Hochschule für Technik und Wirtschaft Dresden
Fakultät Informatik/Mathematik

Semi-autonome Klassifizierung archäologischer Strukturen

Masterarbeit

zur Erlangung des akademischen Grades Master of Science im Studienfach Angewandte Informatik

Eingereicht von:

Huy Do Duc

Betreuer:

Prof. Dr. Marco Block-Berlitz

Dr. phil. Hendrik Rohland

Eingereicht am: 13.04.2021

Inhaltsverzeichnis

Kapitel 1

Motivation und Einführung

Für Archäologen gehört die Mongolei zu den interessantesten Ländern der Welt. In dem einstmals größten zusammenhängenden Herrschaftsgebiet der Erde gibt es noch heute außergewöhnlich viele archäologische Stätten. Nicht wenige sind von Zerstörung oder Verfall bedroht, etwa durch Klimawandel, aggressiven Bergbau und Plünderungen. Um das archäologische Erbe der Mongolei zu erforschen und zu dokumentieren, werden Satellitendaten mittels Künstlicher Intelligenz ausgewertet, womit sich auch diese Arbeit beschäftigt.

Ziel der Arbeit "Semi-autonome Klassifizierung archäologischer Strukturen" ist es, mithilfe von Machine-Learning, Stadt- und Wallanlagen im Gebiet der heutigen Mongolei zu erforschen. Für die Objekterkennung sollen Daten von Sentinel-2, einem Satelliten des Copernicus Projekts der ESA (European Space Agency), verwendet werden. Diese Satellitenbilder besitzen mehr Spektralbereiche als herkömmliche RGB-Bilder (Rot-Grün-Blau-Bilder), und können somit detailliertere Informationen über die Geologie liefern. Auf Grund der geringen Anzahl bereits dokumentierter archäologischer Stätten in der Mongolei wird das Verfahren des Transfer-Learnings angewandt. Dabei werden neu gewonnene Datensätze mit sogenannten vortrainierten Modellen (engl. Pretrained-Models) kombiniert.

1.1 Das Untersuchungsgebiet Mongolei

Die Mongolei ist flächenmäßig eins der größten Länder der Welt (siehe Abbildung 1.1). Die Bevölkerungsdichte ist hingegen äußerst gering. Sie beträgt nur

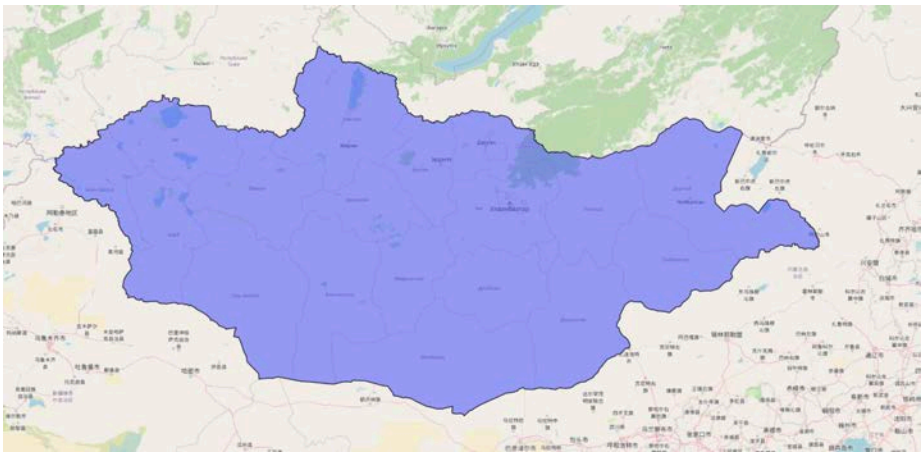

Abbildung 1.1: Die Mongolei ist das Untersuchungsgebiet. Es sollen Stadt- und Wallanlagen entdeckt werden. Dafür werden Satellitenbilder mit Machine-Learning-Algorithmen gescannt.

2 Einwohner pro Quadratkilometer. Die meisten Mongolen leben in oder um die Hauptstadt Ulaanbaatar. In dem riesigen Land gibt es mehrere Vegetationszonen. Dabei überwiegen Steppen und Halbwüsten. Die Vegetationsbedingungen und die unbewohnten weiten Flächen bieten gute Voraussetzungen für die Fernerkundung aus dem All. Die Satelliten, die in über 600 km Höhe die Erde umkreisen, sammeln Daten über Veränderungen der Landflächen, Landnutzung, Naturkatastrophen und Klima. Die Satellitenaufnahmen und künstliche Intelligenz sollen dabei helfen, die Arbeit der Archäologen zu vereinfachen.

1.2 Aufbau der Arbeit

In Kapitel 2 befasst sich mit verwandten Arbeiten und Grundlagen über Satellitenbilder und Machine-Learning vorgestellt. In Kapitel 3 wird das Wissen aus den Themengebieten Satellitenbilder und Machine-Learning angewandt. Es sollen eigene Datensätze aus Satellitenbildern erstellt. Im Kapitel 4 sollen die letzten Vorbereitungsschritte durchgeführt und mit den erstellten Daten trainiert. In Kapitel 5 werden die Ergebnisse ausgewertet und Optimierungsversuche getestet. Im letzten Kapitel folgt eine Zusammenfassung mit einen Ausblick, für weitere mögliche Schritte.

Kapitel 2

Verwandte Arbeiten und Theorien

In diesem Kapitel werden verwandte Arbeiten mit ähnlichen Forschungszielen vorgestellt. Es folgt ein Überblick der Eigenschaften und Verarbeitungsmethoden von Satellitenbildern. Außerdem werden Grundlagen von Machine-Learning, mit dem Fokus auf Objekterkennung, erklärt.

2.1 Verwandte Arbeiten

Werkzeuge aus der Informatik finden immer öfter Anwendung in der Archäologie. Fast verschwundene Straßen werden mit Modellierungen rekonstruiert [1]. In Drohnen-Luftbildern werden Autos erkannt, trotz extremer Wetterbedingungen. Such- und Rettungsteam können dadurch unterstützt werden [2]. Die Oberflächenbedeckungen in Satellitenbildern können durch Machine-Learning in mehrere Klassen eingeordnet werden, z.B. Bäume, Gebäude, steinige Flächen oder niedrige Vegetation [3]. Ein weiteres Beispiel ist die Vorhersage der Chronologie von mittelalterlichen Tempel in Angkor, Kambodscha [4].

Die Erfassung archäologischer Daten ist zeitintensiv und teuer. Die Analysen sind häufig durch schlechte Probegrößen und kleine Datensätze begrenzt. Angesichts der Art der archäologischen Daten ist es oft schwierig, korrekte Informationen zu erhalten [4]. Für die Verwendung datengetriebener Ansätze wurden Datensätze von Satellitenbildern für die Objekterkennung erstellt [5]. Dem Fern-

erkundungsfeld fehlt ein Bewertungsmaßstab, der z.B. dem ImageNet [6] gleicht. Es gab Versuch einen Maßstab für die Bildklassifizierungen von Satellitenbilder zu erstellen. Der Datensatz basiert auf vielfältigen Crowdsourcing-Daten. Mithilfe von Open-Street-Map-Daten konnten Bodenobjekte in Bildern markiert werden. In der Fernerkundung können diese Daten für die Bildklassifizierung nützlich sein [7].

Methoden aus der Fernerkundung werden mit Computer-Vision und Machine-Learning kombiniert [8]. Die Verwendung von Satellitenbildern können in der Fernerkundung für archäologische Forschung behilflich sein [9]. In den letzten Jahren wurde Machine-Learning immer mehr als Hilfsmittel in unterschiedlichen Disziplinen und Fachbereichen verwendet. Das Projekt WODAN verwendet Deep-Learning, um mit LiDAR-Daten in der Niederlande archäologische Objekte zu klassifizieren. Die Herausforderung, die Lokalisierung der Objekte in großen Bildern, soll mittels des Faster-R-CNN-Modells gelöst werden [10][11][12].

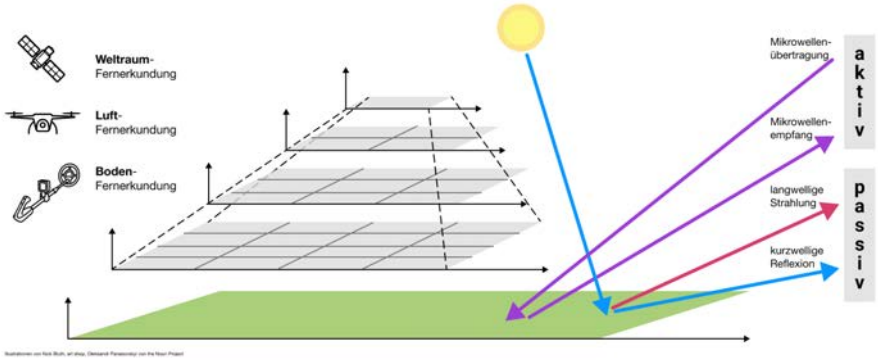

Abbildung 2.1: Bei der passiven Fernerkundung werden kurzwellige und langwellige Strahlung von einer Lichtquelle aufgenommen. Bei der aktiven Fernerkundung wird Energie von einer eigenen Strahlungsquelle ausgesendet. Mit steigender Entfernung zur Erdoberfläche werden unterschiedliche Werkzeuge verwendet. Passive Aufnahmen sind zum Beispiel Fotographie, multispektrale und hyperspektrale Aufnahmen. Synthetic-Aperture-Radar (SAR) ist ein Beispiel für aktive Aufnahmen [13].

2.2 Fernerkundung mit Satelliten

Auch wenn die Fernerkundungstechnologien nicht ursprünglich für archäologische Zwecke konzipiert wurden, sind sie in der Archäologie zu einem unverzichtbaren Werkzeug geworden [13]. Über eine Vielzahl von Sensoren sammeln Satelliten umfassende Informationen über den Zeitpunkt der Aufnahmen, den Sonnenstand und über den Zustand der Atmosphäre. Bei der Fernerkundung wird zwischen aktiver beziehungsweise passiver Aufnahme und der Entfernung der Aufnahmegeräte unterschieden (siehe Abbildung 2.1). Die passive Fernerkundung benötigt das Sonnenlicht, um kurzwellige oder langwellige Strahlungen aufzunehmen. Bei der aktiven Aufnahme wird Energie von einer Strahlungsquelle, meist Mikrowellen oder Laserlicht, ausgesendet und die reflektierten Strahlen wieder aufgenommen [13]. Satellitenbilder decken im Verhältnis zu LiDAR-Daten eine weitaus größere Fläche ab. LiDAR (Light-Detection-and-Ranging) ist eine optische Fernerkundungstechnik, bei der Laser für ein dichtes Abtasten der Erdoberfläche verwendet wird, um ein hochauflösendes digitales Modell der Geländeoberfläche zu erhalten.

Es gibt einige Anbieter, die Satellitenbilder frei zur Verfügung stellen. So werden zum Beispiel die Daten der Landsat- und Sentinel-Satellitenprogramme unent-

geltlich zur Verfügung gestellt [14]. Die Qualität, z.B. die Auflösung, kann dabei stark variieren. Frei verfügbare Daten haben oft den Nachteil einer geringen Auflösung. Ein Pixel bei einem Bild des Senitel-2-Satelliten repräsentiert 10×10 m. Bei einer Auflösung von 10980×10980 px lässt sich so eine Fläche von ca. 100 km² darstellen. Oft sind die hochauflösenden Satellitenbilder nur von privaten Unternehmen erhältlich. So bietet das Rapid-Eye-Satellite-Archive (RESA) Bilder in der Auflösung von $3,5$ m pro Pixel an.

2.2.1 Sentinel-Satelliten

Die hier behandelte archäologische Erkundung der Mongolei basiert auf Daten der Sentinel-Satelliten des Copernicus-Projektes der Europäischen Weltraumorganisation (ESA). Sie sind für verschiedenste Aufgabenbereiche konzipiert, beispielsweise für Meteorologie, Meeresforschung, Umwelt- und Klimaüberwachung.

Sentinel-1 ist ein Radarsatelliten-Paar. Es ist in der Lage bei Tag und Nacht unter allen Wetterbedingungen aussagekräftige Daten zu liefern. Diese werden unter anderem benötigt bei der Waldbewirtschaftung, der Meereskartierung oder bei humanitären Hilfsaktionen in Krisengebieten. Die Sentinel-1-Satelliten verfügen, abhängig von der Verwendung, über spezifische Modi. So gibt es einen Wave-Mode, der speziell für die Datenaufnahme bei der Untersuchung der Ozeane entwickelt wurde. Das Sentinel-3-Satellitenpaar beobachtet die sich ständig verändernde Atmosphäre in Abhängigkeit vom Zustand der Ozeane, der Eisflächen und Landmassen. Dafür werden Farbwerte, Temperatur und die Höhe des Meeresspiegels gemessen. Die Sentinel-5-Satelliten sammeln Daten zur Luftqualität, beispielsweise die Konzentration von Ozon, Methan, Formaldehyd, Kohlenmonoxid, Stickoxid und Schwefeldioxid in den verschiedenen Schichten der Atmosphäre [14].

2.2.2 Sentinel-2

Besonders wichtig für die archäologische Erkundung der Mongolei sind die von Sentinel-2 erstellten Daten. Die Sentinel-2-Mission besteht ebenfalls aus einer Konstellation von zwei Satelliten, die sich in der gleichen sonnensynchronen Umlaufbahn befinden. Sentinel-2 navigiert mit einem dualen GPS-Empfänger und verwendet ein Laser-System für die Kommunikation mit der Erde [14].

Die Auflösung der Satellitenbilder ist räumlich, temporal und radiometrisch zu betrachten:

Räumlich: Die Satelliten sind in einem Winkel von 180 Grad zueinander angeordnet. Die Schwadbreite beträgt 290 km. Dadurch decken die generierten Bilder die gesamte Erdoberfläche ab. In der Fernerkundung wird mit dem Schwad der Aufnahmestreifen eines Satelliten bezeichnet.

Temporal: Die Satelliten benötigen für eine Umrundung der Erde etwa 100 Minuten. Sie fliegen immer im gleichen Winkel zur Sonne, während sich die Erde unter ihnen dreht. So brauchen die Satelliten fünf Tage, um einmal von der kompletten Erdoberfläche Daten zu sammeln [14]. Durch die relativ schnelle Datenaufnahme können plötzlich eingetretene Veränderungen auf der Erdoberfläche registriert werden.

Radiometrisch: Mit Radiometrisch sind unterschiedliche Spektralbereiche gemeint. Multispektrale Bilder enthalten Informationen über reflektierte Strahlungen unterschiedlicher Wellenlänge.

Durch die vielfältigen Daten können Projekte in der Wald- und Vegetationsüberwachung, Raumplanung oder Erkundung von natürlichen Ressourcen unterstützt werden. Sentinel-2 ist geeignet, um größere Regionen in kurzen Zeitabständen zu überprüfen und vergleichende Zustandsanalysen von historischen Anlagen durchzuführen [15].

2.2.3 Multispektrale Bilder

Die meisten digitalen Bilder besitzen nur drei RGB-Kanäle (R steht für Rot, G für Grün und B für Blau). Die Satellitenbilder von Sentinel-2 und anderen Satelliten mit multispektralen Messinstrumenten umfassen hingegen dreizehn Kanäle, auch Bänder genannt (siehe Abbildung 2.2). Diese Bilder werden auch als multispektrale Bilder bezeichnet. Es gibt zwei Sentinel-2-Satelliten, die beide über ein multispektrales Aufnahmegerät verfügen. Dabei wird das Prinzip der Zeilenkamera verwendet. Als Zeilenkamera bezeichnet man einen Kameratyp, der nur eine lichtempfindliche Zeile aufweist, im Gegensatz zu einem zweidimensionalen Sensor, der über eine Vielzahl von Zeilen verfügt [14]. Das von der Erde reflektierte Licht wird von einem Drei-Spiegel-Teleskop gesammelt. Das Aufnahmegerät verfügt über zwei Brennebenen, zwölf Detektoren für

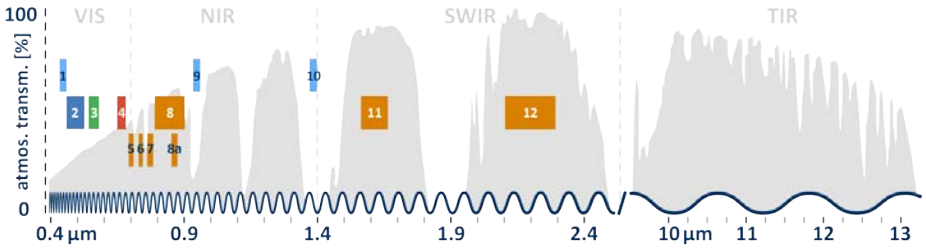

Abbildung 2.2: Als Y-Achse wird die atmosphärische Transmission angegeben und als X-Achse das Wellenspektrum. Die TIR-Daten (thermal-infrarot) werden von dem Sentinel-2-Satellitenpaar nicht aufgenommen [14].

sichtbares und nahes Infrarotspektrum und zwölf Detektoren für kurzwelliges Infrarotspektrum. Die Detektoren sind staffelförmig angeordnet, um die komplette Schwadbreite abzudecken. Die Satellitenbilder werden in zwei räumlichen Auflösungen im Bereich von 490 bis 2200 nm aufgenommen [14]. Vier der Spektralkanäle, Band 2, 3, 4 und 8, haben eine Auflösung von 10 m. Die Bänder 2, 3 und 4 sind die Bänder für Blau, Grün und Rot und werden für die Erstellung eines True-Color-Bildes kombiniert. Das Band 8 repräsentiert die nah-infraroten Wellenlängen. In der Tabelle 2.1 sind auch die Bänder mit der 20 m und 60 m Auflösung und ihre Verwendung dargestellt.

2.2.4 Atmosphärische Korrektur

Die Umlaufbahn wird in Granulate aufgeteilt. Ein Granulat ist der kleinste unteilbare Abschnitt, der alle möglichen Spektralbänder enthält. Die Aufnahmen, die 100×100 km abbilden, befinden sich in der UTM/WGS84-Projektion. Das UTM-System (Universal-Transverse-Mercator) unterteilt die Erdoberfläche in 60 Zonen. Jede UTM-Zone hat eine vertikale Breite von 6 Längengraden und eine horizontale Breite von 8 Breitengraden. Die Pixelkoordinaten, die die Georeferenz angeben, liegen in der oberen linken Bildecke. Die Satellitendaten werden von der ESA in unterschiedlich vorbearbeiteten Standards angeboten.

Produkt 1C: Die 1C-Daten sind seit Juni 2015 verfügbar. Diese Daten beinhalten Informationen über Meeresspiegel, Wolkenmaskierungen und den Wasserdampf in der Atmosphäre. Aus den Rohdaten werden digitale Höhenmodelle erstellt, um die Bilder zu kartografieren. Mit einer konstanten Ground-Sampling-Distanz werden die Daten abhängig von der Auflösung der einzelnen

Tabelle 2.1: Sentinel-2-Satellitenbilder sind multispektrale Informationen. Die Kanäle haben verschiedene räumliche Auflösungen und Verwendungszwecke. Beispielsweise haben die Bänder 2, 3, 4 und 8 eine Auflösung von 10 m pro Pixel [14].

Band	Spektral	Auflösung [m]	Verwendung
1	Aerosole	60	Aerosole
2	Blau	10	Aerosole, Landnutzung, Vegetation
3	Grün	10	Landnutzung, Vegetation
4	Rot	10	
5	Red-edge	20	
6	Red-edge	20	
7	Red-edge	20	
8	NIR	10	Wasserdampf, Landnutzung, Vegetation
8a	Nahes NIR	20	
9	Wasserdampf	60	Wasserdampf
10	SWIR Zirrus	60	Zirruswolken
11	SWIR	20	Landnutzung, Vegetation
12	SWIR	20	Aerosole, Landnutzung, Vegetation

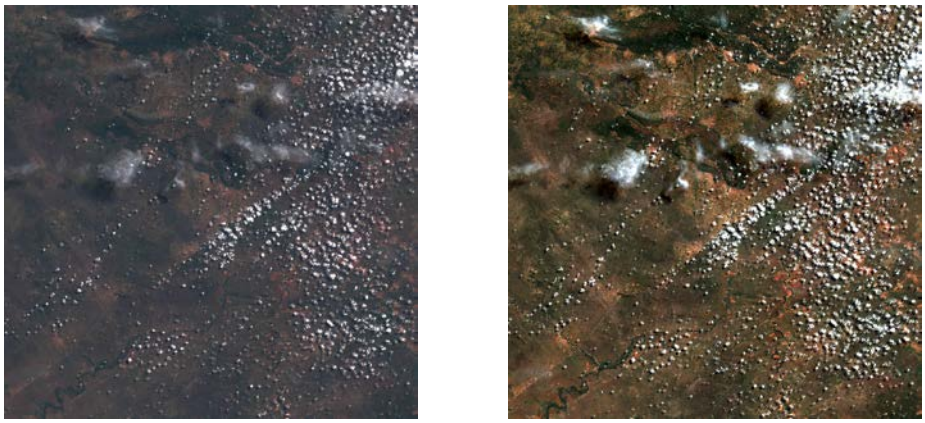

Abbildung 2.3: Hier wird der Unterschied zwischen TOA (links) und BOA (rechts) sichtbar. Vor der atmosphärischen Korrektur hat das TOA-Bild eine blau-graue Verfärbung.

Spektralbänder abgetastet. Die Ground-Sampling-Distanz bestimmt die Auflösung der Satellitenbilder. Je weiter die Satelliten von der Erde entfernt sind, desto größer ist die Ground-Sampling-Distanz (siehe Abbildung 2.1).

Die radiometrischen Messungen werden pro Pixel mit Reflexionen oberhalb der Atmosphäre gespeichert. Es wird von Top-of-Atmosphere (TOA) gesprochen. Aus dem Weltraum betrachtet, ist die Oberfläche der Erde neblig und bläulich (siehe Abbildung 2.3). Der Grund dafür ist die Mischung der Lichtstrahlen, die jeweils von der Erdoberfläche und von der Atmosphäre reflektiert werden. Dabei kommt im blauen Bereich des sichtbaren Lichts der größere Teil aus der Atmosphäre und nicht von der Erdoberfläche.

Produkt 2A: Die atmosphärische Korrektur ist eine Methode, mit der versucht wird, den Einfluss des von der Atmosphäre reflektierten Lichts aus dem Bild zu entfernen. Die 2A-Daten werden von dem Produkt 1C abgeleitet. Level-2A-Produkte werden seit März 2018 systematisch über Europa generiert. Seit Dezember 2018 wurde dies auf die gesamte Erdoberfläche ausgeweitet.

2.2.5 Verarbeitung von Satellitenbildern

Würde der Mensch von einem Satelliten aus auf die Erde schauen, wären die Farben des TCI sichtbar. Eine weitere Methode die Erdoberfläche zu sehen, ist die Erstellung von Falschfarbenbildern. Eine häufig verwendete Kombination ist

die von Nah-Infrarot, Grün und Rot. Diese Falschfarbendarstellung wird oft benutzt, um die Gesundheit von Pflanzen zu analysieren. Um Überschwemmungen und verbranntes Land visuell sichtbar darzustellen, wird kurzwelliges Infrarot, Nah-Infrarot und Grün verwendet.

Durch die multispektralen Messinstrumente können die Daten aus dem breiteren Wellenspektrum für Indexberechnungen genutzt werden. Spektrale Vegetationsindizes werden häufig bei der Klassifizierung der Landbedeckung und der Landnutzung verwendet [16]. Für die Erkundung der Mongolei wurden von den über 200 existierenden Indizes[1] folgende drei ausgewählt: NDVI, SAVI und NAI.

NDVI: Mit dem Normierten Differenzierten Vegetationsindex (NDVI) wird der Gesundheitszustand der Vegetation visuell dargestellt. Dafür werden die Spektralbereiche Rot und Nah-Infrarot benutzt. Je vitaler die Pflanze, desto intensiver sind die Reflexionen in diesen Spektralbereichen. Andere Oberflächenmaterialien, wie Sandboden, Fels oder auch tote Vegetation, zeigen andere Reflexionsgrade. Der NDVI wird mit folgender Formel berechnet:

$$NDVI = \frac{NIR - Rot}{NIR + Rot}$$

Durch die Normierung ergibt sich ein Wertebereich zwischen -1 und $+1$. Negative Werte bezeichnen Wasserflächen. Ein Wert zwischen 0 und 0.2 entspricht nahezu vegetationsfreien Flächen, während ein Wert nahe 1 auf eine dichte Vegetation mit grünen Pflanzen schließen lässt [17].

SAVI: Die Bodenhelligkeit in Gebieten mit geringer Vegetation verfälscht die Ergebnisse des NVDI. Der Soil-Adjusted-Vegetation-Index (SAVI) versucht dieses Problem zu korrigieren. Zusätzlich wird ein Bodenhelligkeitskorrekturfaktor (L) in die Berechnung einbezogen:

$$SAVI = \frac{NIR - Rot}{NIR + Rot + L} \times (1 + L)$$

Der Faktor für die Korrektur der Bodenhelligkeit wird abhängig von unterschiedlichen Bedeckungen angepasst. Standardmäßig ist der Faktor L mit $0,5$

[1] https://www.indexdatabase.de

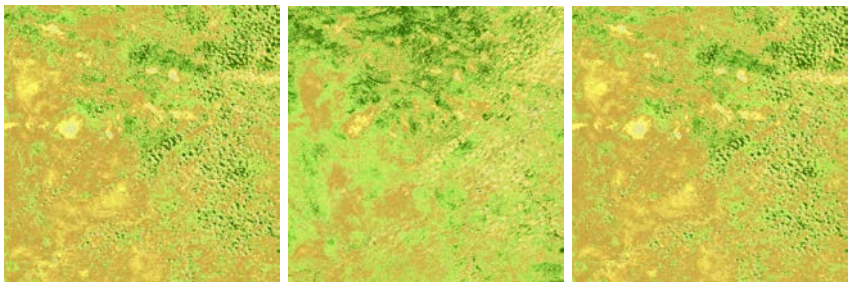

Abbildung 2.4: NVDI, SAVI und NAI. Für die Berechnung wurden Daten vom 30.03.2021 verwendet. Der SAVI mit einem Bodenhelligkeitsfaktor von 0,5 berechnet.

definiert.

NAI: Mit dem normalisierten Archäologieindex (NAI) sollen zugewachsene archäologische Stätten erkundet werden. Das Vegetationswachstum kann auf archäologische Überreste hinweisen, die sich nahe an der Erdoberfläche befinden [18]. NAI wird mit folgender Gleichung definiert:

$$NAI = (p800 - p700)(p800 + p700)$$

Die Spektralbereiche um 700 nm und 800 nm werden verwendet, um signifikante Merkmale hervorzuheben [19]. Die Berechnung von Indizes unterstützen die Identifizierung von archäologischen Merkmalen von spektralen Daten [20]. In der Abbildung 2.4 ist jeweils ein Beispiel von den drei genannten Indizes angezeigt.

2.2.6 Google-Earth-Engine

Unter Verwendung von Karten, 3D-Objekten und Satellitenbilder ist es mit Google-Earth möglich, die Erde zu erforschen. So können Archäologen auf der ganzen Welt ihre Daten und Forschungsergebnisse austauschen [21]. Die Google-Earth-Engine ist ein Tool zur Analyse von georeferenzierten Daten. Die Analysen können für die Landnutzung, Klimamonitoring oder Wald- und Wasserbedeckung benutzt werden. Der freie Zugang erlaubt es, schnell Informationen visuell darzustellen, zu filtern und zu analysieren [22]. Google verfügt über die Daten von mehreren Satellitenbildanbietern. Beispielweise durch das Landsat-Archiv

und Copernicus-Programm haben Analysen auf der Grundlage von Zeitreihendaten ermöglicht, die globale Dynamik der Landbedeckung zu untersuchen, die zuvor durch die geringe Datenverfügbarkeit begrenzt war. Eine solche Methode ist der Continuous-Change-Detection-and-Classification-Algorithmus (CCDC) [23]. Dabei werden Landsat-Daten verwendet, um zeitlich-spektrale Merkmale zu modellieren und Trends und spektrale Variabilität zu messen (siehe Abbildung 2.5). Für die Erkennung von Veränderungen werden Bilddaten aus unterschiedlichen Zeiten verglichen. Andere Projekte verwenden Machine-Learning, um ihre Ziele zu realisieren.

Das Training und Testen innerhalb der Google-Earth-Engine ist auf 100 MB pro Anfrage begrenzt. Die bessere Alternative ist TensorFlow, wo eine direkte Schnittstelle schon vorhanden ist. TensorFlow ist eine Open-Source-ML-Plattform, die fortschrittliche ML-Methoden wie Deep-Learning unterstützt. Obwohl Modelle mit dem Framework TensorFlow außerhalb von Earth Engine entwickelt und trainiert werden, bietet die Earth Engine-API Methoden zum Exportieren von Trainings- und Testdaten im TFRecord-Format und zum Import und Export von Bildern im TFRecord-Format. Das Thema Machine-Learning wird im nächsten Kapitel näher erläutert, mit speziellem Fokus auf faltenden neuronalen Netzen (engl. Convolutional-Neural-Network).

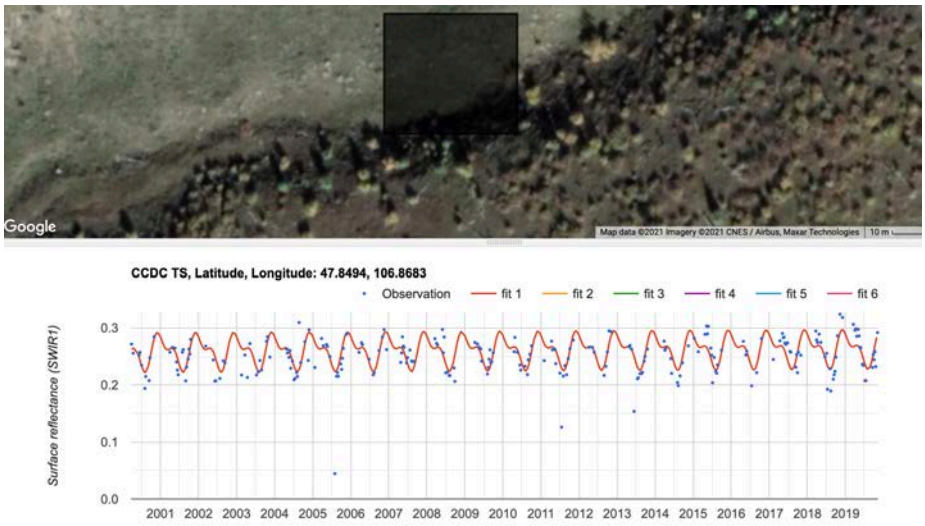

Abbildung 2.5: Der Bildausschnitt zeigt die zeitliche Änderung der Oberflächenreflektion von Anfang 2001 bis Ende 2019. Auf der Karte in Google-Earth wird ein Pixel dafür ausgewählt, wofür die zeitlichen Daten gesammelt werden. Es ist möglich verschiedene Punkte bzw. Pixel miteinander zu vergleichen.

2.3 Machine-Learning

Machine-Learning ist ein Teilgebiet der Künstlichen Intelligenz. Computer werden nicht direkt programmiert, sie erlernen bestimmte Fähigkeiten. Machine-Learning kann anhand unterschiedlicher Ansätze in drei Arten unterteilt werden. Das überwachte Lernen (engl. Supervised-Learning) ist eine Methode, wo einem Machine-Learning-Algoritmus (auch als Modell bezeichnet) ein Datensatz vorgegeben wird, bei dem die Zielvariablen bereits bekannt sind. Der Algorithmus erkennt die Zusammenhänge und Abhängigkeiten der Daten.

In dieser Arbeit werden Satellitenbilder von der Mongolei als Daten verwendet. In diesen Bildern sollen archäologische Objekte erkannt werden. Die Daten werden in Trainings-, Validierungs- und Testdaten eingeteilt. Jeder Datensatz hat eine bestimmte Aufgabe. Zum Erlernen der Aufgabe dienen die Trainingsdaten. Die Validierungsdaten werden für die Evaluierung des Modells verwendet. Mit den Testdaten wird evaluiert, wie das Modell auf unbekannte Bilder reagiert.

Das unüberwachte Lernen (engl. Unsupervised-Learning) wird ohne explizites Training verwendet, z.B. das Clustering-Verfahren. Es nimmt eine automatische Kategorisierung der Informationen vor und bildet Ansammlungen (Cluster).

Die Daten werden anhand ähnlicher Eigenschaften sortiert und kategorisiert. Beim bestärkenden Lernen oder verstärkenden Lernen (engl. Reinforcement-Learning) wird das Modell durch positive oder negative Rückmeldungen auf ein optimales Verhalten innerhalb einer Situation trainiert. In dieser Arbeit wird sich ausschließlich mit dem Supervised-Learning auseinandergesetzt.

2.3.1 Bildklassifizierung

Für den Menschen ist es eine einfache Aufgabe, ein Bild schnell einzuordnen. Dem Computer zu erklären, was auf einem Bild zu erkennen ist, ist eine komplexe Aufgabe. Als erstes werden die Klassen bestimmt, die angeben wonach auf dem Bild gesucht wird. Wird zum Beispiel ein Bild mit einem Hund betrachtet, muss der Klassifizierer, zwischen den Klassen Hund oder kein-Hund unterscheiden.

Herausforderungen sind Deformationen, Beleuchtung, verdeckte Objekte, Background-Clutter oder Intraclass-Variation. Beim Background-Clutter sind die Eigenschaften des Hintergrundes dem gesuchten Objekt zu ähnlich. Das Problem der Intraclass-Variation ist, dass es innerhalb der Klasse viele Variationen gibt, die zu unterscheiden sind. Lösungsansätze für diese Probleme werden in dem Fachbereich Computer-Vision entwickelt.

Es wurden Filter erstellt, um Kanten in Bildern zu entdecken. Mit den Kanten können Ecken erkannt werden. Für ein bestimmtes Objekt müssten eine Reihe von Regeln aufgestellt werden, um dies in einem Bild zu erkennen. Falls ein neues Objekt klassifiziert wird, müssen neue Regeln aufgestellt werden. Dafür wurden Machine-Learning-Algorithmen erstellt, die mit einem datengetriebenen Ansatz (Data-Driven-Approach), die die Klassifizierung automatisch durchführen können. Bei der Klassifizierung mit Trainingsbildern werden am Ende die Bilder einem Label zugeordnet. Labels sind die Zuweisungen der jeweiligen Klasse, zu dem das Bild oder Objekt gehört. Der allgemeine Ablauf vollzieht sich in fünf Schritten:

1. Trainingsdaten mit passenden Labels sammeln.
2. Einen Klassifizierer auswählen und gegebenenfalls Parameter einstellen.
3. Den Klassifizierer mit den Trainingsdaten füttern und trainieren.
4. Bilder klassifizieren.

5. Klassifikationsfehler mit einem Validationsdatensatz evaluieren.

Lernt man die Trainingsdaten auswendig, kann es zum Overfitting kommen. Die Erstellung von Test- und Trainingsdatensätzen ist eine Methode um Overfitting zu erkennen. Der Score des Testdatensatzes ist ein Versuch, ein objektives Maß für die Modellgüte zu finden, da diese Daten dem Netzwerk noch nicht bekannt sind. Ist der Score des Trainingdatensatzes signifikant besser, ist es ein Zeichen von Overfitting. Das bedeutet, der Algorithmus ist nicht generisch genug und kann neue Daten nicht richtig einordnen. Bei komplexeren Problemen können künstliche neuronale Netze weiterhelfen.

2.3.2 Neuron

Künstliche neuronale Netze sind Systeme, die von den biologischen neuronalen Netzen inspiriert sind. Ein neuronales Netz basiert auf einer Sammlung verbundener Knoten, die als künstliche Neuronen bezeichnet werden. Jede Verbindung kann wie die Synapsen in einem Gehirn ein Signal an andere Neuronen übertragen. Ein künstliches Neuron, das ein Signal empfängt, verarbeitet es und kann es einem verbundenen Neuronen weitergeben. Die Ausgabe eines Neurons bildet sich aus der Berechnung einer Funktion mit den Eingabewerten. Die Neuronen und Verbindungen sind mit Gewichtungen annotiert, die sich anpassen. Der Einfluss des Signals wird durch diese Gewichtungen bestimmt. Ein Schwellenwert bestimmt, ob ein Signal gesendet wird. Die Neuronen werden zu Schichten zusammengefasst und die Signale werden vom Eingang über die Schichten bis zum Ausgang transportiert.

Der elementare Grundbaustein jedes neuronalen Netzes ist das Neuron. Ein Neuron ist ein Knotenpunkt im neuronalen Netzwerk an dem ein oder mehrere Eingangssignale (Inputs) zusammentreffen und verarbeitet werden. Nach der Verarbeitung der Eingangssignale werden diese als Output an die nachfolgenden Neuronen weitergegeben.

In der Abbildung 2.6 ist der grundsätzliche Aufbau dargestellt. Auf der linken Seite der Abbildung treffen die Inputs x_1,...,x_n am Neuron ein. Input steht dabei für einen beliebigen numerischen Wert (z.B. vorhandene Daten oder Signale vorhergehender Neuronen). Das Neuron bewertet gewichtet den Input, berechnet den Output o_j und gibt diesen an die darauffolgenden, verbundenen Neuronen weiter.

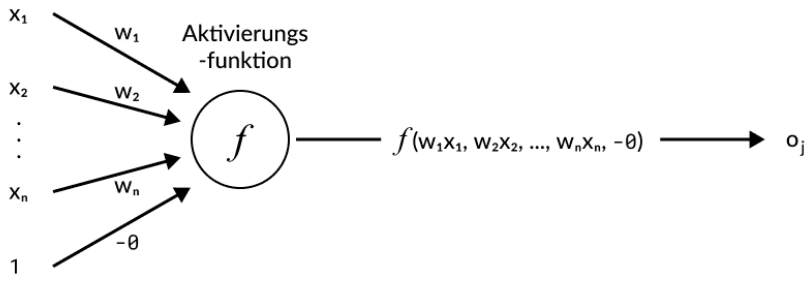

Abbildung 2.6: Hier ist ein Neuron dargestellt. Die Eingabewerte werden mit einer jeweiligen Gewichtung aufgenommen. Die Aktivierungsfunktion f das Ergebnis bzw. Output o_j.

2.3.3 Künstliche neuronale Netze

Die Fähigkeiten eines einzelnen Perzeptrons sind begrenzt. Komplexere Aufgaben werden mit mehrschichtigen Perzeptronen gelöst. Zwischen den Eingangs- und Ausgangsschichten liegen sogenannte Hidden-Layer. Durch die Hidden-Layer können nicht-lineare Funktionen realisiert werden. Verläuft der Informationsfluss nur in die nachfolgende Schicht, nennt man das ein Feed-Forward-Netz. In einem neuronalen Network werden die Ergebnisse der Schichten durch eine ReLU-Funktion (Rectified-Linear-Unit) aktiviert. Das bedeutet beispielsweise, dass alle Werte, die kleiner als Null sind, zu Null werden und alle Werte, die größer als Null sind 1:1 zu erhalten bleiben. Eine weitere Aktivierungsfunktion ist z.B die Sigmoid-Funktion. Die letzte Schicht erhält im Fall von Klassifizierungs-Problemen eine Softmax-Aktivierung. Das Ergebnis der Output-Neuronen wird normalisiert und gibt die Wahrscheinlichkeit an.

Weitere häufig verwendete Architekturen sind Fully-Connected- und Short-Cuts-Netze. Bei einem Fully-Connected-Netz sind alle Neuronen einer Schicht direkt mit allen Neuronen der folgenden Schicht verbunden. Short-Cuts-Netzen besitzen sogar noch mehr Verbindungen, denn dort haben die Neuronen direkte Verbindungen mit den übernächsten Schichten. Falls im Netz Neuronen existieren, die mit Neuronen der selben oder einer vorangegangen Schicht verbunden sind, nennt man dies ein rekurrentes neuronales Netz.

2.3.4 Backpropagation

Bei einem Feed-Forward-Netz fließt der Informationsfluss nur von der Eingabe- zur Ausgabeschicht und die Informationen können nicht rückwärts geschickt werden. Mithilfe von rückwärts laufenden Verbindungen wird die Möglichkeit geschaffen, nochmal in Neuronen von vorausgehenden Schichten zurückzukehren.

Die Backpropagation ist ein Spezialfall eines allgemeinen Gradientenverfahrens in der Optimierung. Es basiert auf dem mittleren quadratischen Fehler. Der Backpropagation-Algorithmus läuft in folgenden Phasen ab:

1. Ein Eingabemuster wird angelegt und vorwärts durch das Netz transportiert.
2. Die Ausgabe des Netzes wird mit der gewünschten Ausgabe verglichen. Die Differenz der beiden Werte wird als Fehler des Netzes erachtet.
3. Der Fehler wird nun wieder über die Ausgabe- zur Eingabeschicht zurück propagiert. Dabei werden die Gewichtungen der Neuronenverbindungen abhängig von ihrem Einfluss auf den Fehler geändert. Dies garantiert bei einem erneuten Anlegen der Eingabe eine Annäherung an die gewünschte Ausgabe.

2.3.5 Gradientenverfahren

Zur Quantifizierung des Modellfehlers wird eine Kostenfunktion berechnet. Bei neuronalen Netzen wird die Kostenfunktion E verwendet, die den mittleren quadratischen Fehler (Mean-Squared-Error, MSE) mit folgender Gleichung berechnet:

$$E = \frac{1}{n} \sum (y_i - o_i)^2$$

Der MSE berechnet zunächst für jeden Datenpunkt die quadratische Differenz zwischen y und o und bildet anschließend den Mittelwert. Das Ziel ist die Minimierung des Fehlers durch Anpassung der Gewichtungen. Die Kostenfunktion hängt von vielen Parametern im neuronalen Netz ab. Wird auch nur eine Gewichtung verändert, hat dies Auswirkungen auf nachfolgende Neuronen. Die Anpassung der Gewichtungen wird von der Lernrate beeinflusst. Die Lernrate steuert dabei, wie groß die Schritte in Richtung der Fehlerminimierung aus-

fallen. Das Vorgehen, die Kostenfunktion iterativ auf Basis von Gradienten zu minimieren, wird als Gradient-Descent bezeichnet [24].

Lernrate: Idealerweise würde die Lernrate so gesetzt werden, dass mit nur einer Iteration das Minimum der Kostenfunktion erreicht wird (2.7oben links). Da in der Praxis der korrekte Wert nicht bekannt ist, muss die Lernrate frei gewählt werden. Im Falle einer kleinen Lernrate kann man relativ sicher sein, dass ein Minimum der Kostenfunktion erreicht wird. Die Anzahl der Iterationen steigt in diesem Szenario bis zum Minimum deutlich an (2.7 oben rechts). Wenn die Lernrate größer als das theoretische Optimum gewählt wird, destabilisiert sich der Pfad zum Minimum. Die Anzahl der Iterationen sinkt, aber es gibt keine Sicherstellung, dass das lokale oder globale Optimum erreicht wird (2.7 unten links). In aktuellen Optimierungsschemata finden auch adaptive Lernraten Anwendung, die die Lernrate im Laufe des Trainings anpassen. So kann zu Beginn eine höhere Lernrate gewählt werden, die dann im Laufe der Zeit weiter reduziert wird, um dem lokalen Optimum am nächsten zu kommen [25].

2.3.6 Convolutional-Neural-Network (CNN)

Convolutional-Neural-Network ist eine Art neuronaler Netze zur Verarbeitung von Bildern. CNNs verwenden Convolution- und Pooling-Layer. Eine Konvolution ist eine Faltung, z.B. ein Matrixmultiplikation. Die Bilddaten werden mit einem Filter bzw. Kernel nach bestimmten Eigenschaften untersucht, wie z.B. der Kontrast. Der Kernel definiert die Größe der Matrix, mit der das Bild gefiltert wird. Die Tiefe eines Convolution-Schichts wird durch die Anzahl der Filter definiert. Das Bild durchläuft die Filter.

Mithilfe der Pooling-Schichts können Bilder komprimiert und überflüssige Informationen entfernt werden. Es gibt mehrere Arten von Pooling-Schichts. Beim Max-Pooling wird der höchste Wert einer Kernel-Matrix verwendet und alle anderen werden verworfen. Beispielsweise bei einer 3×3 Kernel erstellten Matrix, werden die Informationen auf den höchsten Wert reduziert (siehe Abbildung 2.8). Damit werden die relevantesten Signale an die nächsten Schichten weitergegeben. Beim Average-Pooling wird anstatt des höchsten Wertes, der zusammengerechnete Durchschnittswert verwendet.

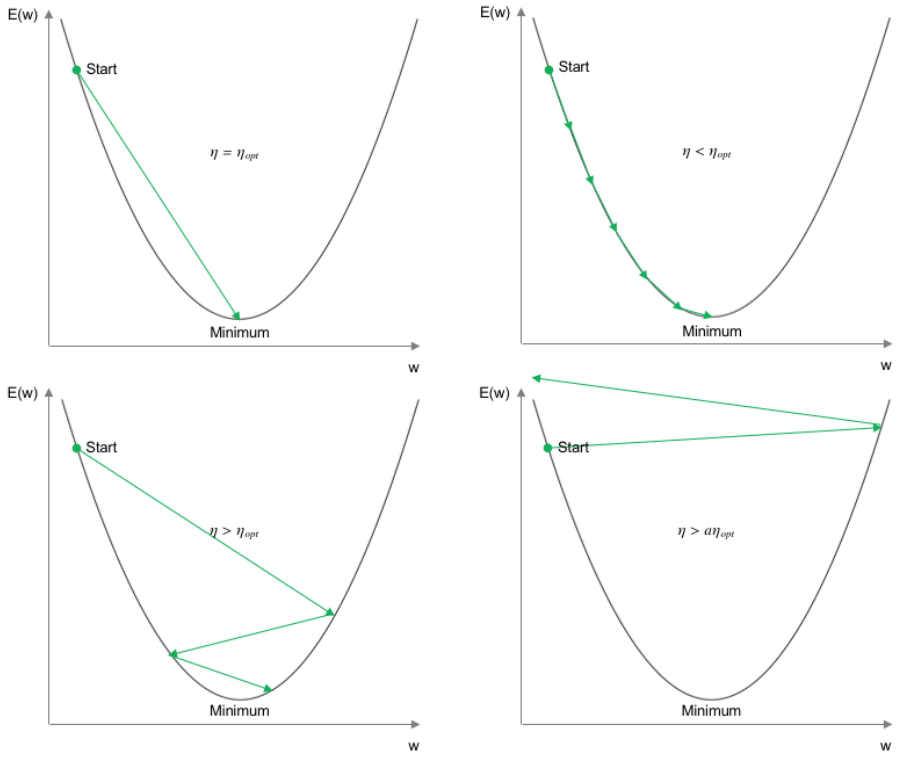

Abbildung 2.7: Es werden vier Beispiele angezeigt. Die unterschiedliche Ansätze zeigen, worauf bei Wahl der Lernrate geachtet werden muss. Beispielsweise im Falle einer zu hohen Lernrate wird keine Lösung für die Anpassung der Gewichtungen gefunden (unten rechts).

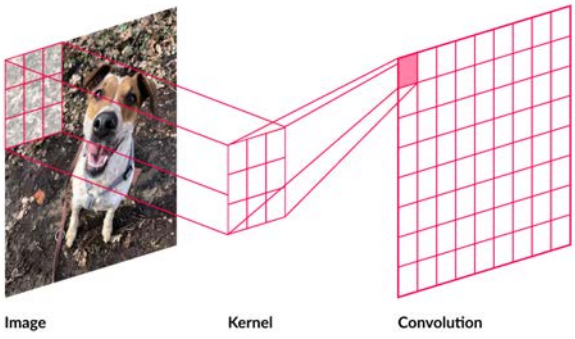

Abbildung 2.8: In einem Convolution-Layer wird das Bild mit einem Kernel, diesem Beispiel mit einer Dimension von 3×3, multipliziert.

2.4 Objekterkennung

Bei der Objekterkennung werden mithilfe von Computer-Vision Objekte in einem Bild erkannt. Ein Unterschied zur einfachen Klassifizierung von Bildern ist, dass die Anzahl der Objekte nicht festgelegt ist. Das Ziel ist die Klassifizierung und Lokalisierung aller Objekte innerhalb eines Bildes. Objektdetektoren können in zwei Kategorien unterteilt werden, in zweistufige und einstufige Detektoren. Zweistufige Detektoren weisen auf eine hohe Lokalisierungs- und Objekterkennungsgenauigkeit vor, während die einstufigen Detektoren eine hohe Geschwindigkeit erreichen [26]. Die einstufigen Detektoren sind geeignet für Echtzeitanwendungen [27]. Ein Vertreter einstufiger Detektoren ist der YOLO-Algorithmus. Beispiele für zweistufige Detektoren sind R-CNN [28], Fast-RCNN [29] und Faster-RCNN [30]. Im Folgenden werden diese detailliert erläutert.

2.4.1 R-CNN

Bei der Objekterkennung mit R-CNN werden in einem Bild als erstes Region-Of-Interests (ROI) erstellt, die angeben wo sich ein Objekt befinden könnte. Position und Größenverhältnis der Objekte sind unbekannt. Das Problem dabei ist die Erstellung der vielen möglichen Regionen, die schnell zu einem rechentechnischen Problem führen kann. R-CNN verwendet die Methode der selektiven Suche (engl. Selective-Search [31]). Diese Methode schlägt Regionen vor, die für die Objekterkennung interessant sein könnten. Diese Regionen werden an ein CNN weitergegeben. Das CNN schaut nach signifikanten Eigenschaften, extrahiert diese und gibt die Daten an einen Klassifizierer weiter. R-CNN versucht zwei Probleme zu lösen. Als Erstes versucht es, in einem Bild mithilfe den vorgeschlagenen Regionen Objekte zu identifizieren. Als Zweites will es die Wahrscheinlichkeiten berechnen, dass in diesen Regionen Objekte von einer bestimmten Klasse vorhanden sind.

Der Nachteil dieser Architektur ist die intensive Berechnung der Regionen mittels der selektiven Suche. Ein weiteres Problem ist, dass diese Regionen einzeln an das CNN weitergegeben werden, welches zu langen Trainingszeiten führt [28].

2.4.2 Fast R-CNN

In Fast-RCNN werden die Trainingsbilder direkt an ein CNN übergeben. Das Ergebnis sind Bilder, die nur signifikante Eigenschaften der Objekte anzeigen.

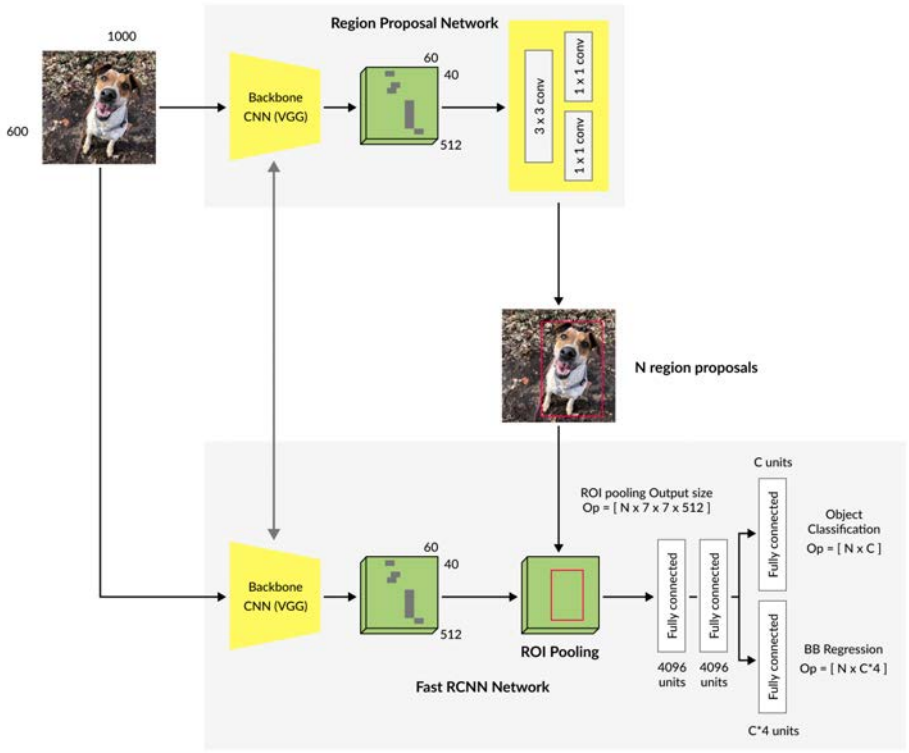

Abbildung 2.9: Faster-R-CNN besteht aus einem Region-Propsal-Network (RPN) und einem Fast-R-CNN. Das RPN ist ein selbstständiges neuronales Netz für die Erstellung von Region-Of-Interests [30].

Diese Bilder werden auch Feature-Maps genannt. Anschließend erfolgt die Erstellung der ROIs mittels der selektiven Suche [29]. Das Ergebnis umfasst die Klassifizierungsergebnisse und die Lokalisierung der Objekte. Mit einer Softmax-Schicht werden die Vorhersagen für die Klassifizierung ausgegeben und zusätzlich die Koordinaten für die jeweilige Box. Das Problem an diesem Algorithmus ist die lange Rechenzeit der ROI .

2.4.3 Faster R-CNN

R-CNN und Fast-R-CNN verwenden die selektive Suche, die rechenintensiv ist. In der Arbeit [30] wurde ein Algorithmus entwickelt, der die selektive Suche umgeht und das Netzwerk die ROIs selbst erlernen lässt. Ähnlich wie beim Fast-R-CNN werden die Bilder direkt in das CNN übergeben. Anstatt die selektive

Suche auf die Feature-Maps anzuwenden, wird ein separates Netzwerk benutzt, um die ROIs vorherzusagen. Die Vorschläge werden dann unter Verwendung einer Pooling-Schicht transformiert, um das Objekt innerhalb der Region zu klassifizieren und die Koordinaten vorherzusagen. Faster-R-CNN wird in dieser Arbeit für die Satellitenbilder angewandt. Aus diesem Grund wird die Abfolge von diesem Algorithmus näher erklärt.

Head: Mit Head ist hier ein vortrainiertes Modell gemeint. Dieses Modell dient als Feature-Extractor. Aus den Trainingsbildern werden Feature-Maps erstellt. Diese Feature-Maps komprimieren die Daten der Eingangsbilder, indem sie nur die besonderen Merkmale speichern.

Anchor-Generation-Layer: Hier werden in dem Trainingsbild eine feste Anzahl von Punkten erstellt, die ein Gitter bilden und die gleichen Abstände zueinander haben. Zu jedem einzelnen Punkt werden dann neun Anker bzw. Bounding-Boxen mit jeweils unterschiedlichen Seitenverhältnissen angelegt.

Region-Proposal-Layer: Der Region-Proposal-Layer besteht aus dem Proposal-Layer, Anchor-Target-Layer und dem Proposal-Target-Layer. Diese Schichten stellen das Region-Proposal-Network (RPN) dar. Der Proposal-Layer transformiert die Anker gemäß den Bounding-Boxen. Boxen die über den Seitenrand gehen, werden entfernt.

Anschließend folgt der Anchor-Target-Layer. Hier werden eine Reihe von potenziellen Bounding-Boxen auf Basis der Ground-Truth-Daten erzeugt. Das Ergebnis dieser Schicht wird zum Trainieren des RPN-Netzwerks verwendet. Für die Klassifizierung der Objekte werden diese potenziellen Bounding-Boxen nicht verwendet. Der Anchor-Target-Layer teilt die Bounding-Boxen in Vorder- und Hintergrund auf. Der Vordergrund steht für die gesuchten Objekte. Vielversprechende Bounding-Boxen für die Objekterkennung sind die, die mit den Ground-Truth-Daten überlappen. Je höher die Überlappung ist, desto wahrscheinlicher wird es, innerhalb der Bounding-Box ein Objekt zu erkennen. Der Anchor-Target-Layer gibt an, wie weit jedes Ankerziel von der nächstgelegenen Bounding-Box entfernt ist. Diese Regressoren sind nur für die Vordergrundboxen vorhanden, da keine Informationen über den Hintergrund existieren. Die RPN-Kostenfunktion wird während des Trainings des RPN-Netzwerks minimiert. Die Kostenfunktion ist eine Kombination aus dem Anteil der erzeugten Bounding-

Abbildung 2.10: Im ROI-Pooling-Layer wird das Bild entzerrt und die Region wird vom Rest des Bildes ausgeschnitten.

Boxen, die korrekt als Vorder- bzw. Hintergrund klassifiziert wurden und einem Abstandsmaß zwischen den vorhergesagten und den Zielregressionskoeffizienten.

$$RPN\ Loss = Classification\ Loss + Bounding\ Box\ Regression\ Loss$$

Nachfolgend bereitet der Proposal-Target-Layer die Vorschläge für den Classification-Layer vor. Zusätzlich werden mit dem Non-Maximum-Suppression-Verfahren (NMS) die Bounding-Boxen zurechtgeschnitten [30]. Bei zu großen Überlappungen werden manche Bounding-Boxen entfernt bzw. abgeschnitten.

ROI-Pooling-Layer: Die Aufgabe des ROI-Pooling-Layers ist die Extraktion der Regionen. Durch eine Transformationsmatrix werden die Feature-Maps entzerrt.

Classification-Layer: Der Classification-Layer leitet die erzeugten Feature-Maps an eine Reihe von Faltungsschichten weiter. Der Ausgang wird durch zwei Fully-Connected-Layer gespeist. Die erste Schicht erzeugt die Wahrscheinlichkeitsverteilung der Klassen für jeden Vorschlag vom RPN. Die zweite Schicht erzeugt die klassenspezifische Bounding-Box. Die Kostenfunktion bei Klassifizierung bildet sich aus erzeugten Bounding-Boxen und dem Abstandsmaß zwischen dem vorhergesagten und dem Zielregressionskoeffizienten ab.

Tabelle 2.2: Die Lernstrategien sind abhängig von der Menge und Ähnlichkeit der Daten.

	ähnliche Daten	unterschiedliche Daten
großer Datensatz	Fine-Tuning	Fine-Tuning oder neu trainieren
kleiner Datensatz	Am Ende des CNN	Am Anfang des CNN

2.4.4 Transfer-Learning

Beim Transfer-Learning werden fertig trainierte Schichten auf neue Datensets wiederverwendet. Entweder bleiben alle Schichten außer dem Output-Layer unverändert oder es werden einige gemäß dem aktuellen Trainingsstand angepasst und neu trainiert. Transfer-Learning ist eine Methode, den Rechenaufwand zu reduzieren und Trainingszeit einzusparen [32]. Das CNN lernt zunächst die relevanten Strukturen zu unterscheiden und mit diesem Wissen abstrakte Objekte daraus abzuleiten und neue zu erkennen. Trotz einer geringen Anzahl von Trainingsbildern kann ein vortrainiertes Modell dabei helfen, akzeptable Ergebnisse zu erreichen. Es gibt zwei Möglichkeiten, mit einem vortrainierten Modell umzugehen, Feature-Extraction und Fine-Tuning:

Feature-Extraction: Um aussagekräftige Merkmale aus neuen Beispielen zu extrahieren. Sie fügen einfach einen neuen Klassifikator über dem vorab trainierten Modell hinzu, der von Grund auf neu trainiert wird, damit Sie die zuvor für den Datensatz erlernten Feature-Maps neu verwenden können. Das gesamte Modell muss nicht neu trainiert werden. Der letzte Klassifizierungsteil des vortrainierten Modells ist jedoch spezifisch für die ursprüngliche Klassifizierungsaufgabe und muss auf die neuen Klassen angepasst werden [32].

Fine-Tuning: Bei der Feature-Extraction werden nur einige Ebenen auf einem Basismodell trainiert. Die Gewichtungen des vortrainierten Modells werden während des Trainings nicht aktualisiert. Eine Möglichkeit, die Leistung zu steigern, ist die Gewichtungen der obersten Schichten des vortrainierten Modells zusätzlich anzupassen [32]. In den meisten CNN ist eine Schicht umso spezialisierter, je höher sie liegt. In den ersten Ebenen werden sehr einfache und allgemeine Funktionen erlernt, die sich auf fast alle Arten von Bildern verallgemeinern lassen. Je höher Sie steigen, desto spezifischer werden die Funktionen

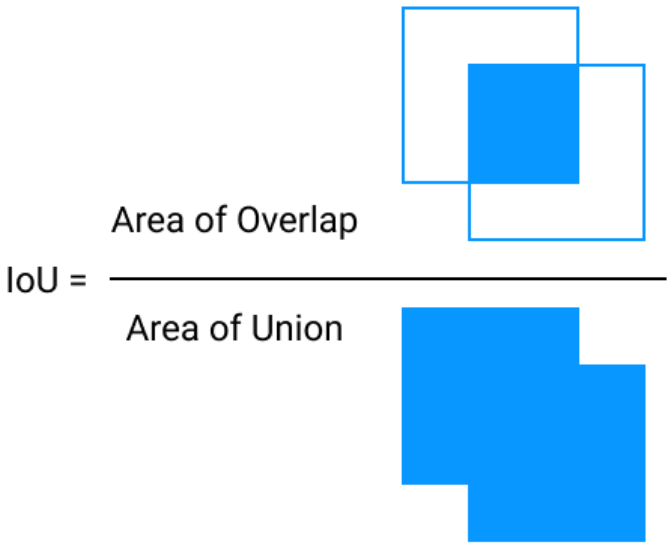

Abbildung 2.11: Der IoU wird aus der Überlappung und der gemeinsamen Menge von zwei Rechtecken berechnet.

für den Datensatz.

Die Tabelle 2.2 zeigt an, in welchen Fällen Fine-Tuning oder Feature-Extraction passend ist. Wenn sich die Daten zwischen vortrainierten und neuem Datensatz ähneln und ein großer Datensatz vorhanden ist, eignet sich das Fine-Tuning. Bei unterschiedlichen Daten und einem kleinen Datensatz von neuen Bildern sollte eine Feature-Extraction am Ende des CNN durchgeführt werden.

2.4.5 Evalidierungsmetriken

Für die Objekterkennung müssen die zwei Maße beachtet werden. Ob die Klassifizierung der Daten stimmt, d.h. werden die Bilder den richtigen Klassen zugeordnet. Außerdem müssen die Genauigkeit der Lokalisierung (der Bounding-Boxen) geprüft werden.

IoU: IoU (Intersection-Over-Union) misst die Überlappung zwischen zwei Grenzen. Es wird verwendet, um zu messen, wie sehr sich die vorhergesagte Grenze mit dem Ground-Truth (der realen Objektgrenze) überschneidet (siehe Abbildung 2.11).

Tabelle 2.3: True-False-Tabelle: tp bedeutet true-positive, fp false negative und fn false negative.

		actual	
		positive	negative
predicted	positive	true positive (tp)	false positive (fp)
	negative	false negative (fn)	true negative (tn)

Precision: Precision zeigt, wie genau die Vorhersagen sind:

$$Precision = \frac{tp}{tp + fp}$$

Die Abkürzungen für tp und fp stehen in der Tabelle 2.3.

Recall: Mit dem Recall wird angegeben, wie gut alle Positiven gefunden wird:

$$Recall = \frac{tp}{tp + fn}$$

Der Average-Precision (AP) ist die Fläche unter der Precision-Recall-Kurve. Der AP-Wert liegt ebenfalls wie die zwei anderen Werte, Precision und Recall, zwischen 0 und 1.

mAP: Des Weiteren gibt es den Mean-Average-Precision (mAP). In dieser Arbeit wird sich auf die Erklärung von COCO beschränkt. COCO und Pascal-VOC [33] sind Bilddatenbanken. Durch die Arbeit mit den Bildannotationen wurden bestimmte Formate und Evaluierungsmetriken entwickelt. Der mAP definiert sich durch die Berechnung des Mittelwertes von allen AP-Werten jeder Klasse. Des Weiteren gibt es noch Spezifikationen (siehe Tabelle B.1). Ein Beispiel ist Precision/mAP-Small. Hier wird der mAP für kleine Objekte berechnet, die kleiner als 32×32 Pixel sind.

Kapitel 3

Erstellung der Trainingsdaten

In der Weite der Mongolei sind herkömmliche Felderkundungen nicht effektiv genug. Hier ist die verstärkte Auswertung von Satellitenbildern mittels Machine-Learning eine vielversprechende Alternative. Basis dafür sind die Ergebnisse bisheriger archäologischer Forschungen. Knapp hundert historische Stadt- und Wallanlagen aus unterschiedlichen Epochen sind bereits dokumentiert. Zur Erstellung von Trainingsdaten werden aktuelle Satellitenaufnahmen nach den bereits erforschten Anlagen durchsucht.

3.1 Sammeln der Daten

Im Rahmen dieser Arbeit werden die Daten der Sentinel-2-Satelliten verwendet. Entscheidender Grund dafür ist ihre kostenlose Nutzung. Sie müssen nicht extra beantragt werden, das bedeutet ein Ersparnis von Zeit und Kosten. Eine direkte Bearbeitung kann sofort stattfinden. In der Abbildung 3.1 sind alle Satellitenbilder zu sehen, die für die Erstellen der Trainingsdaten akquiriert wurden.

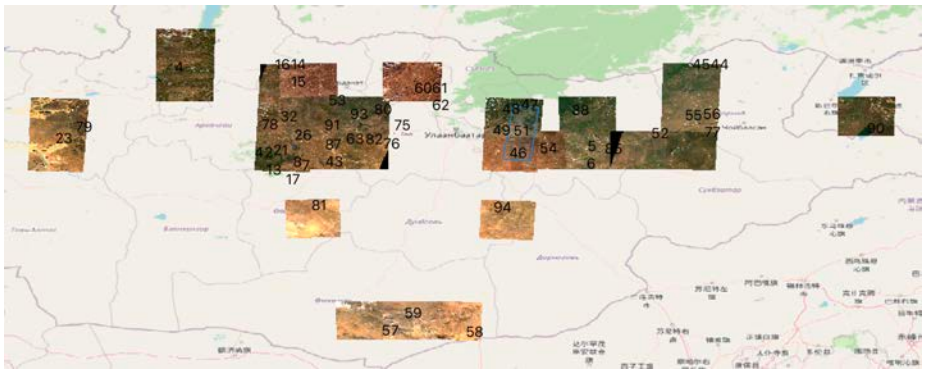

Abbildung 3.1: Alle angezeigten Satellitenbilder für die Erstellung der Trainingsdaten. Es ist zu sehen, dass sich die Objekte über die gesamte Fläche der Mongolei verteilen.

3.1.1 ESA Copernicus Open-Access-Hub

Die Sentinel-Daten können von der Webseite des Open-Access-Hub[1] heruntergeladen werden. Für die Objekterkennung werden nur die Daten des Sentinel-2 verwendet. Erste eingrenzende Suchkriterien sind die Sensing-Periode und die Ingestion-Periode. Die Sensing-Periode zeigt den Zeitraum an, in dem die Daten aufgenommen wurden. Die Ingestion-Periode ist der Zeitraum, in dem die Daten der Datenbank hinzugefügt wurden. Bei der Datensuche ist außerdem der Produkttyp auszuwählen. Zur Verfügung stehen S2MSI1C, S2MSI2A und S2MSI2Ap. Die ersten fünf Zeichen der Produkttypen bedeuten, dass es sich um multispektrale Daten vom Sentinel-2 handelt. Die restlichen Zeichen stehen für den Datentyp: Top-Of-Atmosphere oder Bottom-Of-Atmosphere. Das p bei S2MSI2Ap steht für Pilot-Produkt und bedeutet, dass nur Daten angezeigt werden, die bis März 2018 aufgenommen wurden.

Im Open-Access-Hub erfolgt die Auswahl des ROI durch die visuelle Markierung des Gebietes auf einer geografischen Karte. Alternativ sind die Daten auch über die EarthExplorer-Anwendung[2] des US-Geological-Surveys frei verfügbar. Hier liegt der Vorteil darin, dass die Angabe der Koordinaten der gewünschten Gebiete mit KML-Dateien (Keyhole-Markup-Language) angegeben werden können. KML-Dateien dienen zur Annotation von geografischen Informationen.

[1] https://scihub.copernicus.eu/dhus/#/home
[2] https://earthexplorer.usgs.gov/

Tabelle 3.1: Sentinel-2 Sucheinstellungen: Die relative Orbitnummer gibt an wie hoch die Anzahl der Umlaufbahnen ist um einen Wiederholungszyklus zu vervollständigen.

Suchkriterium	Auswahl	Erklärung
Satellitenplatform	S2A_*	Satelliteplfatform
	S2B_*	Satelliteplfatform
Produkttyp	S2MSI1C	Top-Of-Atmosphere (TOA)
	S2MSI2A	Bottom-Of-Atmosphere (BOA)
	S2MSI2Ap	BOA, Aufnahmen bis 03.2018
Relative Orbitnummer	1 bis 143	Umlaufbahnzahl für ein Wiederholungszyklus
Wolkenbedeckung	0 bis 100%	Der Prozentanteil der Wolkenbedeckung

3.1.2 Alternativer Zugang

Die Google-Earth-Engine ist eine Web-Applikation mit einem Javascript-Editor, mit dem man sich Satellitenbilder-Kollektionen anzeigen lassen kann. Die angezeigten Informationen können durch bestimmte Algorithmen gefiltert werden. Unter der Verwendung des Pakets Geemap[3] lässt sich eine Schnittstelle zu Python-Bibliotheken herstellen, z.B. Rasterio oder NumPy. Auf diesem Weg ist eine interaktive Kartierung innerhalb der Google-Earth-Engine in Python möglich.

In QGIS ist ein Google-Earth-Engine-Plugin vorhanden, welches eine Schnittstelle zum Import von Bildkollektionen von Googe-Earth herstellt. QGIS ist eine freie Geoinformationssystem-Software zum Betrachten, Bearbeiten, Erfassen und Analysieren von räumlichen Daten. Die einzige Bedingung hierfür ist die Google-Earth-Authentifizierung, die über die interne Python-Console vollzogen wird. Ein weiteres QGIS-Plugin, SentinelHub, stellt ebenfalls den direkten Zugang zu den Sentinel-Daten her. Die direkte Einbindung innerhalb von QGIS hat den Vorteil, dass die Zwischenschritte für Download und Konvertierung der Daten eingespart werden können.

Abbildung 3.2: Die Jahreszeiten beeinflussen die Vegetation. Dadurch sind manche Strukturen zu bestimmten Monaten besser zu erkennen.

3.1.3 Einfluss der Jahreszeiten

Bei der Auswahl der Daten spielen Wolkenbedeckung und Jahreszeiten eine wichtige Rolle. Im Open-Access-Hub kann der Prozentanteil der Wolkenbedeckung angegeben werden. Für eine zusätzliche Wolkenmaskierung oder eine atmosphärische Korrektur muss der Produkttyp S2MSI1C verwendet werden.

Die Jahreszeiten bestimmen die Witterung und beeinflussen damit die Vegetation. Diese wiederum hat Folgen für die Sichtbarkeit von archäologischen Spuren im Boden. Frühling und Herbst sind für die Erkennung der Strukturen besonders geeignet [17]. Der Frühling in der Mongolei dauert vom März bis Mai, der Herbst vom September bis Oktober. Beide Jahreszeiten sind besonders interessant, da in diesen Zeiträumen die Vegetation im Wandel und stark von unterschiedlichen Farben gekennzeichnet ist (siehe Abbildung 3.2). Farben sind insofern wichtig, da bei der Fernerkundung die Erkennung eines archäologischen Merkmals vom Kontrast zur Umgebung abhängt [34]. Im Winter, der von November bis Februar dauert, sind viele Flächen mit Schnee bedeckt. So sind Strukturen auf Satellitenbildern kaum zu erkennen, was diese Jahreszeit für diese Arbeit ungeeignet macht.

3.2 Vorverarbeitung der Spektralbänder

Die Vorverarbeitung der Geodaten findet in QGIS statt. Für eine bessere Übersicht der Satellitenbilder wurde eine Weltkarte von OpenStreetMap importiert. Die multispektralen Satellitenbilder müssen nun analysiert und für die Objekterkennung vorbereitet werden. Die Spektralbänder sind separat vorhanden. Ein True-Color-Bild (TCI) enthält die Bänder 2, 3 und 4. Um die RGB

[3] https://github.com/giswqs/geemap

Bilder herzustellen, kann die GDAL-Bibliothek verwendet werden. Dafür wird eine virtuelle Ebene erstellt und die gewünschten Bänder zusammengefügt:

```
1 gdalbuildvrt -separate stacked_bands.vrt red_band.jp2
2         green_band.jp2
3         blue_band.jp2
4 gdal_translate stacked_bands.vrt stacked_bands.tif
```

Nur die Bänder 2, 3, 4 und 8 haben die Auflösung von 10 m pro Pixel. Die Auflösung der restlichen Bänder haben eine Auflösung von 20 m oder 60 m pro Pixel. Für die Erkennung der Stadt- und Wallanlagen ist eine hohe Bildauflösung notwenig. Deshalb wurden für die Erstellung der Trainingsbilder nur die Bänder 2, 3, 4 und 8 verwendet.

3.2.1 Wolkenmaskierung

Auch bei günstigen Wetterbedingungen sind Wolken in Satellitenbildern nicht zu verhindern. Eine Lösung dieses Problems bietet die Wolkenmaskierung. Mithilfe der SNAP-Anwendung (Sentinels-Application-Platform) der ESA ist es möglich, Wolken automatisch zu maskieren, d.h. die Wolken aus den Bildern zu entfernen. Um die Spektralbänder miteinander kombinieren zu können, werden diese auf die gleiche Auflösung interpoliert. Mit dem IdePix-Werkzeug können die Wolken automatisch erkannt werden (siehe Abbildung 3.3). IdePix[4] ist ein Plugin für SNAP, womit Oberflächen identifiziert werden können, gleich ob es sich um Wasser, Land oder Wolken handelt.

3.2.2 Falschfarbendarstellung

Falschfarbendarstellungen von Satellitenbildern können sehr nützlich sein, um verschiedene Landschaftsmerkmale zu analysieren. Durch multispektrale Informationen können weitere Erkenntnisse aus den Satellitenbildern gewonnen werden. In QGIS werden aus Satellitenbildern zusammengesetzte Falschfarbenbilder erstellt, indem die Bänder für Rot, Grün und Blau durch andere Spektralbänder ersetzt werden. Das geschieht in Kombination von Nah-Infrarot, Grün und Rot, in eben dieser Reihenfolge. Die roten Pixel in der Abbildung 3.3 repräsentieren den Wachstumszustand der Pflanzen. Da das Spektralband für Nah-Infrarot die gleiche Auflösung mit 10 m pro Pixel aufweist wie die RGB-Bänder,

[4] https://www.brockmann-consult.de/portfolio/idepix/

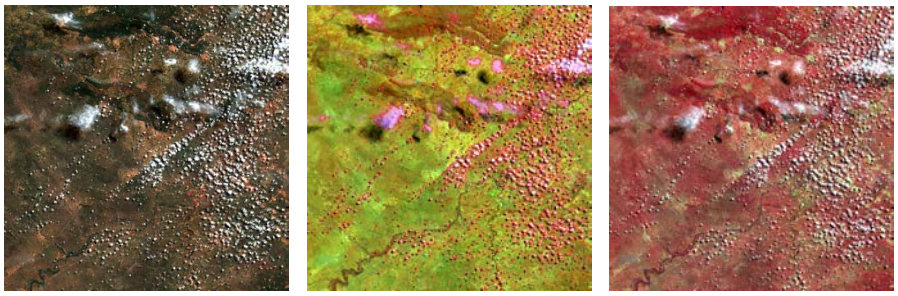

Abbildung 3.3: Die drei Bildern zeigen drei verschiedene Kombinationen von Spektralbändern an. Das Bild links ist ein True-Color-Image. Die Farbwerte wurden atmosphärisch korrigiert. Das Bild rechts ist eine Falschfarbendarstellung mit den Bändern Nah-Infrarot, Grün und Rot (in der Reihenfolge). Das Bild in der Mitte ist ein Ergebnisse einer Wolkenmaskierung. Die Wolken sind mit roter Farbe markiert.

verschlechtert sich die Auflösung nicht. Durch die Generierung von Falschfarbenbildern können die Trainingsdaten erweitert werden.

3.2.3 Berechnen von Indizes

Die Verwendung von Daten aus verschiedenen Jahreszeiten hat unterschiedliche Auswirkungen auf die Indizes. Eigenschaften wie die Feuchtigkeit des Bodens oder die Lebendigkeit der Pflanzen sind hilfreich bei der Beobachtung der Erdoberfläche. Dabei bietet vor allem der Frühling mehr Raum für die Untersuchungen [17]. Das Pflanzenwachstum kann durch bestimmte Wetterbedingungen, wie starkem Wind, so beeinflusst werden, dass sichtbare Strukturen entstehen, die bei der Suche nach archäologischen Objekten helfen können. Diese Objekte können sich auch durch ihre Materialdichte von der natürlichen Umgebung abheben. Die Berechnung von NDVI hilft dabei, archäologische Überreste besser zu erkennen [18].

In der Abbildung 2.4 vom 30.03.2021 wurden die Indizes NDVI, SAVI und NAI mithilfe der SNAP-Anwendung berechnet. Für die Berechnung des NDVI und SAVI wurden die Bänder 4 und 8 verwendet. Für den SAVI wurde zusätzlich der Bodenhelligkeitsfaktor ausgewählt. Für die Berechnung des normalisierten Archäologieindex (NAI) werden die Wellenlängen 800 nm und 700 nm benötigt. Die Bänder 5 und 7 sind diesen Wellenlängen ausreichend nah und können dafür verwendet werden [19]. Eine Bedingung bei der Berechnung der Indizes ist die gleiche Bildauflösung der Bänder. Wie bei den Falschfarbenbildern können die

Tabelle 3.2: Exemplarisch sind zusätzliche Daten der Stadt- und Wallanlagen zu sehen. Die Orientierung in der letzten Zeile beschreibt die Himmelsausrichtung.

name	Bay Baliq 1	Donojn Balgas
description	Enclosure with several buildings	Enclosure with few buildings
epoch	Uyghur	Uyghur
surveyed	true	true
excavated	false	false
has_wall	true	true
gates_number	1	4
orientation	177	115

Ergebnisse der Indexberechnungen als Erweiterung der Trainingsdaten dienen, solange die Auflösung der Bilder sich nicht verringert.

3.3 Archäologische Informationen

Für die Generierung der Trainings-, Validierungs- und Testdaten wurden für die jeweiligen Koordinaten Satellitenbilder akquiriert. Je mehr Trainingsdaten, desto vielfältiger sind die Informationen, die auf das Modell trainiert werden können. In dieser Arbeit sind Stadt- und Wallanlagen in der Mongolei die gesuchten Objekte. Berichte über durchgeführte Ausgrabungen, Feld- und Geländebegehungen liefern dabei die sichersten Informationen über die Lage der Stätten. Weitere Informationen sind die Namen, die Himmelsrichtung, die Anzahl der Tore, die zugehörige Epoche und noch weitere detaillierte Beschreibungen (siehe Tabelle 3.2).

3.3.1 Erstellung der Bilder

Zu beachten ist, dass die Satellitenbilder den gesamten Bereich der Objekte umfassen, damit das Modell nicht mit unvollständigen Objekten trainiert wird.

Die Auflösung der Sentinel-2-Daten begrenzt die Verwendung aller Spektralbereiche. Nur die Spektralbänder 2, 3, 4 und 8 sind in der Auflösung 10m pro Pixel vorhanden. Bei dieser Auflösung muss man davon ausgehen, dass die Objekte

nicht kleiner als 10 m sein dürfen, weil sie sonst nur durch einen Pixel repräsentiert werden. Insgesamt wurden 94 Objekte als bekannte Anlagen aufgezeichnet. Für die Trainingsbilder sind 55 Objekte davon nicht verwendbar. Viele Anlagen sind für die Auflösung der Sentinel-2-Daten zu klein oder klare Strukturen sind nicht sichtbar (siehe Abbildung 4.1). Die Anlagen, die über mehrere hundert Meter lang sind, können für die Trainingsdaten genutzt werden.

Einige Satellitenbilder enthalten potenzielle Objekte, die bisher unbekannt sind. Da es erstmal unklar war ob es sich bei diesen Objekten um echte Stadt- oder Wallanlagen handelt, wurden diese erstmal ausgeschlossen.

3.3.2 Bearbeitung in QGIS

Vektoren bzw. Polygone wurden verwendet, um die Anlagen in QGIS anzuzeigen. Die Eintragung der umrahmenden Polygone dienten als Bounding-Boxen. Die Daten über die Anlagen wurden in Geopackages übergeben. Das Geopackage ist ein offenes, plattformunabhängiges und standardisiertes Datenformat für geografische Informationssysteme. Sie sind eine alternative zum alten Standard der Shapefiles.

In QGIS sind die Objekte mit unterschiedlich geformten Polygonen markiert. Für die Trainingsdaten sind rechteckige Bilder mit dem Objekt im Zentrum das Ziel. Die QGIS-Werkzeuge Buffer und BoundingBox wurden dafür verwendet. Mit dem Buffer-Werkzeug kann ein Abstand zum Objekt erstellt werden. Dieser Abstand ist standardmäßig in Winkelgrad anzugeben. Um die Einheit auf Meter oder Kilometer einzustellen, muss das Polygon in ein lokales Koordinatensystem projiziert werden. Oft ist das globale Koordinatensystem WG84 zu ungenau. Die Informationen sollten sich auf das Objekt begrenzen. Ein größerer Abstand zum Objekt bewirkt, dass mehr Informationen zur Umgebung hinzugefügt wird. Mehr Informationen bedeuten eine längere Berechnungsdauer.

Die Funktion BoundingBox konvertiert jegliche Polygone zu Rechtecken. Dabei ist zu beachten, dass die Ecken nicht abgerundet sind. Beide Funktionen, Buffer und BoundingBox, können als Batch-Vorgang für eine Vielzahl von Objekten ausgeführt werden (siehe Abbildung 3.4).

3.3.3 Export der Bilder

Die Daten mit jeweiligen Annotationen befinden sich in QGIS. Es gibt drei Methoden in QGIS, die Bilder zu exportieren:

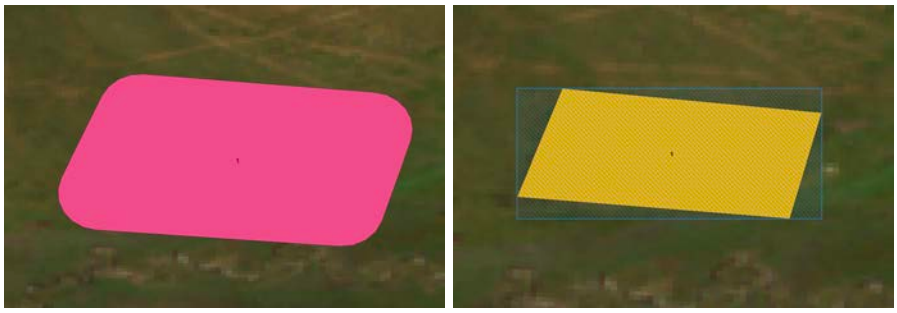

Abbildung 3.4: Erstellung des Buffers - Für das Beispiel A wurde die Distanz von $0,01$ in der Einheit degree ausgewählt. Mit diesem Buffer haben wir genug Bildinformationen um das richtige Bildverhältnis zu erhalten. Ein Problem der eingezeichneten Polygone sind die unterschiedlichen Längen- und Breitenverhältnisse. Der Join-Style in der BoundingBox-Funktion ist auf Miter einzustellen. Das bewirkt, dass die Ecken nicht abgerundet sind, sondern eckig.

1. Die Vektordaten bzw. die Polygone, die die Objekte markieren, werden als Masken verwendet. Das Satellitenbild ist das Ausgangsbild. Die Pixel, die sich innerhalb der Maske befinden, werden als neue Ebene extrahiert. Beim Export dieser Ebene müssen die Daten als gerendertes Bild gespeichert werden, damit die Farben des Bildes auch außerhalb des GIS angezeigt werden. Die exportierte Datei ist ein GeoTIFF, das bedeutet, dass das Bild georeferenziert ist. Die Auflösung entspricht der originalen Auflösung des Satellitenbildes.

2. Das Bild wird direkt aus der Ansicht von QGIS exportiert, dabei muss die neu generierte Box als Maske verwendet werden. Der Vorteil ist, dass das Datei-Format direkt als JPEG oder PNG auswählbar ist. Diese Formate sind später für die weitere Verarbeitung geeignet. Zu beachten ist, dass die originale Auflösung verändert wird. Dieses Problem ist zu vernachlässigen, da alle Bilder später nochmal auf die gleiche Auflösung skaliert werden. Zusätzlich zu dem exportierten Bild wird auch eine Tiff-World-Datei erstellt, die Informationen für die Georeferenz des Ausschnittes beinhaltet.

3. Es ist möglich zu testen, ob die Boxen auf den Satellitenbildern liegen. Zu beachten ist, dass mehrere Satellitenbilder aus verschiedenen Jahreszeiten dieselbe Fläche bedecken. Aufgrund dieser Überlagerung musste getestet werden, welche Boxen auf den einzelnen Satellitenbildern liegen. Dafür wurden alle Überschneidungen zwischen Box und Satellitenbild direkt ex-

portiert. Durch dieses Testen entstehen mehrheitlich schwarze Bilder, die anschließend entfernt werden.

Kapitel 4

Erkennung archäologischer Objekte

In diesem Kapitel wird erläutert, welchen Algorithmen bzw. welches Machine-Learning-Modell verwendet wurde, um die Stadt- und Wallanlagen in den Satellitenbildern zu erkennen und zu lokalisieren. Dabei werden die letzten Vorbereitungsschritte wie Annotation der Bilder und Konfigurierung der Parameter erklärt, bevor das Training startet.

4.1 Auswahl für Faster-RCNN

Die Stadt- und Wallanlagen sollen in den Satellitenbildern erkannt werden. Es musste ein Algorithmus verwendet werden, der die Objekte lokalisiert und erkennt. Dabei spielt die Genauigkeit der Lokalisierung eine wichtige Rolle. Echtzeit bzw. eine schnelle Bearbeitung der Daten ist dabei ein zweitrangiges Kriterium. Die R-CNN-Architektur liefert gute Ergebnisse bei unterschiedlichen Aufgaben in der Objekterkennung [26]. Das Grundkonzept eines R-CNN besteht darin, mehrere Objektvorschläge innerhalb eines Bildes zu generieren, Merkmale aus jedem Vorschlag mithilfe eines CNN zu extrahieren und diese dann zu klassifizieren [11]. Zurzeit ist Faster-R-CNN von den R-CNN Modellen die weitest entwickelte Version. Das Modell wurde schon in vielen anderen Projekten für die Objekterkennung verwendet, wie z.B. zur Erkennung von Passanten[35] oder die Identifizierung von Autos in Luftbildern [2]. Faster R-CNN führt tendenziell zu

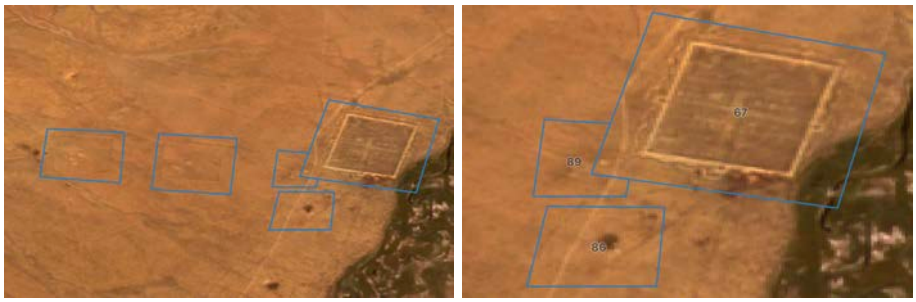

Abbildung 4.1: Hier sind mehrere Anlagen markiert. Das rechte Bild zeigt drei Bounding-Boxen. Erkennbar ist nur die große Anlage auf der rechten Seite.

langsameren, aber genaueren Ergebnissen im Vergleich zu anderen Modellen[26]. Aus diesen Gründen, wurde Faster-R-CNN für die Objekterkennung der Stadt- und Wallanlagen ausgewählt.

4.2 Annotation und Konfiguration

Aus den Satellitendaten mit den RGB Bändern sind 70 Trainingsbilder entstanden. Um die Bilder mit der Object-Detection API von Tensorflow zu verwenden, müssen die Bilder in ein TFRecord Datenformat konvertiert werden. Zu jedem Datensatz ist eine Label-Map notwendig. Diese Label-Map definiert wie die Klassen-Beschreibungen (String) auf Zahlen (integer) abgebildet werden. Im Folgendem wird ein Beispiel für eine Label-Map gezeigt:

```
1 item{
2         id:1
3         name: 'Temple'
4 }
5 item{
6         id: 2
7         name: 'Grab'
8 }
```

Die Identifizierungsnummern beginnen immer mit der Nummer 1. Die meisten Modelle werden mit RGB-Bildern trainiert, dadurch erwarten auch die meisten Modelle ein Bild mit drei Kanälen als Input. Des Weiteren wird eine Liste von den Bounding-Boxen mit den zugehörigen Klassen benötigt, die durch die

Annotationen erstellt wird. Die Objekte werden in den Satellitenbildern durch diese Annotationen gekennzeichnet. Eine automatische Generierung der Annotationen innerhalb QGIS ist geplant, wurde aber nicht in Rahmen dieser Arbeit realisiert. Mithilfe von Roboflow[1] konnten Trainingsdaten annotiert werden. Bei der Annotation der Trainingsdaten ist zu beachten, dass die Bounding-Boxen nur die Objekte umrahmen und bei mehreren Objekten in einem Bild, müssen alle Objekte einzeln annotiert werden.

Die Liste mit den Annotationen kann in verschiedenen Formaten erstellt werden. Die meist verwendeten Formate sind COCO und Pascal-VOC. Da Faster-R-CNN mit dem Framework Tensorflow implementiert wurde, muss das Format TFRecord verwendet werden. Es besteht trotzdem noch die Möglichkeit, die Daten in andere Formate zu konvertieren. Da es nur den Hintergrund und eine Klasse gibt, muss nicht auf eine Klassenbalance geachtet werden. Gäbe es mehrere Kategorien, z.B. moderne Häuser, archäologische Anlagen und Gräber, sollten die Anzahl der Bilder und Objekte ausgeglichen sein. Vor der Einteilung in Test-, Trainings- und Validierungsdatensatz sollten die Daten gemischt werden, um einen unwillkürlichen Bias zu verhindern.

4.2.1 Trainingspipeline

Tensorflow bietet auf GitHub eine Auswahl von vortrainierten Objekterkennungsmodellen an. Die meisten Modelle wurden mit COCO-Datensets trainiert. Für die spätere Evaluierung werden die COCO-Metriken verwendet. Für die Erstellung der Feature-Maps wird ein vortrainiertes Modell ausgewählt. Das Inception-V2 [36] ist als Modell für die Feature-Extraktion voreingestellt.

In der Config-Datei werden einige Hyperparameter eingestellt. Die Datei wird in fünf Abschnitte unterteilt. Als erstes wird das Modell konfiguriert, z.B. der Feature-Extractor. Anschließend wird ausgewählt, welche Parameter zum Trainieren von Modellparametern verwendet werden sollen. Der Abschnitt *eval_config* bestimmt, welche Metriken für die Auswertung genutzt werden. Der Abschnitt *train_input_config* definiert den Datensatz für das Training. Der Abschnitt *eval_input_config* gibt an, welcher Datensatz für die Evaluierung verwendet wird.

Für das Fine-Tuning muss nur die Checkpoint-Datei ausgewählt werden. Diese Datei beinhaltet die Informationen über das vortrainierte Modell und dessen

[1] https://roboflow.com

Gewichtungen.

4.2.2 Entwicklungsumgebung

Als Framework wurde TENSORFLOW ausgewählt. Eine mögliche Alternative ist PYTORCH. Die Implementierung von Modellen ist mit PYTORCH oberflächlich und somit einfach zu realisieren. Doch durch die Entwicklung von Tensorflow 2.0 ist die Verwendung von Machine-Learning-Modellen mit Tensorflow verständlicher geworden. Vorteile von Tensorflow sind die vorhandenen Schnittstellen zu Keras und der Google-Earth-Engine. Keras ist eine Deep-Learning-Bibliothek.

GOOGLE bietet als kostenlosen Service COLABORATORY an, womit im Browser auch Python-Code ausführbar ist. Es ist keine Konfiguration erforderlich und der Zugriff zu GPUs ist kostenfrei. Es wurde eine NVIDIA K80 mit einem GPU-Speicher von 12 GB verwendet. Nach Stand Januar 2021 werden auch T4 angeboten, mit einem GPU-Speicher von 16 GB.

Ein Nachteil an Google Colab ist die geringe Upload-Geschwindigkeit. Nach 12 Stunden fährt die Maschine herunter und alle Daten sind verloren. Außerdem kann es dazu kommen, dass der Google-Service überlastet ist und die GPU-Leistung nicht immer angeboten werden kann.

4.3 Umgang mit wenigen Trainingsdaten

Exakte Ergebnisse setzen ausreichende Datenmengen voraus. Oft fehlt es an vielfältigen und großen Datensätzen. Der Mangel beschränkt das Potential von Machine-Learning-Modellen. Einfache Algorithmen, wie z.B. Lineare-Regression, benötigen weniger Daten für gute Ergebnisse, aber diese Daten haben einen hohen Bias und eine niedrige Varianz. Bei Algorithmen, die viele Trainingsdaten benötigen, um gute Ergebnisse zu erreichen, ist der Bias geringer und die Verteilung der Gewichtungen ausgeglichener. Unausgeglichene Datensätze können zu geringerer Leistung und Bias führen [37].

Die Anzahl der bekannten archäologischen Anlagen in der Mongolei ist gering. Das ist ein Problem, denn viele Anlagen sind zu klein, um auf den Satellitenbildern erkannt zu werden (siehe Abbildung 4.1). Google-Earth wurde durch eine visuelle Überprüfung als zusätzliches Validierungstool verwendet [15].Die Trainingsdaten von den ausreichend großen Anlagen muss erweitert werden.

4.3.1 Datenerweiterung

Bei kleinen Datensätze können durch bestimmte Bildverarbeitungoperationen wie Rotation, Transformation und Spiegelung die Datensätze vergrößert werden. Faster-RCNN ist invariant zu Rotationen. Nach der Rotation des Bildes erkennt das CNN das Objekt nicht mehr [30]. Das bedeutet, dass die Bilder vor dem Training noch rotiert werden können, um ein besseres Ergebnis zu erlangen. Die data_augmentation_options in der train_config geben die Operationen für die Datenerweiterung an.

Es wurden mehrere Satellitenbilder vom selben Erdabschnitt aus verschiedenen Jahreszeiten verwendet. Dadurch entstehen mehr Trainingsdaten und das Modell wird auf unterschiedliche Wetterbedingungen trainiert. Des Weiteren wurden Indexberechnung der Satellitenbilder erstellt. Die Anzahl der Daten kann durch eine Indexberechnung verdoppelt werden. Die Auswirkungen auf die Leistung des Modells ist dabei unbekannt.

Es wurden keine zusätzlichen Trainingsdaten für den Hintergrund erstellt. Die Annotation der Klassen für Faster-R-CNN, wird immer mit vier Koordinaten der Bounding-Box gekennzeichnet. Der Hintergrund ist durch alle Strukturen definiert, die nicht von den Trainingsdaten inkludiert werden.

4.4 Training

Als erstes werden die Boxen erstellt, die das Objekt eingrenzen. Dabei werden vier Punkte verwendet: ymin, xmin, ymax, xmax. Anschließend werden die Trainingsbilder für das Region-Proposal-Netzwerk (RPN) auf eine einheitliche Input-Größe skaliert. Die Boxen werden zu dem Format angepasst. In der Abbildung 4.2 wurde für die Input-Größe 800×800 px gewählt. Diese Bilder durchlaufen ein CNN, z.B. das VGG16. Der Output sind Feature-Maps mit einer Auflösung von $50 \times 50px$ und einer Tiefe von 512 Kanälen. Das RPN soll mithilfe dieser Feature-Maps Vorschläge für Bounding-Boxen erstellen. Für die ROIs werden Anker generiert. Diese Anker sind Punkte im Bild, die zur Erstellung der Boxen dienen. Als Beispiel werden 2500 Punkte generiert, um diese

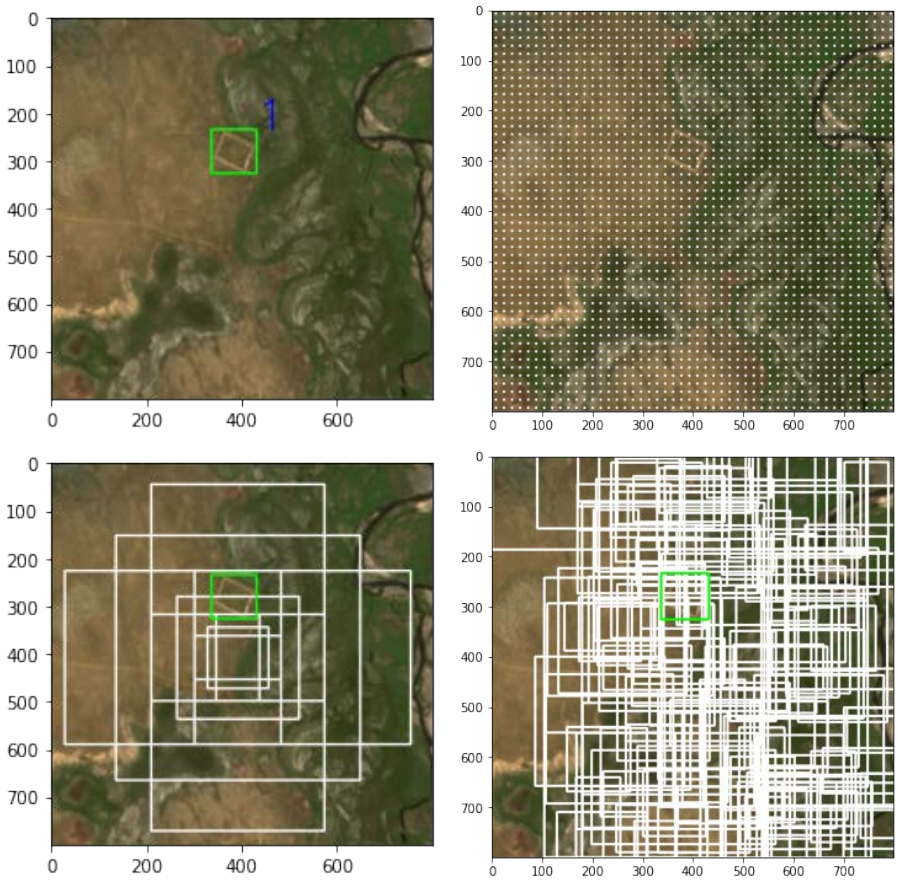

Abbildung 4.2: Hier wird der Ablauf eines Bildes innerhalb des Modell Faster-RCNN angezeigt. Oben rechts werden die Ankerpunkte angezeigt. Unten links werden exemplarisch Bounding-Boxen mit unterschiedlichen Seitenverhältnisse visualisiert.

Ankerboxen zu erstellen:

$$anchor_{total} = 16 \cdot 16 = 2500$$

Die 16 ergibt sich, wenn die Input-Größe mit der Auflösung der Feature-Maps dividiert wird. Für jeden einzelnen dieser 2500 Punkte werden jeweils 9 Ankerboxen erstellt. Insgesamt wurden 22500 Ankerboxen generiert:

$$bounding\,box_{total} = 2500 \cdot 9 = 22500$$

Ankerboxen, die über den Rand des Bildes gehen, werden entfernt und es bleiben 8940 Ankerboxen übrig. Anschließend wurden die Überlappungen mit den Ground-Truth-Boxen überprüft. Die Boxen, die unter dem Schwellenwert für die Überlappungen liegen, werden ebenfalls entfernt. Danach wird geprüft, ob Objekte innerhalb der Boxen vorhanden sind. Es gibt zwei Klassen, die 1 steht für das Objekt und die 0 für den Hintergrund. Der Schwellenwert für eine Überlappung mit einer Ground-Truth-Box liegt bei 0.7, d.h. 70% muss die Überlappung mindestens betragen. Bei weniger als 30% werden die Boxen als Hintergrund klassifiziert. Der Rest der Boxen erhält den Index -1, diese weder für weitere Berechnungen ausgeschlossen. Für die Klassifizierung wird der Cross-Entropy-Loss berechnet. Danach werden Boxen, die eine hohe Überlappung haben, mittels NMS zusammengefasst. In dem Beispiel wurden empfohlene von 2000 Vorschläge für die Bounding-Boxen verwendet [29]. Die Boxen werden nach der höchsten Wahrscheinlichkeit sortiert. Die Feature-Maps werden an die Bounding-Boxen angegepasst. Anschließend werden sie auf die Auflösung 7×7 px konvertiert (siehe Abbildung 4.3). Danach verlaufen die Feature-Maps durch zwei Fully-Connected-Layers. Als Ausgabe folgen die Ergebnisse für die Klassifizierung und die Lokalisierung.

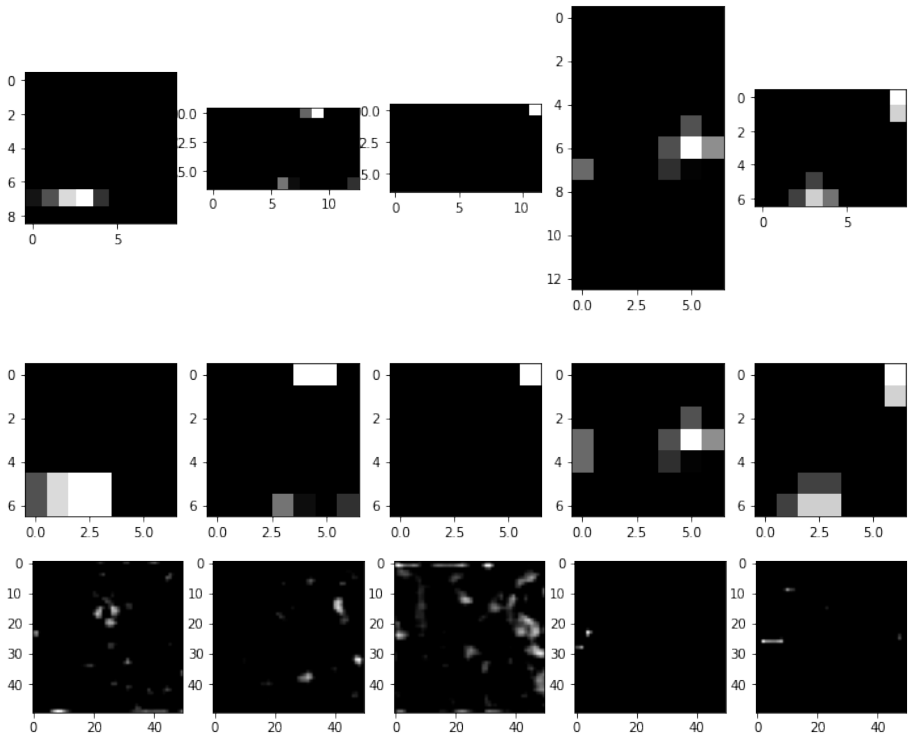

Abbildung 4.3: Abbildung unterschiedlicher Feature-Map Auflösungen. Die erste Reihe sind die untransformierten Feature-Maps. Die zweite Reihe zeigt die Feature-Maps nach der Konvertierung. Die Feature-Maps in der letzten Reihe zeigen die Ergebnisse des Feature-Extractors.

Kapitel 5

Experimente und Auswertung

Faster-R-CNN-Modelle eignen sich besser für Fälle, in denen eine hohe Genauigkeit gewünscht wird und die Latenz eine geringere Priorität hat. Beim Training mit Google-Colab musste die zeitliche Begrenzung von 12 Stunden eingehalten werden.

5.1 Experimente

Die Auflösung hat einen direkten Einfluss auf die Genauigkeit der Objekterkennung [26]. Alle Modelle liefern bessere Ergebnisse je größer die Objekte sind [26]. In Folgenden soll herausgefunden werden, ob bestimmte Hyperparameter die Leistung des Modells verbessern können.

5.1.1 Hyperparameter

Alle Einstellungen, die vor dem Trainieren des Modells manuell veränderbar sind, können Hyperparameter sein. Die Epochen, die Iterationen und Batch-Size sind Parameter, die vor dem Training auszuwählen sind. Die Werte können die Laufzeit stark beeinflussen. Die Batch-Size bestimmt wie viele Trainingsdaten auf einmal durch das Netz geschickt werden. Eine Epoche ist vorbei, wenn alle Trainingsdaten durch das Modell gelaufen sind. Die Anzahl der Iterationen

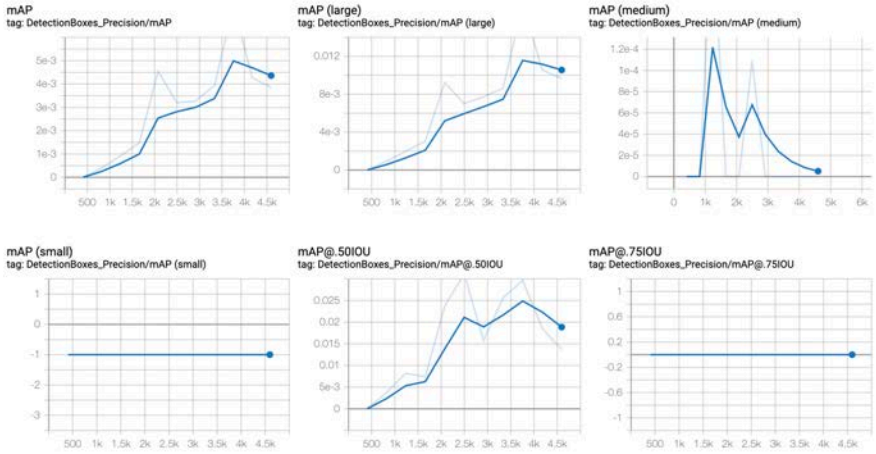

Abbildung 5.1: Testergebnisse von mAP: Die Ergebnisse werden nach bestimmten Kriterien eingeordnet. Die genauen Definitionen sind in der Tabelle B.1 zu finden. Das Diagramm oben links zeigt die allgemeinen mAP-Werte. Die restlichen Diagramme werden entweder nach den Pixelgrößen der Objekte oder nach dem IoU-Schwellenwert betrachtet.

gibt die benötigten Durchläufe für alle Trainingsdaten an. Bei 70 Trainingsbildern und einer Batch-Size von 10 werden 7 Iterationen benötigt. Eine kleinere Batch-Size benötigt weniger Speicherplatz und die Gewichtungen werden öfter aktualisiert. Die Größen 12, 10 und 8 wurden mit dem Inception-Modell getestet. Je kleiner die Batch-Size wurde, desto länger hatte das Trainieren gedauert. Bei einer Größe von 8 wurde das 12 Stunden Zeitlimit sogar überschritten.

5.1.2 Konfigurationen am Modell Faster-R-CNN

Im Allgemeinen gehört die Auswahl der Klassifizierer und Optimierungsalgorithmen ebenfalls zu den Hyperparametern. Bei Faster-R-CNN wird für die Erstellung der Feature-Maps ein vortrainiertes Modell verwendet. Um das Modell zu modifizieren, werden Protocol-Buffers verwendet. Protocoll-Buffers dienen zur effizienten Serialisierung strukturierter Daten. Neben der Auswahl des Feature-Extractors, kann die Größe der Feature-Map ebenfalls die Leistung oder Berechnungsdauer des Modells beeinflussen [26]. Neben Inception [36] stehen noch andere Modelle im TENSORFLOW Model-Zoo zur Auswahl.

Im RPN werden potenzielle Bounding-Boxen erstellt und für die Objekterken-

nung vorgeschlagen. Bei der Erstellung können ebenfalls eine Reihe von Parameters eingestellt bzw. ausgewählt werden. So kann der Schwellenwert bei der Überlappung der Bounding-Boxen einen Einfluss auf die Ergebnisse haben. Ebenso spielt die Größe der Bounding-Boxen eine Rolle [11]. Optimal sind 300 Proposals [30]. Nach den Experimenten von [26], kann die Anzahl der Proposals auch reduziert werden, ohne Verlust beim mAP zu verursachen.

5.2 Auswertung mit Tensorboard

Mithilfe von Tensorboard werden die COCO-Metriken berechnet und visualisiert. TensorBoard ist ein ein Werkzeug zur Darstellung der Modellgraphen und des Monitorings von Evaluierungsmetriken.

Es wurden unterschiedliche Parameter für das Testen ausprobiert. Die Testergebnisse in der Abbildung 5.1 zeigen die unterschiedlichen mAP-Verläufe. Es gibt Metriken, die sich an den IoU-Schwellenwert orientieren. Andere beziehen sich auf die Größe des Objekts. In der Abbildung 5.1 zeigt das Diagramm mAP-Small den Wert -1 durchgängig an. Das bedeutet, dass keine potenziellen Bounding-Boxen in dieser Pixelgröße (32×32) erkannt wurden. Im Diagramm unten rechts werden ebenfalls keine Objekte gefunden, die sich über dem IoU-Schwellenwert von 75% befinden. Bei einem Schwellenwert von 50% steigt die Quote. Interessant ist, dass in den Diagrammen von großen und mittelgroßen Objekten bessere Werte angezeigt werden. Die Metriken mit den unterschiedlichen Pixelgrößen zeigen, dass die Objekte eher größer als 96×96 Pixel groß sind.

Zusätzlich zu den mAP-Diagrammen wurde noch Recall-Diagramme erstellt(siehe Abbildung B.1). Sie zeigen ähnliche Ergebnisse. In der Tabelle 5.1 sind die Verluste bzw. Kosten aufgelistet. Der größte Verlust ergibt sich beim RPN-Loss/Objectness_loss. Der Gesamt Verlust liegt bei fast 30%. Insgesamt zeigen die Werte keine ausreichenden Ergebnisse.

5.3 Post-Processing

Abschließend müssen die Ergebnisse wieder in ein geografisches Informationssystem (GIS) transportiert werden. Das rechte Bild in der Abbildung 5.2 muss in die Originalgröße transformiert werden. Dafür müssen die Bilder georeferenziert

Tabelle 5.1: Hier sind Trainingsergebnisse für die Kosten-/Verlustfunktion aufgelistet, die bei dem Modell Faster–CNN entstehen.

BoxClassifierLoss/Classification_loss	0.07131558
BoxClassifierLoss/Localization_loss	0.07040951
RPNLoss/Localization_loss	0.019063829
Loss/RPNLoss/Objectness_loss	0.13564777
Loss/Total_loss	0.2964367

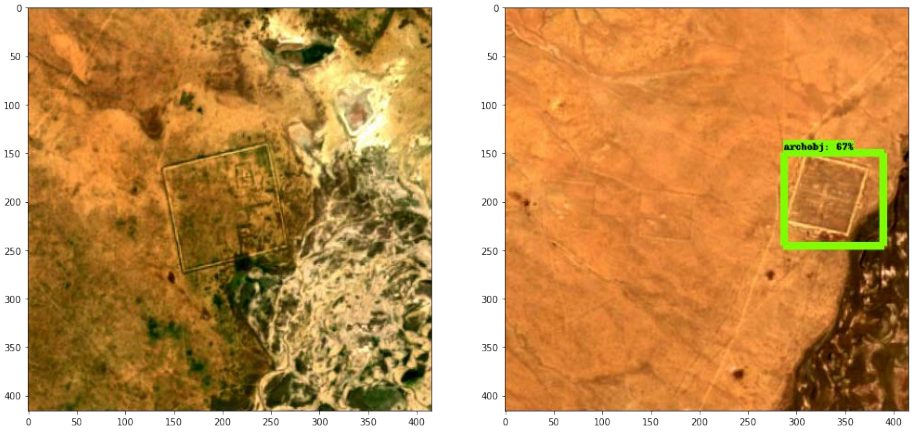

Abbildung 5.2: Hier sind zwei Testbilder zu sehen. Auf dem rechten Bild wurde ein Objekt mit einer Wahrscheinlichkeit von 67% erkannt. Auf dem linken Bild wurde das eigentlich klar erkennbare Objekte nicht identifiziert.

und die Bounding-Boxen in Vektoren umgewandelt werden.

Eine Möglichkeit, die Bilder zu georeferenzieren ist die Verwendung von Tiff-World-Dateien. Dafür werden die Ergebnisse des Modells in das GIS importiert. Der einzige Zwischenschritt ist die Skalierung in die ursprüngliche Auflösung. Der Nachteil ist, die Ergebnisse werden nur als Rasterbilder angezeigt. Um die Boxen korrekt zu vektorisieren, müssen sie in Koordinaten umgewandelt werden. Außerdem müssen die Punkte georeferenziert werden, damit die Bounding-Box passend zum Satellitenbild weiterverwendet werden kann.

Kapitel 6

Zusammenfassung und Ausblick

In dieser Arbeit wurde die Erforschung archäologischer Stätten mit Hilfe von Satellitenbildern und Machine-Learning von der gesamten Mongolei behandelt. Dafür wurden kostenfreie Daten von Senitel-2-Satellliten akquiriert. Zusätzlich zu den Satellitenbildern wurden Informationen von ca. 100 bereits dokumentierten archäologischen Stadt- und Wallanlagen verwendet. Auf dieser Basis konnten die Trainingsdaten erstellt werden. Die Satellitenbilder mussten verarbeitet und für das Machine-Learning-Modell angepasst werden. Ebenfalls wurden multispektrale Informationen genutzt. Falschfarbenbilder wurden generiert und Indexberechnungen durchgeführt. Durch die Veränderung der Farbinformationen sollten stärkere Kontraste erzeugt werden, die eine bessere Erkennung von archäologischen Merkmalen ermöglichen können.

Es wurde das Modell Faster-R-CNN mit dem selbst erstellten Datensatz getestet. Die Ergebnisse des Trainings sind leider nicht gut genug, da die Genauigkeit in der Erkennung von archäologischen Objekten unzureichend ist. Gründe dafür sind meiner Meinung nach die geringe Anzahl der zur Verfügung stehenden Trainingsdaten und die nicht ausreichende Auflösung von 10 m pro Pixel. Viele der bekannten archäologischen Objekte in der Mongolei sind relativ klein und damit unter den in dieser Arbeit beschriebenen Methoden nicht nutzbar.

Um bessere Ergebnisse zu generieren sollte der Datensatz erweitert werden. Eine Alternative wäre die Nutzung von Satellitenbildern aus anderen Quellen, z.B.

von kostenpflichtigen Anbietern. Zusätzlich sollte getestet werden, ob hochauflösende Bilder zu besseren Ergebnissen führen könnten. Neben qualitativ besseren Trainingsdaten sollten auch andere Methoden von Machine-Learning getestet werden. Auch wenn Faster-R-CNN immer noch zu den genauesten Modellen für die Objekterkennung zählt, sollten neuere Modelle getestet werden, z.B. EfficientDet [38], BiDet [39] und R-FCN [40].

Literaturverzeichnis

[1] Willem F. Vletter and Rowin J. Van Lanen. Finding Vanished Routes: Applying a Multi-modelling Approach on Lost Route and Path Networks in the Veluwe Region, the Netherlands. *Rural Landscapes: Society, Environment, History*, 5(1):2, 2018.

[2] Dunja Boži-Stuli, Stanko Kruži, Sven Gotovac, and Vladan Papi. Complete model for automatic object detection and localisation on aerial images using convolutional neural networks. *Journal of Communications Software and Systems*, 14(1):82–90, 2018.

[3] Leena Matikainen, Kirsi Karila, Juha Hyyppä, Paula Litkey, Eetu Puttonen, and Eero Ahokas. Object-based analysis of multispectral airborne laser scanner data for land cover classification and map updating. *ISPRS Journal of Photogrammetry and Remote Sensing*, 128:298–313, 2017.

[4] Sarah Klassen, Jonathan Weed, and Damian Evans. Semi-supervised machine learning approaches for predicting the chronology of archaeological sites: A case study of temples from medieval angkor, Cambodia. *PLoS ONE*, 13(11):1–17, 2018.

[5] Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna Nemani. DeepSat - A learning framework for satellite imagery. *GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems*, 03-06-Nove:1–22, 2015.

[6] Teofilo F. Gonzalez. Handbook of approximation algorithms and metaheuristics. *Handbook of Approximation Algorithms and Metaheuristics*, pages 1–1432, 2007.

[7] Haifeng Li, Xin Dou, Chao Tao, Zhixiang Wu, Jie Chen, Jian Peng, Min Deng, and Ling Zhao. Rsi-cb: A large-scale remote sensing image classification benchmark using crowdsourced data. *Sensors (Switzerland)*, 20(6):28–32, 2020.

[8] Qun Liu, Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna Nemani. DeepSat V2: feature augmented convolutional neural nets for satellite image classification. *Remote Sensing Letters*, 11(2):156–165, 2020.

[9] Sara Zanni, Université Bordeaux Montaigne, Biljana LUČIĆ, Zaštitu Spomenika, Kulture Sremska, and Alessandro D E Rosa. From the Sky to the Ground : A Spatial Approach to the Archaeological Research in the Srem Region (Serbia), the Case Study of Pusta Dreispitz site. (660763), 2020.

[10] Karsten Lambers, Wouter B. Verschoof-van der Vaart, and Quentin P.J. Bourgeois. Integrating remote sensing, machine learning, and citizen science in dutch archaeological prospection. *Remote Sensing*, 11(7):1–20, 2019.

[11] Wouter Baernd Verschoof-van der Vaart and Karsten Lambers. Learning to Look at LiDAR: The Use of R-CNN in the Automated Detection of Archaeological Objects in LiDAR Data from the Netherlands. *Journal of Computer Applications in Archaeology*, 2(1):31–40, 2019.

[12] Wouter B. Verschoof-Van Der Vaart, Karsten Lambers, Wojtek Kowalczyk, and Quentin P.J. Bourgeois. Combining deep learning and location-based ranking for large-scale archaeological prospection of LiDAR data from the Netherlands. *ISPRS International Journal of Geo-Information*, 9(5), 2020.

[13] Lei Luo, Xinyuan Wang, Huadong Guo, Rosa Lasaponara, Xin Zong, Nicola Masini, Guizhou Wang, Pilong Shi, Houcine Khatteli, Fulong Chen, Shahina Tariq, Jie Shao, Nabil Bachagha, Ruixia Yang, and Ya Yao. Airborne and spaceborne remote sensing for archaeological and cultural heritage applications: A review of the century (1907–2017). *Remote Sensing of Environment*, 232(March):111280, 2019.

[14] ESA. *ESA's Optical High-Resolution Mission for GMES Operational Services*. 2015.

[15] Deodato Tapete and Francesca Cigna. Appraisal of opportunities and perspectives for the systematic condition assessment of heritage sites with co-

pernicus Sentinel-2 high-resolution multispectral imagery. *Remote Sensing*, 10(4):1–22, 2018.

[16] Abdulhakim Mohamed Abdi. Land cover and land use classification performance of machine learning algorithms in a boreal landscape using Sentinel-2 data. *GIScience and Remote Sensing*, 57(1):1–20, 2020.

[17] Tuna Kalayci, Rosa Lasaponara, John Wainwright, and Nicola Masini. Multispectral contrast of archaeological features: A quantitative evaluation. *Remote Sensing*, 11(8):1–23, 2019.

[18] Athos Agapiou, Dimitrios D. Alexakis, and Diofantos G. Hadjimitsis. Spectral sensitivity of ALOS, ASTER, IKONOS, LANDSAT and SPOT satellite imagery intended for the detection of archaeological crop marks. *International Journal of Digital Earth*, 7(5):351–372, 2014.

[19] Athos Agapiou, Dimitrios D. Alexakis, Apostolos Sarris, and Diofantos G. Hadjimitsis. Evaluating the potentials of sentinel-2 for archaeological perspective. *Remote Sensing*, 6(3):2176–2194, 2014.

[20] KW Wegmann. Assessing Coastal Landscape Change for Archaeological Purposes: Integrating Shallow Geophysics, Historical Archives and Geomorphology at Port Angeles,. *Archaeological …*, 252(February):229–252, 2012.

[21] Lei Luo, Xinyuan Wang, Huadong Guo, Rosa Lasaponara, Pilong Shi, Nabil Bachagha, Li Li, Ya Yao, Nicola Masini, Fulong Chen, Wei Ji, Hui Cao, Chao Li, and Ningke Hu. Google earth as a powerful tool for archaeological and cultural heritage applications: A review. *Remote Sensing*, 10(10):1–33, 2018.

[22] Meisam Amani, Arsalan Ghorbanian, Seyed Ali Ahmadi, Mohammad Kakooei, Armin Moghimi, S. Mohammad Mirmazloumi, Sayyed Hamed Alizadeh Moghaddam, Sahel Mahdavi, Masoud Ghahremanloo, Saeid Parsian, Qiusheng Wu, and Brian Brisco. Google Earth Engine Cloud Computing Platform for Remote Sensing Big Data Applications: A Comprehensive Review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13(September):5326–5350, 2020.

[23] Paulo Arévalo, Eric L. Bullock, Curtis E. Woodcock, and Pontus Olofsson. A Suite of Tools for Continuous Land Change Monitoring in Google Earth Engine. *Frontiers in Climate*, 2(December):1–19, 2020.

[24] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 7700 LECTU:437–478, 2012.

[25] Leslie N. Smith. Cyclical learning rates for training neural networks. *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017*, (April):464–472, 2017.

[26] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:3296–3305, 2017.

[27] Licheng Jiao, Fan Zhang, Fang Liu, Shuyuan Yang, Lingling Li, Zhixi Feng, and Rong Qu. A survey of deep learning-based object detection. *IEEE Access*, 7(3):128837–128868, 2019.

[28] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, U C Berkeley, and Jitendra Malik. R-CNN. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1:5000, 2014.

[29] Ross Girshick. Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:1440–1448, 2015.

[30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[31] Taylan Sen, Md Kamrul Hasan, Minh Tran, Yiming Yang, and Mohammed Ehsan Hoque. Say CHEESE: Common human emotional expression set encoder and its application to analyze deceptive communication. *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, pages 357–364, 2018.

[32] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, 4(January):3320–3328, 2014.

[33] Mark Everingham, S. M.Ali Eslami, Luc Van Gool, Christopher K.I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015.

[34] A. R. Beck. Archaeological site detection: the importance of contrast. 2007.

[35] Weijie Kong, Nannan Li, Thomas H. Li, and Ge Li. Deep Pedestrian Detection Using Contextual Information and Multi-level Features. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10704 LNCS(January):166–177, 2018.

[36] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 07-12-June:1–9, 2015.

[37] Wouter B. Verschoof-van der Vaart and Juergen Landauer. Using CarcassonNet to automatically detect and trace hollow roads in LiDAR data from the Netherlands. *Journal of Cultural Heritage*, 2020.

[38] Mingxing Tan, Ruoming Pang, and Quoc V. Le. EfficientDet: Scalable and efficient object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 10778–10787, 2020.

[39] Ziwei Wang, Ziyi Wu, Jiwen Lu, and Jie Zhou. BiDet: An Efficient Binarized Object Detector. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2046–2055, 2020.

[40] Region-based Fully Convolutional Networks and Jifeng Dai. Aron cap II - A crítica filosófica.pdf.

Anhang A

Dateiformate

A.1 Annotationen

A.1.1 COCO

```
<object>
        <name>fig</name>
        <truncated>0</truncated>
        <bndbox>
                <xmin>256</xmin>
                <ymin>27</ymin>
                <xmax>381</xmax>
                <ymax>192</ymax>
        </bndbox>
</object>
```

A.1.2 Pascal-VOC

```
filename,width,height,class,xmin,ymin,xmax,ymax
000001.jpg,500,375,wallanlage,111,144,134,174
000002.jpg,500,375,wallanlage,178,84,230,143
000003.jpg,500,466,wallanlage,115,139,180,230
```

A.2 Config-Datei

```
batch_size: 1
optimizer {
  momentum_optimizer: {
    learning_rate: {
      manual_step_learning_rate {
        initial_learning_rate: 0.0002
        schedule {
          step: 0
          learning_rate: .0002
        }
        schedule {
          step: 900000
          learning_rate: .00002
        }
        schedule {
          step: 1200000
          learning_rate: .000002
        }
      }
    }
    momentum_optimizer_value: 0.9
  }
  use_moving_average: false
}
fine_tune_checkpoint:
        "/usr/home/username/tmp/model.ckpt-#####"
from_detection_checkpoint: true
load_all_detection_checkpoint_vars: true
gradient_clipping_by_norm: 10.0
data_augmentation_options {
  random_horizontal_flip {
  }
}
```

Anhang B

Evaluierungsmetrik

Tabelle B.1: Diese Tabelle gibt einen Überblick von den unterschiedlichen mAP- und AR-Metriken, nach COCO.

Precision/mAP	mAP over classes averaged over IOU thresholds - from .5 to .95 with .05 increments.
Precision/mAP@.50IOU	mean average precision at 50% IOU
Precision/mAP@.75IOU	mean average precision at 75% IOU
Precision/mAP small	mean average precision for small objects (area $<$ 32^2 pixels)
Precision/mAP medium	mean average precision for medium sized objects (32^2 pixels $<$ area $<$ 96^2 pixels)
Precision/mAP large	mean average precision for large objects (96^2 pixels $<$ area $<$ 10000^2 pixels).
Recall/AR@1	average recall with 1 detection
Recall/AR@10	average recall with 10 detection
Recall/AR@100	average recall with 100 detection
Recall/AR@100 small	average recall for small objects with 100
Recall/AR@100 medium	average recall for medium objects with 100
Recall/AR@100 large	average recall for large objects with 100 detections

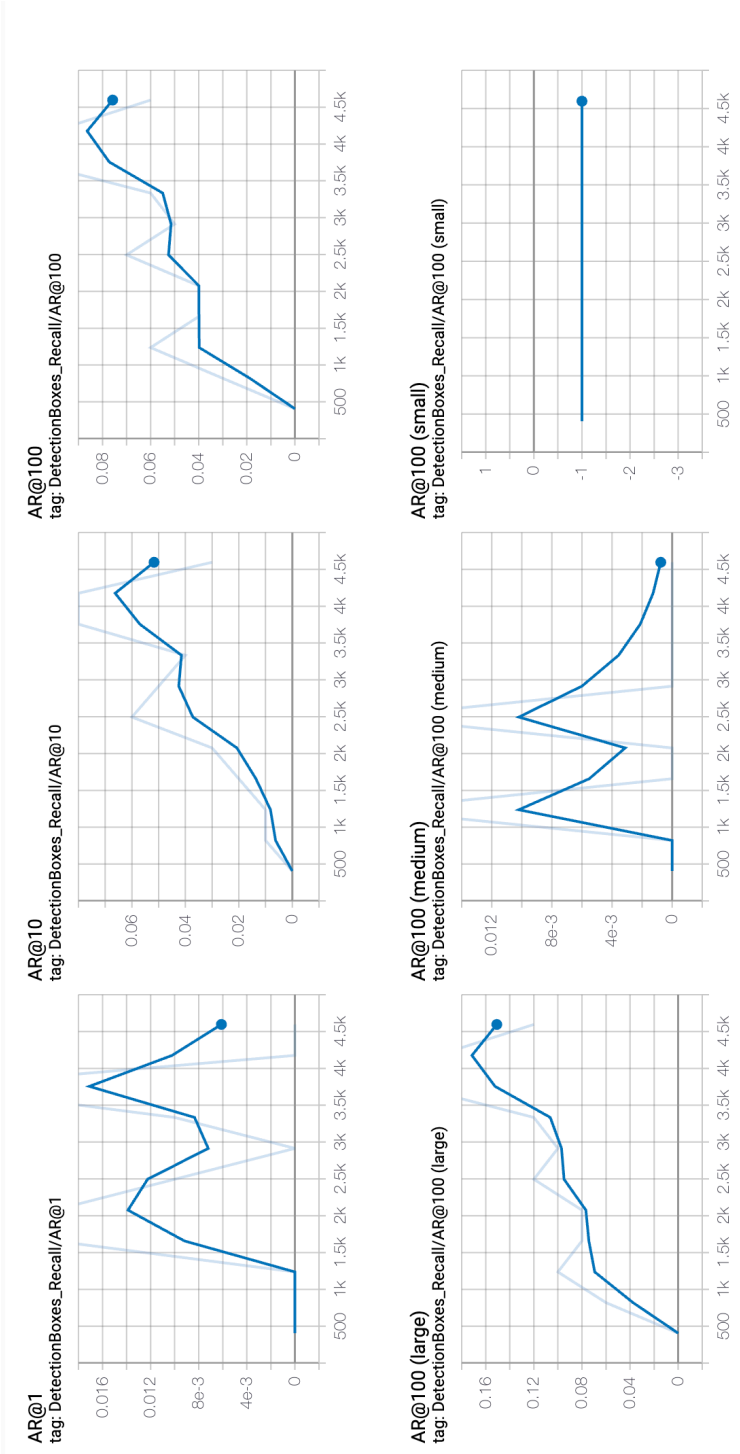

Abbildung B.1: Ein Übersicht der AR-Diagramme wird hier angezeigt.

Selbstständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit mit dem Titel „Semi-autonome Klassifizierung archäologischer Strukturen“ selbstständig angefertigt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, bildliche Darstellungen, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Dresden, den 13.04.2021 Huy Do Duc