte erste Ergebnisse, die einen Ausblick auf eine Einsatzmöglichkeit verschaffen. Auf den Karten der anderen Jahre ist die Segmentierung wie zu erwarten nicht genauso gut, wie auf den für das Training verwendeten Rasterdaten. Dennoch werden neue Wohngebiete der 90er Jahre in dieser Karte auch zusätzlich erkannt. Die fehlenden 10 Gebäude werden in allen Karten nicht detektiert. Auffällig ist außerdem in beiden anderen Karten jeweils ein kleines Waldstück, von dem ein Teil fälschlicherweise als Siedlung erkannt wird (Abb. 20).



Abb. 20 Erkannte Siedlungsflächen von links nach rechts: 1972, 1983, 1992  
(Maßstab 1:10 000, Quelle: eigene Darstellung, ArcGIS Pro)

Nach visueller Beurteilung ist zwischen den Karten von 1972 und 1983 bis auf wenige Häuser keine Veränderung am Gebäudebestand zu erkennen. Die Karte von 1992 unterscheidet sich nicht nur in der Symbolisierung stärker von den anderen, sondern zeigt auch eine sichtbar verdichtete Bebauung und neu entstandene Wohngebiete. Dies lässt sich auch aus einem Flächenvergleich der Siedlungsobjekte für die unterschiedlichen Jahre entnehmen (Tab. 6).

	1972	1983	1992
Fläche [ha]	137.89	135.73	172.44
Änderung		-1.6 %	+27.0 %
Fläche ohne FP [ha]	135.02	135.73	168.88
Änderung		+0.5 %	+24.4 %

Tabelle 6 Erkannte Siedlungsflächen auf dem Testgebiet im Jahresvergleich  
(Prozentangaben jeweils im Vergleich zum vorhergehenden Zeitstand, Quelle: eigene Darstellung und Berechnung)

Die beiden optisch auffallend falsch detektierten Waldgebiete (*False Positives FP*) wurden mithilfe einer manuell digitalisierten Maske entfernt, um einen Eindruck vom Ausmaß der Fehldetektionen zu schaffen. Die Polygone der Karte von 1992 verringern sich damit um ca. 4 ha. Trotz diesem Fehler ist ein signifikanter Anstieg der Siedlungsfläche zwischen 1983 und 1993 nachweisbar.

## 8 Diskussion

Mit fast allen verwendeten Backbone-Modellen konnte eine Erkennung und Segmentierung von Siedlungsgebieten erreicht werden. Lediglich das VGG-11-Modell und das mit Mask-RCNN trainierte ResNet-50-Modell detektierten keine Objekte. Da auch das ResNet-34-Modell mit Mask-RCNN nur eine unzureichende Erkennung der Siedlungsgebiete erreichte, wurde auf eine weitere Verwendung dieses Modelltyps verzichtet. Max-DeepLab sorgte bei fast allen Backbones für gute Segmentierungsergebnisse und wurde daher auch für den Großteil der Modelle eingesetzt. U-Net konnte erst am Ende dieser Arbeit getestet werden und stellte sich als ebenfalls sehr leistungsfähig heraus. Die Verbesserung des ResNet-34-Modells gegenüber dem anderen auf U-Net basierendem Modell liegt vermutlich nicht nur an der Tiefe des Netzes, sondern insbesondere an der erhöhten Epochenanzahl. Diese Annahme ist auf die Trainingsdetails zurückzuführen, in denen eine geringfügige, aber stetige Verbesserung der *Training* und *Validation Loss* – Werte bis Epoche 45 zu beobachten war. Bei anderen Modellen ließ sich kein positiver Einfluss einer erhöhten maximalen Epochenanzahl erkennen.

Die mit U-Net trainierten Modelle heben sich in der Segmentierungsqualität von den anderen ab. U-Net ist der einzige Modelltyp, der auf Grauwertbildern aus lichtmikroskopischen Aufnahmen basiert. Die Zellen unterscheiden sich in diesen nicht primär in ihrer Textur, sondern sind durch feine Kontraste an ihren Angrenzungen zu differenzieren.

Im Trainingsprozess der U-Net-Architektur wurde den Randpixeln der Zellen von Anfang an ein höheres Gewicht zugewiesen als den anderen (Ronneberger et al., 2015). Denkbar wäre, dass die Architektur daher stärker auf Straßen und Gartenumgrenzungen als anzunehmende Grenzlinie reagiert. Max-DeepLab erkannte zwar viele Objekte richtig, scheint die Grenzen jedoch eher anhand der Nähe zu Gebäuden bzw. sicher zugehörigen Elementen zu orientieren.

Da ein Großteil der besten Modelle auf einem, durch eine verringerte Schrittweite in der Kachelerstellung, vergrößertem Datensatz trainiert worden ist, ist die Annahme zu bestätigen, dass eine größere Menge an Trainingsdaten einen positiven Einfluss auf die Leistung hat. Durch eine Digitalisierung von Objekten anhand mehrerer Kartenblätter hätten die Modelle womöglich noch verbessert werden können. Die Kachelgröße selbst wirkte sich ebenfalls auf die Modelle aus: Das testweise Verringern der Kachelgröße auf ein Format von 128x128 Pixeln, sorgte für einen signifikanten Einbruch der Leistungsfähigkeit. Die Standardeinstellung im Trainingswerkzeug von 256x256 Pixeln erwies sich als am besten. Die Leistung der mit kleinen Batch-Größen trainierten Modelle war tendenziell schlechter als die der auf dem gleichen Backbone basierenden Modelle mit größeren Batch-Größen. Aufgrund der Größe der Kacheln musste sich hardwarebedingt auf eine maximale Batch-Größe von 16 beschränkt werden. Die 12GB VRAM der Grafikkarte waren bei einigen Modellen nahezu vollständig ausgeschöpft.

Problematisch während der Auswertung war, dass manche Angaben zum Trainingswerkzeug in ArcGIS Pro seitens des Herstellers nur unzureichend dokumentiert sind. So ist beispielsweise leider nicht geklärt, welche Layer in einem Backbone-Modell fixiert und verändert werden, je nach Auswahl der Option „Backbone-Layer fixieren“. Fraglich war außerdem, welches Vertrauen der Nutzer in die vom Trainingswerkzeug ausgegebenen Genauigkeitsangaben setzen kann. Ob die ausgegebenen Panoptic Quality-Werte anhand der *Intersection over Union* der umgebenden Rechtecke oder der pixelweisen Überlagerung berechnet werden, ist nicht zu finden. Da es sich um eine Segmentierungsaufgabe handelt ist anzunehmen, dass eine pixelweise Berechnung stattfindet. Für das VGG-11-Modell, das kein einziges Objekt erkannte, wurde jedoch eine PQ von 0.6579 zurückgegeben.

Die für die mit U-Net trainierten Modelle zurückgegebene OA beträgt jeweils 0.98 – ein Wert, der in Anbetracht der detektierten Objekte nicht realistisch erscheint. Zwar betrug die aus der Fehlermatrix entnommene OA für die bekannten Rasterdaten ebenfalls knapp über 90 Prozent, die 8 Prozent Differenz können in diesem Wertebereich jedoch einen großen Unterschied bedeuten. Verschiedene Backbone-Modelle leisteten bezüglich der OA gute Ergebnisse, allerdings lag bei der Berechnung der Fehler eine ungleiche Klassenverteilung vor. Für die Klasse „Keine Siedlung“ wurden für die *Ground Truth* - Polygone im Durchschnitt das Doppelte bis Dreifache an Punkten erstellt, wie für die Klasse „Siedlung“. Die visuelle Beurteilung zeigte, dass die Segmentierung für die besten Modelle nach Entfernen der Kleinstobjekte sehr gut ausfällt. Manche Umgrenzungen, die von den manuell digitalisierten abweichen, könnten dennoch als korrekt angesehen werden (Abb. 21).

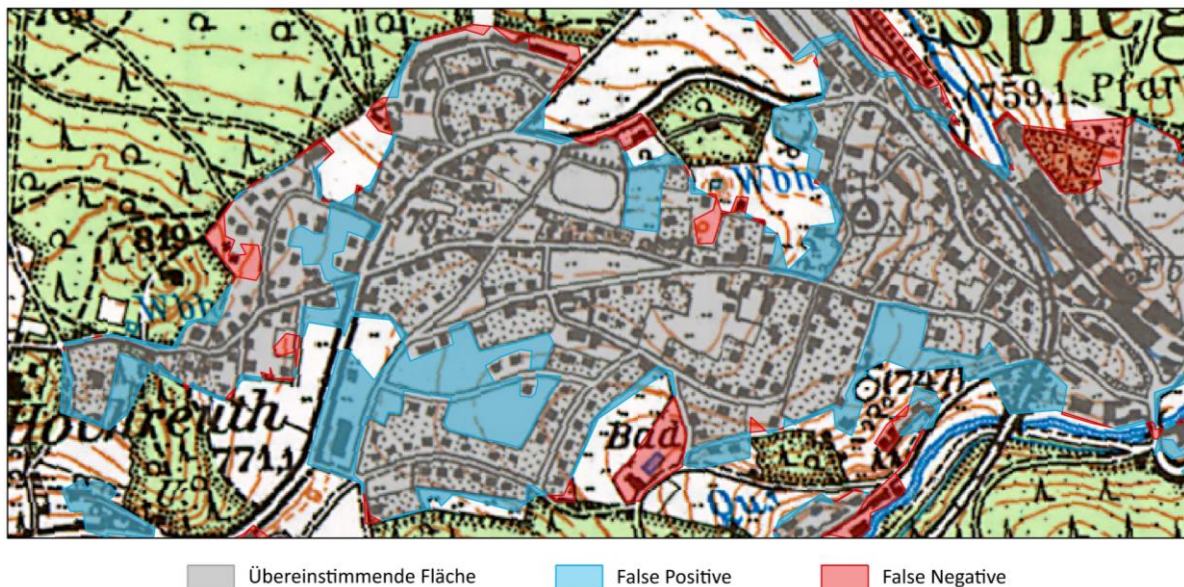


Abb. 21 Differenzflächen: SOLL – Segmentierte Siedlungsflächen mit ResNet-34, U-Net  
Maßstab: 1:10 000 (Quelle: eigene Darstellung, ArcGIS Pro)

Die im digitalisierten Layer enthaltenen Freiflächen zwischen benachbarten Wohngebieten hätten von einer anderen digitalisierenden Person womöglich als Siedlungsfläche eingestuft werden können und wären damit richtig gewesen. Außerdem ist fraglich, ob die an das Fabrikgelände im Osten Spiegelaus angrenzende Grünfläche hätte Siedlung sein müssen – das Modell grenzt dieses Gebiet nicht unbedingt zu Unrecht aus. Dies unterstreicht die Schwierigkeiten, die schon bei der Digitalisierung von Siedlungsflächen auftreten. Wo genau die Grenzen und Verbindungen zwischen benachbarten Wohngruppen zu ziehen sind, ist nicht definiert. Eine strenge Einhaltung genaustens definierter Richtlinien würde den Zeitaufwand für die Digitalisierung jedoch weiter in die Höhe treiben und ist daher nicht anwendbar.

## 9 Fazit

Diese Arbeit zeigt, dass einige vortrainierte Open-Source-Modelle für die Erkennung von Siedlungsgebieten in historischen Topographischen Karten einsetzbar sind. 5 Modelle erreichten eine Gesamtgenauigkeit von über 87 Prozent, wobei 4 davon auf unterschiedlichen Backbone-Modellen basieren. Der zu erwartende Fehler beim besten Modell liegt bei etwa 11 Prozent. Obwohl die mit U-Net trainierten Modelle im Gegensatz zu manchen auf Max-DeepLab basierenden einen Teil eines Wohngebiets nicht erkennen, sind die Abgrenzungen im Allgemeinen qualitativ besser. Unter der Voraussetzung, dass Gartenflächen und Straßen als Anhaltspunkt dienen, ist dieser Modelltyp zu bevorzugen, da er sich dem Digitalisierungsvorgehen am ehesten annähert und ein optisch logisches Segmentierungsbild erzeugt.

Die in ArcGIS Pro implementierten Werkzeuge ermöglichen eine intuitive GUI-basierte Möglichkeit, ohne Programmierkenntnisse Deep-Learning-Modelle zu trainieren. Die Dokumentation ist an mancher Stelle jedoch unzureichend und die quantitative Genauigkeitsbewertung zeigte, dass eine eigene Bemessung der Modellperformance unumgänglich ist. Die berechneten Genauigkeitsangaben des Trainingswerkzeugs können gänzlich irreführend sein.

Die Digitalisierung der Trainingsgebiete und das Ermitteln der besten Trainingsparameter stellten den größten Arbeitsaufwand des Modelltrainings dar. Das Training des besten Modells war nach etwas über fünf Stunden abgeschlossen, die Dauer für die Erkennung und Segmentierung für das Testgebiet lag nur bei wenigen Sekunden. Eine manuelle Digitalisierung von Siedlungen würde vermutlich einen geringeren Fehler verursachen, der Zeitaufwand ist jedoch unverhältnismäßig hoch. Zudem ist zu berücksichtigen, dass die genauen Abgrenzungen im individuellen Ermessen des Digitalisierenden liegen und ein vollständig einheitlicher Datensatz ebenfalls nicht zu erreichen ist.

Die Informationsgewinnung aus historischem Kartenmaterial kann nicht nur zeitliche Analysen der Siedlungs- und Landschaftsentwicklung ermöglichen, sondern auch helfen, kausale Zusammenhänge zu erkennen. Je nach Anforderungen kann das beste trainierte Modell für solche Aufgabenstellungen eingesetzt werden. Der nächste Schritt könnte darin bestehen, statt einer binären Klassifikation, eine lückenlose Segmentierung der historischen Karte inklusive Straßen, Wiesen und Waldgebiete umzusetzen. Unter Verwendung eines größeren Trainingsdatensatzes und leistungsfähigerer Hardware könnte die Genauigkeit weiter verbessert werden. Letztlich zeigte die Transferstudie, dass die Siedlungserkennung auch auf Karten sehr ähnlicher Symbolisierung noch funktioniert, die sichtlich auffälligen durch manuelle Abgrenzung bestimmten Fehler lagen für beide Karten (1972, 1992) nur bei ca. zwei Prozent (2 ha auf 100 ha).

Das mit einer OA von 89.6 Prozent mit U-Net trainierte ResNet-34-Modell wurde als Deep-Learning-Paket in ArcGIS Online zum Herunterladen freigegeben. Ergänzend steht auch das in Kapitel 7.3 beschriebene Nachverarbeitungswerkzeug als Geoverarbeitungs-Paket zur Verfügung.



## Literaturverzeichnis

- Adv. (2008, April 11). Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens (GeoInfoDok) - ATKIS-Objektartenkatalog Basis-DLM. Zugriff am 28.11.2022. Verfügbar unter: <https://www.adv-online.de/icc/extdeu/broker.jsp?uMen=4ad505ea-127b-b941-2df2-65a572e13d63>
- Adv. (2019, Mai 21). Dokumentation zur Modellierung der Geoinformationen des amtlichen Vermessungswesens - Erläuterungen zum ATKIS Basis-DLM. Zugriff am 28.11.2022. Verfügbar unter: <https://www.adv-online.de/icc/extdeu/broker.jsp?uMen=4ad505ea-127b-b941-2df2-65a572e13d63>
- Aloysius, N. & Madathikulangara, G. (2017). A review on deep convolutional neural networks (S. 0588–0592). doi:10.1109/ICCSP.2017.8286426
- Bavarikon. (o. J.). Topographischer Atlas vom Königreiche Baiern 1:50 000, 1812-1867. *Topographischer Atlas vom Königreiche Baiern 1:50 000, 1812-1867 | bavarikon*. Zugriff am 26.11.2022a. Verfügbar unter: <https://www.bavarikon.de/object/bav:BSB-CMS-0000000000002414?locale=de&p=&lang=de>
- Bavarikon. (o. J.). Karten und Pläne aus dem Bestand der Bayerischen Staatsbibliothek. Zugriff am 26.11.2022b. Verfügbar unter: <https://www.bavarikon.de/object/bav:BSB-CMS-0000000000000619>
- Bhatt, C., Kumar, I., Vijayakumar, V., Singh, K. U. & Kumar, A. (2021). The state of the art of deep learning models in medical science and their challenges. *Multimedia Systems*, 27 (4), 599–613. doi:10.1007/s00530-020-00694-1
- Bhattacharyya, S., Pan, I., Mukherjee, A. & Dutta, P. (2017). *Hybrid Intelligence for Image Analysis and Understanding*. John Wiley & Sons.
- Cui, L., Huang, S., Wei, F., Tan, C., Duan, C. & Zhou, M. (2017). Superagent: A customer service chatbot for e-commerce websites. *Proceedings of ACL 2017, system demonstrations* (S. 97–102).
- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition* (S. 248–255). IEEE.
- Erb, R. J. (1993). Introduction to Backpropagation Neural Network Computation. *Pharmaceutical Research*, 10 (2), 165–170. doi:10.1023/A:1018966222807
- Grigorescu, S., Trasnea, B., Cocias, T. & Macesanu, G. (2020). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37 (3), 362–386. Wiley Online Library.
- He, K., Gkioxari, G., Dollár, P. & Girshick, R. (2018, Januar 24). Mask R-CNN. arXiv. doi:10.48550/arXiv.1703.06870
- He, K., Zhang, X., Ren, S. & Sun, J. (2015, Dezember 10). Deep Residual Learning for Image Recognition. arXiv. doi:10.48550/arXiv.1512.03385
- Hecht, R., Herold, H. & Meinel, G. (2008). Gebäudescharfe Analyse der Siedlungsentwicklung auf Grundlage mittelmaßstäbiger Karten.

- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T. et al. (2017, April 16). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv. doi:10.48550/arXiv.1704.04861
- Huang, G., Liu, Z., van der Maaten, L. & Weinberger, K. Q. (2018, Januar 28). Densely Connected Convolutional Networks. arXiv. doi:10.48550/arXiv.1608.06993
- Hussain, M., Bird, J. J. & Faria, D. R. (2019). A Study on CNN Transfer Learning for Image Classification (Advances in Intelligent Systems and Computing). In A. Lotfi, H. Bouchachia, A. Gegov, C. Langensiepen & M. McGinnity (Hrsg.), *Advances in Computational Intelligence Systems* (S. 191–202). Cham: Springer International Publishing. doi:10.1007/978-3-319-97982-3\_16
- Kalirane, M. (2023, Januar 2). Gradient Descent vs. Backpropagation: What's the Difference? *Analytics Vidhya*. Zugriff am 29.3.2023. Verfügbar unter: <https://www.analyticsvidhya.com/blog/2023/01/gradient-descent-vs-backpropagation-whats-the-difference/>
- Kirillov, A., He, K., Girshick, R., Rother, C. & Dollár, P. (2019, April 10). Panoptic Segmentation. arXiv. doi:10.48550/arXiv.1801.00868
- LDBV. (2017). Positionsblätter 1:25 000 - Kartendokumente des Bayerischen Königreichs. Zugriff am 26.11.2022. Verfügbar unter: <https://www.ldbv.bayern.de/produkte/historisch/position.html>
- LDBV. (2022). Historische Topographische Karten. *Landesamt für Digitalisierung, Breitband und Vermessung - Produkte - Historische Karten*. Zugriff am 26.11.2022. Verfügbar unter: <https://www.ldbv.bayern.de/produkte/historisch/tk.html>
- Lecun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. et al. (1990). Handwritten digit recognition with a back-propagation network. *Advances in Neural Information Processing Systems (NIPS 1989)*, Denver, CO. Morgan Kaufmann.
- Lei, S., Zhang, H., Wang, K. & Su, Z. (2022). How Training Data Affect the Accuracy and Robustness of Neural Networks for Image Classification.
- Meinel, G., Herold, H. & Hecht, R. (2014). Analyse und Visualisierung der Siedlungsentwicklung mit SEMENTA®-CHANGE. *Flächennutzungsmonitoring II - 52*.
- Micro-Star Int'l Co.Ltd & NVIDIA Corporation (Hrsg.). (2022, September 21). GeForce RTX 3060 GAMIN x12G Data Sheet. Zugriff am 27.2.2023. Verfügbar unter: <https://storage-asset.msi.com/datasheet/vga/global/GeForce-RTX-3060-GAMING-X-12G.pdf>
- Miller, G. A. (1995, November). WordNet: A Lexical Database for English.
- Mohd Aszemi, N. & Panneer Selvam, D. D. D. (2019). Hyperparameter Optimization in Convolutional Neural Network using Genetic Algorithms. *International Journal of Advanced Computer Science and Applications*, 10, 269–278. doi:10.14569/IJACSA.2019.0100638
- Oppermann, A. (2021, August 9). Batch-Normalisierung in Deep Learning. *KI Tutorials*. Zugriff am 13.3.2023. Verfügbar unter: <https://artemoppermann.com/de/batch-normalisierung-in-deep-learning/>

- Radiuk, P. (2017). Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets. *Information Technology and Management Science*, 20, 20–24. doi:10.1515/itms-2017-0003
- Redmon, J. & Farhadi, A. (2016, Dezember 25). YOLO9000: Better, Faster, Stronger. arXiv. doi:10.48550/arXiv.1612.08242
- Redmon, J. & Farhadi, A. (2018, April 8). YOLOv3: An Incremental Improvement. arXiv. doi:10.48550/arXiv.1804.02767
- Ronneberger, O., Fischer, P. & Brox, T. (2015, Mai 18). U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv. doi:10.48550/arXiv.1505.04597
- Rukundo, O. (2023). Effects of Image Size on Deep Learning. *Electronics*, 12 (4), 985. doi:10.3390/electronics12040985
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A. & Chen, L.-C. (2019, März 21). MobileNetV2: Inverted Residuals and Linear Bottlenecks. arXiv. doi:10.48550/arXiv.1801.04381
- Sarkar, D. (DJ). (2018, November 17). A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning. *Medium*. Zugriff am 1.4.2023. Verfügbar unter: <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
- Sifre, L. & Mallat, S. (2014, März 7). Rigid-Motion Scattering for Texture Classification. arXiv. doi:10.48550/arXiv.1403.1687
- Simonyan, K. & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Thompson, N. C., Greenewald, K., Lee, K. & Manso, G. F. (2022, Juli 27). The Computational Limits of Deep Learning. arXiv. doi:10.48550/arXiv.2007.05558
- Voithofer, M. (2019, Juni 25). Verwendung von Deep Learning für die Erkennung von Werbung am Straßenrand in Dashcam-Videos. Zugriff am 6.12.2022. Verfügbar unter: <https://theses.fh-hagenberg.at/system/files/pdf/Voithofer19.pdf>
- Wang, H., Zhu, Y., Adam, H., Yuille, A. & Chen, L.-C. (2021, Juli 12). MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers. arXiv. doi:10.48550/arXiv.2012.00759
- Wang, H., Zhu, Y., Adam, H., Yuille, A. & Chen, L.-C. (o. J.). Supplementary Material for MaX-DeepLab: End-to-End Panoptic Segmentation with Mask Transformers.
- Ying, X. (2019). An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*, 1168, 022022. doi:10.1088/1742-6596/1168/2/022022

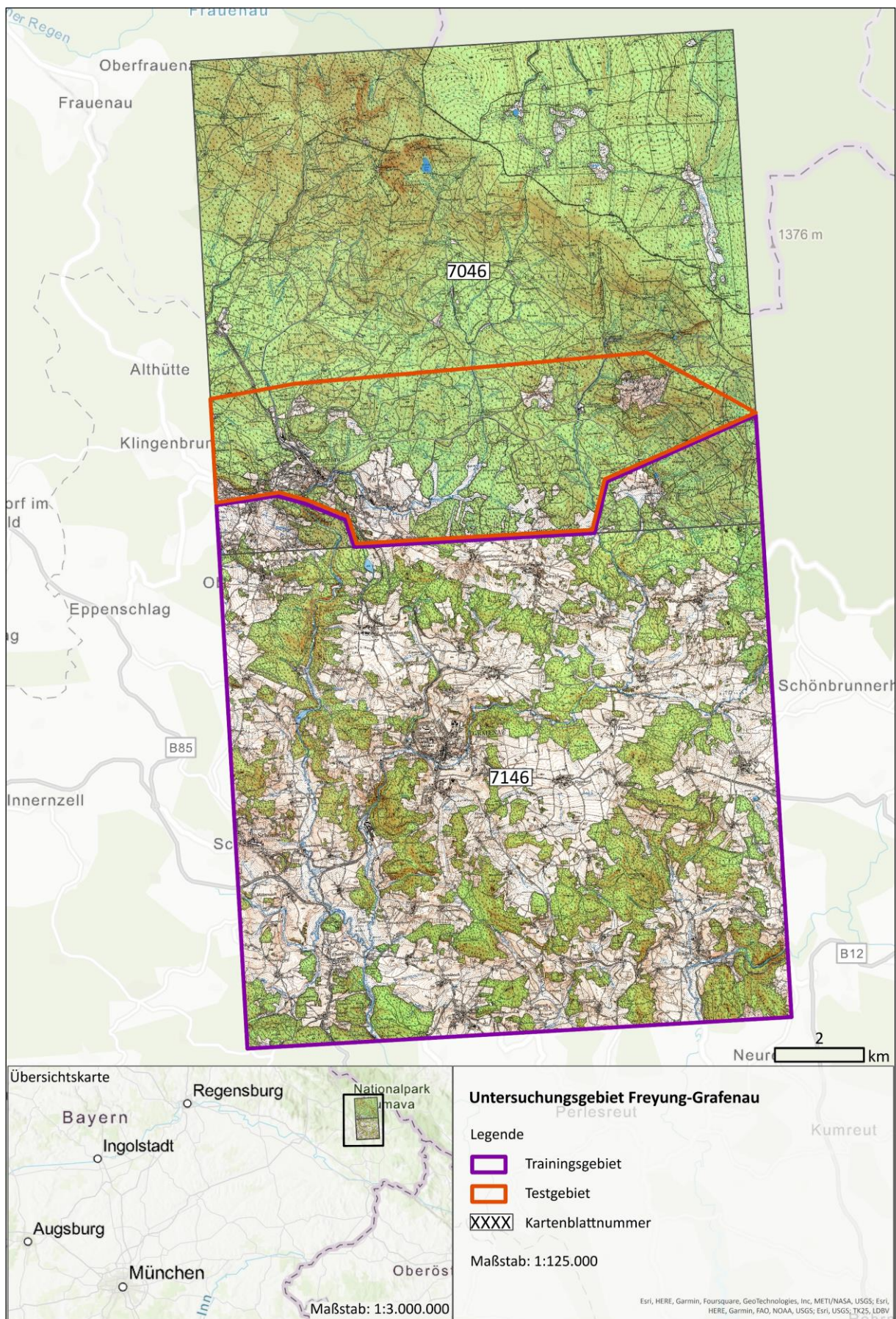
## Anlagen

- 1 Übersichtskarte: Lage des für die Analyse verwendeten Interessengebiets
- 2 Tabelle E1: Dokumentation der exportierten Trainingsdaten
- 3 Tabelle E2: Dokumentation der trainierten Modelle, Parametereinstellungen und Ergebnisse
- 4 Tabelle E3: Genauigkeitsübersicht der Modelle
- 5 Prozessgrafik zur Berechnung der Fehlermatrix
- 6 Prozessgrafik zur Nachverarbeitung
- 7 Weblinks zu den freigegebenen Paketen

Weitere Anlagen sind in digitaler Form auf der beigelegten Speicherkarte abgelegt und werden im Folgenden benannt.

- I Dokumentation der Export-/Parametereinstellungen (.xlsx)
- II Nachtrainierte Modelle
- III Kartenblätter 7046 und 7146 der TK25 als TIF
- IV Digitalisierte Siedlungsgebiete und Gebietsabgrenzungen (.shp)
- V Tabelle mit berechneten Konfusionsmatrizen
- VI Freigegebenes Deep-Learning- und Geoverarbeitungspaket

Anlage 1: Übersichtskarte: Lage des für die Analyse verwendeten Interessengebiets



Anlage 2: Tabelle E1: Dokumentation der exportierten Trainingsdaten

Nr	Ausgabeordner Name	Einstellungen Parameter							Ergebnisse Ausgabeordner						
		Klassenwertfeld	Puffer-radius	Bild-format	Kachel-größe XY	Schritt XY	Drehwinkel	NoFeature Kacheln	Metadatenformat	Min Polygon-überlappung	Größe Images [GB]	Anzahl Image Dateien	Anzahl TIFFS	Größe Labels [MB]	Anzahl Label Dateien
1	256_0,5_30_PS	None	0	TIFF	256	128	30	X	Panoptische Segmentierung	-	1.77	22344	11172	31.1	33516
2	128_0,5_30_PS_mU0,2	None	0	TIFF	128	64	30	X	Panoptische Segmentierung	0.2	0.50	23264	11632	19.8	34896
4	256_0,4_20_PS_mU0,2	None	0	TIFF	256	96	20	X	Panoptische Segmentierung	0.2	4.36	54432	27216	77.5	81648
5	256_0,5_30_RCNN	None	0	TIFF	256	128	30	X	RCNN-Masken	-	1.77	22344	11172	31.2	33432
8	S_224_0,5_30_RCNN_mU0,2	None	0	TIFF	224	112	30	X	RCNN-Masken	0.2	1.02	16320	8160	23.1	24480
10	S_256_0,4_30_PS_mU0,2	None	0	TIFF	256	96	30	X	Panoptische Segmentierung	0.2	3.62	44752	22376	62.6	67128
11	Klasskacheln	None	0	TIFF	256	128	30	X	Klassifizierte Kacheln	-	1.93	24744	12372	34.2	37116

Anlage 3: Tabelle E2: Dokumentation der trainierten Modelle, Parametereinstellungen und Ergebnisse

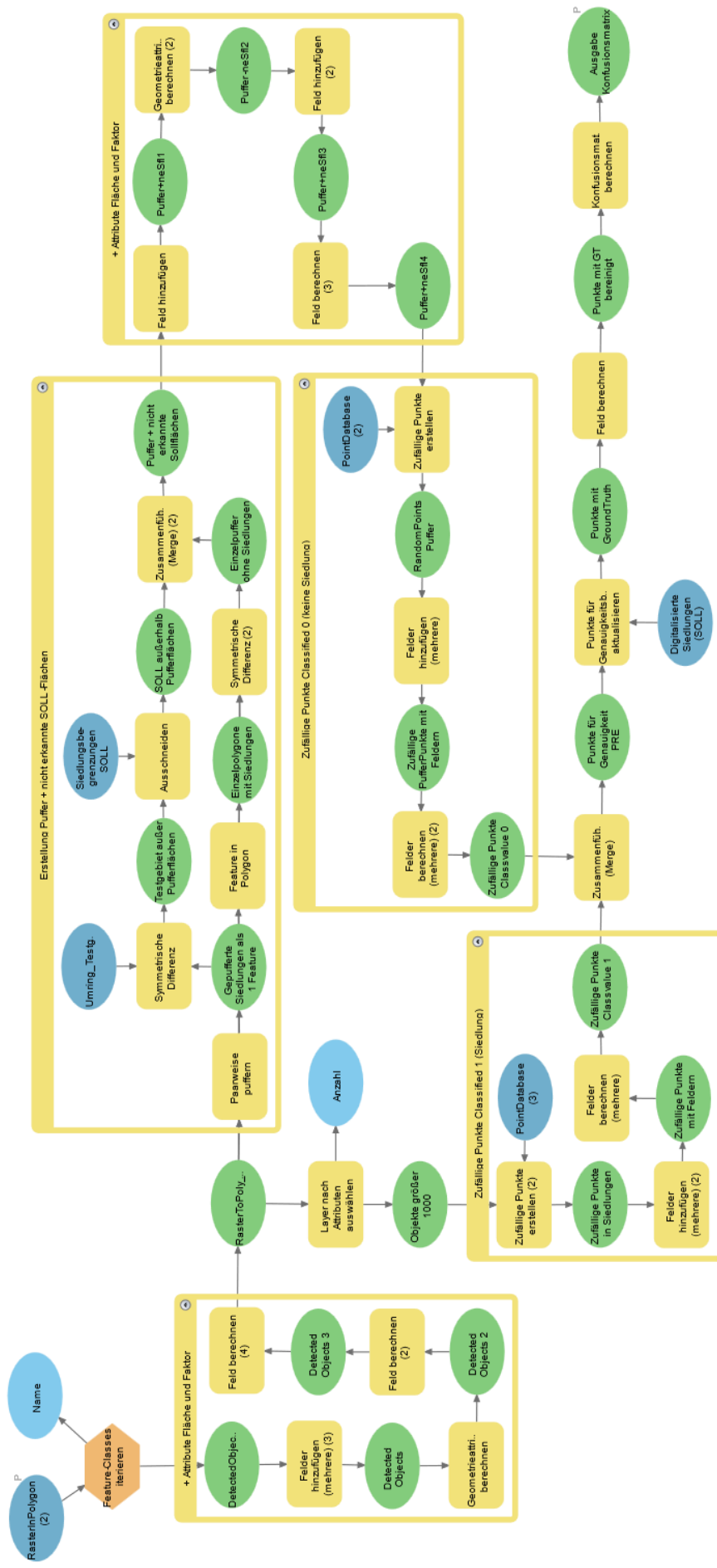
Nr	Ausgabemodell Name	Einstellungen Parameter [Processor:GPU]										Ergebnisse Ausgabebereiner						
		Allgemein			Modellargumente				Erweitert			Panoptic Quality /OA	Anzahl erkannte Objekte	Durchgeführte Epochen	Letzter Training Loss Wert	Letzter Validation Loss Wert	Dauer Training [hrs]	
		Eingabe Trainingsdaten	Backbone Modell	Max Epoch	Modelltyp	Batch-Größe	num-chip	Monitor Size	Valid-Lernrate	Vorzeitiges Beenden	Modell-frieren							
1	ResNet18_TrainedOn_256.0.5_30_PS_BatchSize16	256.0.5_30_PS	ResNet-18	20	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.6979	7/6	20	0.6979	0.7073	4:05:28
2	VGG19BN_TrainedOn_256.0.5_30_PS_BatchSize16	256.0.5_30_PS	VGG-19 BatchN	20	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.7059	3/6	11	0.8741	0.9697	2:22:01
3	DarkNet53_TrainedOn_256.0.5_30_PS_BatchSize16	256.0.5_30_PS	DarkNet-53	20	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.7807	4/5	20	0.6398	0.698	3:59:28
4	MobileNet2_TrainedOn_256.0.5_30_PS_BatchSize16	256.0.5_30_PS	MobileNet-V2	20	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.7783	4/5	20	0.7301	0.6896	3:47:24
<i>Batch Size verändert</i>																		
5	VGG11_TrainedOn_256.0.5_30_PS	256.0.5_30_PS	VGG-11	20	Max-Deeplab	8	30	256	valid_loss	Default	20	✓	0.6579	0/6	9	1.1989	1.1485	3:54:25
6	ResNet18_TrainedOn_256.0.5_30_PS_BatchSize4	256.0.5_30_PS	ResNet-18	20	Max-Deeplab	4	10	256	valid_loss	Default	20	✓	0.7507	7/5	7	0.9529	0.9356	2:13:08
7	MobileNet2_TrainedOn_256.0.5_30_PS_BatchSize8	256.0.5_30_PS	MobileNet-V2	20	Max-Deeplab	8	10	256	valid_loss	Default	20	✓	0.7813	5/6	20	0.4639	0.6783	4:49:21
<i>Kachelgröße/Trainingsdaten verändert, RCNN-Maske</i>																		
8	ResNet18_TrainedOn_128.0.5_30_PS_mu0.2_BatchSize32	128.0.5_30_PS_mu0.2	ResNet-18	20	Max-Deeplab	32	10	128	valid_loss	Default	20	✓	0.4857	2/5	20	2.859	2.5784	1:57:58
9	ResNet50_TrainedOn_224.0.5_30_RCNN_BatchSize2_30	S_224.0.5_30_RCNN_mu0.2	ResNet-50	30	MaskRCNN	32	-	224	valid_loss	Default	20	✓	0.0000	0/5	30	0.0843	0.0879	2:28:37
10	ResNet34_TrainedOn_256.0.5_30_RCNN_BatchSize16	256.0.5_30_RCNN	ResNet-34	30	MaskRCNN	16	-	256	valid_loss	Default	20	✓	0.3908	6/4	30	2.5598	1.0978	4:03:06
<i>Erhöhte Epochenanzahl</i>																		
11	DarkNet53_TrainedOn_256.0.4_20_PS_BatchSize16_100	256.0.4_20_PS_mu0.2	DarkNet-53	100	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.7055	8/11	19	0.3222	0.9428	8:31:04
12	MobileNet2_TrainedOn_256.0.4_20_PS_mu0.2_BatchSize16_150E	256.0.4_20_PS_mu0.2	MobileNet-V2	150	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.7222	6/8	16	0.2246	0.9011	7:43:12
13	ResNet18_TrainedOn_256.0.5_30_PS_BatchSize16_50	256.0.5_30_PS	ResNet-18	50	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.7357	4/5	18	0.9196	0.8409	3:48:42
<i>Veränderte Trainingsdaten, Standardisiert</i>																		
14	ResNet18_TrainedOn_S_256.0.4_30_PS_mu0.2_BatchSize16_50	S_256.0.4_30_PS_mu0.2	ResNet-18	50	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.6854	4/7	16	0.7052	0.9726	6:50:54
15	VGG16B_TrainedOn_S_256.0.4_30_PS_mu0.2_BatchSize16_20	S_256.0.4_30_PS_mu0.2	VGG-16 BatchN	20	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.7265	6/8	20	0.6264	0.8008	8:07:17
16	DenseNet121_TrainedOn_S_256.0.4_30_PS_BatchSize16_30	S_256.0.4_30_PS_mu0.2	DenseNet-121	30	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.7623	4/7	30	0.5442	0.7923	12:14:57
17	MobileNet2_TrainedOn_S_256.0.4_30_PS_mu0.2_BatchSize16_30E	S_256.0.4_30_PS_mu0.2	MobileNet-V2	30	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.7680	5/7	30	0.5272	0.8075	13:20:28
18	DarkNet53_TrainedOn_S_256.0.4_30_PS_mu0.2_BatchSize16_V10	S_256.0.4_30_PS_mu0.2	DarkNet-53	20	Max-Deeplab	16	10	256	valid_loss	Default	10	✓	0.7392	7/8	20	0.7935	0.8413	8:26:23
19	DarkNet53_TrainedOn_S_256.0.4_30_PS_mu0.2_BatchSize16_50E	S_256.0.4_30_PS_mu0.3	DarkNet-53	50	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	0.6564	4/7	10	1.2314	1.3758	5:17:42
<i>Löyer Net Fixed</i>																		
20	DarkNet53_TrainedOn_256.0.5_30_PS_BatchSize16_30E_NF	256.0.5_30_PS	DarkNet-53	30	Max-Deeplab	16	6	256	valid_loss	Default	20	✓	x	2/5	30	0.7827	0.7386	5:13:59
21	MobileNet2_TrainedOn_256.0.5_30_PS_BatchSize16_50E_NF	256.0.5_30_PS	MobileNet-V2	50	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	x	4/5	50	0.7127	0.6979	8:41:46
22	ResNet18_TrainedOn_256.0.5_30_PS_BatchSize16_50E_NF	256.0.5_30_PS	MobileNet-V3	50	Max-Deeplab	16	10	256	valid_loss	Default	20	✓	x	4/6	16	1.9676	1.7979	2:43:21
<i>U-Net</i>																		
23	ResNet18_TrainedOn_KlassKacheln_BatchSize16	KlassKacheln	ResNet-18	20	U-Net	16	10	256	valid_loss	Default	20	✓	0.9800	2/3	20	0.0389	0.0533	2:10:36
24	ResNet34_TrainedOn_KlassKacheln_BatchSize16_50E	KlassKacheln	ResNet-34	50	U-Net	16	10	256	valid_loss	Default	20	✓	0.9800	3/3	47	0.0319	0.0407	5:13:51

Anlage 4: Tabelle E3: Genauigkeitsübersicht der Modelle absteigend nach OA sortiert – Angaben aus der Konfusionsmatrix für das Testgebiet, Training & Validation Loss aus den Ergebnissen des Trainingswerkzeugs, ohne Modell 5 & 9 (Tabelle E2)

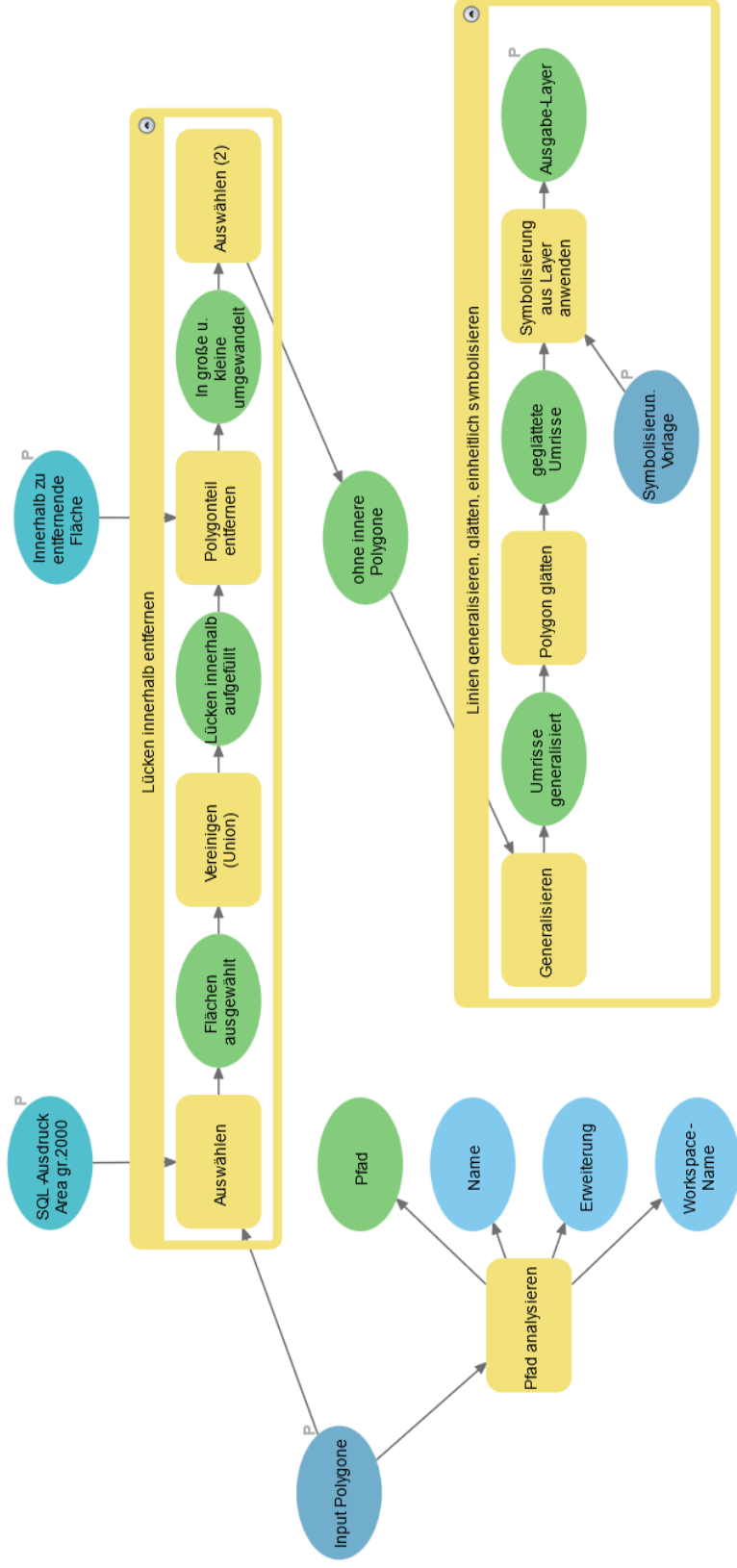
Nr	Name	OA	Kappa	TN	FN	FP	TP	PA_CO	PA_C1	UA_CO	UA_C1	TrainLoss	ValidLoss	Tr:Val
1	ResNet34_TrainedOn_KlassKachelIn_BatchSize16_50E	0.8961	0.7332	24002	1376	2273	7465	0.913	0.844	0.946	0.767	0.03	0.04	1:1
2	ResNet18_TrainedOn_KlassKachelIn_BatchSize16_	0.8785	0.6903	23425	1713	2523	7214	0.903	0.808	0.932	0.741	0.71	0.97	1:1
3	DarkNet53_TrainedOn_s_256_0,4_30_PS_mU0,2_BatchSize16_V10CP	0.8785	0.7197	17275	1645	1733	7141	0.909	0.813	0.913	0.805	0.79	0.84	1:1
4	VGG16B_TrainedOn_s_256_0,4_30_PS_mU0,2_BatchSize16_30ECP	0.8719	0.7030	17849	1584	2096	7196	0.895	0.820	0.918	0.774	0.08	0.09	1:1
5	DenseNet121_TrainedOn_s_256_0,4_30_PS_mU0,2_BatchSize16_30ECP	0.8682	0.6913	17973	1738	2049	6977	0.898	0.801	0.912	0.773	0.54	0.79	1:1
6	MobileNet2_TrainedOn_256_0,5_30_PS_BatchSize16CP	0.8674	0.7081	15832	1082	2491	7548	0.864	0.875	0.936	0.752	0.71	0.70	1:1
7	MobileNet2_TrainedOn_256_0,5_30_PS_BatchSize16_50E_NFCP	0.8663	0.7042	15324	1506	1976	7240	0.886	0.828	0.911	0.786	0.73	0.69	1:1
8	DarkNet53_TrainedOn_256_0,5_30_PS_BatchSize16CP	0.8637	0.7043	14933	1113	2434	7544	0.860	0.871	0.931	0.756	0.78	0.74	1:1
9	DarkNet53_TrainedOn_256_0,5_30_PS_BatchSize16_30E_NFCP	0.8637	0.6987	15561	1344	2270	7336	0.873	0.845	0.920	0.764	0.64	0.70	1:1
10	ResNet18_TrainedOn_256_0,4_30_PS_mU0,2_BatchSize16_50ECP	0.8622	0.6397	23444	2216	2578	6560	0.901	0.747	0.914	0.718	0.70	0.71	1:1
11	ResNet18_TrainedOn_256_0,5_30_PS_BatchSize16CP	0.8608	0.6873	15442	1962	1643	6853	0.904	0.777	0.887	0.807	0.95	0.94	1:1
12	ResNet18_TrainedOn_256_0,5_30_PS_BatchSize16_50ECP	0.8569	0.6711	18869	1129	3287	7564	0.852	0.870	0.944	0.697	1.97	1.80	1:1
13	MobileNet2_TrainedOn_256_0,4_20_PS_mU0,2_BatchSize16_150ECP	0.8549	0.6821	15189	1400	2422	7335	0.862	0.840	0.916	0.752	0.22	0.90	1:4
14	MobileNet2_TrainedOn_256_0,5_30_PS_BatchSize8CP	0.8543	0.6777	14731	1841	1850	6904	0.888	0.789	0.889	0.789	0.46	0.68	1:1
15	MobileNet2_TrainedOn_s_256_0,4_30_PS_mU0,2_BatchSize16_30ECP	0.8535	0.6626	16756	2025	2011	6760	0.893	0.769	0.892	0.771	0.53	0.81	1:2
16	DarkNet53_TrainedOn_256_0,4_20_PS_mU0,2_BatchSize16_100ECP	0.8510	0.6759	15802	930	3178	7663	0.833	0.892	0.944	0.707	0.32	0.94	1:3
17	VGG19B_TrainedOn_256_0,4_30_PS_BatchSize16CP	0.8313	0.6209	14662	2533	1716	6273	0.895	0.712	0.853	0.785	1.20	1.15	1:1
18	ResNet18_TrainedOn_256_0,5_30_PS_BatchSize4CP	0.8093	0.5380	20676	2270	4136	6515	0.833	0.742	0.901	0.612	0.04	0.05	1:1
19	DarkNet53_TrainedOn_s_256_0,4_30_PS_mU0,2_BatchSize16_50ECP	0.8060	0.5758	18731	708	5693	7870	0.767	0.917	0.964	0.580	1.23	1.38	1:1
20	ResNet18_TrainedOn_256_0,5_30_PS_BatchSize16_50E_NFCP	0.7277	0.4430	9056	4314	778	4551	0.921	0.513	0.677	0.854	0.92	0.84	1:1
21	ResNet18_TrainedOn_128_0,5_30_PS_BatchSize32CP	0.6193	0.0243	14403	8112	1324	948	0.916	0.105	0.640	0.417	2.86	2.58	1:1
22	ResNet34_TrainedOn_256_0,5_30_RCNN_BatchSize16	0.6071	0.2191	4034	5120	1941	6878	0.675	0.573	0.441	0.780	2.56	1.10	2:1



Anlage 5: Workflow zur Erstellung einer Konfusionsmatrix im ArcGIS Model Builder



Anlage 6: Postprocessing Workflow zur Generalisierung der Polygongrenzen erstellt im ArcGIS Modelbuilder



## Anlage 7: Weblinks

Folgende Dateien wurden vorläufig für die Organisation THWS in ArcGIS Online freigegeben:

### I. **Deep-Learning-Paket**

- mit U-Net trainiertes ResNet-34-Modell als .dlpk



[\[https://thws.maps.arcgis.com/home/item.html?id=ce5665a9f7884e41bbf30634f2f9c4c0\]](https://thws.maps.arcgis.com/home/item.html?id=ce5665a9f7884e41bbf30634f2f9c4c0)

### II. **Geoverarbeitungspaket**

- kurzer Arbeitsablauf zum Schließen von Lücken und Generalisieren der erkannten Siedlungspolygone



[\[https://thws.maps.arcgis.com/home/item.html?id=c902cc2b48764a42b0f0091a11828b81\]](https://thws.maps.arcgis.com/home/item.html?id=c902cc2b48764a42b0f0091a11828b81)