Hochschule Darmstadt

Fachbereiche Mathematik und Naturwissenschaften & Informatik

Active Learning für die 3D-Objekterkennung in Punktwolken

Abschlussarbeit zur Erlangung des akademischen Grades
Master of Science (M.Sc.)
im Studiengang Data Science

Vorgelegt von:
Michael Trei
Matrikelnummer: 759297

Referent: Prof. Dr. Andreas Weinmann
Korreferentin: Prof. Dr. Elke Hergenröther

Ausgabedatum: 03.05.2024

Abgabedatum: 17.10.2024

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht. Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen. Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Darmstadt, den

Abstract (deutsch)

In der Objekterkennung sind Deep Learning Modelle zum Standard geworden. Durch das Lernen anhand von Beispielen können diese eine beeindruckende Leistungsfähigkeit erzielen. Der größte Nachteil des Deep Learning besteht darin, dass eine große Menge an annotierten Daten benötigt wird. Um die Menge an annotierten Daten zu reduzieren, wurde das Active Learning (AL) entwickelt. Das Ziel des AL ist es, die Daten mit dem größten Lerneffekt zu identifizieren und nur diese mit Annotationen zu versehen. Während das AL in der Objekterkennung in Bildern bereits Gegenstand vieler wissenschaftlicher Veröffentlichungen ist, besteht für die Anwendung von AL in der dreidimensionalen Objekterkennung noch Forschungsbedarf.

Die Effektivität von Active Learning hängt signifikant von der eingesetzten Modellarchitektur sowie der Charakteristik des zugrundeliegenden Datensatzes ab. Infolgedessen führt eine Erweiterung des methodischen Repertoires zu einer verbesserten Adaptivität von AL auf neue Datensätze und Modellarchitekturen. Die vorliegende Arbeit befasst sich mit der Anwendung von Active Learning Methoden im Kontext der dreidimensionalen Objekterkennung in Punktwolken. Hierbei steht die Übertragbarkeit von AL-Algorithmen aus dem zweidimensionalen in den dreidimensionalen Bereich im Fokus.

Im Rahmen der Arbeit wurden zunächst die spezifischen Herausforderungen bei der Anwendung von AL-Methoden auf Punktwolken identifiziert und analysiert. Basierend auf diesen Erkenntnissen erfolgte die Entwicklung einer Taxonomie zur systematischen Selektion geeigneter Algorithmen. Die ausgewählten Methoden wurden implementiert und unter Verwendung des Fully Convolutional Neural Networks FCAF3D auf einem nicht öffentlichen Datensatz von Rohrleitungssystemen evaluiert.

Die empirische Untersuchung umfasste neun AL-Algorithmen, von denen vier (Region of Interest Matching, Localization Stability, PPAL und eine Variation der Localization Stability) eine konsistente Reduktion der benötigten Datenmenge für einzelne Objektklassen demonstrieren. Die Localization Stability erwies sich mit einer Reduktion von 28% als besonders effektiv.

Abstract (englisch)

Deep learning models have become the standard in object detection. By learning from examples, they can achieve impressive performance. The biggest disadvantage of deep learning is that a large amount of annotated data is required. Active Learning (AL) was developed to reduce the amount of annotated data. The aim of AL is to identify the data with the greatest learning effect and only annotate this data. While AL in object detection in images is already the subject of many scientific publications, there is still a need for research into the application of AL in three-dimensional object detection.

The effectiveness of active learning depends significantly on the model architecture used and the characteristics of the underlying data set. Consequently, an extension of the methodological repertoire leads to an improved adaptivity of AL to new data sets and model architectures. This thesis deals with the application of active learning methods in the context of three-dimensional object detection in point clouds. The focus here is on the transferability of AL algorithms from the two-dimensional to the three-dimensional domain.

As part of the work, the specific challenges in the application of AL methods to point clouds were first identified and analyzed. Based on these findings, a taxonomy was developed for the systematic selection of suitable algorithms. The selected methods were implemented and evaluated using the Fully Convolutional Neural Network FCAF3D on a non-public dataset of pipeline systems.

The empirical study included nine AL algorithms, four of which (Region of Interest Matching, Localization Stability, PPAL and a variation of Localization Stability) demonstrated a consistent reduction in the amount of data required for individual object classes. Localization Stability proved to be particularly effective with a reduction of 28%.

Inhalt

Abbildungsverzeichnis

Formelverzeichnis

Abkürzungsverzeichnis

AL	-	Active Learning
BB	-	Bounding Box
IoU	-	Intersection over Union
RoI	-	Region of Interest
NN	-	Neuronales Netz
OD	-	Object Detection
AP	-	Average Precision
GPS	-	Global Positioning System
KI	-	Künstliche Intelligenz
RGB	-	Red, Green, Blue
RGBD	-	Red, Green, Blue, Depth
CAD	-	Computer-Aided Design
BLAD	-	Box Level Active Detection
LOCAL	-	Localization-based active learning
CV	-	Computer Vision
ReDAL	-	Region-based and Diversity-aware Active Learning

1 Einleitung

1.1 Hintergrund & Problemstellung

Die Objekterkennung (engl. object detection, kurz OD) ist in vielen Bereichen, wie in der Gesichtserkennung, Qualitätskontrolle oder in Fahrassistenzsystemen, vertreten [1], [2], [3]. Bei der OD werden Objekte mit Hilfe einer Bounding Box (BB) lokalisiert und eine Klassenzugehörigkeit bestimmt. In zweidimensionalen Bilddaten lässt sich eine BB als Rechteck repräsentieren, das ein Objekt umschließt. Die kontinuierliche Weiterentwicklung von Deep Learning Methoden aus dem Bereich der Künstlichen Intelligenz (KI) hat zu leistungsfähigen Modellen der Objekterkennung geführt, da diese anhand von Beispielen die gewünschte Ausgabe lernen [4]. Dies führt zu der zentralen Voraussetzung annotierter Datensätze, was den größten Nachteil von Deep Learning basierten Methoden darstellt. Das Annotieren von Datensätzen wird manuell von Menschen durchgeführt, wobei zur Schätzung des Aufwands etwa 35s pro Bounding Box zugrunde gelegt werden können [5]. Bezogen auf den Umfang von State-of-the-Art KI-Modellen mit ca. 2,5 M Objekten in einem Datensatz müssen etwa 13 Personenjahre aufgewendet werden, um die dafür erforderliche Trainingsbasis zu labeln. Somit ist für ein robustes KI-Modell in der Regel erheblicher Aufwand erforderlich. Um die benötigte Datenmenge zu reduzieren und somit den Aufwand und die Kosten der Datenannotation zu senken, wurde das Active Learning (AL) entwickelt. Das Ziel des AL besteht darin, die Datenpunkte mit dem größten Lerneffekt eines KI-Modells zu identifizieren und nur diese mit Annotationen zu versehen [6]. Aufgrund des hohen Aufwands für die Datenannotation ist das AL in der zweidimensionalen Objekterkennung in Bildern bereits häufig Gegenstand der Forschung gewesen [7].

Eine weitere Einsatzmöglichkeit von Objekterkennung ist der 3D-Bereich. So ermöglichen u.a. 3D Laserscanner die Erfassung der räumlichen Umgebung inklusive Position von Oberflächen. Die Distanz zu den Oberflächen wird, wie bei einem Radar, mit dem Time-of-Flight-Verfahren bestimmt und die einzelnen Messpunkte werden in Punktwolken zusammengefasst. Durch den Vorteil, Tiefeninformationen präzise zu erfassen, ergeben sich viele potenzielle Einsatzbereiche wie zum Beispiel in der Forstwirtschaft, Städteplanung und dem autonomen Fahren [8], [9], [10]. Durch diese zusätzliche Dimension erhöht sich die benötigte Zeit zum Labeln der Daten auf über 100s [11], wodurch der Einsatz von 3D-Objekterkennung sehr aufwändig und kostspielig ist.

Trotz des großen Potenzials von AL ist es in dem aufstrebenden Gebiet der 3D-Objekterkennung unterrepräsentiert. In der vorliegenden Arbeit wird die Anwendung von Active Learning für die Objekterkennung in Punktwolken untersucht. Hierbei steht die Übertragung von Algorithmen aus der Domäne von 2D Bildern auf 3D Punktwolken im Fokus.

1.2 Zielsetzung

Im Rahmen der vorliegenden Masterarbeit wird die Übertragung von Active Learning Methoden aus dem zweidimensionalen Bereich auf dreidimensionale Punktwolken untersucht. Die Relevanz dieser Untersuchung ergibt sich daraus, dass die Effektivität von AL von der Wahl des eingesetzten Modells sowie der Charakteristik des zugrundeliegenden Datensatzes abhängt (siehe Kapitel 3.3). Infolgedessen führt eine Erweiterung des methodischen Repertoires zu einer verbesserten Adaptivität von AL auf neue Datensätze und Modellarchitekturen. Hierfür erfolgt eine Identifizierung relevanter Active Learning Ansätze, die sich in der zweidimensionalen Objekterkennung bereits als effektiv erwiesen haben. Um die Anwendbarkeit der Techniken zu belegen, werden ausgewählte Algorithmen implementiert. Diese werden in einer Machbarkeitsstudie auf ihre Leistungsfähigkeit geprüft. In dieser experimentellen Untersuchung wird die Datenauswahl für das Trainieren der KI-Modelle von den AL-Algorithmen bestimmt. Zur Evaluierung wird ein Modell auf Basis randomisiert gelabelter Daten trainiert. Durch einen Vergleich der Modelle lässt sich die potenzielle Einsparung bestimmen, wodurch die Leistungsfähigkeit der Algorithmen empirisch evaluiert werden kann.

1.3 Aufbau

In Kapitel 2 erfolgt zunächst eine Einführung in die Thematik des Active Learning. Hierfür wird eine Taxonomie für AL-Algorithmen der Objekterkennung erstellt. Dies bietet eine Übersicht der verschiedenen Aspekte der Algorithmen und wird als Einleitung in die Thematik verwendet. Anschließend werden AL-Algorithmen für die 2D Objekterkennung dargelegt und darüber hinaus bestehende AL-Algorithmen betrachtet, welche für den Einsatz in Punktwolken entwickelt wurden.

Anhand dieser Grundlagen werden in Kapitel 3 Herausforderungen bei der Anwendung von AL in Punktwolken identifiziert. Diese dienen als Basis für die Identifizierung von Lösungsansätzen. Des Weiteren werden die Hürden bei der Auswahl von AL-Algorithmen anhand von Forschungsergebnissen erörtert.

Der Aufbau der Versuchsreihe wird in Kapitel 4 vorgestellt. Hierfür wird der verwendete Datensatz und das Modell beschrieben und anschließend wird auf den generellen Ablauf der Experimente eingegangen. Hier erfolgt die Auswahl der Algorithmen für die Experimente. Die zuvor dargestellte Taxonomie wird dabei als Instrument zur Selektion der Algorithmen herangezogen. Der Selektionsprozess basiert dabei auf dem Prinzip, ein möglichst breites Spektrum der Taxonomie abzudecken.

Die Ergebnisse der Experimente werden in Kapitel 5 vorgestellt und in Kapitel 6 erfolgt eine Interpretation der Resultate. Abschließend erfolgt in Kapitel 7 eine Zusammenfassung der Resultate dieser Arbeit und es werden Anstöße für zukünftige Forschung gegeben.

2 Stand der Technik

In diesem Kapitel werden sowohl die Grundlagen des Active Learning erörtert als auch eine Übersicht über die relevanten wissenschaftlichen Arbeiten von AL in der Computer Vision dargestellt. Zunächst werden alternative Maßnahmen dargestellt (2.1), welche zur Reduktion des Datenbedarfs eingesetzt werden können. Dies ermöglicht die Kontextualisierung von AL, wodurch die Auswahl von AL gegenüber den Alternativen begründet wird. Im weiteren Verlauf des Kapitels erfolgt die Fokussierung auf Active Learning.

Zunächst werden die Grundlagen des Active Learning vorgestellt (2.2), welche für jeden AL-Algorithmus von Bedeutung sind. Darauf folgt eine Übersicht über die Grundlegenden Szenarien zur Generierung von Anfragen im AL (2.3). In Kapitel (2.4) wird eine selbst erstellte Taxonomie von AL-Techniken vorgestellt, welche dazu dient die einzelnen Bestandteile der vorgestellten Algorithmen besser einordnen zu können. Anschließend folgt eine Einführung in AL-Techniken (2.5), welche öfters wiederverwendet wurden. Ein Großteil dieses Kapitels ist die Übersicht über Arbeiten aus dem 2D (2.6) und 3D (2.7) Bereich der Computer Vision. Abschließend wird noch die Evaluation von AL-Algorithmen dargestellt (2.8).

2.1 Techniken zur Reduktion des Datenbedarfs

In der Ära des Deep Learning stellt die Verfügbarkeit großer Datenmengen sowohl eine Chance als auch eine Herausforderung dar. Während umfangreiche Datensätze die Leistungsfähigkeit von Modellen steigern können, bringen sie auch erhebliche Kosten in Bezug auf Speicherung, Verarbeitung und manuelle Annotation mit sich. Um diese Herausforderungen zu bewältigen, wurden verschiedene Techniken zur Datenreduktion entwickelt. Eine häufig eingesetzte Technik ist die Data Augmentation. Hierbei wird der Datensatz künstlich vergrößert, indem verschiedene Transformationen auf die Daten und Annotationen angewendet werden. Diese umfassen in der zwei- und dreidimensionalen Computervision Operationen wie z. B. Rotation, Spiegelung, oder das künstliche Hinzufügen von Rauschen. Diese künstliche Vergrößerung des Datensatzes verbessert die Generalisierbarkeit durch eine Verringerung des Overfittings. Das Overfitting äußert sich, durch eine gute Performance auf den Trainingsdaten, welche nicht auf die Testdaten übertragbar ist. Hierfür ist die Ursache, dass das Modell sich auf nicht signifikante Merkmale der Trainingsdaten anpasst. Die Funktionsweise von Data Augmentation ist, dass durch ständige Transformationen das Modell weniger Möglichkeiten besitzt sich auf irrelevante Merkmale zu stützen und stattdessen die zugrundeliegende Struktur der Daten erlernt [12].

Während Data Augmentation Strategien versuchen die Generalisierbarkeit zu steigern, nutzt das Transfer Learning die Fähigkeit von Neuronalen Netzen (NN) sich auf neue Daten anzupassen. Die Vorgehensweise ist, dass ein Modell auf einem großen Datensatz vortrainiert wird, gefolgt von einem Finetuning auf dem Zieldatensatz. Der Kerngedanke besteht darin, dass das

Modell vorhandenes Wissen nutzen und übertragen kann, anstatt bei jeder neuen Aufgabe von Grund auf neu zu lernen [13]. Ein Nachteil dieser Technik ist die Zunahme der benötigten Rechenleistung. Soll eine neue Modellarchitektur trainiert werden, muss sowohl das Vortraining als auch das Finetuning erneut durchgeführt werden. Außerdem schließen viele öffentliche Datensätze explizit den kommerziellen Einsatz aus, was den Einsatz von Transfer Learning für kommerzielle Zwecke verhindert [14].

Sowohl Data Augmentation als auch das Transfer Learning nutzen nur bereits gelabelte Daten. Im Gegensatz dazu ermöglicht das Self-Supervised Learning Daten ohne Annotationen für das Training zu nutzen. Die Grundidee besteht darin, dass automatisch Ziele für das Training generiert werden, welche es dem Modell erlauben die Struktur der Daten zu erlernen. Dies wird z. B. bei Autoencodern angewendet.

Autoencoder bestehen aus zwei Hauptkomponenten: Encoder und Decoder. Der Encoder komprimiert die Eingabedaten in eine kompakte Repräsentation, während der Decoder diese Repräsentation zur Rekonstruktion der ursprünglichen Eingabe nutzt. Diese Architektur ermöglicht ein unüberwachtes Trainingsverfahren, bei dem die Fehlerfunktion die Abweichung zwischen Ein- und Ausgabe quantifiziert. Zur erfolgreichen Bewältigung dieser Aufgabe ist das Erlernen kompakter Repräsentationen erforderlich, die die essenziellen Charakteristika der Daten erfassen. Nach Abschluss des Trainings kann der Encoder beispielsweise zur Merkmalsextraktion eingesetzt werden. Nachteilig bei dieser Art des Trainings ist der Zwang zu einer vorgegebenen Modellarchitektur. Ist die Kapazität des Encoders zu groß lernt der Autoencoder nur die Eingabedaten zu kopieren, ohne dass nützliche Repräsentationen gelernt werden. Folglich muss für andere Modellarchitekturen eine alternative Technik genutzt werden um den Bedarf an gelabelten Daten zu verringern [15].

Das Active Learning ist eine weitere Technik den Bedarf an gelabelten Trainingsdaten zu verringern. Dazu werden die Daten aus einem Pool von ungelabelten Daten herausgesucht, die den größten Nutzen hinsichtlich des Trainingserfolgs eines KI-Modells bieten und nur die ausgewählten Daten werden im Anschluss annotiert [6]. Dieses Vorgehen bringt Vorteile mit sich. AL ist im Gegensatz zum Transfer Learning nicht auf externe Datenquellen angewiesen. Auch der große Bedarf an Rechenleistung, welcher für das Transfer Learning und dem Self-Supervised Learning wird durch AL verringert.

2.2 Active Learning

Active Learning ist ein maschinelles Lernverfahren, das auf einigen grundlegenden Annahmen basiert. Eine zentrale Annahme ist, dass nicht alle Daten den gleichen Informationsgehalt aufweisen. Hieraus resultiert, dass sich die Datenpunkte in ihrem Nutzen für den Lernprozess unterscheiden. Auf Basis dieser Annahme kann die Schlussfolgerung gezogen werden, dass durch das geschickte Auswählen der Trainingsdaten auch eine geringere Menge an Daten zu einem guten Modell führt [6].

Um eine geschickte Auswahl der Trainingsdaten durchzuführen, ist eine Metrik notwendig. Diese Metrik muss den Informationsgehalt bestimmen, um die Datenpunkte nach Ihrem Nutzen für den Lernprozess zu bewerten. Die Existenz einer solchen Metrik ist eine weitere Voraussetzung für das AL.

Eine weitere Voraussetzung für Active Learning ist die Existenz eines sogenannten Orakels. Das Orakel muss dazu in der Lage sein, die wahren Annotationen für ausgewählte Datenpunkte zu liefern. Zur Illustration des Orakels kann es als Lehrer dargestellt werden, der dem Schüler (dem AL-Algorithmus) in unklaren Fällen mit der korrekten Antwort weiterhilft.

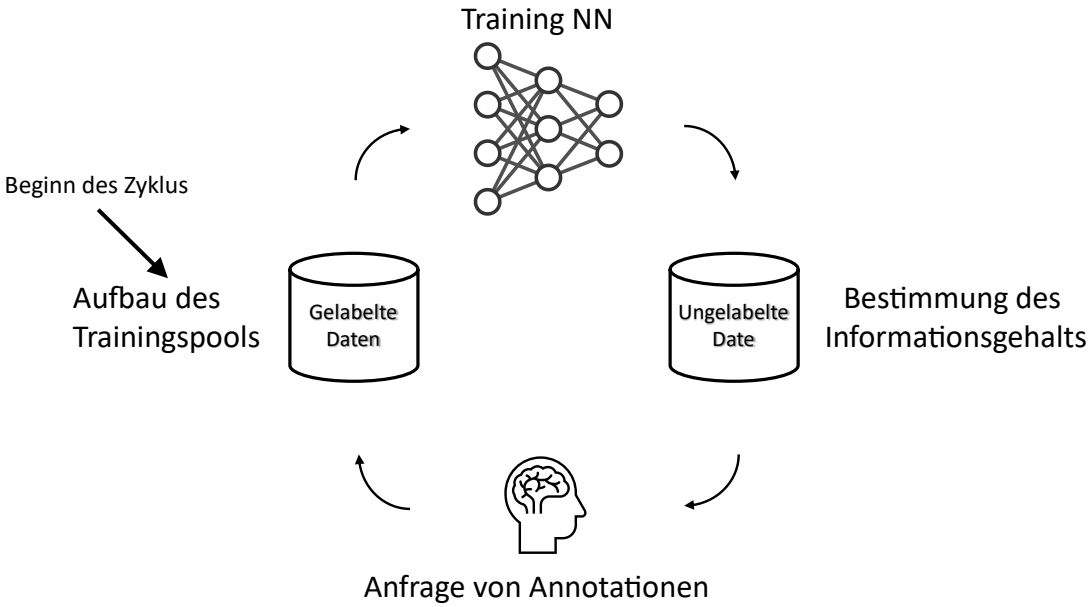


Abbildung 1: Der Active Learning Zyklus

Dieses Orakel kann ein menschlicher Experte oder eine andere zuverlässige Informationsquelle sein. Ein Beispiel für ein nicht menschliches Orakel ist das Testen von AL-Algorithmen. Dies wird auf bereits annotierten Daten durchgeführt, wobei die meisten Annotationen dem Algorithmus zunächst vorenthalten werden. Hier wird die Rolle des Orakels von einer Datenbank übernommen, welche die wahren Label erst auf Anfrage übergibt.

Aus diesen Annahmen lässt sich der generelle Rahmen für AL-Algorithmen ableiten, dargestellt in Abbildung 1.

1. Ein Modell wird auf den bereits annotierten Daten trainiert
2. Der Informationsgehalt von nicht annotierten Daten wird bestimmt
3. Frage das Orakel nach den Annotationen für die informativsten Daten
4. Füge die angefragten Annotationen dem Trainingsdaten hinzu

Dieser Prozess wird iterativ ausgeführt, bis das Modell die gewünschte Leistungsfähigkeit erreicht hat, das Budget zum Annotieren der Daten ausgeschöpft ist oder der Pool an ungelabelten Daten ausgeschöpft wurde.

Trotz der Vorteile gibt es jedoch einige Herausforderungen, die bei der Anwendung von AL berücksichtigt werden müssen. Eine zentrale Annahme im AL ist, dass das Orakel, welches hauptsächlich durch einen Menschen realisiert wird, immer korrekte Annotationen liefert. In der Realität machen Menschen jedoch Fehler. In der Objekterkennung können ähnlich wirkende Klassen leicht verwechselt werden oder teilweise verdeckte Objekte werden übersehen. Arbeiten mehrere Personen an demselben Datensatz können unterschiedliche Meinungen oder Annahmen zu abweichenden Annotationen führen. Außerdem sind die Personen, die das Labeln ausführen nicht immer Domänenexperten, was zu Fehlern durch fehlendes Fachwissen führt. Das alles kann die Qualität der gelabelten Daten negativ beeinflussen und somit die Leistung des Modells beeinträchtigen. Ein fehlerfreies Orakel ist die Annahme der meisten wissenschaftlichen Literatur, welche sich mit AL befasst. Der Umgang mit fehlerbehafteten Orakeln erfordert den Einsatz von anderen Techniken, welche nicht im Umfang dieser Arbeit behandelt werden. Für eine Übersicht wie mit fehlerhaften Labeln umgegangen werden kann, wird auf [16] verwiesen.

Ein weiteres Problem sind die variierenden Annotationskosten. Die Komplexität der zu annotierenden Daten beeinflusst den Zeitaufwand erheblich. Komplexe Daten erfordern oft mehr Zeit, da sie einen höheren Anspruch an den Daten-Annotierer stellen, Rücksprachen mit Domänenexperten oder Fachabteilungen erfordern sowie zusätzliche Recherchen notwendig machen können.

Das Kaltstartproblem stellt eine weitere Herausforderung dar. AL-Algorithmen gehen von einem initial gelabelten Datensatz für das Training und die Evaluierung aus. Wenn dieser Datensatz unterrepräsentierte Klassen enthält, kann dies die Auswahl der Daten für die Annotation und somit die Leistung des Modells beeinflussen.

Schließlich ist die Trainingszeit ein wichtiger Aspekt. Deep Learning-Verfahren zeichnen sich durch hohe Rechenintensität und folglich erheblichen Zeitaufwand aus. Bei der Integration von AL in existierende Annotationsprozesse können so Verzögerungen auftreten. Benötigt eine AL-Iteration mehr Zeit als das Annotieren der Daten kommt es zu Leerlaufzeiten für die Fachkräfte der Datenannotation. Dies kann die Kosten für die Annotation erhöhen oder erfordert weitere Anpassungen im Annotationsprozess. Insbesondere Ensemble-Methoden (siehe Kapitel 2.5), die auf dem Training multipler Modelle basieren, können die Trainingszeit vervielfachen.

2.3 Query Szenarien

Active Learning kann grob in drei Szenarien eingeteilt werden. Diese basieren auf der Art und Weise wie die ungelabelten Daten dem Modell zur Bestimmung

des Informationsgehalts präsentiert und zur Auswahl weiter an das Orakel übergeben werden [6]. Für einen Überblick werden hier alle Szenarien vorgestellt.

Pool-based

Das Pool-based Active Learning ist die am häufigsten angewendete Art des AL. Hier steht ein großer, statischer Pool an Daten ohne Annotationen zur Verfügung. Der Informationsgehalt wird in jeder Iteration für den gesamten Pool bestimmt und die informativsten Datenpunkte werden dem Orakel übergeben. Dies ist die Ausgangslage für viele Anwendungsfälle, in denen viele Daten auf einmal gesammelt werden können z. B. wenn mit Data-Mining Bilder aus Online-Quellen gesammelt werden [17].

Stream-based

Hier treffen kontinuierlich neue Daten ein und können nicht in der Gesamtheit gespeichert werden, bis ein großer Pool an Daten zur Verfügung steht. Anwendungsfälle sind beispielsweise das Aufdecken von Kreditkartenbetrug [18] oder die automatisierte Qualitätskontrolle in der Industrie [19], bei denen die Daten sequentiell eintreffen und ein schnelles Reagieren auf neue Situationen von entscheidender Bedeutung ist.

Membership Query Synthesis

In diesem Szenario werden die Daten zum Labeln nicht aus einem Bestand ausgewählt, sondern durch generative Methoden erstellt. Hier entfällt der Aufwand für das Sammeln von Daten, birgt aber das Risiko, dass die erzeugten Daten für den menschlichen Daten-Annotierer schwer zu interpretieren sind und in ihrer Verteilung von denen der echten Welt abweichen können [20]

In dieser Arbeit wird ausschließlich das Pool-based AL betrachtet. Dies liegt einerseits an dem Aufbau des Datensatzes, welcher in Kapitel 4.1 im Detail beschrieben wird. Außerdem sind die meisten AL-Algorithmen im Bereich der Computer Vision für das Pool-based AL entwickelt worden, wodurch eine große Auswahl an diesen Besteht.

2.4 Taxonomie von Active Learning Techniken

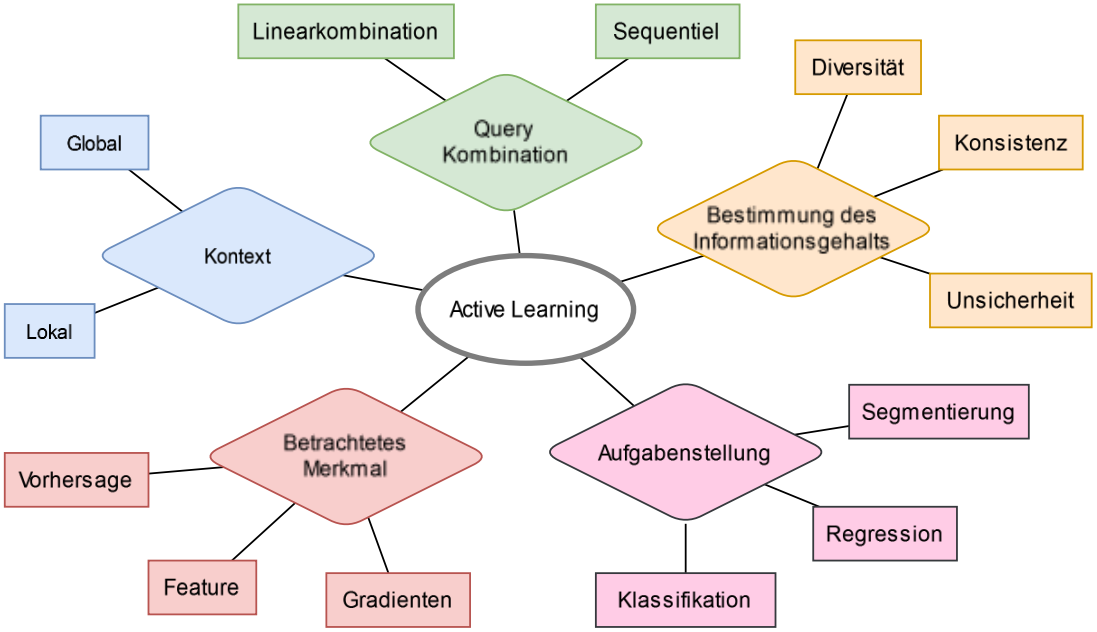


Abbildung 2: Taxonomie der Active Learning Techniken

Die Entwicklung von Taxonomien dient der Strukturierung diverser Aspekte von Themenbereichen. Existierende Klassifikationssysteme für Active Learning-Algorithmen beziehen sich auf spezifische Bereiche des AL, wodurch diese Defizite in der umfassenden Abbildung von AL-Techniken im Bereich der Computer Vision aufweisen [7], [21], [22]. Aus diesem Grund erfolgte die Konzeption einer spezifischen Taxonomie. Diese adressiert die Besonderheiten des Active Learning im Kontext der Computer Vision.

Eine solche domänenspezifische Kategorisierung ermöglicht eine präzisere Einordnung und Analyse von AL-Methoden für Bildverarbeitungsaufgaben. Die neu entwickelte Taxonomie berücksichtigt ihre grundlegenden Prinzipien und Auswahlkriterien der Computer Vision. Dadurch wird eine detailliertere Betrachtung der Facetten von AL-Strategien im visuellen Bereich realisiert.

Die Einteilung in die verschiedenen Klassen ist nicht exklusiv, die meisten Algorithmen umfassen mehrere Klassen. Dargestellt in Abbildung 2 ist eine Übersicht, welche die allgemeinen Oberklassen im Zentrum und die Unterklassen am Rand darstellen. Die folgende Übersicht stellt die Hauptkategorien dieser Taxonomie vor und bildet die Grundlage für eine Erläuterung der einzelnen Ansätze.

2.4.1 Bestimmung des Informationsgehalts

Die Bestimmung des Informationsgehalts ist die Hauptaufgabe im AL. Die Unsicherheit eines Modells zu nutzen ist die einfachste und eine am häufigsten verwendete Möglichkeit den Informationsgehalt zu bestimmen [6]. Zum Beispiel wird im Fall einer binären Klassifikation für jeden Datenpunkt eine Wahrscheinlichkeit zwischen 0 und 1 ausgegeben. Eine einfache Möglichkeit

die Unsicherheit zu schätzen, ist es die Datenpunkte auszuwählen, welche eine Wahrscheinlichkeit nahe der Klassifikationsgrenze, meist 0.5, besitzen. Dieser Ansatz fokussieren sich primär auf die Selektion von Datenpunkten nahe der Entscheidungsgrenze.

Die Konzentration auf die Entscheidungsgrenze resultiert in einer Verfeinerung der Klassifikationsgrenzen, wodurch sich die Leistungsfähigkeit des Modells für ebendiese Fälle steigert. Allerdings kann dies zu einer suboptimalen Abdeckung des gesamten Merkmalraums führen, da Regionen mit hoher Sicherheit unterrepräsentiert bleiben. Da DL-Modelle für die Klassifikation häufig eine zu hohe Sicherheit für die erkannte Klasse ausgeben, selbst wenn die Eingabe nicht den Trainingsdaten ähneln, erweist sich als problematisch.

Dies führt dazu, dass die angefragten Daten nicht der Verteilung der Grundgesamtheit entsprechen und durch den iterativen Charakter von AL entfernt sich die Verteilung der Gelabelten Daten immer weiter von der dieser [23]. Dies kann zu einem sich selbst verstärkenden Bias führen, bei dem das Modell kontinuierlich Datenpunkte in bereits überrepräsentierten Bereichen des Merkmalsraums auswählt, da die Exploration nur an der Entscheidungsgrenze stattfindet. Folglich wird die Diversität des Datensatzes eingeschränkt, was die Generalisierungsfähigkeit des Modells beeinträchtigt.

Vergleicht eine Technik verschiedene Ausgaben miteinander, wird diese in die Klasse der Konsistenz eingeordnet. Diese hat ihren größten Nutzen für die Aufgabe der Regression, da hier keine Wahrscheinlichkeitsverteilung für die Bestimmung der Unsicherheit zur Verfügung steht. Eine Möglichkeit hierfür besteht in der Quantifizierung der Konsistenz zwischen verschieden transformierten Versionen der Eingabedaten. Die Idee dieser Methode ist, dass ein Modell, dessen Erkennungen stabil bleiben, wenn die Daten transformiert werden, bereits ein gutes Verständnis für die Daten besitzt. Treten hingegen starke Änderungen in der Vorhersage auf, konnte das Modell noch nicht die inhärente Struktur der Daten erfasst hat. Auch die Verwendung eines Ensembles (Seite 22) kann für die Bestimmung der Konsistenz verwendet werden.

Techniken, welche die Struktur der Daten betrachten fallen unter die Klasse der Diversität. Das Ziel dieser Techniken ist es Anhand der Struktur der Daten eine Teilmenge auszuwählen, die den Eingaberaum möglichst gut abdeckt. Dem liegt die Annahme zu Grunde, dass diverse Daten den Informationsgehalt des Trainingsdatensatzes erhöhen. Wird das Modell im Trainingsverlauf mit möglichst verschiedenen Daten konfrontiert führt dies zu einer gesteigerten Generalisierbarkeit, da die Notwendigkeit in unbekannte Bereiche zu extrapolieren verringert wird. Außerdem zeigt sich die Diversitätsmaximierung in der initialen Phase des AL-Prozesses als effektiver im Vergleich zur Unsicherheitsbestimmung. Bei limitiertem Trainingsdatensatz resultiert die Unsicherheitsschätzung häufig in geringerer Zuverlässigkeit, was zu Instabilitäten im AL-Prozess führen kann [21].

Eine Herausforderung besteht in der potenziellen Bevorzugung von Ausreißern, was durch den Fokus der Exploration geschieht. Besonders in den

ersten Iterationen eines AL-Algorithmus kann sich dies als nachteilig erweisen, da es mehrere brauchen kann, bis genug Datenpunkte in den unsicheren Regionen gefunden werden. Es existieren verschiedene Methoden zur Quantifizierung der Diversität. Diese basieren auf der Analyse des Feature-Raums, der geometrischen Struktur oder der Gradienten.

2.4.2 Query Kombination

Durch die Komplexität der Daten ist eine einzelne Methode meist nicht in der Lage den Informationsgehalt angemessen zu bestimmen. Deshalb ist die Kombination von verschiedenen Methoden Bestandteil vieler AL-Algorithmen. Gängige Kombinationsstrategien umfassen sequenzielle Auswahl und Linearkombination. Methoden, welche eine Metriken mit reellwertigem Wertebereich erzeugen eignen sich für Linearkombinationen. Hierbei ist auf ähnliche Wertebereiche zu achten, um Methodendominanz zu vermeiden. Clustering-basierte Verfahren erfordern sequenzielle Anwendung.

2.4.3 Betrachtetes Merkmal

Ein wichtiger Punkt für die Bewertung einer AL-Technik sind die Betrachteten Merkmale. Um die Unsicherheit der Klassifikation zu bestimmen, wird häufig die Vorhersage herangezogen. Der Vorteil von Methoden, welche die Vorhersage betrachten ist, dass sie leicht auf neue Modelle und Aufgaben übertragen werden können. Zum Beispiel sind Techniken, die für die Klassifikation von Bildern entwickelt wurden wie das Entropie Sampling auch noch in Punktwolken anwendbar.

Techniken, die auf der Diversität basieren, können ein breiteres Spektrum an Merkmalen betrachten. Durch die Komplexität der Eingabedaten wird für die Diversität häufig der Feature-Raum verwendet. Der Feature-Raum bietet den Vorteil, dass dort abstrakte Repräsentationen vorliegen, bei welchen semantisch ähnliche Objekte nahe beieinander liegen [24]. Dies ermöglicht es die Ähnlichkeit mit einem Ähnlichkeitsmaß wie beispielsweise dem Skalarprodukt zu bestimmen. Neue Datenpunkte können durch Minimierung des Ähnlichkeitsmaßes angefragt werden. Als Alternative zu den Features können auch die Gradienten eines Modells zur Bestimmung der Ähnlichkeit herangezogen werden. Die Voraussetzung von Annotationen zur Bestimmung der Gradienten macht es notwendig, hypothetische Annotationen zu schätzen und diese für die Berechnung der Gradienten zu nutzen. Eine weitere Anwendung für die hypothetischen Gradienten ist, die Norm als Maß für die Unsicherheit zu Nutzen. Die Annahme dahinter ist, wenn das Modell sicher über seine Vorhersage ist, dann ist die Länge des Gradienten klein, wodurch ein Training mit der Vorhersage nur eine geringe Änderung am Modell bewirken würde. Folglich würde der Datenpunkt nur einen geringen Informationsgehalt besitzen. Umgekehrt impliziert ein großer Gradient eine hohe Unsicherheit des Modells, was auf einen potenziell informativen Datenpunkt für das Training hindeutet [25].

2.4.4 Aufgabenstellung

Active Learning hat sich in verschiedenen Bereichen der Computer Vision etabliert, darunter Klassifikation, Objekterkennung und Segmentierung. Die verschiedenen Bereiche besitzen sowohl Gemeinsamkeiten als auch Unterschiede, was sich sowohl auf die eingesetzten Modellarchitekturen als auch auf die AL-Techniken auswirkt.

Die Klassifikation ist eine fundamentale Aufgabe in der Computer Vision. In der Bildverarbeitung ist ihre Aufgabe einem Bild eine Klasse zuzuordnen. Da die Klassifikation eine große Relevanz aufweist, beschäftigen sich auch viele Arbeiten im Bereich des AL mit dieser Aufgabe [17]. Da sowohl statistische als auch Deep Learning Modelle für die Vorhersage der Klasse eine Wahrscheinlichkeitsverteilung erzeugen, nutzen AL-Techniken diese Verteilung, um den Informationsgehalt durch die Unsicherheit in der Vorhersage zu bestimmen. Obwohl diese Vorgehensweise den inhärenten Nachteil besitzt, dass der Fokus auf der Entscheidungsgrenze liegt wird sie durch ihre einfache Umsetzung und gute Nachvollziehbarkeit häufig eingesetzt [21]. Die Semantische Segmentierung wird in der Computer Vision (CV) häufig eingesetzt. Im Gegensatz zur Klassifikation wird hier für jeden Pixel ein Label zugeordnet. Durch die Ähnlichkeit zu der Klassifikation haben sich hierfür ähnliche AL-Techniken etabliert, welche allerdings mit Methoden zur Aggregation oder Bereichsauswahl erweitert werden müssen [26].

Die Objekterkennung setzt sich aus den Teilaufgaben der Klassifizierung und Lokalisierung für jedes gefundene Objekt zusammen. Die AL-Techniken für die Klassifizierung werden häufig über die erkannten Objekte aggregiert und als ein Auswahlkriterium herangezogen [27]. Auch wenn diese Herangehensweise die Regressionsaufgabe der Objektlokalisierung außer Acht lässt, kann dies schon zu einer Einsparung der benötigten Daten führen. Durch die geteilten Parameter in den Hidden Layern besteht ein Zusammenhang zwischen den Aufgaben wodurch sich auch die Lokalisierung verbessern kann, obwohl nur die Unsicherheit in der Klassifikation für das AL betrachtet wurde [28]. Während sich für die AL-Techniken, welche die Klassifizierung nutzen ein gewisser Konsens besteht und Techniken häufig wiederverwendet werden ist das AL für die Regression noch in der Entwicklung. Algorithmen für die Objekterkennung unterscheiden sich häufig in der Art wie die Unsicherheit der Lokalisierung genutzt wird.

2.4.5 Betrachteter Kontext

Der betrachtete Kontext spielt eine entscheidende Rolle bei der Entwicklung von Active Learning Algorithmen für verschiedene Aufgaben des Deep Learning. Für den betrachtete Kontext sieht die Taxonomie zwei Unterklassen vor, einen globalen und einen lokalen Kontext. Die globale Betrachtung zeichnet sich dadurch aus, dass das betrachtete Merkmal in seiner Gesamtheit für das Auswahlkriterium herangezogen wird. Wird beispielsweise der Feature Raum mit einem globalen Kontext betrachtet, wird die Feature Map in ihrer Gesamtheit zur Bestimmung des Informationsgehalts genutzt. AL-Ansätze, welche für die Klassifikation entwickelt wurden, berücksichtigen

typischerweise einen globalen Kontext, was sowohl auf die Unsicherheit in der Bestimmung der Klasse als auch bei der Nutzung der Diversität geschieht.

Demgegenüber zeichnet sich der lokale Kontext dadurch aus, dass der Einsatz von spezifischen lokalen Regionen im Vordergrund steht. Sowohl die Objekterkennung als auch die Semantische Segmentierung charakterisiert sich als lokale Aufgabe, da sowohl für die Unterscheidung von nah beieinander liegenden Objekten in der OD als auch für die Trennung von Klassengrenzen in der Segmentierung die Features genaue räumliche Informationen beinhalten müssen.

Folglich kann sich der Informationsgehalt für diese Aufgaben auf lokale Regionen beschränken. So kann eine globale Betrachtung der Diversität in Szenarien scheitern, in denen der Großteil des Bildes oder der Punktwolke dem Modell bereits bekannt ist, einzelne Ausschnitte sich aber von den Trainingsdaten unterscheiden. Dies kann dazu führen, dass in einer lokalen Region keine ausreichende Trennung im Feature Raum zur Unterscheidung von Objekten vorliegt. Eine solche Region besitzt einen hohen Informationsgehalt. Wird in einem solchen Fall die globaler Feature Map zur Bestimmung der Diversität eingesetzt, kann dies zum Verlust relevanter Details führen weil das Ähnlichkeitsmaß von dem größtenteils bekannten Features dominiert wird [29], [30],

Der betrachtete Kontext wirkt sich auf die Verwendung von Unsicherheitsmaßen aus. Die Aggregation durch Durchschnittsbildung von Unsicherheitsmaßen wird primär von der Mehrheit der Objekte bestimmt. Dies führt dazu, dass einzelne Objekte mit hoher Unsicherheit in der Gesamtbetrachtung untergehen.

2.5 Active Learning Techniken

Nach dem im vorausgegangenen Kapitel die Taxonomie von Active Learning Techniken vorgestellt wurde, erfolgt nun die Betrachtung fundamentaler Methoden, die als Ausgangspunkt für fortgeschrittene Algorithmen dienen. Das Kapitel dient als Grundlage für die anschließende Analyse aktueller Forschungsarbeiten im Bereich AL für das Deep Learning und orientiert sich an etablierten Verfahren, die in der Literatur breite Akzeptanz finden.

Beim **Least Confidence Sampling**, einer Methode des Active Learning, wählt das Modell diejenigen Datenpunkte für das Labeling aus, bei denen es am unsichersten in seiner Vorhersage ist. Das bedeutet, die Datenpunkte mit der geringsten Vorhersagewahrscheinlichkeit werden priorisiert.

$$x_{LC} = \underset{x}{argmin}\, \mathrm{P}(\mathrm{y}^*|\mathrm{x})$$

Formel 1: Least Confidence Sampling

Mit $y^* = \underset{y}{\operatorname{argmax}}\, P(y|x)$ ist die vorhergesagte Klasse für die Eingabedaten x

Die Idee dahinter ist, dass das Modell von diesen unsicheren Fällen am meisten lernen kann. Durch die Annotation und das Training mit diesen Datenpunkten kann die Entscheidungsgrenze des Modells verbessert und die Unsicherheit in diesen Bereichen reduziert werden.

Least Confidence Sampling fokussiert sich somit auf die Datenpunkte nahe der Entscheidungsgrenze, um die Genauigkeit und Robustheit des Modells in diesen kritischen Bereichen zu erhöhen. Dies kann sich aber auch nachteilig auf die Auswahl auswirken. Ein Satz Trainingsdaten der mit dieser Methode erstellt wird, bildet nicht zwingend die wahre Verteilung der Daten ab.

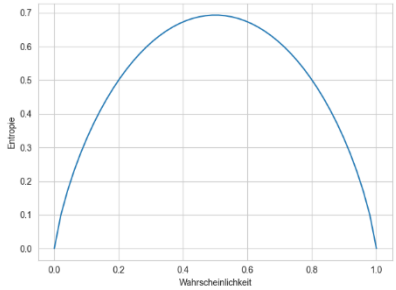


Abbildung 3: Entropie für eine binäre Klassifikation

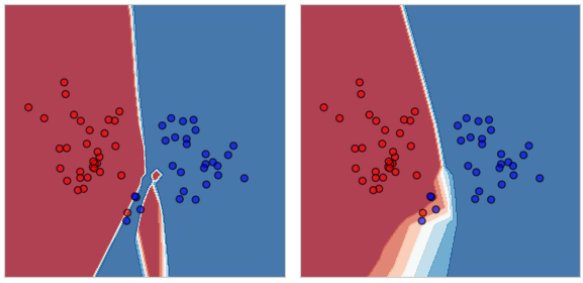


Abbildung 4: Beispiel für die Unsicherheit in einem Ensemble auf den Testdaten

Die **Entropie** ist Maß aus der Informationstheorie für den Informationsgehalt und kann als Unsicherheit interpretiert werden. In Abbildung 3 ist die Entropie für verschiedene Wahrscheinlichkeiten für den Fall einer Binären Klassifikation dargestellt. Es lässt sich deutlich erkennen, dass Sie für den Fall der größten Unsicherheit maximal wird. Um sie für das Active Learning einzusetzen, werden die Daten ausgewählt, für die die Entropie maximal wird.

$$x_{Entropie} = \underset{x}{\operatorname{argmax}} - \sum_i P(y_i^*|x_i)\, log\, P\,(y_i^*|x_i)$$

Formel 2: Entropie Sampling

Die **Kullback-Leibler-Divergenz** misst den Unterschied von zwei Wahrscheinlichkeitsverteilungen p und q. Sie ist nicht direkt zur Datenauswahl geeignet, wird aber in Kombination mit anderen Techniken eingesetzt.

$$D(P||Q) = \sum_x p(x)\, log\, \frac{p(x)}{q(x)}$$

Formel 3: Kullback-Leibler-Divergenz

Query-By-Committee, auch als **Ensemble-Methode** bekannt, ist ein Ansatz im Active Learning, bei dem die Unsicherheit durch die Uneinigkeit mehrerer Modelle geschätzt wird. Bei diesem Verfahren werden alle Modelle auf dem

aktuellen gelabelten Datensatz trainiert und anschließend verwendet, um Vorhersagen auf den ungelabelten Daten durchzuführen. Der Datenpunkt mit der größten Uneinigkeit zwischen den Modellen wird dann zum Labeln ausgewählt. Für die Klassifikation erfolgt die Bestimmung der Uneinigkeit häufig durch Metriken wie die Entropie (Formel 2) oder die Kullback-Leibler-Divergenz (Formel 3).

Jedes Modell im Ensemble stellt eine Hypothese dar, die konsistent mit dem aktuell gelabelten Datensatz ist. Durch das Finden von Bereichen, in denen die Modelle unterschiedliche Vorhersagen treffen, können die möglichen Hypothesen reduziert und die Generalisierbarkeit verbessert werden. Abbildung 4 zeigt ein Beispiel von zwei neuronalen Netzen, die auf die Klassifikation von zwei Klassen trainiert wurden. Die Farbe der Punkte repräsentiert die zugehörige Klasse, während der Farbverlauf des Hintergrunds von Rot über Weiß zu Blau die Entscheidungen der beiden neuronalen Netze visualisiert und die Entscheidungsgrenze deutlich macht.

Durch die gezielte Auswahl von weiteren Trainingsdaten aus Bereichen, in denen die Modelle unterschiedliche Vorhersagen treffen (z.B. im unteren Bereich der Abbildung), kann die Generalisierbarkeit der Modelle gesteigert und ihre Leistungsfähigkeit verbessert werden. Der Nachteil dieser Methode ist, dass mehrere Modelle zur Bestimmung der Unsicherheit trainiert werden, wodurch sich die Trainingszeit vervielfacht. Besonders im Deep Learning kann dies zu einem erheblichen Mehraufwand führen.

Monte-Carlo Dropout ist eine Methode zur Bestimmung der Unsicherheit in Neuronalen Netzen, die auf der Dropout-Regularisierungstechnik basiert. Dropout ist eine Technik, bei der während des Trainings zufällig ausgewählte Gewichte in einem NN temporär deaktiviert werden, um eine Überanpassung des Modells zu vermeiden und die Generalisierungsfähigkeit zu verbessern.

Bei der Anwendung von Monte-Carlo Dropout zur Unsicherheitsbestimmung wird das trainierte Neuronale Netz mehrfach mit aktiviertem Dropout ausgewertet, auch wenn es sich um die Inferenzphase handelt. Durch die zufällige Deaktivierung von Neuronen bei jeder Auswertung entstehen leicht unterschiedliche Vorhersagen für die gleichen Eingabedaten. Diese Variationen in den Vorhersagen spiegeln die Unsicherheit des Modells wider [31]. Eine hohe Varianz deutet darauf hin, dass das Modell für die gegebenen Eingabedaten unsicher ist, während eine niedrige Varianz eine höhere Sicherheit in der Vorhersage widerspiegelt.

Der Vorteil von Monte-Carlo Dropout ist, dass es leicht in bestehende Neuronale Netze integriert werden kann, da es keine Änderungen an der Architektur erfordert. Es ermöglicht eine Abschätzung der Unsicherheit ohne zusätzlichen Rechenaufwand während des Trainings. Allerdings erhöht die mehrfache Auswertung des Modells beim Erstellen der Vorhersagen die Berechnungszeit.

2.6 Active Learning in der 2D Computer Vision

Nachdem wir die grundlegenden Konzepte und Methoden des Active Learning betrachtet haben, wenden wir uns nun der Anwendung dieser Techniken im Bereich der zweidimensionalen Computer Vision zu. Die Erkenntnisse aus diesem verwandten Forschungsgebiet bieten Einblicke und Ansatzpunkte, die sich auf die 3D-Objekterkennung übertragen lassen.

Wang et al. veröffentlichten 2014 das erste Paper, welches Active Learning mit neuronalen Netzen kombinierte. In ihrer Arbeit untersuchten sie die Klassifikation von handgeschriebenen Ziffern auf dem bekannten MNIST Datensatz. Der Ansatz bestand darin, zunächst einen Autoencoder mit 2 hidden Layern auf dem gesamten ungelabelten Datensatz vorzutrainieren und anschließend ein Finetuning des Decoders mit möglichst wenig gelabelten Daten durchzuführen [32]. Es wurden die Strategien Least Confidence Sampling, Margin Sampling und Entropie Sampling mit dem zufälligen Sampling verglichen.

Durch Anwendung dieser Methode konnten Wang et al. zeigen, dass alle Methoden zu einer höheren Genauigkeit im Vergleich zum Referenzwert führen. Darüber hinaus konnte gezeigt werden, dass das Margin Sampling eine höhere Genauigkeit im Vergleich zu den anderen Methoden erzielt. Da der Erfolg von AL-Techniken modellabhängig ist und die eingesetzte Architektur mittlerweile durch CNNs abgelöst wurde ist dieses Ergebnis nur bedingt übertragbar. Es zeigt jedoch auf, dass auch AL auch unter suboptimalen Bedingungen zu Einsparungen der benötigten Datenmenge führen kann.

Kao et al. stellen in ihrer Arbeit zwei neue Metriken vor, um die Unsicherheit der Lokalisierung von Objekterkennungsmodellen für Bilder zu bewerten: Localization Tightness und Localization Stability [33].

Die Localization Tightness soll messen, wie gut die erkannte Box das Objekt umschließt. Im idealen Fall würde man hierfür die Intersection over Union (IoU) zwischen der Erkennung und dem Ground Truth bestimmen. Da der Ground Truth für Bilder ohne Annotationen nicht zur Verfügung steht, wird eine Schätzung der Localization Tightness vorgenommen. In Zwei Stufen Detektoren erstellt das Modell in der ersten Stufe zunächst Empfehlung (Proposals) für BB, welche Objekte enthalten könnten. Hierdurch werden Regionen mit Objekten von ihrem Hintergrund separiert. Die Verfeinerung der Vorschläge erfolgt in einer zweiten Stufe, welche die endgültige Position und Klasse bestimmt. Hier setzt die Localization Tightness an und nutzt das Proposal als Schätzung für die Annotation. Somit ergibt sich die Localization Tightness als $T\left(B_0^j\right) = IoU\left(B_0^j, R_0^j\right)$ wobei B_0^j die erkannte BB darstellt und R_0^j das dazugehörige Proposal. Für die Auswahl der Daten wird noch der zugehörige Erkennungswahrscheinlichkeit P_{max} mit einbezogen und die Anfrage Funktion ergibt sich

$$J\left(B_0^j\right) = \left|T\left(B_0^j\right) + P_{max}\left(B_0^j\right) - 1\right|$$

Formel 4: Localization Tightness

Dadurch werden BB angefragt, bei welchen die Localization Tightness hoch aber die Erkennungswahrscheinlichkeit niedrig ist und umgekehrt, folglich das Modell dem eigenen Proposal nicht vertraut. Da jedes Bild mehrere Objekte enthalten kann wird zur Aggregation das Minimum verwendet.

Die Localization Stability hingegen bewertet, wie robust die Lokalisierung der Bounding Boxes gegenüber Rauschen im Eingabebild ist. Die Idee dieser Methode ist, dass ein Modell dessen Erkennungen stabil bleiben, wenn Rauschen zu den Daten hinzugefügt wird, bereits ein gutes Verständnis für die Daten besitzt.

Hierzu wird künstliches Rauschen zum Bild hinzugefügt und gemessen, wie stark sich die Lokalisierung der Bounding Boxes dadurch verändert. Zunächst werden Vorhersagen für die originalen Daten ohne Rauschen und mit N verschiedenen stärken an Rauschen durchgeführt. Zum Finden der übereinstimmenden BB $C_n\left(B_0^j\right)$ wird für jedes Niveau an Rauschen die BB bestimmt, die die höchste IoU mit der unveränderten BB B_0^j besitzt. Der Durchschnitt dieser IoU Werte ergibt die Unsicherheit einer BB

$$S_B\left(B_0^j\right) = \frac{\sum_{n=1}^{N} \mathrm{IoU}\left(B_0^j, C_n\left(B_0^j\right)\right)}{N}$$

Formel 5: Localization Stability einer Bounding Box

Da Fehlerkennungen mit einer geringen Erkennungswahrscheinlichkeit durch die Veränderung der Eingabedaten schnell zu einer großen Änderung der Lokalisierung führen können, wird der gewichtete Durchschnitt dieser IoU-Werte über alle Bounding Boxes der Bilder I_i gebildet, wobei die Gewichte durch die höchste Klassifizierungswahrscheinlichkeit der jeweiligen Bounding Box bestimmt werden.

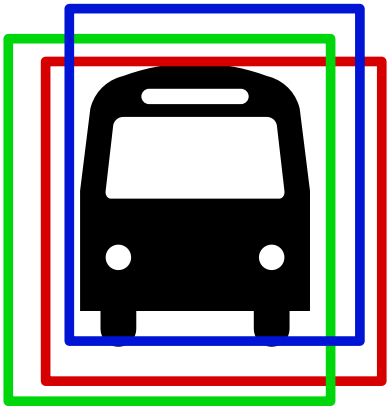
$$S_I(I_i) = \frac{\sum_{j=1}^{M} P_{max}\left(B_0^j\right) S_B\left(B_0^j\right)}{\sum_{j=1}^{M} P_{max}\left(B_0^j\right)}$$

Formel 6: Localization Stability

Durch diese Gewichtung wird der Einfluss von Hintergrunddetektionen reduziert und der Fokus auf die für die Objekterkennung relevanten Bounding Boxes gelegt.

In ihrer Arbeit stellen Yoo et al. die Methode Learning Loss für Active Learning vor, die unabhängig von der Modellarchitektur und der spezifischen Aufgabe anwendbar sein soll [34]. Der Kern ihres Ansatzes besteht darin, den Loss für die Auswahl der Daten zu schätzen, anstatt sich auf herkömmliche Unsicherheitsmaße zu verlassen. Hierzu schlagen sie vor, dass ein bestehendes Modell um einen zusätzlichen Hilfs-Zweig erweitert wird, der

darauf trainiert wird, den Loss für eine gegebene Eingabe vorherzusagen. Durch diese Erweiterung kann das Modell selbst beurteilen, welche Datenpunkte den höchsten Informationsgewinn versprechen und somit für das Active Learning ausgewählt werden sollten.



	Rot	Grün	Blau
Rot		0.68	0.66
Grün	0.68		0.57
Blau	0.66	0.57	

Abbildung 5: Illustration des Consensus Score

Der Consensus Score von Schmidt et.al (2020) versucht die Unsicherheit zu schätzen, indem die Uneinigkeit der Erkennungen in einem Ensemble bestimmt wird [35]. Der Consensus Score basiert auf der Localization Stability ersetzt aber die Erkennung unter verschiedenen Niveaus von Rauschen durch ein Ensemble. Die Modelle im Ensemble sind homogen in ihrer Architektur und in ihrem Trainingsablauf. Die Modelle unterscheiden sich nur durch die unterschiedliche Initialisierung. Folglich bestimmt der Score die Uneinigkeit von verschiedenen Versionen desselben Modells in der Lokalisierung.

Um den Consensus Score zu bestimmen, wird zunächst die IoU Matrix $\Omega^{ij} \in R^{N \times N}, i, j \in [1, M]$ erstellt, wobei M die Anzahl der Modelle und N die Anzahl der Erkennungen pro Modell ist. Mit der IoU Matrix werden die Regions of Interest (RoI) gefunden. Diese stellen Regionen dar, welche in mehrere Modelle im Ensemble ein Objekt erkannt haben. Die RoIs werden bestimmt als das Maximum jeder Zeile von Ω^{ij} was als $\max(\omega_n^{ij}), n \in [1, N]$ bezeichnet wird. Das Nutzen der maximalen IoU verhindert, dass nahe Objekte mit geringfügig überlappenden BB fälschlicherweise als Unstimmigkeit im Ensemble erkannt werden. Der Consensus Score wird definiert als:

$$\text{Consensus Score} = 1 - \frac{1}{N} \sum_{n=1}^{N} \min_{i,j \in [1,M]} \{\max(\omega_n^{ij})\}$$

Formel 7: Consensus Score

Durch die Verwendung des minimalen Wertes der RoI genügt bereits eine abweichende Lokalisierung im Ensemble, um den Consensus Score zu erhöhen. In Abbildung 5 ist die Bestimmung des Consensus Score mit einer RoI dargestellt. Anhand der IoU Werte rechts vom Bild kann der Consensus Score als 0.57 bestimmt werden.

Neben dem Consensus Score haben Schmidt et al. Das RoI-Matching als weiteren Ansatz entwickelt. Ähnlich zum Consensus Score basiert diese Methode auf den Erkennungen eines Ensembles. Hierbei werden die Regionen, in welchen mehrere Modelle ein Objekt erkannt haben als Region of Interrest bezeichnet und die Klassenverteilung innerhalb der RoI wird betrachtet.

Der Algorithmus Box Level Active Detection (BLAD) von Lyu et al. kombiniert den AL-Prozess mit Self Supervised Learning. Das Kriterium für die Anfrage der Bilder basiert auf der Evaluierung der Konsistenz zwischen dem Originalbild und dessen augmentierter Version. Hierbei wird die Unsicherheit anhand der durchschnittlichen Varianz der Bounding-Box-Regressionsparameter sowie der Kreuzentropie der Klassenverteilung quantifiziert. Komplementär dazu integriert die Strategie Elemente des Self-Supervised-Learning, wodurch eine Erweiterung der Annotationen unter Einbeziehung des Pools ungelabelter Daten ermöglicht wird. In diesem Kontext erfolgt die Generierung von Pseudo-Labels für bereits hinreichend gelernte Objekte, was zu einer effizienteren Nutzung des vorhandenen Datenmaterials führt.

Yu et al. stellen in ihrem Paper eine neue Methode für Active Learning vor, welche speziell auf die Herausforderungen der 2D Objekterkennung zugeschnitten sein soll [36]. Der Kern Ihrer Methode besteht auf der Bestimmung der Unsicherheit, welche durch die Konsistenz der Erkennung zwischen dem Originalbild und einer augmentierten Version bestimmt wird.

Die Autoren argumentieren, dass eine einzelne Metrik wie bei Learning Loss nicht ausreicht, um die Unsicherheit für Klassifikations- und Regressionsaufgaben in diesem Kontext adäquat zu bestimmen. Aus diesem Grund wird die Unsicherheit für beide Aufgaben bestimmt. Um die Unsicherheit in der Lokalisierung der BB zu bestimmen, wird die IoU herangezogen. Zur Bestimmung der Änderung in der Klassenverteilung verwenden die Autoren die Jensen-Shannon-Divergenz, eine auf der Kullback-Leibler-Divergenz basierende, symmetrische Metrik mit begrenztem Wertebereich von 0 bis 1. Da die Jensen-Shannon-Divergenz denselben Wertebereich besitzt wie die IoU, können diese leicht miteinander kombiniert werden, da die Metrik von keinem der beiden Werte dominiert wird.

Das Ziel dieser Methode ist es, die lokale Region mit der größten Unsicherheit zu bestimmen. Um dies zu erreichen, wird zur Aggregation der Werte eines Bildes das Minimum verwendet. Ein Mittelwert über das gesamte Bild kann eine niedrige Unsicherheit erzeugen, obwohl sich dort informative Regionen befinden.

Hierdurch stellt der niedrigste Wert der Metrik nicht zwangsläufig das informativste Bild dar. Bereits eine inkonsistente Erkennung in einem Bild führt zu einer hohen Unsicherheit, selbst wenn andere Erkennungen eine hohe Konsistenz aufweisen. Daher sollte die Metrik der ausgewählten Bilder einen gewissen Abstand zur Untergrenze aufweisen, um eine ausgewogene Auswahl zu gewährleisten.

In der Regel sind in Bildern mehrere Objekte enthalten. Die Auswahl der Bilder für das Training kann hierbei Einfluss auf die Klassenverteilung im Datensatz führen. Ist bspw. ein Objekttyp überproportional häufig in den Trainingsbildern enthalten, kann dies zur Verzerrung der Klassenverteilung im Datensatz führen, was sich nachteilig auf die Erkennung der unterrepräsentierten Klasse auswirken kann. Um eine ausgewogene Auswahl der Bilder zu gewährleisten, schlagen die Autoren einen weiteren Schritt vor. Nach der Bestimmung der Unsicherheit für jedes Bild, werden die gewählten Daten gefiltert. Hierbei wird die Jenson-Shannon-Divergenz zwischen der Verteilung Klassenverteilung der bereits gelabelten Daten und der Verteilung der einzelnen Bilder berechnet. Ziel ist es, Bilder auszuwählen, deren Klassenverteilung sich möglichst stark von der Verteilung der bereits gelabelten Daten unterscheidet. Durch dieses Vorgehen soll eine Verzerrung der Klassenverteilung im Datensatz vermieden werden.

Ein rein diversitätsbasierter Ansatz für die Klassifizierung von Bildern wurde von Sener et al. vorgestellt. Dieser Ansatz zielt darauf ab, ein Core-Set des Datensatzes zu identifizieren. Dieses Core-Set stellt eine möglichst repräsentative Teilmenge dar, deren Training einen ähnlichen Loss wie das Training auf dem gesamten Datensatz erzielt [37].

Für den AL-Algorithmus werden die gelabelten Daten durch das Core-Set repräsentiert. In jeder Iteration werden die Daten für das labeln ausgewählt, die die dem Core-Set hinzugefügt werden. In diesem Rahmen ist die exakte Bestimmung des Core-Sets nicht möglich. Die Daten, die zur Auswahl für das Core-Set stehen besitzen noch keine Label und somit kann der Loss nicht berechnet werden, weshalb eine Approximation zum Einsatz kommt. Hierfür werden die Daten aus dem Pool der ungelabelten Daten ausgewählt, welche den bereits gelabelten Daten am unähnlichsten sind.

Für die Ähnlichkeitsbestimmung wird der Feature-Vektor des finalen Layers betrachtet. Eine direkte Nutzung der Eingabedaten ist aufgrund ihrer Komplexität und hohen Dimensionalität nicht praktikabel. Der Feature-Raum bietet hingegen den entscheidenden Vorteil, dass dort abstrakte Repräsentationen vorliegen, bei denen semantisch ähnliche Objekte in räumlicher Nähe zueinander positioniert sind.

Zur Erstellung des Core-Sets wird ein Greedy Farthest First Clustering Algorithmus verwendet. Hierbei werden die Datenpunkte ausgewählt, welche die größte Distanz zu den bereits gelabelten Daten aufweisen. Als Distanzmaß findet die l2-Norm Anwendung.

Das Ziel beim Clustering ist die Auswahl von Datenpunkten als Clustercentren, welche die größte Distanz zwischen den verbleibenden Datenpunkten und dem nächsten Clusterzentrum verringern. Diese Distanz wird als Abdeckradius bezeichnet. Eine Visualisierung dieses Konzepts findet sich in Abbildung 6 die die Reduktion des Abdeckradius durch Hinzunahme eines weiteren Datenpunkts illustriert.

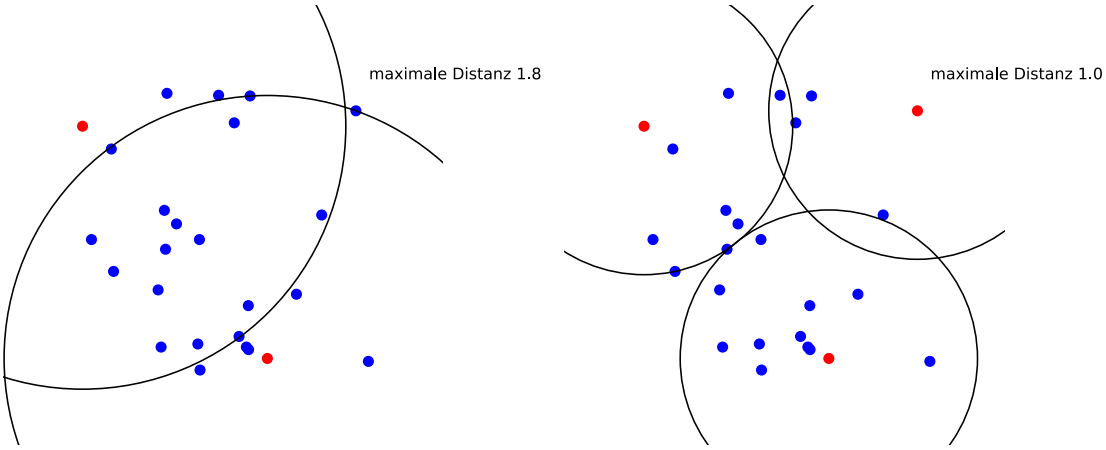


Abbildung 6: Core-Set Clustering

Die Idee hinter diesem Ansatz ist, dass die nicht annotierten Datenpunkte von dem nächsten Clusterzentrum respektive dem annotierten Datenpunkt repräsentiert werden. Demnach führt eine Verringerung des Abdeckradius zu einer besseren Abbildung der Grundgesamtheit durch den Trainingsdatensatz. Dies resultiert aus der Repräsentation einer geringeren Anzahl von Datenpunkten pro Clusterzentrum und deren erhöhter Ähnlichkeit zum Zentrum. Das Greedy Farthest First Clustering gewährleistet eine ausreichende Differenzierung der angefragten Datenpunkte von den bereits annotierten Daten. Diese Differenzierung ist essenziell, da Datenpunkte mit hoher Ähnlichkeit nur einen geringen Informationsgewinn für das Training bieten. Die Methode zielt somit darauf ab, die Diversität des Trainingsdatensatzes zu maximieren, wobei gleichzeitig redundante Informationen minimiert werden.

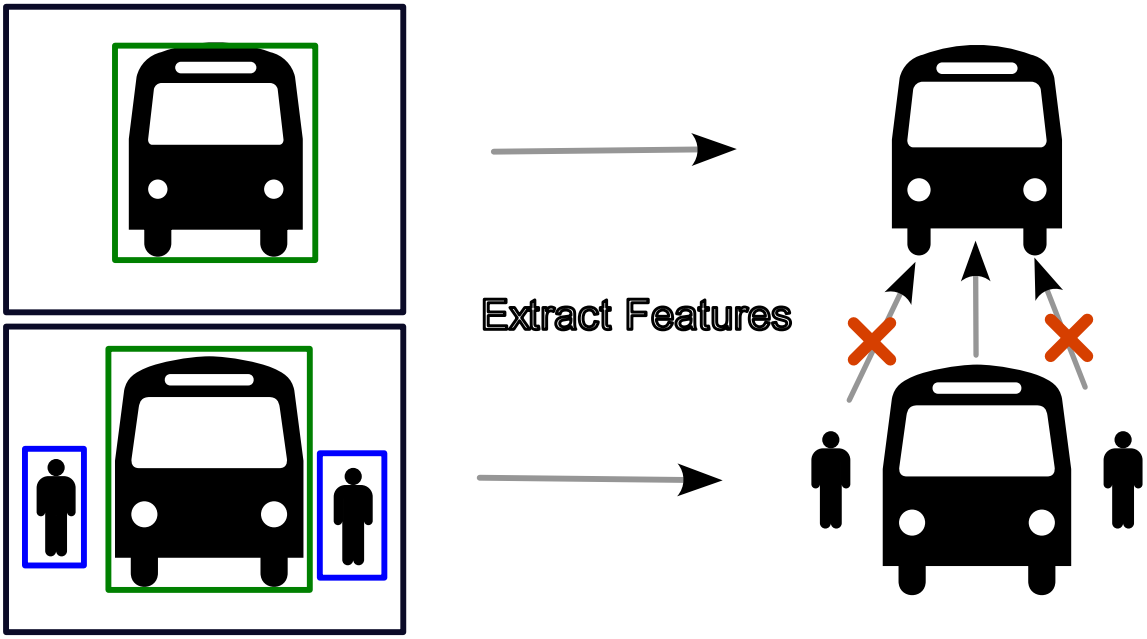


Abbildung 7: Illustration der Bestimmung der Ähnlichkeit von PPAL

Der Plug and Play Active Learning (PPAL) Algorithmus wurde für die Objekterkennung in Bildern entwickelt und kombiniert eine gewichtete Entropie-basierte Unsicherheitsmetrik mit einem Diversitätsmaß im Feature Space. Durch eine lokale Betrachtung der Objekt Features soll das Diversitätsmaß besser für die Objekterkennung geeignet sein [30]. Die Kombination der beiden Anfrage Methoden erfolgt sequenziell. Die erste Stufe ist ein gewichtetes Entropie Sampling.

Zum Gewichten der Entropie wird ein Maß vorgestellt, welches die Schwierigkeit der Klassen bestimmt. Hierzu wird ein Koeffizient berechnet, welcher sowohl die Unsicherheit der Klassifizierung wie auch der Lokalisierung berücksichtigt. Dies erhöht die Auswahlwahrscheinlichkeit für schlecht erkannte Klassen. Der Schwierigkeitskoeffizient wird definiert als:

$$w = 1 - P\left(b\middle|\hat{b}\right)^{\xi} \cdot IoU\left(b, \hat{b}\right)^{1-\xi}$$

Formel 8: Schwierigkeitskoeffizient von PPAL

Der Koeffizient wird auf den Testdaten berechnet, wobei b für das erkannte Objekt steht, $\hat{b}$ für die Annotation und $P\left(b\middle|\hat{b}\right)$ für die Wahrscheinlichkeit der korrekten Klasse. Um den Fokus des Koeffizienten zwischen der Klassifikation und der Lokalisierung abzuwägen ist ξ ein Hyperparameter im Bereich $[0,1]$. Der Koeffizient wird auf den Testdaten berechnet und während des Trainings durch einen gleitenden Mittelwert angepasst. Die Bildauswahl für die erste Stufe erfolgt anhand der summierten Klassenentropie für jedes Objekt, gewichtet mit dem Schwierigkeitskoeffizient.

In der zweiten Stufe wird die Diversität im Feature-Space betrachtet. Es erfolgt ein Clustering der Objektfeatures basierend auf einer Variation des Core-Set Algorithmus. Im Gegensatz zu Core-Set wird die Diversität nur für die Bilder

sichergestellt, welche von der ersten Stufe ausgewählt wurden, die Daten aus dem gelabelten Pool werden nicht zur Bestimmung der Diversität berücksichtigt. Dies verringert den Rechenaufwand zur Erstellung der Distanzmatrix, welche für das Greedy Farthest First Clustering notwendig ist.

Des Weiteren wird nicht die gesamte Feature Map zur Bestimmung der Ähnlichkeit verwendet. Es werden nur die Teile der Feature Map genutzt, welche sich innerhalb der BB der detektierten Objekte befinden. Die Autoren argumentieren, dass das Nutzen der gesamten Feature Map aus einer tiefen Schicht die globale Ähnlichkeit bestimmt. Diese globale Ähnlichkeit soll für die Aufgabe der Objekterkennung nicht geeignet sein, da detaillierte Features mit exakten Räumlichen Informationen durch den steigenden Abstraktionsgrad und die geringere räumliche Auflösung in den tieferen Schichten verloren gehen. Um die Ähnlichkeit zwischen zwei Bildern zu bestimmen, werden die enthaltenen Objekte miteinander verglichen. Das Vorgehen ist es, zunächst die Objektähnlichkeit zu bestimmen. Hierfür wird die extrahierte Feature Map für jedes Objekt in Bild a mit allen Objekten in Bild b mit der Kosinus Ähnlichkeit verglichen. Der größte Wert gibt die Objektähnlichkeit an. Dabei werden nur Objekte derselben Klasse miteinander verglichen, wie in Abbildung 7 dargestellt. Die Ähnlichkeit der Bilder ergibt sich als Durchschnitt der Objektähnlichkeiten.

2.7 Active Learning für Punktwolken

Im Folgenden werden wissenschaftliche Arbeiten zu Active Learning für Punktwolken vorgestellt. Diese ermöglichen es Herausforderungen zu ermitteln, die bei der Übertragung von AL-Algorithmen von der 2D in die 3D Objekterkennung auftreten können und dienen als Basis zur Identifizierung von potenziellen Lösungen.

Der Ansatz von Feng et al. untersuchte die Effizienzsteigerung multimodaler 3D Objekterkennungsmodelle durch Active Learning-Techniken [28]. Der vorgestellte Ansatz nutzt ein 2D-Objekterkennungsmodell zur Generierung von Region Proposals aus 2D-Bildern. Diese dienen der Bestimmung der 3D-Punkten innerhalb des Frustums, welches aus dem Region Proposal erstellt wird. Ein Frustum (engl. Pyramidenstumpf) wird in der Computergrafik eingesetzt, um den sichtbaren Bereich zu definieren. Die Frustum-Punkte werden zur Erzeugung von RGBD-Bildern (RGB-Bilder mit Tiefeninformationen in einem zusätzlichen Kanal) verwendet. Diese fungieren als Input für den Modell-Head, basierend auf einem vierschichtigen Convolutional Neural Network zur Objektklassifikation und -lokalisierung.

Diese Methodik zielt auf eine Reduktion der erforderlichen Punktmenge ab, da der Head nicht mehr zwischen Objekt und Hintergrund differenzieren muss. Zudem verringert sich der Annotationsaufwand, da nur die Punkte innerhalb des Frustums annotiert werden.

Um die Wahrscheinlichkeiten der Objektklassen zu bestimmen, verwenden sie zwei Ansätze: Monte-Carlo-Dropout und Ensemble-Methoden. Zur Bestimmung der Unsicherheit in der Objekterkennung untersuchen Feng et al.

zwei Maße: Mutual Information und Entropie. Die Ergebnisse ihrer Experimente zeigen, dass die auf Mutual Information basierende Unsicherheit konsistent bessere Ergebnisse bei der Lokalisierung von Objekten liefert. Im Gegensatz dazu erweisen sich entropiebasierte Unsicherheitsmaße als vorteilhafter für die Klassifizierung der Objekte.

Basierend auf dem Ansatz von Feng et al. wurde von Moses et al. die Localization-based Active Learning (LOCAL) Methode entwickelt, welche zusätzlich noch die Unsicherheit in der Lokalisierung berücksichtigt [38]. Hierfür werden die aus den stochastischen Erkennungen detektierten Objekte mittels der IoU einander zugeordnet und die Varianz der Bounding Box Parameter mit der Unsicherheit der Klassifizierung kombiniert

$$\mathrm{LOCAL} = \frac{\text{Total Variance for } x_i}{\max\limits_{x_j \in X} \text{Total Variance for } x_j} + \frac{\text{Classification Uncertainity for } x_i}{\max\limits_{x_j \in X} \text{Classification Uncertainity for } x_j}$$

Formel 9: Unsicherheit von LOCAL

Für die Unsicherheit der Klassifizierung wurde sowohl die Mutual Information als auch die Entropie getestet, wobei beide ähnlich gute Resultate erzielt haben.

Wu et al. stellen den Region-based and Diversity-aware Active Learning Algorithmus (ReDAL) für die semantische Segmentierung von Punktwolken vor. Die Methode basiert auf der Aufteilung der Punktwolke in kleine Regionen, für welche der Informationsgehalt separat bestimmt wird. Anschließend werden nur die ausgewählten Regionen annotiert und für das Training verwendet [39].

Der Kern des Verfahrens umfasst zwei Hauptkomponenten: die Schätzung der Regionsinformation und die Berücksichtigung der Diversität. Zur Schätzung der Regionsinformation werden drei Metriken herangezogen: Entropie, Farbdiskontinuität und strukturelle Komplexität. Während die Bestimmung der Entropie aus der Modellvorhersage erfolgt, wird sowohl die Farbdiskontinuität als auch die strukturelle Komplexität direkt aus den Eingabedaten berechnet.

Die Farbdiskontinuität wird als durchschnittliche l^1-Norm der Farbabweichung zwischen den k nächsten Nachbarn berechnet. Hierdurch werden Regionen bevorzugt, in denen es zu häufigen Farbwechseln kommt. Ein Farbwechsel geht häufig mit der Änderung der Klasse einher. Dies soll dazu führen, dass mehr Regionen mit Klassengrenzen ausgewählt werden.

Die strukturelle Komplexität ergibt sich aus dem Verhältnis der Eigenwerte der Kovarianzmatrix der k nächsten Nachbarn, wobei der kleinste durch den größten Eigenwert geteilt wird. Da der kleinste Eigenwert der Kovarianzmatrix für flache Oberflächen null ist, ist der Wert der strukturellen Komplexität für flache Oberflächen ebenfalls null. Folglich kann mit der strukturellen Komplexität festgestellt werden, ob eine Region zu einer flachen Ebene oder einer Kante gehört. Vergleichbar zu der Farbdiskontinuität sollen hierdurch Klassengrenzen gefunden werden. Außerdem soll es die Auswahl von großen

Objekten reduzieren. Große Objekte bestehen zumeist aus ähnlichen zusammenhängenden Flächen und besitzen durch die Redundanz einen geringeren Informationsgehalt.

Zur Sicherstellung der Diversität wird der Core-Set-Algorithmus im Feature-Raum angewendet. Hierbei erfolgt ein Average Pooling der vorletzten Feature Map in den Regionen. Im Gegensatz zum Core-Set Algorithmus werden die Cluster nicht unmittelbar für die Datenauswahl eingesetzt. Stattdessen wird die Zugehörigkeit zu einem Core-Set Cluster als Kriterium für die Gewichtung der Information Estimation Values verwendet. Falls mehrere Regionen zum selben Core-Set Cluster gehören, werden alle Regionen außer der mit dem höchsten Information Estimation Value mit einem Konstanten Faktor heruntergewichtet. Anschließend können die Regionen durch den Information Estimation Value für die Annotation ausgewählt werden.

Das Paper von Liang et al. aus dem Jahr 2022 befasst sich mit der Nutzung multimodaler Informationen in autonomen Fahrzeugen [40]. Die Autoren schlagen Maße vor, um die Vielfältigkeit der Daten zu bestimmen und so eine effiziente Auswahl von repräsentativen Datenpunkten zu ermöglichen. Dieser Ansatz zielt darauf ab, den Annotationsaufwand zu minimieren und gleichzeitig eine hohe Abdeckung der verschiedenen Szenarien zu gewährleisten.

Es werden zwei Maße vorgestellt, um die Vielfältigkeit der Daten zu quantifizieren: die räumlichen und die temporale Vielfältigkeit. Für die räumlichen Vielfältigkeit nutzen die Autoren die GPS-Positionen, die in autonomen Fahrzeugen vorliegen. Sie erstellen eine Adjazenz Matrix, die die quadrierten euklidischen Distanzen zwischen den k nächsten Nachbarn enthält. Der Wert der räumlichen Vielfältigkeit entspricht der kürzesten Distanz auf dem resultierenden k-Nächsten-Nachbarn-Graphen. Die temporale Vielfältigkeit wird anhand der Aufnahmezeiten bestimmt. Wenn zwei Datenpunkte an derselben Position aufgenommen wurden, entspricht die temporale Vielfältigkeit der absoluten Differenz ihrer Zeitstempel. Andernfalls wird die temporale Vielfältigkeit auf unendlich gesetzt. Zusätzlich wird noch die Feature-Vielfältigkeit des CoreSet-Algorithmus verwendet, der die Diversität anhand der Merkmale der Datenpunkte bewertet.

Zur Auswahl einer repräsentativen Teilmenge der Daten schlagen die Autoren eine Modifikation des CoreSet-Algorithmus vor. Anstatt nur die Feature-Distanz zu berücksichtigen, verwenden sie eine Linearkombination der räumlichen, temporalen und Feature-Distanzen, wobei sowohl die räumliche als auch die temporale Distanz zuvor normalisiert wurden.

Liang et al. testen verschiedene Kombinationen der vorgeschlagenen Vielfältigkeitsmaße, um deren Einfluss auf die Qualität der ausgewählten Daten zu untersuchen. Die gleiche Gewichtung der räumlichen, temporalen und Feature Vielfältigkeit erzielt dabei die größte Zunahme was zeigt, dass die Diversität durch ein einzelnes Maß schwer zu bestimmen ist.

Darüber hinaus entwickeln sie eine Kostenfunktion, die den Annotationsaufwand für das Labeln der ausgewählten Daten berücksichtigt.

Dies soll eine bessere Vergleichbarkeit der Methoden ermöglichen, als nur die Anzahl der gelabelten Bounding Boxes oder Frames zu berücksichtigen.

Luo et al. präsentieren in ihrer Arbeit einen Ansatz, der drei wesentliche Kriterien für die Auswahl der Datenpunkte berücksichtigt: Label Conciseness, Feature Representativeness und Geometric Balance [41]. Der vorgeschlagene Auswahlprozess besteht aus drei aufeinanderfolgenden Stufen, um den Rechenaufwand zu reduzieren. In der ersten Stufe, die sich auf die Label Conciseness konzentriert, wird die Entropie der Label-Verteilung berücksichtigt. Ziel ist es, die Unausgewogenheit der Klassen zu verringern, indem die Menge der Datenpunkte gefunden wird, welche die Kullback-Leibler-Divergenz zwischen der Klassenverteilung und einer Gleichverteilung minimiert. Durch die Auswahl von Datenpunkten mit einer ausgewogenen Label-Verteilung kann das Modell eine bessere Generalisierungsfähigkeit entwickeln.

Die zweite Stufe befasst sich mit der Feature Representativeness. Hierbei wird angestrebt, die Menge der Datenpunkte zu finden, die die repräsentativsten Features enthalten. Dies geschieht anhand der Gradient-Vektoren. Um die Gradienten für die ungelabelten Daten zu ermitteln, werden hypothetische Labels auf Basis der Modellvorhersagen zugewiesen. Durch die Verwendung von Markov-Chain-Dropout in der Vorhersage und die Bildung des Durchschnitts der Erkannten Boxen werden hypothetische Bounding-Box-Ziele für die Regression ermittelt. Die hypothetischen Ziele werden genutzt, um die Gradienten zu bestimmen, welche anschließend wie bei dem Core-Set Algorithmus geclustert werden. Durch die Auswahl von Datenpunkten mit repräsentativen Features kann das Modell die wesentlichen Merkmale der Daten besser erfassen.

In der dritten Stufe liegt der Fokus auf der Geometric Balance. Die Idee dahinter ist, dass die Punktdichte der ausgewählten Samples repräsentativ für den gesamten Datensatz sein sollte. Um dies zu erreichen, wird die Kullback-Leibler-Divergenz zwischen der Punktdichte innerhalb der Bounding-Boxes der ausgewählten Samples und der Punktdichte des gesamten Datensatzes minimiert. Dabei wird eine Gleichverteilung für die Punktdichte angenommen, um sowohl Regionen mit hoher als auch niedriger Punktdichte zu berücksichtigen. Durch die Auswahl von Datenpunkten mit einer ausgewogenen geometrischen Verteilung soll das Modell die räumliche Struktur der Daten besser erfassen.

Ein Ansatz der Active Learning, überwachtes und selbstüberwachtes Lernen für die Objekterkennung in 3D kombiniert wurde von Hwang et al. (2023) vorgestellt [42]. Um Daten für das Labeling auszuwählen, vergleicht das Verfahren die Vorhersagen des Modells für die Originaldaten mit den Vorhersagen für transformierte Versionen derselben Daten, um die Unsicherheit des Modells zu bestimmen. Zusätzlich werden auf dieselbe Art Verlustfunktionen für das selbstüberwachte Lernen erzeugt.

Die Konsistenz zwischen zu den transformierten Daten berücksichtigt sowohl die Klassifikation als auch die Lokalisierung. Für die Klassifikation wird die

Kullback-Leibler-Divergenz zwischen den Verteilungen genutzt, während für die Lokalisierung die Summe des Smooth L_1 Loss der absoluten Differenzen der Werte (bzw. des absoluten Sinus der Differenz für die Rotation) verwendet wird. Diese Berechnung wird sowohl für den Auswahlprozess im Active Learning als auch für das selbstüberwachte Lernen verwendet.

Außerdem wird die Verlustfunktion des selbstüberwachten Lernens auch für bereits gelabelte Daten angewendet, indem diese mit der Verlustfunktion des überwachten Lernens kombiniert wird. Ein wesentlicher Vorteil des vorgestellten Ansatzes besteht darin, dass zur Auswahl der Trainingsdaten die Werte der Verlustfunktion für das unüberwachte Lernen herangezogen werden. Dies führt zu einem effizienten Active-Learning-Prozess ohne zusätzlichen Overhead. Darüber hinaus nutzt das Verfahren den Pool der ungelabelten Daten für das selbstüberwachte Lernen, was zu einer verbesserten Leistungsfähigkeit des Modells beiträgt. In Ihren Experimenten auf einem Datensatz zum autonomen Fahren zeigen die Autoren, dass das Verfahren auch ohne das selbstüberwachte Lernen die Baseline mit zufälliger Datenauswahl schlägt und über eine ähnliche Leistungsfähigkeit verfügt, wie Entropie basierte Methoden.

2.8 Metriken zur Evaluation von Active Learning Algorithmen

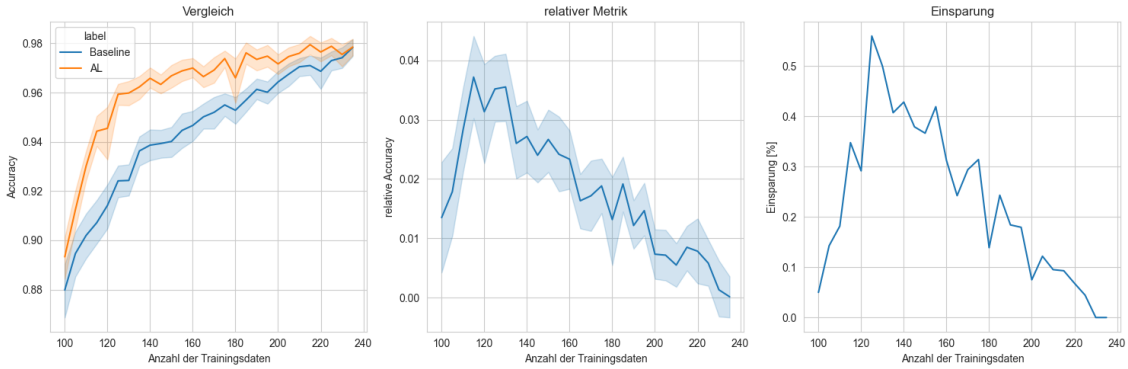


Abbildung 8: Beispiel einer Evaluation

Die Evaluierung von Active Learning Algorithmen ist ein wichtiger Aspekt in der Forschung und Entwicklung dieser Methoden. Allerdings gibt es derzeit keinen Konsens darüber, wie eine solche Evaluierung am besten durchgeführt werden sollte [43]. In den meisten Fällen wird ein Vergleich mit einem Training angestellt, bei dem die Daten zufällig ausgewählt wurden. Hierbei wird oft die relative Einsparung der benötigten Daten als Maß für die Effektivität des Active Learning Ansatzes herangezogen. Abbildung 8 zeigt ein Beispiel, von der Evaluierung eines AL-Algorithmus anhand der Genauigkeit. Der erste Graph zeigt, wie sich die Genauigkeit mit einer zunehmenden Datenmenge im Vergleich zum Basiswert verhält, während der zweite Graph die Differenz zur besseren Übersicht darstellt. Rechts ist die prozentuale Einsparung an annotierten Daten dargestellt.

Darüber hinaus werden häufig Standardverfahren zur Evaluierung von Machine Learning Modellen, wie beispielsweise die Kreuzvalidierung,

eingesetzt. Um die Generalisierbarkeit der Ergebnisse zu zeigen, ist es wichtig, die Algorithmen auf verschiedenen Datensätzen und mit unterschiedlichen Modellen zu testen. Dies bringt jedoch den Nachteil eines höheren Aufwands und gesteigerten Bedarfs an Rechenkapazität mit sich.

In komplexeren Anwendungsgebieten, wie der Objekterkennung, ist es zudem sinnvoll, eine Kostenfunktion zu definieren. Diese berücksichtigt, dass das Labeln von mehreren Objekten in einem Bild mehr Zeit in Anspruch nimmt als das Labeln von Bildern mit nur wenigen Objekten. Auch für Bilder ohne Objekte muss eine gewisse Zeit für die Annotation einkalkuliert werden [40].

Insgesamt bleibt die Evaluierung von Active Learning Algorithmen eine Herausforderung, die weitere Forschung und die Entwicklung standardisierter Methoden erfordert, um eine bessere Vergleichbarkeit und Reproduzierbarkeit der Ergebnisse zu gewährleisten.

3 Methode

Im vorherigen Kapitel wurde der aktuelle Stand der Technik für das Active Learning in Punktwolken und für die 2-dimensionale Computer Vision vorgestellt. Auf dieser Basis können AL-Algorithmen und die enthaltenen Techniken der 2D Computer Vision auf ihre Einsetzbarkeit in Punktwolken untersucht werden. Es ist hierbei das Ziel, das Spektrum der zur Verfügung stehenden Algorithmen zu erweitern und deren Leistungsfähigkeit im dreidimensionalen Raum zu evaluieren. Die Relevanz dieser Untersuchung ergibt sich daraus, dass die Effektivität von AL von der Wahl des eingesetzten Modells sowie der Charakteristik des zugrundeliegenden Datensatzes abhängt. Infolgedessen führt eine Erweiterung des methodischen Repertoires zu einer verbesserten Adaptivität von AL auf neue Datensätze und Modelle. Zu diesem Zweck werden die Problematiken analysiert, welche beim Übertragen von AL-Algorithmen auf die 3D Objekterkennung auftreten können. Darauf folgt eine Typisierung der Lösungsansätze von AL-Algorithmen, welche für den Einsatz in dreidimensionalen Daten entwickelt wurden und die Adaption eines Lösungsansatzes von der Semantischen Segmentierung auf die Objekterkennung. Abschließend werden die Herausforderungen dargestellt, welche bei der Bewertung von AL-Algorithmen auftreten.

3.1 Problematik bei der Anwendung von Diversitätsmaßen auf Punktwolken

Bevor ein AL-Algorithmus aus der 2D Computer Vision für den Einsatz in 3D Punktwolken genutzt werden kann, muss zunächst geprüft werden, ob schwerwiegende Gründe gegen ihren Einsatz sprechen. Als problematisch erweist sich die Anwendung von Methoden, die die Diversität der Features betrachten. In zweidimensionalen Bildern stehe dafür zwei Optionen zur Auswahl. Die einzelnen Features einer Feature Map können zu einem einzelnen Vektor konkateniert werden, welcher zur Ähnlichkeitsbestimmung herangezogen werden kann. Alternativ können Feature Maps miteinander verglichen werden, indem jedes Feature einer Feature Map mit dem korrespondierenden Feature der anderen Feature Map zur Ähnlichkeitsbestimmung herangezogen wird und die Paarweisen Ähnlichkeiten anschließend aggregiert werden. Ermöglicht wird dies durch die dichte Besetzung der Feature Map. Diese bestehen immer aus der gleichen Anzahl an Features, wodurch immer ein Korrespondierendes Feature zur Verfügung steht.

Dies ist für 3D Daten nicht ohne weitere Anpassungen möglich. Für diese ist es Charakteristisch dünnbesetzt im Raum vorzuliegen. Das bedeutet, dass ein Großteil des Raums leer ist und folglich nur an wenigen Positionen Datenpunkte vorliegen. Folglich kann eine paarweise Ähnlichkeit nicht bestimmt werden, da das Auftreten von Features an der Exakt selben Position unwahrscheinlich ist. Des Weiteren kann sich die Anzahl an Punkten in den Eingabedaten Unterscheiden, wodurch das konkateniert und anschließende Vergleichen der einzelnen Features nicht möglich ist. Außerdem kam es durch

die dünne Besetzung der Daten zu einer Adaption der Modellarchitekturen an diese Eigenschaft. Durch den fortgeschrittenen Entwicklungsstand der 2D Computervision wurden einige der dort eingesetzten Konzepte in 3D Modellarchitekturen übernommen. Dies führt dazu, dass sowohl Gemeinsamkeiten als auch Unterschiede zwischen den Modellarchitekturen der 2D und 3D Computer Vision bestehen. Die Architekturen für 3D Daten werden hierbei in die zwei Kategorien punktbasiert und voxelbasiert eingeteilt [44]. Punktbasierte Modelle verarbeiten die Punktkoordinaten mit den zugehörigen Merkmalen direkt, ohne dass eine Änderung des Datenformats oder der Datenstruktur erfolgt. Hierzu wird eine Teilmenge der Punkte ausgewählt und in lokale Regionen gruppiert. Ein neuronales Netz verarbeitet die Gruppen und extrahiert Features, die die Merkmale der vordefinierten Gruppen beschreiben. Durch das mehrfache anwenden dieser Operationen auf den erzeugten Features der vorangegangenen Schicht und einer Vergrößerung der lokalen Region werden zunehmend abstraktere Features gewonnen, die Eigenschaften in einem größeren Bereich erfassen [45].

Voxelbasierte 3D Modelle, die Sparse Convolutions einsetzen, haben die größte Ähnlichkeit zu den zweidimensionalen Architekturen und können diese sogar nachbilden [46]. Der Hauptunterschied zu den Dense Convolutions der Bilderverarbeitung ist die Reduktion des Rechenaufwands durch die selektive Betrachtung für nicht leere Voxel.

Diese Ansätze erschweren die Anwendung von AL-Techniken, die auf Feature-Raum-Diversität basieren. Die dünne Verteilung der Features im Raum führt zum Versagen konventioneller Ähnlichkeitsmaße bei Punktwolken. Variierende Feature-Positionen verhindern Ähnlichkeitsbestimmungen für die Mehrheit der Voxel aufgrund fehlender Vergleichsfeatures in den überwiegend leeren Punktwolken.

Abbildung 9: Konkrete Darstellung der Problematik von Diversitätsmaßen

Eine Illustration dieser Problematik findet sich in der Gegenüberstellung zweier baugleicher Flansche in Abbildung 9. Dargestellt werden die überlappenden Voxel in Grün und die restlichen in Rot. Trotz identischer Bauart und manueller Ausrichtung resultiert die Überlappung der gefüllten Voxel in ca. 20%.

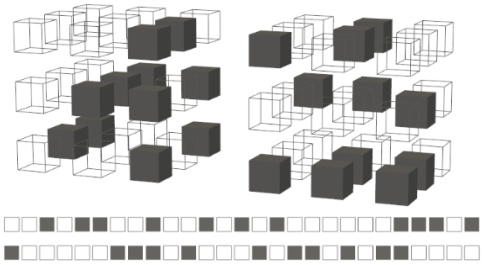


Abbildung 10: Visualisierung der Problematik von Diversitätsmaßen

Beim Scannen von realen Oberflächen kommt es zu zufälligen Abweichungen, welche die Position der Voxel beeinflussen können. Diese Unregelmäßigkeiten können simuliert werden. Abbildung 10 zeigt, Voxel in einem 3×3 Gitter, welche zufällig zu $^1/_3$ gefüllt wurden. Dennoch sind nur zwei nicht leere Voxel an identischen Positionen und stehen für Vergleichsoperationen wie z. B. das Skalarprodukt zur Verfügung. In realistischen Szenarien befinden sich die Objekte in Punktwolken selten an derselben Position und kommen zumeist in unterschiedlichen Orientierungen vor.

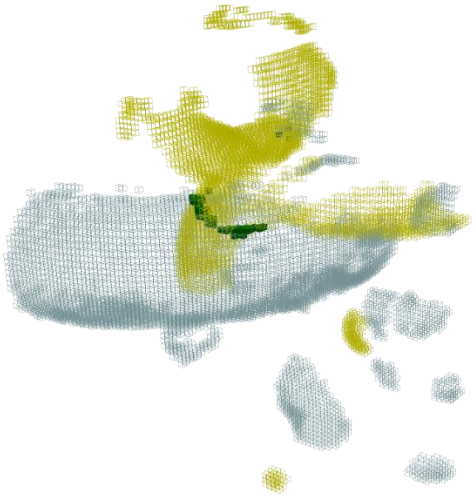


Abbildung 11: Realistische Visualisierung der Problematik

Der Einsatz von Pooling Operationen und Convolutions mit einem Stride größer als eins wirken dem entgegen. Dies hat den Effekt, dass die räumliche Größe antiproportional zu dem Stride abnimmt, bzw. die Voxelgröße proportional Steigt. Zusätzlich kommt es zu einer multiplikativen Vergrößerung des Stride für jede Schicht. So führen fünf Schichten mit einem Stride von zwei bereits zu einem Finalen Stride von 32. Abbildung 11 zeigt zwei übereinandergelegte Punktwolken in den Farben Gelb und Blau. Diese wurden mit einer Voxelgröße von 1mm diskretisiert und anschließend wurde eine Reduktion der Auflösung durch einen Stride von 32 angewendet. Ungeachtet der der reduzierten Auflösung befinden sich von 7932 Voxeln hier nur 36 an derselben Position, was circa 0.4% entspricht. Diese Voxel werden in Grün dargestellt. Diese Diskrepanz resultiert in Informationsverlust da die meisten

Voxel nicht zur Ähnlichkeitsbestimmung herangezogen werden, was die Aussagekraft des Ähnlichkeitsmaßes reduziert.

In punktbasierten Architekturen verschärft sich diese Problematik. Die Beibehaltung kontinuierlicher Positionen der Punkte resultiert in einer geringen Wahrscheinlichkeit exakter Übereinstimmungen. Für das Durchführen der Experimente wurde nur ein Voxelbasiertes Modell verwendet (siehe Kapitel 4.2). Infolgedessen wurde von einer eingehenderen Analyse der spezifischen Herausforderungen punktbasierter Architekturen abgesehen.

3.2 Lösungsansätze für die Anwendung von Diversitätsmaßen in Punktwolken

Im vorangegangenen Kapitel wurde die Problematik von Diversitätsmaßen in Punktwolken erläutert. Basierend auf den in Kapitel 2.7 präsentierten Forschungsarbeiten erfolgt nun eine Darlegung von Lösungsansätzen. Diese zielen darauf ab, die Anwendbarkeit diversitätsbasierter Active Learning-Techniken auf Punktwolkendaten zu ermöglichen

3.2.1 Modellarchitektur-basiertes Vorgehen

Wie im Kapitel 3.1 erläutert wurde, kann der Ursprung der Problematik in den unterschiedlichen Datenformaten und Modellarchitekturen verortet werden. Der Einsatz einer adäquaten Modellarchitektur, die durch den intrinsischen Aufbau das Auftreten der Problematik verhindert ist als potenzielle Strategie naheliegend. Hierbei ist anzumerken, dass im Kontext des Deep Learning für Punktwolken eine Vielzahl von Modellarchitekturen zur Disposition steht. Die Selektion einer geeigneten Architektur stellt somit eine Möglichkeit dar, den Einsatz von Diversitätsmaßen im Bereich der Punktwolkenverarbeitung zu ermöglichen. Es ist zu betonen, dass die Wahl der Modellarchitektur nicht nur zur Lösung der diskutierten Problematik beiträgt, sondern auch maßgeblich die Performance des Objekterkennungsmodells beeinflusst. Des Weiteren ist anzumerken, dass der Wechsel zu einer alternativen Modellarchitektur mit zusätzlichem Aufwand hinsichtlich der Implementierung sowie der Optimierung der Hyperparameter einhergehen kann. Wenngleich dieser Aspekt in der praktischen Umsetzung Beachtung finden muss, wird er in der nachfolgenden Analyse nicht mit einbezogen, da dies von den spezifischen Umständen der Realisierung abhängig ist. Infolgedessen wird exemplarisch eine Analyse einer Modellarchitektur vorgenommen und die Eignung für die Adressierung der vorliegenden Herausforderung evaluiert.

Dass die Problematik bei der Diversitätsbestimmung durch die Wahl der Modellarchitektur vermieden werden kann, wurde von Moses et al. empirisch belegt [47]. Ihre Architektur basiert auf VoxelNet [48], dessen Grundprinzipien zur Feature-Extraktion im Folgenden erläutert werden.

Die Verarbeitung beginnt mit der Voxelisierung der Punktwolke, wobei der dreidimensionale Raum in gleichmäßige Gitterzellen von 0,2 m Größe unterteilt wird. Im Unterschied zu der konventionellen diskretisierung zu Voxeln werden jedem Voxel mehrere Punkte zugewiesen, die um ihre relative

Position zum Voxelzentrum ergänzt werden. Die Feature-Extraktion erfolgt anschließend in zwei Schritten: Zunächst erzeugen mehrere sogenannte Voxel Feature Encoding Layer lokale Features für die Voxel, die als Feed-Forward-Netzwerk mit anschließendem Max-Pooling konzipiert sind. Im zweiten Schritt kommen Sparse Convolutions zum Einsatz, die die finalen Features generieren und durch den Einsatz eines Stride von 2 die Dimension der Feature Map reduzieren.

Durch die Verwendung von großen, unkonventionellen Voxeln mit eigener Feature Extraktion und Convolutions mit Stride kommt es zu einer erheblichen Reduktion der räumlichen Ausdehnung der Feature Map. Hierdurch verbleiben 400 Voxel was einer Ausdehnung von $\sqrt[3]{400} \approx 7$ Voxeln pro Dimension entspricht. Dies reduziert die Wahrscheinlichkeit leere Voxel zu vergleichen und folglich erhöht es die Ausdrucksstärke des Feature Vergleichs.

3.2.2 Gradienten basierte Diversitätsanalyse

Der Ursprung der Problematik liegt in der dünnen Verteilung des betrachteten Merkmals. Demzufolge könnte eine Modifikation des betrachteten Merkmals einen Ansatz darstellen. In diesem Kontext ist auf die bereits in Kapitel 2.4.3 erörterte Möglichkeit zu verweisen, Gradienten als alternative Merkmalsrepräsentation heranzuziehen.

Der Einsatz von Gradienten ermöglicht es, die zuvor beschriebenen Schwierigkeiten beim Vergleich der Features zu umgehen. Dies resultiert aus der Tatsache, dass Gradienten für jeden Modellparameter existieren, wobei die Anzahl der Gradienten unabhängig von der spezifischen Menge der Eingabedaten ist. Infolgedessen wird eine konsistente Basis für Vergleiche geschaffen, die nicht den Limitationen unterliegt, welche sich aus der Struktur des Feature-Raums ergeben. Problematisch bei diesem Vorgehen ist die Notwendigkeit von Annotationen für die Daten. Die Voraussetzung von annotierten Daten steht in einem Konflikt mit dem Grundprinzip des Active Learning, welches auf den Daten ohne Annotationen durchgeführt wird. Zur Lösung dieses Konflikts wird eine Heuristik benötigt, welche hypothetische Annotationen bestimmt. Hierfür stehen verschiedene Techniken zur Verfügung wie das Nutzen der Modellausgabe, Markov-Sampling oder der Einsatz eines Ensembles. Empirisch belegt wurde dieses Verfahren bereits für die Bildklassifikation [25], 2D Objekterkennung [49] und die 3D Objekterkennung in Punktwolken [41].

3.2.3 Feature-Aggregation

Ein weiterer Ansatz zur Bewältigung dieser Herausforderung besteht in der Aggregation der Features. Hierbei wird eine Verdichtung der Informationen durch die Aggregation der Features angestrebt, was z.B. durch Bildung des Durchschnitts oder des Maximums geschehen kann. Bei der Implementierung dieser Methodik lassen sich grundsätzlich zwei Herangehensweisen unterscheiden: die globale und die lokale Aggregation.

Die globale Aggregation zeichnet sich dadurch aus, dass die Features für alle Datenpunkte zu einem einzigen Feature-Vektor zusammengefasst werden. Der Vorzug dieses Verfahrens liegt in seiner einfachen Umsetzung und der universellen Einsetzbarkeit, unabhängig von der gewählten Netzwerkarchitektur oder der spezifischen Aufgabenstellung und kann somit für die Klassifizierung, semantische Segmentierung und die Objekterkennung eingesetzt werden. Es ist jedoch anzumerken, dass diese Methode mit einem Informationsverlust einhergeht. Begründet werden kann dies damit, dass mit einer steigenden Anzahl an Features, die aggregiert werden, der resultierende Feature-Vektor stärker gegen den Durchschnitt konvergiert. Hierdurch nimmt die Ähnlichkeit der Feature-Vektoren zu, wodurch die Bedeutung von Ähnlichkeitsmaßen abnimmt. Darüber hinaus führt das Aggregieren von allen Features zu einem Verlust der räumlichen Informationen, welche für stark lokalisierte Aufgaben wie die Objekterkennung und die semantische Segmentierung von Bedeutung sind [29], [30]. Dennoch wurde das Verfahren bereits für die 3D Objekterkennung eingesetzt und kann Bedarf an Annotation verringern [40].

Im Gegensatz dazu zielt die lokale Aggregation darauf ab, Features in begrenzten räumlichen Bereichen zu aggregieren. Der Vorteil dieses Ansatzes besteht in der Beibehaltung räumlich lokalisierter Features, wodurch der Informationsverlust im Vergleich zur globalen Methode reduziert wird. Die Begründung für die Effektivität dieses Ansatzes liegt in der Aggregation einer geringeren Anzahl von Features. Hierdurch wird vermieden, dass potenziell informative Regionen mit ausgeprägter räumlicher Lokalisierung durch den Aggregationsprozess von dem Durchschnitt dominiert werden. Hieraus resultiert die bessere Abbildung der lokalen Region durch den resultierenden Feature-Vektor.

Ein Anwendungsgebiet dieses Vorgehens ist die semantische Segmentierung für Punktwolken. Hierbei erfolgt die Bestimmung von lokalen Regionen durch ein Clustering, wobei die innerhalb eines Clusters extrahierten Features aggregiert und zur Maximierung der Diversität herangezogen werden [39], [50], [51]. Außerdem erstreckt sich die Anwendung auch auf die 2D Objekterkennung, wo lokale Regionen durch die erkannten BB definiert wurden (Siehe PPAL S. 30).

Es ist jedoch anzumerken, dass die Anwendung dieser Methodik die Entwicklung einer adäquaten Strategie zur Ähnlichkeitsbestimmung durch den Vergleich lokaler Features erfordert. Die Herausforderung hierbei liegt in der potenziell variierenden Anzahl lokaler Regionen, welche eine direkte Anwendung etablierter Ähnlichkeitsmaße ohne weitere Adaption nicht zulässt. Infolgedessen bedarf es der Konzeption und Implementierung spezifischer Verfahren zur Überwindung dieser Beschränkung, um eine effektive Nutzung der lokalen Feature-Informationen zu ermöglichen.

3.3 Bewertungsschema für Active Learning Algorithmen

Die Bewertung und der Vergleich von Active Learning Algorithmen im Kontext des Deep Learning stellen eine komplexe Herausforderung dar. Es existiert

keine universelle Methode zur Vorhersage der Leistungsfähigkeit eines spezifischen AL-Algorithmus, da dessen Erfolg maßgeblich vom verwendeten Datensatz und Modell abhängt [52], [53], [54], [55]. Durch das Fehlen eines Standards für die AL-Evaluation wird die Ergebnispräsentation in wissenschaftlichen Veröffentlichungen mit unterschiedlichen Datensätzen und Modellarchitekturen durchgeführt. Dieser Umstand verhindert somit einen aussagekräftigen Vergleich der Leistungsfähigkeit verschiedener Arbeiten.

Zusätzlich wird die Performance von der Zusammensetzung des initialen Trainingsdatensatzes signifikant beeinflusst. Der Einfluss dieses Effekts kann weitreichender sein als die erwartbare Einsparung durch AL. Trotz identischer Modellarchitektur und Datensatz wurde für die Genauigkeit der Bildklassifikation bereits eine absolute Abweichung von 13% für die zufällige Baseline festgestellt [56]. Somit ist es nicht möglich die Ergebnisse mit den genannten Werten von anderen Arbeiten zu vergleichen, da die Konfigurationen für die Pools der initial gelabelten Daten meist nicht veröffentlicht werden.

Ein beobachtbarer Trend in der Entwicklung von AL-Algorithmen zeigt sich in der Kombination verschiedener Techniken. Während in den frühen wissenschaftlichen Arbeiten in der Regele einzelnen Methoden untersucht wurden, lässt sich hier ein Trend zur Untersuchung von Lösungsansätzen mit kombinierten Techniken erkennen.

4 Implementierung der Versuchsreihe

Nach dem im vorherigen Kapitel der Rahmen für die Übertragung von Active Learning Algorithmen in die Anwendung von Punktwolken gelegt wurde, wird in diesem Kapitel die Versuchsreihe beschrieben. Zunächst folgt eine Erläuterung des Datensatzes (Kapitel 4.1) sowie des verwendeten KI-Modells (Kapitel 4.2). Die Kenntnis von dem Datensatz und Modell ist von Bedeutung, da die Leistungsfähigkeit von AL-Algorithmen maßgeblich von diesen abhängig ist (siehe Kapitel 3.3). Darauf folgt der grundlegende Aufbau der durchgeführten Versuchsreihe (Kapitel 4.3). Abschließend wird die Auswahl der für die experimentelle Untersuchung herangezogenen Algorithmen dargelegt und begründet (Kapitel 4.4).

4.1 Datensatz

Wie bereits in Kapitel 3.3 beschrieben wurde, ist die Auswahl des Datensatzes von zentraler Bedeutung für Performance von Active Learning-Algorithmen. Zur besseren Einordnung der Ergebnisse beschreibt dieses Kapitel den für die Experimente verwendeten Datensatz. Dazu gehören Zweck, Umfang, Struktur und Verteilung der Daten sowie potenzielle Herausforderungen bei der Objekterkennung. Die Darstellung des Datensatzes bildet die Grundlage für das Verständnis der nachfolgenden Experimente und ermöglicht eine Einordnung der erzielten Ergebnisse. Zudem werden mögliche Einschränkungen und Besonderheiten des Datensatzes diskutiert, die bei der Interpretation der Resultate zu berücksichtigen sind.

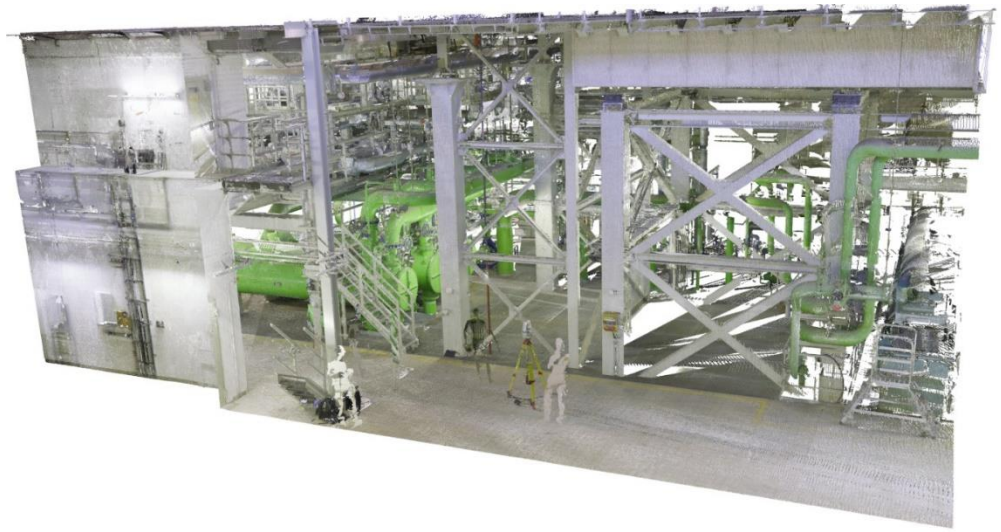


Abbildung 12: Beispielpunktwolke einer Kraftwerksanlage

Vor dem Hintergrund der KI-basierten Objekterkennung in Punktwolken, wird der vorliegende Datensatz als Trainingsdatensatz im Rahmen der Versuchsreihe herangezogen. Ziel ist die automatisierte Erkennung von Rohrleitungssystemen und deren Komponenten. Dies bildet die Grundlage für die Rückführung der Rohrkomponenten in CAD-Modelle. Im Rahmen dieser Arbeit liegt der Fokus ausschließlich auf der KI-basierten Objekterkennung.

Um die hierfür erforderliche KI entsprechend zu trainieren, wurden die Punktwolken von unterschiedlichen industriellen Anlagen genutzt. Abbildung 12 zeigt exemplarisch die Punktwolke eines Kraftwerks. Zur Erfassung dieser Punktwolken werden Laser Scanner eingesetzt. Diese tasten die Umgebung 360° sphärisch ab und speichern jedes Mal, wenn der Laser auf eine Oberfläche trifft, einen Punkt. Informationen zu den eingesetzten Laser Scannern, sowie der Erfassungsmethode (terrestrischer Scanner, mobiler Scanner, Drohne, etc.) liegen nicht vor. Die Rohrsysteme in Industrieanlagen bestehen aus einer Vielzahl von Komponenten. Im Rahmen der Versuchsreihe liegt der Fokus auf den drei folgenden Komponententypen, respektive Klassen:

Abbildung 13: Abbildung eines Flansches in einer Punktwolke

Flansche sind flache Werkstücke, welche mit einem Lochkreis versehen sind. Durch die Kopplung zweier Flaschen lassen sich Rohrabschnitte miteinander verbinden.

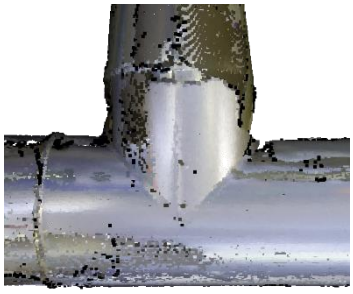


Abbildung 14: Abbildung eines T-Stück in einer Punktwolke

Ein T-Stück kann zum Verbinden oder Trennen von Rohrleitungen eingesetzt werden. Sie können als definierte Standardteile oder als Eigenbau in einem Rohrsystem vorkommen.

Abbildung 15: Abbildung eines Rohrbogens in einer Punktwolke

Rohrbögen dienen der Richtungsänderung in einem Rohrleitungssystem und können als Fertigteil mit genormtem Winkel oder als Sonderanfertigung mit beliebigem Winkel vorkommen.

Wie bereits erwähnt, erfolgt die Durchführung der Versuchsreihe mit einer oben genannten Auswahl an drei Klassen. Die Klassen Rohrbogen und T-Stück wurden aufgrund ihrer Bedeutung im Datensatz ausgewählt. Als richtungsändernde Bauteile sind diese von zentraler Bedeutung für den Verlauf und die topologische Struktur von Rohrleitungssystemen. Die Auswahl

der Klasse Flansch erfolgte, um kleine Bauteile zu repräsentieren. Durch die flache Bauweise besitzen diese verhältnismäßig kleine Ausmaße. Außerdem sind Flansche häufig in unmittelbarer Nähe zu anderen Bauteilen positioniert, wodurch der Anspruch an die Lokalisierung erhöht wird. Allerdings kommen diese in einer großen Anzahl vor, weshalb eine gute Erkennung erwartet wird. Die getroffene Auswahl an Klassen soll eine differenzierte Analyse der Leistungsfähigkeit der Active-Learning-Algorithmen ermöglichen.

Weitere Klassen wurden nicht berücksichtigt. Die manuelle Annotation der Daten erfordert einen hohen Zeitaufwand und das Annotieren von weiteren Klassen würde nicht im Verhältnis zum Umfang der Masterarbeit stehen.

Während des Labelns wurden diejenigen Punkte aus dem Datensatz entfernt, die nicht zu den zu trainierenden Objekten und damit nicht zu den Klassen gehören, die später erkannt werden sollen. Dies umfasst alle Punkte des Hintergrunds, welcher hauptsächlich den Boden, Stützstrukturen wie Stahlträger und Wände ausmacht. Weiterhin wurden angrenzende Komponenten sowie Messgeräte oder Rohrhalterungen entfernt. Neben dem verringerten Rechenaufwand vereinfacht dies auch die Aufgabe der Objekterkennung. Ohne Hintergrund wird die korrekte Lokalisierung in den Vordergrund gestellt, da die Unterscheidung zwischen Hintergrund und den gesuchten Objekten entfällt.

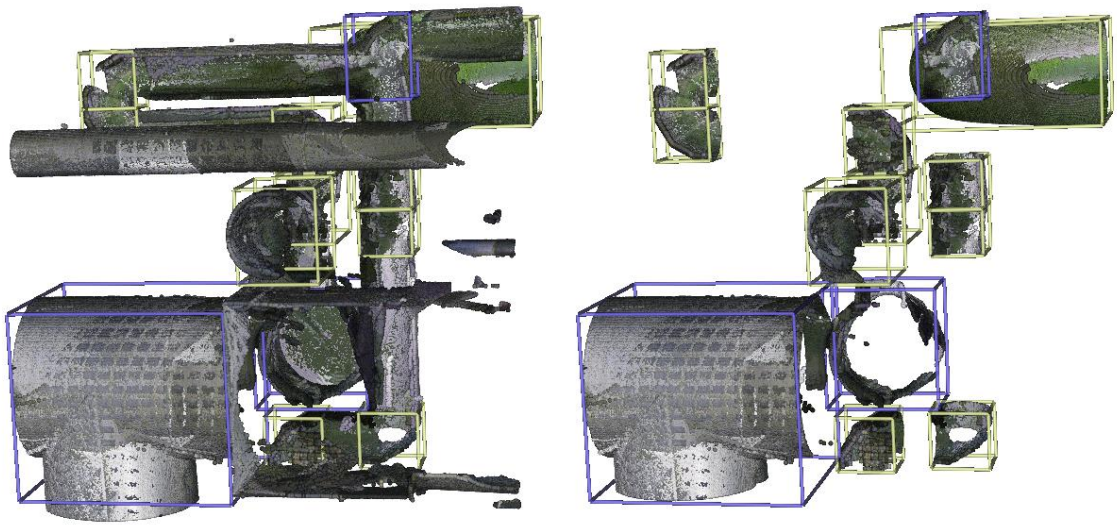


Abbildung 16: Beispiel Punktwolken für das Training. Links sind alle Punkte vorhanden, während rechts nicht verwendete Punkte entfernt wurden.

Abbildung 16 zeigt einen Ausschnitt einer Punktwolke, in welcher die Umrisse der Annotationen farblich illustriert sind. Auf der linken Seite sind alle Punkte enthalten, während auf der rechten Seite ausschließlich Punkte innerhalb der annotierten Bounding Boxes vorhanden sind.

Aus den Punktwolken der Industrieanlagen wurden kleinere Ausschnitte für das Durchführen der Versuchsreihe erstellt. Dies wurde aus drei Gründen durchgeführt:

1. Große Ausmaße der Scans: Die Scans der Industrieanlagen weisen teilweise große Ausmaße auf. Die Aufnahmen können eine Größe von mehreren Terabyte besitzen und tausende zu erkennende Objekte beinhalten. Diese Größe ist zum Trainieren von DL-Modellen nicht praktikabel und könnte mit den zur Verfügung stehenden Resources nicht bewerkstelligt werden.

2. Objekterkennung erfolgt lokal: Die Objekterkennung ist eine lokale Aufgabe. Zum einen haben die zu erkennende Objekte im Verhältnis zu der Umgebung kleine Ausmaße und zum anderen sollen Datenpunkte, die weit entfernt vom Objekt liegen, nicht zur Erkennung beitragen.

3. Unterschiedlicher Informationsgehalt von Objekten: Es wurde die Annahme getroffen, dass nicht alle Objekte in einer Punktwolke den gleichen Informationsgehalt und damit den gleichen Mehrwert hinsichtlich des Trainingseffekts eines KI-Modells aufweisen. Hintergrund ist zum einen der hoher Standardisierungsgrad der verwendeten Rohrkomponenten und zum anderen eine hohe Übereinstimmung bzgl. des Formfaktors (bspw. existieren Rohrbögen in unterschiedlichen Durchmessern bei gleichem Formfaktor).

Zum Aufteilen der Punktwolken wurde ein K-Means Clustering der BB-Zentren durchgeführt und die Clusterzugehörigkeit als Kriterium für die Erzeugung von Ausschnitten genutzt. Der Parameter für die Anzahl an Clustern wurde so gewählt, dass die Anzahl an BB pro Ausschnitt 10 beträgt. Durch das Clustering entstehen insgesamt 1812 Ausschnitte, welche für das Sampling von AL-Algorithmen, Trainieren und zur Evaluation zur Verfügung stehen.

Zur Datenbereinigung wurden zwei Maßnahmen ergriffen. Zum einen wurden BB mit weniger als 30 Punkten entfernt. Dies kann auftreten, wenn die Distanz zwischen der gescannten Oberfläche und dem Laserscanner groß ist, da sich die resultierende Auflösung antiproportional zu dieser Distanz verhält. Außerdem ist das gesamte Erfassen der Anlage durch die engen Verhältnisse in industriellen Anlagen nicht immer möglich, wodurch Bauteile in Regionen mit einer hohen Bauteildichte nur partiell erfasst werden. Durch diese Maßnahme wurden 507 BB und die darin enthaltenen Punkte aus dem Datensatz entfernt.

Des Weiteren wurden alle Punkte entfernt, die nicht innerhalb einer Bounding Box liegen. Dies wurde durchgeführt, da aufgrund von Rauschen nicht alle Punkte zweifelsfrei einem Objekt zugeordnet werden können. Die Ursachen für das Rauschen können unterschiedlich sein. So kann der Reflexionsgrad der Oberflächen zu Spiegelungen und damit zu Rauschen führen. Dies kann dazu führen, dass ein Punkt in einiger Entfernung zu der Oberfläche erkannt wird und als Rauschen keinem Objekt mehr zugeordnet werden kann. In einigen Regionen ist dieser Effekt so stark, dass die korrekte Lokalisierung und die Bestimmung der Klasse selbst für Menschen nicht mehr möglich sind, da die Bauteile ineinander übergehen. Im Labeling Prozess wurden solche unklaren Fälle vom Trainingsdatensatz ausgeschlossen und es wurden keine Annotationen für diese erstellt. Durch das Entfernen aller Punkte, welche nicht

innerhalb einer BB liegen, konnte sichergestellt werden, dass diese ungelabelten Punkte keinen Einfluss auf das Modelltraining besitzen.

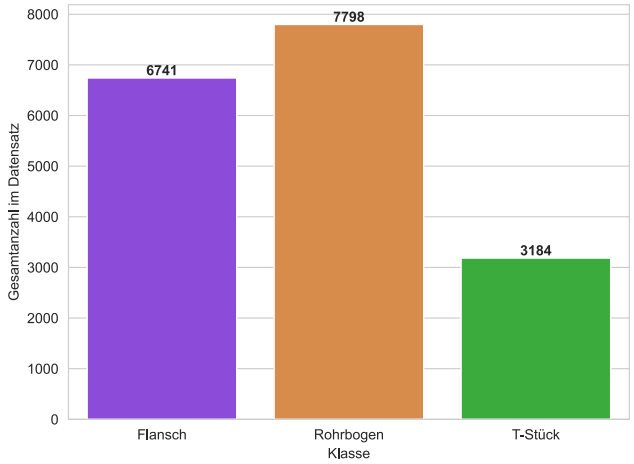


Abbildung 17: Verteilung der Klassen des verwendeten Datensatzes

Abbildung 17 zeigt die Verteilung der drei definierten Klassen innerhalb des Datensatzes. Dieser besteht insgesamt aus 17723 Objekten, die sich in 7798 Objekte der Klasse Rohrbogen, 6741 Objekte der Klasse Flansch und 3184 Objekte der Klasse T-Stück aufteilen. Hieraus ergibt sich, dass die Klasse T-Stück im Verhältnis zu den anderen beiden Klassen unterrepräsentiert ist. Um die Erkennung dieser Klasse zu verbessern, wurde ein Dataset Sampling durchgeführt [57]. Hierbei werden die Instanzen der unterrepräsentierten Klasse in einer Datenbank gespeichert und die Punkte und Annotationen während des Trainings an zufälligen Positionen wieder eingefügt. Hierdurch vergrößert sich die gesehene Anzahl an Objekten der unterrepräsentierten Klasse, was die Erkennung verbessern kann.

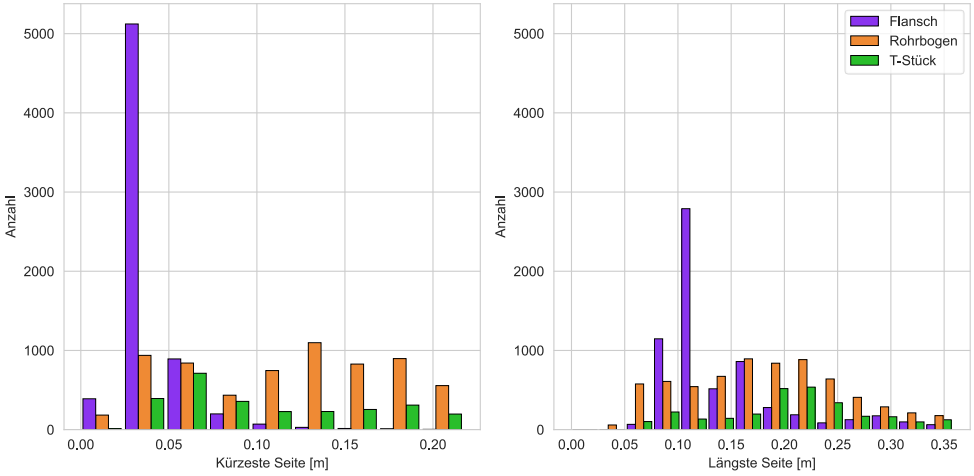


Abbildung 18: Verteilung der Bounding Box Ausmaße. Auf der linken Seite ist die Verteilung der kürzesten Seite dargestellt und rechts die der längsten Seite.

Abbildung 18 zeigt die Verteilung der BB Größen. Dargestellt ist die Größe der kürzesten (links) und der längsten (rechts) Seite für 90% der Daten. Bauartbedingt sind Flansche schmal. Über 50% der Flansche sind kleiner als 5cm in der Tiefe. Generell besitzen 90% der Teile Ausmaße von weniger als 40cm. Gleichzeitig kommen im Datensatz auch deutlich größere Objekte vor, das größte Objekt ist über 2m groß. Dementsprechend muss das Modell zur Objekterkennung dazu in der Lage sein sowohl kleine Teile präzise zu lokalisieren sowie große Teile zu erkennen.

4.2 Modell

Nach der Betrachtung des Datensatzes erfolgt nun die Betrachtung der Modellarchitektur sowie den Spezifikationen des für die Versuchsreihe eingesetzten Objekterkennungsmodells. Es werden die grundlegenden Anforderungen an das Modell dargelegt und dessen struktureller Aufbau erläutert. Hierdurch soll die Einordnung der Resultate ermöglicht werden, da die Performance von Active Learning Algorithmen von dem eingesetzten Modell abhängig ist [21].

Für die Objekterkennung in Punktwolken steht eine Reihe an Modellarchitekturen zur Auswahl. Durch die Analyse dieser und durch die Betrachtung des zur Verfügung stehenden Datensatzes werden im Folgenden Anforderungen herausgearbeitet.

Durch die Relevanz vom autonomen Fahren und die Verfügbarkeit von entsprechenden Datensätzen wurde eine Vielzahl von Modellarchitekturen für dieses Anwendungsgebiet entwickelt [58], [59], [60], [61], [62], [63], [64], [65]. Dies führte zu der Adaption der Modellarchitekturen an die Gegebenheiten der Datensätze. Hierzu gehört unter anderem das sich Objekte überwiegend in der Ebene befinden. Dies steht im Kontrast zu dem zur Verfügung stehenden Datensatz. Aufgrund der Topologie von Industrieanlagen kommt es zu einer hohen Dichte an Objekten in der z-Achse. Das Modell darf folglich keine Annahme über die Position der Objekte treffen (Anforderung 1).

Durch die dünne Bauweise der Flansche ist die Erkennung ebendieser besonders herausfordernd, da bereits eine Verschiebung von wenigen Zentimetern entlang der Rotationsaxe einen drastischen Einfluss auf die IoU (Intersection over Union) besitzt. Des Weiteren sind Flansche als Verbindungsteil meistens direkt neben anderen Bauteilen positioniert. Diese Konstellation kann bei suboptimaler Lokalisierung zur Elimination relevanter Bounding Boxes durch den Non-Maximum-Suppression-Algorithmus führen. Infolgedessen ist die Präzision der Lokalisierung als kritischer Faktor für die Leistungsfähigkeit des Systems zu betrachten (Anforderung 2).

Außerdem besitzen die Objekte im Datensatz eine große Spannweite von Größen, welche von wenigen Zentimetern bis hin zu mehreren Metern reicht. Hieraus folgt die Voraussetzung, dass Objekte mit unterschiedlichen Ausmaßen erkannt werden (Anforderung 3). Da die meisten Teile ähnliche Ausmaße besitzen, wird diese Anforderung als optional betrachtet.

Zusammenfassend werden die Anforderungen hier aufgelistet:

1. Keine Annahme über Position von Objekten
2. Genaue Lokalisierung
3. Erkennung von Objekten mit unterschiedlichen Ausmaßen

Als Modellarchitektur für die Objekterkennung wurde FCAF3D (engl. fully convolutional anchor-free 3D object detection) gewählt. Es ist ein Single-Stage-Objekterkennungsmodell, d.h. es führt die Erkennung in einem einzigen Schritt durch, ohne Region Proposal oder Segmentierung von Hintergrundpunkten [66].

Als ankerfreies Modell verzichtet es auf die Verwendung vordefinierter Anker und bestimmt die Ausmaße direkt aus den Features. Anker basierte Modelle dagegen bestimmen die Lokalisierung und Ausmaße relativ zu vorgegebenen Ankern und das Modell lernt die Anker an die Objekte anzupassen. Durch das Verzichten auf Anker wird eine höhere Flexibilität bei der Erkennung von Objekten mit unterschiedlen Seiten- und Größenverhältnissen gewonnen.

Ein Fully Convolutional Neural Network zeichnet sich dadurch aus, dass ausschließlich Convolutional Layer zum Einsatz kommen. Diese Architektur erfordert eine räumlich strukturierte Eingabe. Zur Verarbeitung von Punktwolken erfolgt daher eine Diskretisierung in Voxel. Voxel sind das dreidimensionale Äquivalent zu Pixeln.

Die Transformation der 3D-Koordinaten in Voxel erfolgt unter Verwendung einer Voxelgröße von 1mm, im Gegensatz zur Größe von 50mm im Originalmodell. Diese Modifikation stammt aus der Beobachtung, dass circa 50% der Flansche in ihrer geringsten Ausdehnung unter 50mm liegen. Diese flachen Flansche stellen etwa 35% aller Objekte dar.

Die Wahl einer reduzierten Voxelgröße von 1mm optimiert die Lokalisierung kleinerer Objekte, was insbesondere bei Flanschen aufgrund ihrer flachen Bauweise relevant ist. Eine Voxelgröße, welche die Objektdimensionen übersteigt, würde zu einer Reduzierung auf zweidimensionale Strukturen führen, was eine präzise Lokalisierung verhindert.

FCAF3D nutzt Sparse Convolutions, um den Speicherbedarf zu reduzieren. Das bedeutet, dass nur die Informationen von Voxeln gespeichert werden, die einen Punkt enthalten. Dies ist effizient, da in 3D-Daten oft große Bereiche leer sind.

Das Modell folgt der typischen Architektur von zweidimensionalen Modellen zur Objekterkennung, bestehend aus einem Backbone, Neck und Head. Als Backbone dient ResNet [67], bei welchem die Dense Convolutional Layer durch Sparse Convolutions ersetzt wurden [46]. Während das FCAF3D im Original ein Backbone mit 34 Schichten verwendet, wurden die Schichten für die Experimente auf 18 reduziert. Dies verringert die Anzahl der Parameter und reduziert somit die Neigung zum Overfitting. Außerdem wird durch die

Reduzierung der Parameter das Training beschleunigt, wodurch mehr Experimente in derselben Zeit durchgeführt werden können.

Im Neck kommt ein Feature Pyramid Network zum Einsatz. Dies ist eine Architektur zur Objekterkennung in CNNs [68]. Es adressiert das Problem der Skalenvarianz durch Erstellung einer mehrstufigen Merkmalspyramide. Hierzu werden die Feature Maps aus verschiedenen Schichten des Backbones extrahiert und die hochaufgelösten Feature Maps der früheren Schichten mit den semantisch ausdruckstärkeren aber räumlich geringer aufgelösten Features der späteren Schichten angereichert. Anschließend wird im Head die Erkennung für jede Schicht der Feature-Pyramide durch drei Convolutional Layer durchgeführt. Dies ermöglicht die Erkennung von Objekten verschiedener Größen in einem einzelnen Netzwerk bei gleichzeitiger Beibehaltung der Inferenzgeschwindigkeit.

Durch den Einsatz von Sparse Convolutions und die ankerfreie Architektur eignet sich FCAF3D besonders für Anwendungen, bei denen die Form der Objekte stark variiert oder im Vorfeld nicht bekannt ist.

4.3 Aufbau der Experimente

Um die Anwendung von Active Learning Algorithmen aus dem zweidimensionalen in Punktwolken zu evaluieren, wurde eine Reihe von Experimenten durchgeführt. Die experimentelle Ausgestaltung folgt dabei einem systematischen Ansatz, der im Folgenden näher erläutert wird.

Mit circa 17.000 annotierten Instanzen ist der Datensatz kleiner als andere Datensätze für die Objekterkennung in Punktwolken wie beispielsweise SUN RGBD (64.000) [11] oder nuScenes (1.4 Millionen) [69]. Zur Reduzierung von Effekten durch Overfitting und um die Reproduzierbarkeit zu gewährleisten, wurde hierzu für alle Experimente eine Kreuzvalidierung durchgeführt.

Bei einer K-fachen Kreuzvalidierung wird der gesamte Datensatz in K partitionen mit einer ähnlichen Größe aufgeteilt. Von diesen Partitionen wird eine für die Validierung genutzt und das Training wird auf den verbleibenden $\mathrm{K}-1$ Partitionen durchgeführt. Dieser Vorgang wird K-mal durchgeführt, wobei jede Partition einmal für die Validierung verwendet wird. Für die Experimente wurde $\mathrm{K}=5$ gewählt wodurch 20% der Daten für die Validierung genutzt werden.

Vor Beginn der Experimente wurde für jede Partition der Kreuzvalidierung eine Konfiguration an initialen Trainingsdaten festgelegt und diese über die Experimente hinweg konstant gehalten. Somit verfügen alle Experimente über dieselbe Ausgangslage, was einen adäquaten vergleich der Algorithmen ermöglichen soll.

Um die Schwankung der Modellperformance durch unterschiedliche Initialisierungen der Modellparameter zu reduzieren, wurde in jeder Iteration des AL-Zyklus das Modelltraining dreifach wiederholt. Hierbei unterscheiden sich die Trainingsdurchläufe nur in der Initialisierung der Modellparameter. Für AL-Algorithmen, die auf einem Ensemble basieren, wurden stattdessen die

Modelle des Ensembles zur Evaluierung verwendet. Die Wahl einer dreifachen Wiederholung stellt einen Kompromiss dar. Hierbei wurde der erforderliche Rechenaufwand gegen potenzielle Fehlinterpretationen durch den Einfluss der Varianz abgewogen. Die Entscheidung für drei Wiederholungen wurde anhand von Erfahrungswerten getroffen.

Wie bereits in Kapitel 3.3 erörtert wurde, kann die Performance eines AL-Algorithmus stark von dem initialen Trainingspool beeinflusst werden. Um die Laufzeit der Experimente nicht weiter zu erhöhen, wurden keine Maßnahmen wie das mehrfache Durchführen mit unterschiedlichen initialen Daten durchgeführt. Stadtessen wird angenommen, dass durch das Durchführen der Kreuzvalidierung und folglich unterschiedlichen initialen Trainingsdaten, weitere Effekte durch die Wahl der initialen Daten vernachlässigbar sind.

Die dargelegte experimentelle Struktur soll durch die Einbeziehung verschiedener Varianzquellen eine differenzierte Betrachtung der Ergebnisse ermöglichen. Hierbei wird sowohl die Varianz innerhalb einzelner Trainingsläufe als auch die Konsistenz der Modellperformanz über diverse Datenpartitionen und der initialen Trainingsdaten hinweg berücksichtigt. Diese Vorgehensweise soll die Reproduzierbarkeit der experimentellen Resultate erhöhen und potenzielle systematische Verzerrungen minimieren.

4.4 Auswahl der Active Learning Algorithmen für die Experimente

Das Ziel dieser Arbeit besteht darin, Active Learning Algorithmen aus der 2D Objekterkennung auf ihre Anwendbarkeit für die 3D Objekterkennung in Punktwolken zu überprüfen. Zu diesem Zweck wird im Folgenden die Auswahl, der für die experimentelle Untersuchung relevanten Algorithmen erörtert.

In den vorangegangenen Kapiteln wurden hierfür grundlegende Vorüberlegungen angestellt, die als Basis für die nachfolgende Selektion der Algorithmen dienen. Zunächst wurde eine Taxonomie entwickelt, die eine systematische Einordnung der zu untersuchenden Algorithmen ermöglicht (Kapitel 2.4). Diese Kategorisierung stellt ein wesentliches Instrument dar, um die charakteristischen Eigenschaften und Anwendungsbereiche der verschiedenen AL-Ansätze zu erfassen und zu strukturieren.

Darüber hinaus erfolgte eine Analyse der spezifischen Problematiken, die bei der Übertragung von AL-Methoden in den dreidimensionalen Raum auftreten können (Kapitel 3.1 und 3.2). Diese analytische Betrachtung fokussierte sich insbesondere auf die Adaption von Diversitätsmaßen, welche eine zentrale Rolle in vielen AL-Algorithmen einnehmen. Die gewonnenen Erkenntnisse bilden die Grundlage für die Entwicklung geeigneter Strategien zur Anpassung dieser Maße an die Anforderungen der 3D Objekterkennung.

Auf Basis dieser Vorarbeiten wird die Selektion der Algorithmen vorgenommen, die im empirischen Teil der Arbeit untersucht werden. Die berücksichtigten AL-Algorithmen sollen hierbei ein möglichst breites Spektrum der zuvor erstellten Taxonomie abdecken. Im Folgenden werden zunächst Algorithmen aufgeführt, welche in ihrer Anfragefunktion nur die Vorhersage

des KI-Modells betrachten. Diese zeichnen sich durch ihre gute Übertragbarkeit auf die Domäne von 3D Daten aus. Anschließend werden die Algorithmen betrachtet, welche die Diversität der Trainingsdaten berücksichtigen. Hierdurch können die identifizierten Herausforderungen bei der Diversitätsbestimmung adressiert werden.

Die tabellarische Darstellung (Tabelle 1) zeigt die Einordnung von Algorithmen, deren Anfragefunktion sich ausschließlich auf die Vorhersage stützt. Die Evaluation dieser Verfahren soll Aufschluss darüber geben, inwiefern die alleinige Betrachtung der Vorhersage für effektives Active Learning in Punktwolken ausreichend ist. Die Matrix ordnet Taxonomie Kategorien (Zeilen) den entsprechenden Algorithmen (Spalten) zu. Hierbei indiziert das Symbol • eine Kategorie Zugehörigkeit, während - einen Ausschluss kennzeichnet.

		RoI	**Consensus Score**	**Localization Stability**	**BLAD**
Kontext	Lokal	-	-	-	-
	Global	-	-	-	-
Betrachtetes Merkmal	Vorhersage	•	•	•	•
	Feature	-	-	-	-
	Gradienten	-	-	-	-
Bestimmung des Informationsgehalts	Unsicherheit	-	-	-	-
	Diversität	-	-	-	-
	Konsistenz	•	•	•	•
Aufgabenstellung	Klassifikation	•	-	-	•
	Regression	-	•	•	•
Query Kombination	Sequenziell	-	-	-	-
	Linearkombination	-	-	-	•

Tabelle 1: Einordnung der Algorithmen in die Taxonomie

Hierzu zählen die beiden Ensemble Methoden RoI Matching und der Consensus Score. Während bei dem RoI Matching ausschließlich die Konsistenz in der Klassifizierung berücksichtigt wird, betrachtet der Consensus Score die Konsistenz in der Lokalisierung. Für beide Algorithmen wurde ein Ensemble Größe von drei Modellen gewählt. Diese Größe wurde aus der Referenzimplementierung entnommen. Demgegenüber stehen die transformationsbasierten Methoden Localization Stability und BLAD. Diese zeichnen sich dadurch aus, dass sie die Konsistenz zwischen verschiedenen Transformationen der Eingabedaten bestimmen. Für die Localization Stability ist diese Transformation das Hinzufügen von Rauschen. Es wird ein normalverteiltes Rauschen mit Erwartungswert 0 und Standardabweichungen von 0.01 und 0.02 eingesetzt, wobei die Parameterwahl auf einer empirischen Versuchsreihe basiert. Für BLAD wurden, basierend auf der Referenzimplementierung, horizontale und vertikale Spiegelungen als

Transformationen gewählt. Für einen fairen Vergleich zu den anderen Algorithmen wird nur die AL-Komponente des BLAD Algorithmus verwendet und das Self Supervised Learning wurde nicht implementiert.

Die Einordnung von Algorithmen, welche die Betrachtung der Diversität verwenden ist in Tabelle 2 gegeben. Die Deutung dieser erfolgt analog zu Tabelle 1.

		Core-Set	**PPAL**	**PPAL Global**	**Localization Stability Core-Set**	**CRB**
Kontext	Lokal	-	•	-	-	-
	Global	•	-	•	•	•
Betrachtetes Merkmal	Vorhersage	-	•	•	•	•
	Feature	•	•	•	•	-
	Gradienten	-	-	-	-	•
Bestimmung des Informationsgehalts	Unsicherheit	-	•	•	-	•
	Diversität	•	•	•	•	•
	Konsistenz	-	-	-	•	-
Aufgabenstellung	Klassifikation	-	•	•	-	•
	Regression	-	-	-	•	-
Query Kombination	Sequenziell	-	•	•	•	•
	Linearkombination	-	-	-	-	-

Tabelle 2: Einordnung der Algorithmen in die Taxonomie

Durch die Untersuchung des Core-Set Algorithmus soll die Problematik der globalen Feature Aggregation untersucht werden. Hier erfolgt ein Mean Pooling der Features, die dem Head des Modells für die Erkennung zur Verfügung stehen. Die Ähnlichkeit der resultierenden Feature-Vektoren wird durch die Cosinus-Ähnlichkeit quantifiziert.

Mit dem Ziel, die Relevanz des lokalen Kontexts und der lokalen Feature Aggregation zu evaluieren, wurde zudem PPAL in die experimentelle Untersuchung einbezogen. Hierbei wird ein Mean Pooling auf die Features innerhalb der Bounding Boxes angewendet, wobei die Ähnlichkeit der Feature-Vektoren ebenfalls durch die Cosinus-Ähnlichkeit bestimmt wird.

Im Rahmen der experimentellen Untersuchungen wurde zur potenziellen Verbesserung vorhersagebasierter Methoden durch Diversitätssteigerung eine Kombination von Location Stability und Core-Set implementiert. Der Selektionsprozess gestaltet sich hierbei zweistufig: Zunächst erfolgt mittels Location Stability eine Bewertung der Ausschnitte nach ihrem Informationsgehalt, wobei die 300 informativsten Instanzen extrahiert werden. Aus dieser Teilmenge werden anschließend durch den Core-Set-Algorithmus die finalen 200 Samples selektiert.

Abschließend erfolgt ein Vergleich mit dem CRB-Algorithmus, welcher ursprünglich für 3D-Daten konzipiert wurde. Hierdurch wird eine komparative Evaluation ermöglicht, wodurch sich Erkenntnisse über die Übertragbarkeit von Active Learning Strategien zwischen verschiedenen Datenmodalitäten gewinnen lassen.

5 Evaluation der Ergebnisse

Nachdem im vorangegangenen Kapitel die Durchführung der Testreihen erfolgte, werden in diesem Kapitel die Ergebnisse dargestellt. Anschließend werden diese interpretiert. Zentraler Evaluationskriterium ist die klassenweise Average Precision (AP) nach Pascal VOC. Dies erfolgt unter Verwendung eine Intersection over Union Schwellwertes von 0,5 [70].

Im Folgenden werden die Ergebnisse zu den verwendeten AL-Algorithmen zunächst grafisch dargestellt und im Anschluss beschrieben. In der Darstellung findet sich die Einsparung an benötigten Annotationen in Prozent auf der y-Achse. Negative Werte stellen eine Verringerung dar, während Positive Werte eine Zunahme an benötigten Annotationen entsprechen. Um einen fairen Vergleich der Algorithmen zu ermöglichen wurde die Reduktion über die Gesamtanzahl an Annotationen bestimmt. Dies ist notwendig aufgrund der unterschiedlichen Anzahl an Objekten in den Punktwolken und der Neigung mancher Algorithmen, Punktwolken mit einer großen Anzahl an Objekten auszuwählen. Aufgrund der unterschiedlichen Anzahl an Annotationen zwischen der zufälligen Baseline und den Algorithmen wurden die Werte der Baseline linear interpoliert. Die x-Achse der Graphen zeigt die AL-Iteration an. Zusätzlich ist die die Gesamtanzahl an Annotationen unter der Iteration angegeben. Die Linien zeigen den Durchschnitt über die Partitionen der Kreuzvalidierung an und die schattierte Fläche stellt $\pm$ eine Standartabweichung dar. Die Angabe für die Reduktion der Benötigten Annotationen wurde für die einzelnen Partitionen separat bestimmt. Die Ergebnisse der mehrfach Wiederholten Iterationen (siehe Kapitel 4.3) wurden vor der Bestimmung der Reduktion aggregiert.

Abweichungen in einem Bereich von ±10% werden als nicht signifikant betrachtet. Schwankungen in diese Größenordnung konnten beim Vergleich von Versuchen mit zufälliger Datenauswahl beobachtet werden. Es sollte beachtet werden, dass Abweichungen in der Iteration null nicht durch die Algorithmen entstanden sind. Diese stellt die Ausgangslage dar, die für alle Algorithmen Identisch ist. Große Schwankungen an dieser Stelle stammen einzig von der Initialisierung des Modells und der Instabilität durch das Trainieren auf einer geringen Datenmenge. Des Weiteren besitzen die dargestellten Resultate der dritten Iteration eine eingeschränkte Aussagekraft. Durch die lineare Interpolation über die Datengrenze hinweg ist das Auftreten von Artefakten möglich. Außerdem sollte die Unterrepräsentation der Klasse T-Stück beachtet werden. Durch die geringe Anzahl an Objekten ist eine geringere Leistungsfähigkeit für diese Klasse festgestellt worden. Die vollständige Auflistung aller Ergebnisse Inklusive der AP für jede Klasse findet sich im Anhang in Abbildung 45.

Consensus Score

Der Consensus Score bestimmt den Informationsgehalt einer Punktwolke durch ein Ensemble. Hierbei werden die Erkennungen mehrerer Modelle miteinander verglichen. Starke Abweichungen in der Lokalisierung (ohne Berücksichtigung der Klassenverteilung) deuten auf einen großen

Informationsgehalt der Daten hin. Nachfolgend werden die Ergebnisse der experimentellen Untersuchung vorgestellt.

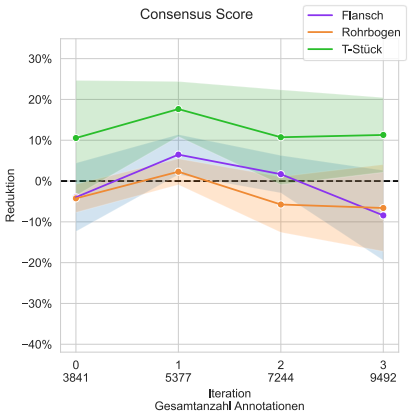


Abbildung 19: Ergebnisse des Consensus Score

Abbildung 19 zeigt die Ergebnisse des Trainings mit dem Consensus Score unter Berücksichtigung der einzelnen Klasse:

Flansch: Die Datenmenge variiert zwischen - 8% und + 6% über die Iterationen, mit Standardabweichungen von 5% bis 11%. Eine eindeutige Optimierung des Trainingseffekts ist nicht festzustellen, wobei die letzte Iteration mit - 8% die stärkste Reduktion aufweist, die sich noch innerhalb einer Standartabweichung befindet.

Rohrbogen: In der ersten Iteration ist kein Unterschied zur Basisline feststellbar. Die zweite Iteration weist eine Reduktion um - 6% auf, wobei die Standardabweichung auf 7% ansteigt. In der dritten Iteration setzt sich dieser Trend fort, mit einer Verringerung des Datenbedarfs um durchschnittlich - 7% bei gleichzeitiger Zunahme der Variabilität (Standardabweichung 11%). Für die Klasse Rohrbogen kann eine zunehmende, wenn auch moderate, Optimierung des Datenbedarfs erzielt werden.

T-Stück: Über die drei Iterationen hinweg schwankt die erforderliche Datenmenge zwischen +11% und +18%, eine Tendenz zu +10% zu erkennen ist. Die Standardabweichung liegt zwischen 7% und 12%. Hierdurch ergibt sich, dass mit dem Consensus Score mehr Datensätze gegenüber einer zufälligen Baseline erforderlich sind. Für die Klasse T-Stück führt der Algorithmus zu keiner Optimierung des Trainingseffekts.

Bei der Auswertung der experimentellen Ergebnisse wurde festgestellt, dass der Consensus Score keine Verbesserung gegenüber der zufälligen Baseline aufweist. Darüber hinaus ist für die Klasse T-Stück sogar eine Verschlechterung der Performance aufgetreten.

RoI Matching

Das RoI Matching bestimmt den Informationsgehalt einer Punktwolke durch ein Ensemble. Hierbei werden die Erkennungen mehrerer Modelle miteinander verglichen. Detektieren mehrere Modelle ein Objekt an derselben Position wird die Entropie der Klassenverteilung als Informationsgehalt der Daten verwendet.

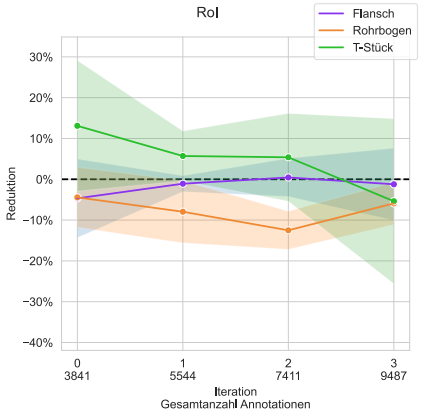


Abbildung 20: Ergebnisse des RoI Matching

Flansch: Über alle drei Iterationen hinweg zeigen sich nur marginale Veränderungen des Datenbedarfs. Die Mittelwerte bewegen sich im Bereich von - 1% bis 0%, was deutlich unterhalb der festgelegten Signifikanzschwelle von 10% liegt. Es ist zu beobachten, dass die Standardabweichung über die Iterationen auf bis zu 9% zunimmt, was auf eine steigende Variabilität in den Ergebnissen hindeutet. Trotz dieser Zunahme bleiben die Schwankungen insgesamt moderat. Somit ergibt sich für die Klasse Flansch keine Veränderung des Datenbedarfs für den RoI-Algorithmus.

Rohrbogen: In der ersten Iteration zeigt sich eine nicht signifikante Reduktion des Datenbedarfs um - 8%. Die zweite Iteration weist mit einem Mittelwert von - 13% eine signifikante Verbesserung auf, da hier die 10%-Schwelle überschritten wird. In der dritten Iteration ist wiederum eine nicht signifikante Reduktion um - 6% zu verzeichnen. Somit lässt sich eine geringe Datenreduktion für die Klasse Rohrbogen feststellen.

T-Stück: In der ersten und zweiten Iteration zeigt sich eine geringfügige Erhöhung des Datenbedarfs ca.+ 6%, die jedoch als nicht signifikant zu bewerten ist. In der dritten Iteration ist eine Reduktion des Datenbedarfs um - 5% zu beobachten, die ebenfalls innerhalb des als nicht signifikant definierten Bereichs liegt. Die Standardabweichung nimmt über die Iterationen von 6% über 11% bis hin zu 20% zu. Hierdurch verringert sich die Aussagekraft der Tendenz zur Datenreduktion in der letzten Iteration.

Localization Stability

Um die Unsicherheit bei der Erkennung zu bestimmen, wird die Erkennung der originalen Daten mit der Erkennung von modifizierten Daten abgeglichen. Als Modifikation wird Rauschen zu den Daten hinzugefügt.

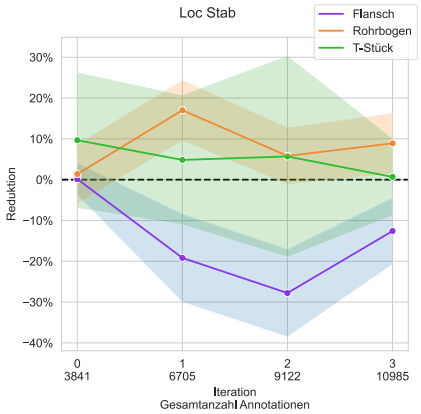


Abbildung 21: Ergebnisse der Localization Stability

Flansch: In der ersten Iteration ist eine signifikante Reduktion des Datenbedarfs um - 19% zu verzeichnen. Diese Verbesserung intensiviert sich in der zweiten Iteration auf - 28%. In der dritten Iteration fällt die Reduktion mit - 13% geringer aus, überschreitet jedoch weiterhin die Signifikanzschwelle von 10%. Die Standardabweichungen bleiben über die ersten beiden Iterationen konstant bei 11% und sinken in der dritten Iteration leicht auf 8%, was auf eine geringfügig höhere Stabilität der Ergebnisse hindeutet.

Der Localization Stability-Algorithmus kann für die Klasse Flansch eine durchgängig signifikante Optimierung des Datenbedarfs erzielen. Wenngleich in der dritten Iteration ein leichter Rückgang zu beobachten ist, bleibt die Verbesserung dennoch auf einem signifikanten Niveau.

Rohrbogen: Die erste Iteration weist eine signifikante Erhöhung des Datenbedarfs um - 17% auf. Diese unerwünschte Zunahme überschreitet deutlich den als signifikant definierten Schwellenwert von 10%. In den darauffolgenden Iterationen fällt die Zunahme des Datenbedarfs mit + 6% bis + 9% innerhalb des nicht signifikanten Bereichs. Es ist hervorzuheben, dass die Standardabweichung über alle Iterationen hinweg konstant bei 7% liegt, was auf eine gleichbleibende Streuung der Messwerte hindeutet.

T-Stück: Die Mittlere benötigte Datenmenge bewegt sich im Bereich von + 1% bis + 6% für alle Iterationen, was unter der 10%-Schwelle liegt. Bemerkenswert ist die Entwicklung der Standardabweichung. Diese steigt auf bis 25% in der zweiten Iteration an, bevor diese in der dritten Iteration auf 9% abfällt. In der zweiten Iteration treten Extremwerte von – 32% und + 37% auf, was auf eine starke Abhängigkeit von der Auswahl der initialen Trainingsdaten hindeutet. Somit kann für die Effektivität des Algorithmus keine Aussage für diese Klasse getroffen werden.

BLAD

Der BLAD Algorithmus bestimmt die Unsicherheit in der Erkennung. Ermittelt wird diese durch die Abweichung der Erkennung zwischen den originalen Daten und modifizierten Versionen. Als Modifikationen werden Spiegelungen eingesetzt. Eine große Unsicherheit soll informative Daten bestimmen.

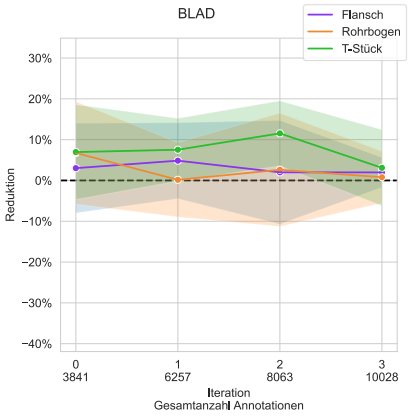


Abbildung 22: Ergebnisse von BLAD

Flansch: Die Auswertung der Daten zeigt, dass über drei Iterationen hinweg keine signifikante Reduktion des Datenbedarfs erzielt wurde. Es ist festzustellen, dass die mittleren Veränderungen mit Werten von 5%, 2% und 4% innerhalb des als nicht signifikant definierten Bereichs von ±10% liegen.

Rohrbogen: In der Analyse der Daten für die Klasse "Rohrbogen" zeigt sich keine signifikante Reduktion des Datenbedarfs über die drei Iterationen hinweg. Die mittleren Veränderungen (0%, 3%, 4%) liegen innerhalb des als nicht signifikant definierten Bereichs von ±10%. Die zunehmende Standardabweichung in den ersten beiden Iterationen (von 9% auf 14%) deuten auf eine gewisse Instabilität des Verfahrens hin.

T-Stück: Es zeigt sich keine signifikante Reduktion des Datenbedarfs über die drei Iterationen. Die mittleren Veränderungen (8%, 12%, 6%) liegen größtenteils im nicht-signifikanten Bereich von ±10%. Lediglich in der zweiten Iteration ist mit 12% eine leichte, unerwünschte Zunahme zu verzeichnen.

Core Set

Der Core-Set Algorithmus basiert ausschließlich auf dem Prinzip der Diversitätsmaximierung. Hierzu werden die Feature Maps der Datenpunkte ohne Annotationen mit denen verglichen die bereits annotiert wurden. Durch ein greedy farthest first sampling werden nur diejenigen Daten zur Annotation ausgewählt, welche sich am meisten von den bereits annotierten Daten unterscheiden.

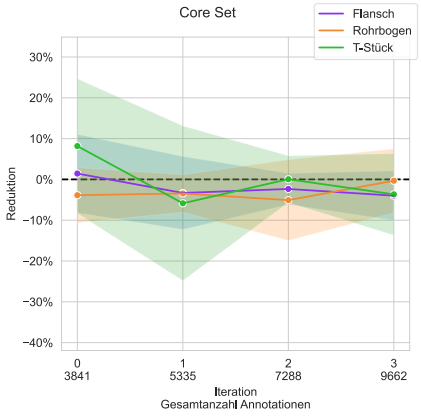


Abbildung 23: Ergebnisse von Core Set

Flansch: Über alle drei Iterationen hinweg zeigen sich geringfügige Reduktionen des Datenbedarfs. In der ersten Iteration liegt diese im Bereich von - 2% bis - 4%. Es ist hervorzuheben, dass keine dieser Veränderungen das festgelegte Signifikanzniveau von 10% erreicht. Die Standardabweichungen variieren zwischen 4% und 9%, wobei die höchste Streuung in der ersten Iteration zu verzeichnen ist. Hier kam es zu einer Reduktion von – 19% für eine Partition der Kreuzvalidierung. Da diese für die anderen Partitionen nicht auftritt, wird dies als Ausreißer betrachtet. In den nachfolgenden Iterationen stabilisieren sich die Werte auf einem niedrigeren Niveau.

Rohrbogen: In der ersten Iteration ist eine Reduktion des Datenbedarfs um - 3% zu verzeichnen. Diese Verringerung intensiviert sich in der zweiten Iteration auf - 5%. In der dritten Iteration zeigt sich praktisch keine Veränderung (0%). Alle diese Werte liegen innerhalb des als nicht signifikant definierten Bereichs von - 10% bis + 10%.

Die Standardabweichungen variieren zwischen 5% und 10%, wobei die höchste Streuung in der zweiten Iteration auftritt. In dieser konnte für zwei Partitionen eine Reduktion von mehr als - 10% erzielt werden. Dies deutet auf eine gewisse Instabilität der Ergebnisse hin.

T-Stück: Die mittlere Datenreduktion liegt zwischen 0% und – 6%. Auffällig ist die hohe Standardabweichung von 19% in der ersten Iteration, die in den folgenden Iterationen auf 6% bzw. 10% absinkt. Dies deutet auf eine initial höhere Variabilität der Ergebnisse hin, die sich in den späteren Durchläufen stabilisiert.

PPAL

Der PPAL-Algorithmus bestimmt den Informationsgehalt der Daten durch eine Kombination eines entropiebasierten Unsicherheitsmaßes und einem anschließenden Diversitätssampling. Zur Bestimmung der Diversität werden die Features verwendet, die sich innerhalb der erkannten BB befinden. Das

Ähnlichkeitsmaß führt einen paarweisen vergleich der Objektfeatures zwischen zwei Punktwolken durch, wodurch sich eine lokale Betrachtung des Kontexts ergibt.

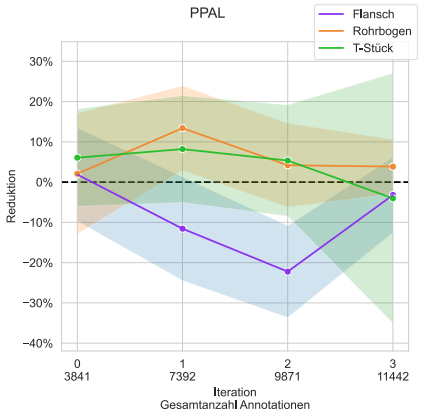


Abbildung 24: Ergebnisse von PPAL

Flansch: Das festgelegte Signifikanzniveau von 10% konnte in zwei Iterationen überschritten werden (- 12% und - 22%). Die dritte Iteration weist mit - 3% keine bedeutsame Änderung auf.

Rohrbogen: In der ersten Iteration zeigt sich eine Erhöhung des Datenbedarfs um + 13%. In den darauffolgenden Iterationen fällt die Erhöhung auf + 4% ab wobei sich die Standardabweichung von initial + 11% auf + 7% verringert.

T-Stück: In keiner Iteration konnte eine mittlere Veränderung größer als das Signifikanzniveau beobachtet werden. Die Standardabweichung ist für die ersten zwei Iterationen bei 13%, wobei diese in der letzten Iteration auf 31% ansteigt.

Der PPAL-Algorithmus konnte die benötigte Datenmenge für die Klasse Flansch reduzieren, was mit einer moderaten Zunahme an Annotationen der Klasse Rohrbogen einhergeht. Die Anzahl an angefragten Annotationen ist in jeder Iteration größer als die anderen Algorithmen.

PPAL mit globalem Kontext

PPAL global entsteht durch das Ersetzen der Diversitätsbestimmung mit dem klassischen Core-Set-Algorithmus. Der Core-Set Algorithmus nutzt sämtliche Features der Eingabedaten, wodurch diese Modifikation einen globalen Kontext betrachtet. Hierdurch soll überprüft werden, wie sich der betrachtete Kontext auf die Leistungsfähigkeit auswirkt.

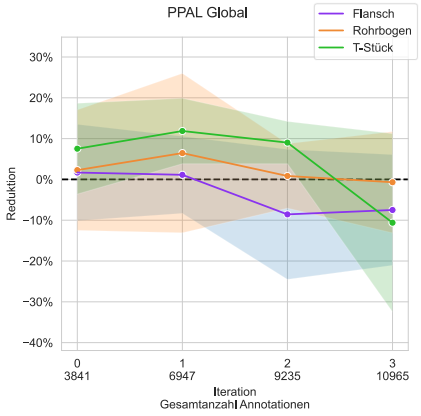


Abbildung 25: Ergebnisse von PPAL mit Globalem Kontext

Flansch: Die erste Iteration weist eine vernachlässigbare Veränderung des Datenbedarfs von + 1% auf. In der zweiten Iteration ist eine Reduktion von - 9% zu verzeichnen, die sich an der Grenze zur Signifikanz bewegt. Dies geht mit einer deutlichen Zunahme der Standardabweichung von 9% auf 16% einher, welche von Extremwerten von – 28% und 6% verursacht wird.

Rohrbogen: Für keine der Iterationen konnte eine signifikante Datenreduktion erreicht werden. Nennenswert ist die starke Standartabweichung von 20% in der ersten Iteration, welche sich auf 8% verringert. Diese rührt von mehreren Extremwerten von circa ± 20%.

T-Stück: In der ersten Iteration zeigt sich eine Erhöhung des Datenbedarfs um + 12%, welche knapp über dem definierten Signifikanzniveau von 10% liegt. Die zweite Iteration weist mit 9% eine nicht signifikante Zunahme auf. In der dritten Iteration ist hingegen eine signifikante Reduktion um - 11% zu verzeichnen. Während die Standardabweichung für die ersten beiden Iterationen geringe Werte von 8% bzw. 5% aufweisen, steigt die Standardabweichung in der dritten Iteration auf 22% an. Die starke Streuung führt zu einer eingeschränkten Aussagekraft der Ergebnisse.

Localization Stability mit Core-Set

Location Stability Core-Set realisiert die Integration diversitätssteigernder Komponenten in vorhersagebasierte Methoden und wurde durch die Kombination der Location Stability und Core-Set realisiert. Der implementierte Selektionsprozess erfolgt zweistufig: Zunächst extrahiert die Location Stability eine Vorauswahl an Punktwolken, aus denen der Core-Set-Algorithmus die finalen Samples selektiert. Diese Vorgehensweise zielt auf eine Optimierung des Informationsgehalts und der Diversität der ausgewählten Daten ab.

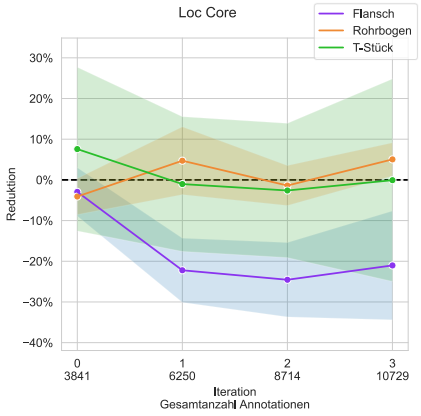


Abbildung 26: Ergebnisse der Localization Stability mit Core-Set

Flansch: Über alle drei Iterationen hinweg zeigt sich eine signifikante Reduktion des Datenbedarfs. Die maximale Datenreduktion von - 25% findet in der zweiten statt und fällt geringfügig in der dritten Iteration auf - 21%.

Die Standardabweichung weist eine leicht steigende Tendenz auf, beginnend bei 8% bis hin zu 13% in der dritten. Dennoch ist die Datenreduktion mehr als eine Standardabweichung von dem Signifikanzniveau entfernt.

Rohrbogen: Für diese Klasse konnte keine signifikante Abweichung von der zufälligen Datenauswahl festgestellt werden. Die Standardabweichungen zeigen eine abnehmende Tendenz über die Iterationen hinweg, was auf eine zunehmende Konsistenz der Ergebnisse hindeutet.

T-Stück: Für keine der Iterationen konnte eine signifikante Datenreduktion erreicht werden. Die Standardabweichung vergrößert sich von 16% in der ersten auf 25% in der vierten Iteration.

CRB

Der CRB-Algorithmus wurde speziell für 3D-Punktwolken entwickelt. Dieser soll den Vergleich der AL-Algorithmen aus der 2D Objekterkennung ermöglichen, welche im Rahmen dieser Arbeit auf die Anwendung in Punktwolken übertragen worden sind. Die Funktionsweise des Algorithmus besteht aus zwei Schritten. In dem ersten Schritt werden die Punktwolken anhand der erkannten Objektklassen gefiltert. Das Ziel hierbei ist, dass die Klassen der ausgewählten Punktwolken einer Gleichverteilung folgen. Hierdurch soll die Überrepräsentation von Klassen vermieden werden. Im zweiten Schritt findet ein Diversitätssampling durch den Core-Set Algorithmus statt. Das Merkmal, welches zur Bestimmung der Ähnlichkeit verwendet wird, sind die Gradienten. Zur Bestimmung der Gradienten werden hypothetische Annotationen erzeugt. Diese entstehen dadurch, dass mehrere Durchgänge mit Dropout durchgeführt werden. Die hierdurch erkannten Bounding Boxes werden aggregiert und als hypothetische Annotationen für die Bestimmung der Gradienten verwendet.

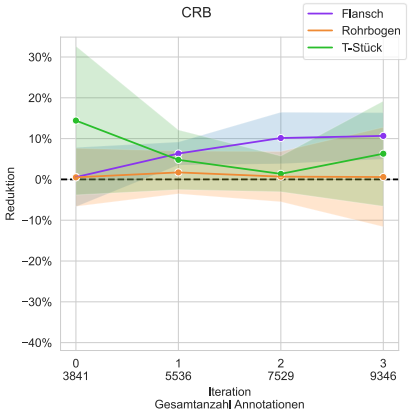


Abbildung 27: Ergebnisse von CRB

Flansch: In der ersten Iteration zeigt sich eine geringfügige Erhöhung des Datenbedarfs um 6%, die bis zur dritten Iteration auf 11% ansteigt, die als signifikant zu bewerten ist. Dies geht mit einer leichten Steigerung der Standardabweichung von 3% auf 6% einher.

Rohrbogen: In allen drei Iterationen zeigt sich im Durchschnitt keine Veränderung zu dem Referenzwert. Die zunehmende Standardabweichung deutet auf eine steigende Instabilität im AL-Prozess hin. Diese ist durch zwei Ausreißer verursacht worden. In der letzten Iteration betragen diese – 14% und 20% für zwei Partitionen der Kreuzvalidierung, während die verbleibenden Partitionen keine Veränderung zum Referenzwert aufweisen.

T-Stück: Über die drei Iterationen hinweg schwankt der Datenbedarf zwischen 1% und 6%, wobei keine signifikanten Veränderungen zu beobachten sind. Die Standardabweichung variiert von 4% bis 13%, mit einem deutlichen Anstieg in der letzten Iteration.

6 Interpretation

Die nachfolgende Analyse widmet sich der Interpretation der zuvor präsentierten Resultate. Dies wird nur für ausgewählte Algorithmen durchgeführt, bei denen ausreichend Anhaltspunkte für eine Interpretation zur Verfügung stehen. Im Fokus steht hierbei die Untersuchung der selektierten Klassen sowie die Reduktion der benötigten Datenmenge in Relation zu der Anzahl annotierter Objekte pro Klasse. Hierbei wird untersucht, inwiefern die Selektion spezifischer Klassen durch den Active-Learning-Algorithmus die Gesamtperformanz der Algorithmen beeinflusst hat. Es soll festgestellt werden, ob bestimmte Klassen überproportional häufig zur Annotation angefragt wurden und welche Implikationen sich daraus ergeben.

Im Zuge der Interpretation wird die Menge an Objekten in den generierten Trainingsdatensätzen betrachtet. Diese wird als prozentuale Abweichung zu dem zufälligen Referenzalgorithmus angegeben. Damit diese relative Angabe in Bezug gesetzt werden kann, ist in Abbildung 28 die absolute Menge an Annotationen für den Referenzalgorithmus dargestellt. Die vollständige Auflistung der Annotationen für jeden Algorithmus befindet sich im Anhang in Abbildung 47.

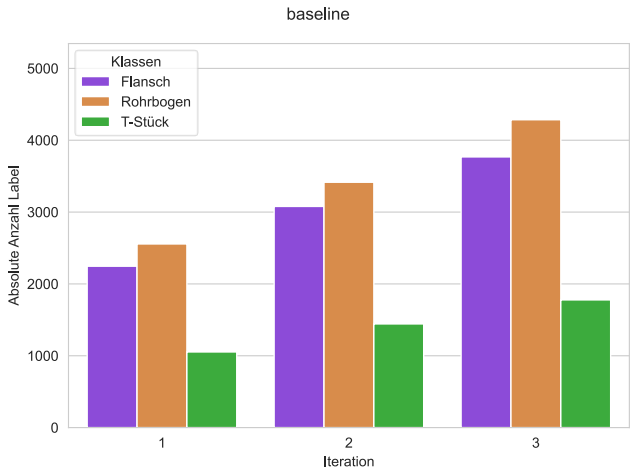


Abbildung 28: Absolute Anzahl an Annotationen der Baseline

Core Set

Bei der Übertragung des Core-Set-Algorithmus auf die Anwendung in Punktwolken ist festgestellt worden, dass die Leistungsfähigkeit des Algorithmus im Vergleich zur Anwendung auf Bilddaten signifikant reduziert ist. Im Folgenden werden potenzielle Ursachen für dieses Ergebnis erörtert.

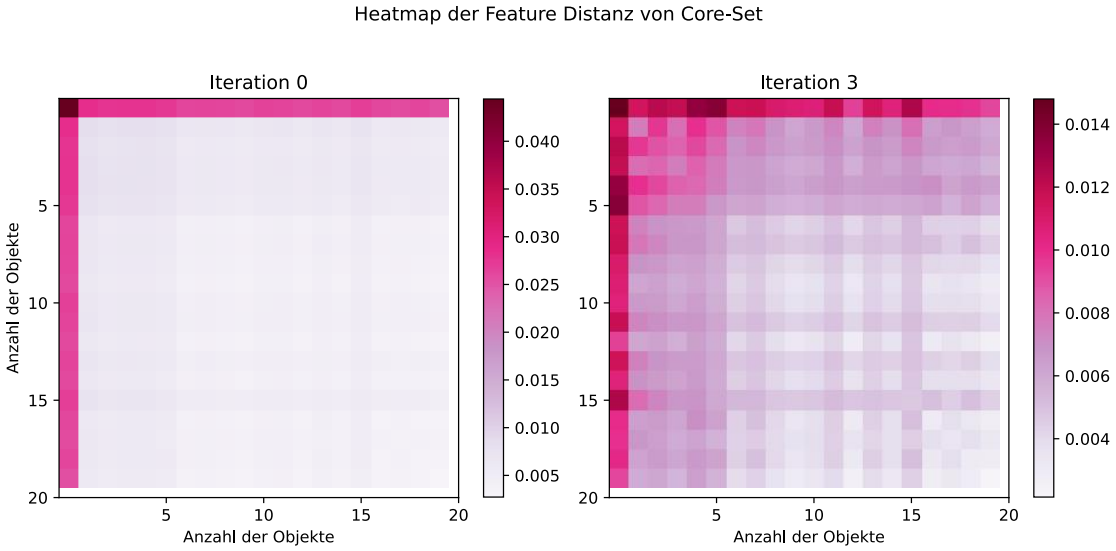


Abbildung 29: Heatmap der Feature Distanz von Core-Set

Der Core-Set Algorithmus basiert auf der Maximierung der Diversität. Hier wird durch Greedy Farthest First Clustering der Feature Maps durchgeführt. Abbildung 29 zeigt eine Heatmap der durchschnittlichen Feature Distanz zwischen Punktwolken. Auf der x und y-Achse ist die Anzahl an enthaltenen Objekten der Punktwolken abgebildet. Diese ist dargestellt für die Ausgangslage (links) und für die letzte Iteration (rechts).

Hierbei kann festgestellt werden, dass die Anwendung der Feature-Distanz in den frühen Iterationsphasen für Punktwolken mit einer Mehrzahl von Objekten nicht die erwartete Effektivität aufweist. Die Distanz zwischen den Feature Maps von Punktwolken mit mehreren Objekten ist klein im Verhältnis zu der Distanz von Punktwolken mit vielen Objekten. Zu erwarten wäre eine stärkere Ausprägung der Distanz zwischen Punktwolken mit mehreren Objekten. Zudem zeigt sich eine große Distanz zwischen Punktwolken mit vielen Objekten und Punktwolken mit nur einem Objekt. Ursächlich hierfür ist vermutlich der Informationsverlust durch das Aggregieren der Features. Werden Features vieler Objekte und somit vieler Punkte aggregiert, konvergiert der resultierende Feature-Vektor stärker gegen den Durchschnitt als der Feature Vektor eines Objektes. Folglich ist die Distanz zwischen Feature-Vektor von Punktwolken mit vielen Objekten kleiner als die Distanz von Feature-Vektoren von Punktwolken mit wenigen Objekten. Hierdurch bevorzugt das Greedy Farthest First Clustering Punktwolken mit nur wenigen Objekten.

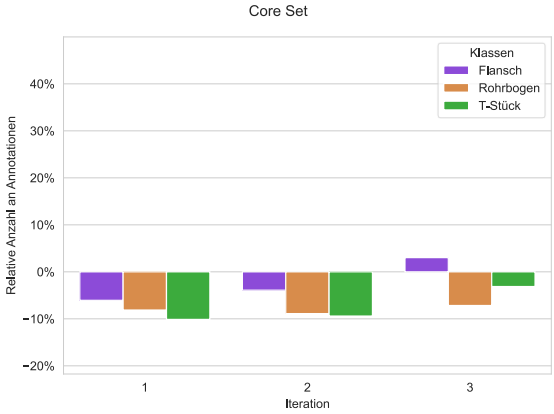


Abbildung 30: Anzahl der Annotationen von Core-Set, relativ zur Baseline

Dies kann in Abbildung 30 beobachtet werden, welche die verwendeten Annotationen relativ zur zufälligen Datenauswahl zeigt. Im Verlauf des AL-Prozesses kommt es zu einer Sättigung. Sobald der Trainingsdatensatz aus genügend Punktwolken mit wenigen Objekten besteht, steigt die Bedeutung der kleineren Distanzen und es werden vermehrt Punktwolken mit einer größeren Anzahl an Objekten ausgewählt.

Localization Stability

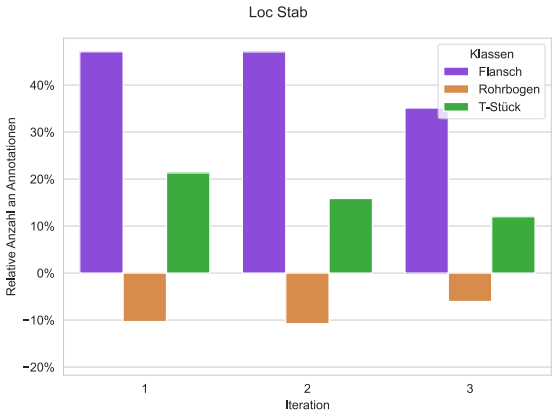


Abbildung 31: Relative Anzahl an Annotationen der Localization Stability

Die gute Leistungsfähigkeit bei der Reduktion der benötigten Datenmenge der Localization Stability motiviert für eine genauere Analyse. Abbildung 31 zeigt die Differenz der Annotationen im erzeugten Trainingsdatensatz zu dem Basiswert pro Klasse für jede Iteration. Es ist deutlich zu erkennen, dass Objekte der Klasse Flansch in jeder Iteration überrepräsentiert sind. Aus diesem Grund soll die Leistungsfähigkeit pro Klasse analysiert werden. Es gilt zu evaluieren, ob der implementierte Algorithmus tatsächlich informative Objekte selektiert oder ob eine systematische Bevorzugung der Klasse Flansch vorliegt.

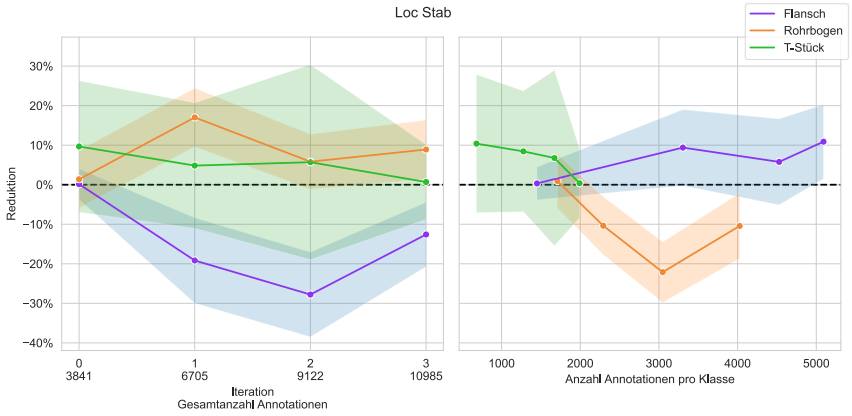


Abbildung 32: Reduktion der Localization Stability relativ zu den Annotationen pro Klasse

Abbildung 32 zeigt auf der rechten Seite die Reduktion im Verhältnis zu der Anzahl an Annotationen pro Klasse. Hierdurch soll festgestellt werden, ob der Algorithmus trotz der Überrepräsentation der Klasse Flansch informative Objekte der anderen Klassen identifizieren konnte. Zum Vergleich ist auf der linken Seite die Datenreduktion im Verhältnis zu der Gesamtanzahl der Annotationen aus Abbildung 24 erneut abgebildet. Durch diese Betrachtungsweise ändert sich die Interpretation. Für die Klasse Flansch wird ersichtlich, dass die Anzahl der benötigten Flansche nicht reduziert werden konnte. Stattdessen werden Punktwolken bevorzugt, welche viele Objekte der Klasse Flansch beinhalten. Diese Entwicklung stoppte erst in der dritten Iteration, als die Punktwolken mit einer überproportional großen Anzahl an Flanschen ausgeschöpft wurde. Die Ursache für die festgestellten Effekte ist mit hoher Wahrscheinlichkeit in der spezifischen Geometrie der Flansche zu verorten. Aufgrund der flachen Konstruktionsweise der Flansche ist bereits eine geringfügige Translation entlang der Rotationsachse ausreichend, um eine deutliche Reduktion der IoU zu bewirken. Da die IoU das zentrale Evaluationskriterium der Localization Stability darstellt, ergibt sich die Schlussfolgerung, dass verstärkt Punktwolken mit einer hohen Dichte an Flanschen für die Annotation selektiert werden.

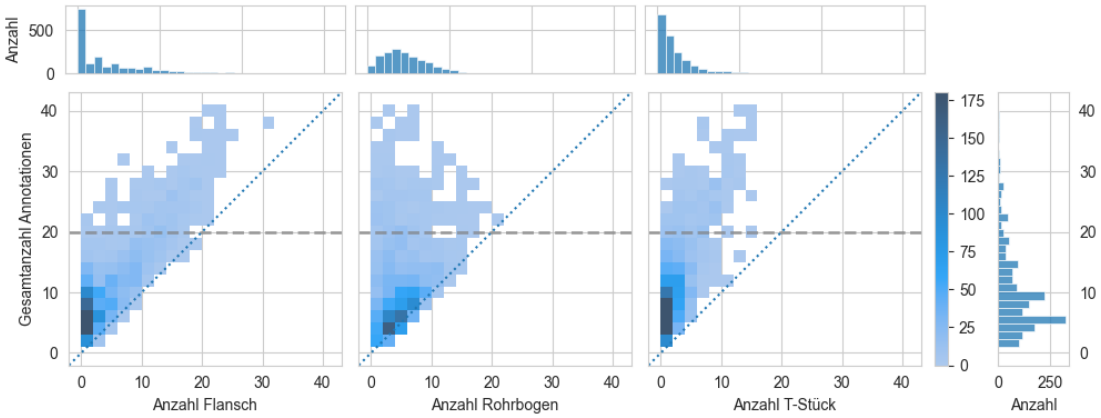


Abbildung 33: Verteilung der Anzahl an Annotationen in den Punktwolken

Hinzu kommt, dass in Punktwolken mit vielen Objekten Flansche überrepräsentiert sind. Abbildung 33 zeigt die Verteilung der Objektanzahl der einzelnen Klassen in dem Verhältnis zu der Gesamtanzahl an Objekten in den einzelnen Punktwolken. Die graue gestrichelte Linie zeigt das 90% Quantil der Gesamtanzahl der Objekte pro Punktwolke an. Die Histogramme in der oberen Zeile und in der rechten Spalte repräsentieren die Randverteilungen. Die blaue gepunktete Line zeigt Punktwolken an, welche ausschließlich Objekte einer Klasse beinhalten. Hier lässt sich erkennen, dass in Punktwolken mit vielen Objekten Flansche überrepräsentiert und in Punktwolken mit wenigen Objekten unterrepräsentiert sind. Durch die Präferenz der Localization Stability Punktwolken mit einer hohen Dichte an Flanschen zu bevorzugen und der Überrepräsentation von Flanschen in großen Punktwolken führt dies zu einer überproportionalen Repräsentation der Flanschobjekte im generierten Trainingsdatensatz.

Für die Klasse der Rohrbögen hingegen ist ein gegensätzlicher Trend zu beobachten. Hier erfolgt eine mittlere Reduktion um - 20% in der zweiten Iteration bei der Betrachtung der Reduktion im Verhältnis zu den Objekten pro Klasse. Demnach konnte der Algorithmus trotz der Bevorzugung von Punktwolken mit einer großen Anzahl an Flanschen informative Objekte für diese Klasse bestimmen. Inwieweit die Auswahl der Objekte der Klasse Flansch zu der gesteigerten AP pro annotiertem Rohrbogen beiträgt, muss weiter untersucht werden.

Localization Stability mit Core-Set

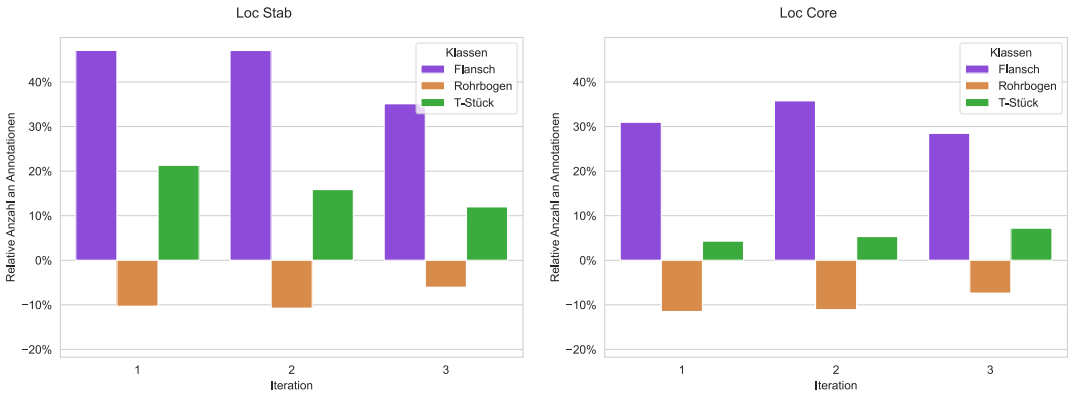


Abbildung 34: Relative Anzahl an Annotationen der Localization Stability Core-Set

Bei der Betrachtung von Core-Set im Verhältnis zu der Gesamtanzahl an Annotationen in Abbildung 30 konnte beobachtet werden, dass der Algorithmus tendenziell weniger Objekte pro Punktwolke im Vergleich zu der zufälligen Datenauswahl bevorzugt. Der prozentuale Unterschied an Annotationen zu der zufälligen Baseline ist in Abbildung 34 dargestellt. Zur Veranschaulichung wurden die Werte der unmodifizierten Localization Stability aus Abbildung 31 auf der linken Seite erneut dargestellt. Die Abschwächung der Überrepräsentation der Klasse Flansch kann in jeder Iteration beobachtet werden.

Dies führt zu einer Auswirkung auf die Average Precision pro annotiertem Objekt der Klassen Flansch und T-Stück in Abbildung 35. Die Effektstärke ist nicht ausreichend, um eine Steigerung der Average Precision pro Flansch gegenüber der zufälligen Datenauswahl zu erreichen. Dennoch konnte die Zunahme an nötigen Annotationen, die für den unmodifizierten Algorithmus aufgetreten ist, beseitigt werden.

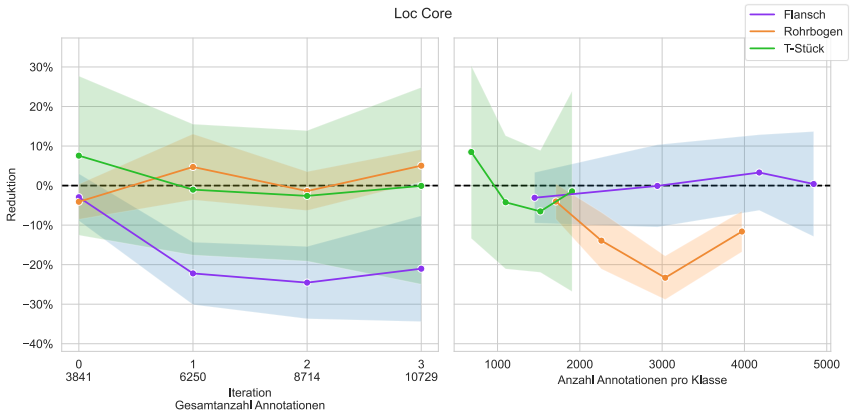


Abbildung 35: Ergebnisse der Localization Stability mit Core-Set relativ zu den Annotationen pro Klasse

Zum Prüfen der These, dass die Überrepräsentation der Klasse Flansch in großen Punktwolken die Datenauswahl des AL-Algorithmus beeinflusst, wurde ein weiterer Versuch durchgeführt. Hierfür wurde der originale Localization Stability Algorithmus modifiziert. Hierbei wurde die Metrik (Formel 6) zur Bestimmung des Informationsgehalts nach der Anzahl der erkannten Objekte B gewichtet mit $min\left(exp\left(-\frac{B-T}{\lambda}\right), 1\right)$, wobei der Parameter T die Anzahl an Bounding Boxen steuert und λ die Gewichtung. Zur Bestimmung der optimalen Parameterwerte wurden empirische Untersuchungen durchgeführt, wobei die Parameter $T = 15$ und $\lambda = 10$ die besten Ergebnisse lieferten.

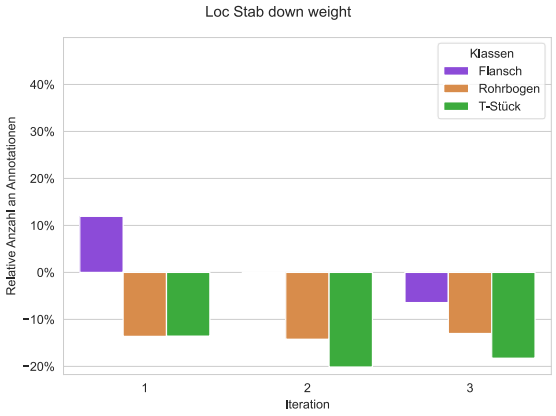


Abbildung 36: Relative Anzahl an Annotationen der gewichteten Localization Stability

Der Einfluss dieser Modifikation auf die angefragten Annotationen ist in Abbildung 36 dargestellt. Das überproportionale Anfragen der Klasse Flansch konnte hierdurch beseitigt werden, wobei die Gesamtanzahl an angefragten Annotationen zurückgegangen ist.

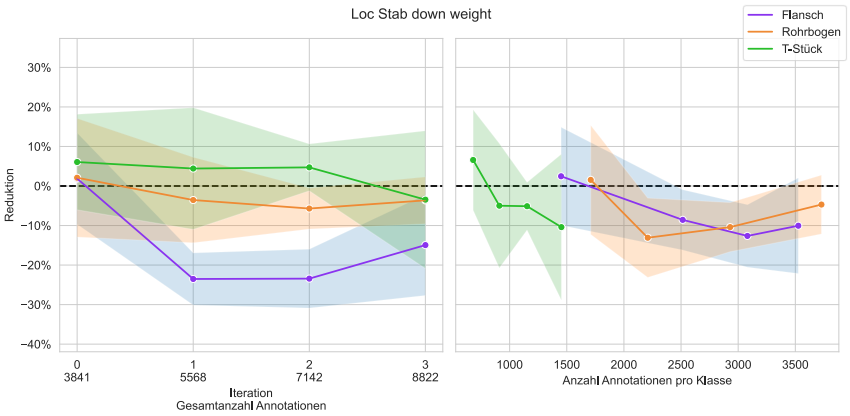


Abbildung 37: Ergebnisse der gewichteten Localization Stability relativ zu den Annotationen pro Klasse

Abbildung 37 zeigt die Ergebnisse der gewichteten Localization Stability für die Gesamtanzahl an Annotationen (links) und die Anzahl der Annotationen pro Klasse (rechts). Hier lässt sich erkennen, dass die Gewichtung der Metrik zu einer verbesserten Auswahl der Klassen Flansch und T-Stück geführt hat. Dies geht jedoch mit einer Verschlechterung der Resultate für die Klasse Rohrbogen einher. Hier sinkt die maximale Reduktion auf – 13% (– 22% in der unmodifizierten Variante) und fällt in der dritten Iteration wieder unter die Signifikanzschwelle von ± 10%. Dies ist der einzige Algorithmus, welcher bei einer Betrachtung der Objektanzahl pro Klasse eine Reduktion für mehr als eine Klasse erzielen konnte.

PPAL

Durch den lokalen Vergleich von Features wurde für den PPAL-Algorithmus eine hohe Leistungsfähigkeit erwartet. Diese konnte sich teilweise bestätigen, wobei aber andere Algorithmen eine größere Reduktion erzielt haben. Vergleichbar zu der Localization Stability kann in Abbildung 38 festgestellt werden, dass die Klasse Flansch im erzeugten Trainingsdatensatz überrepräsentiert ist. Generell kann festgestellt werden, dass der PPAL-Algorithmus die meisten Annotationen anfragt.

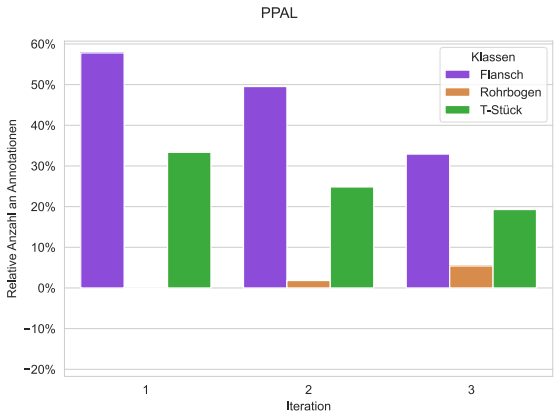


Abbildung 38: Relative Anzahl an Annotationen des PPAL-Algorithmus

Abbildung 39 zeigt die Reduktion im Verhältnis zu der Anzahl der Objekte pro Klasse. Ähnlich zu der Localization Stability zeigt sich hier, dass die Überrepräsentation von Klassen einen negativen Einfluss auf den AL-Prozess bewirkt.

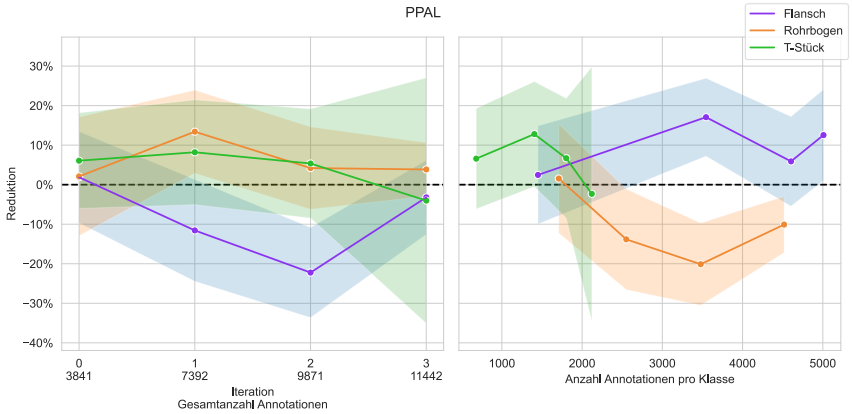


Abbildung 39: Ergebnisse des PPAL-Algorithmus relativ zu den Annotationen pro Klasse

Die Tendenz zu Punktwolken mit vielen Objekten wird in der Bestimmung der Ähnlichkeit vermutet. Diese ergibt sich als Maximum der paarweisen Distanzen zwischen den Objektfeatures. Hierdurch steigt in Punktwolken mit vielen Objekten die Wahrscheinlichkeit, dass sich Objekte voneinander unterscheiden und somit eine große Distanz besitzen.

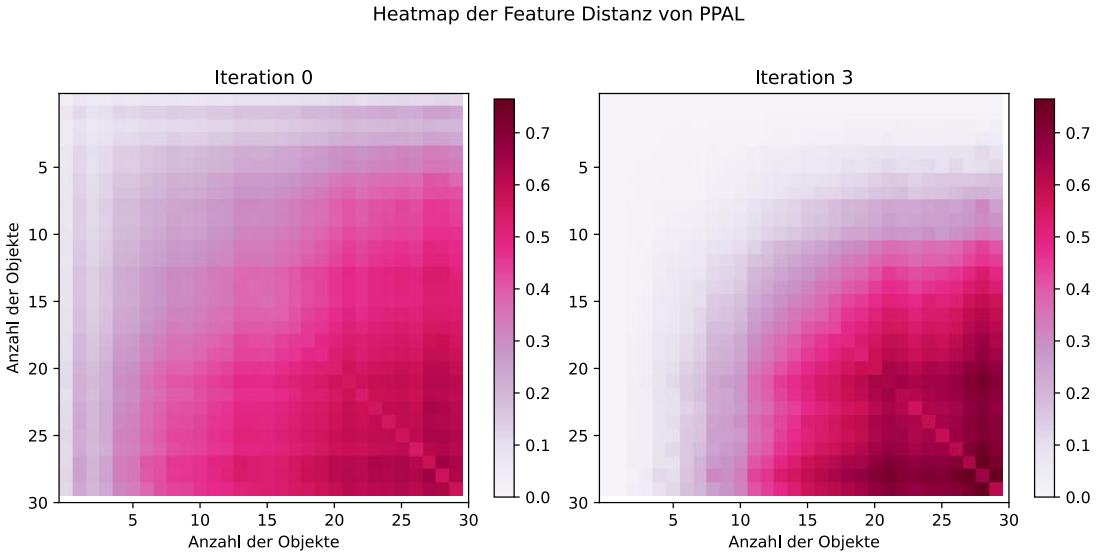


Abbildung 40: Heatmap der Feature Distanz von PPAL

Abbildung 40 zeigt eine Heatmap der durchschnittlichen Feature Distanz zwischen Punktwolken. Auf der x und y-Achse sind die Anzahl an enthaltenen Objekten der Punktwolken abgebildet. Diese ist dargestellt für die Ausgangslage (links) und für die letzte Iteration (rechts). Hier ist deutlich zu erkennen, dass bereits in der Ausgangslage die Distanz zwischen Punktwolken mit vielen Objekten am größten ist. Die Ursache für die Verschärfung des Phänomens am Ende des AL-Prozesses bleibt offen. Als mögliche Erklärung hierfür kann die zunehmende Aussagekraft der Features durch den größeren Trainingsdatensatz am Ende des AL-Prozesses in Betracht gezogen werden. Hierdurch können feinere Unterschiede zwischen den Objekten im Feature-Raum abgebildet werden, was eine Vergrößerung der Distanz zur Folge hat.

PPAL Global

Die Hypothese, dass die Bestimmung der Ähnlichkeit für die überproportionale Anfrage von Punktwolken mit einer hohen Objektdichte verantwortlich ist, kann durch die Betrachtung der modifizierten Version von PPAL mit globalem Kontext überprüft werden. Wie bereits dargestellt wurde, zeigt der Core-Set Algorithmus eine Neigung für Punktwolken mit wenigen Objekten. Folglich sollte der Austausch der lokalen Kontextbetrachtung zu der globalen durch den Core-Set Algorithmus zu einer Abschwächung des Effekts führen. Abbildung 41 zeigt die relative Anzahl der Annotationen im Datensatz im Vergleich zu der Baseline. Zur Veranschaulichung sind auf der linken Seite die Werte des originalen PPAL-Algorithmus und auf der rechten Seite der modifizierten Version abgebildet.

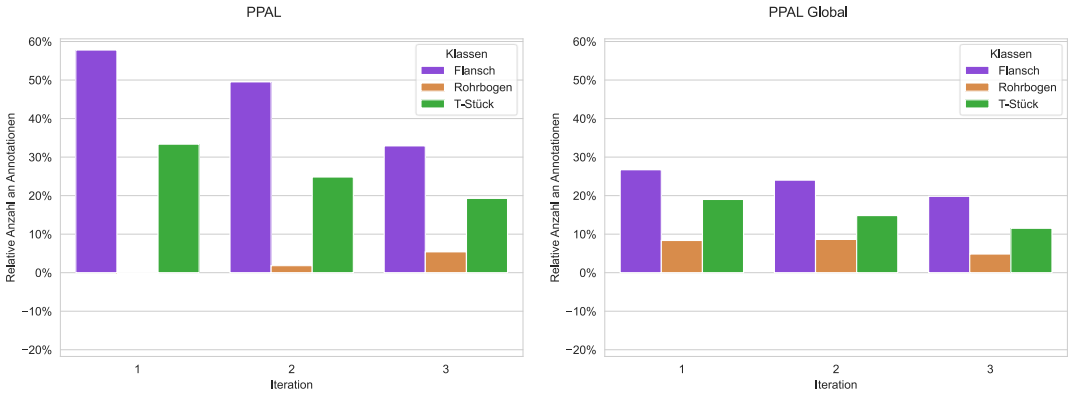


Abbildung 41: Relative Anzahl an Annotationen des PPAL-Global-Algorithmus

Hier lässt sich deutlich erkennen, dass durch die Modifikation die Anzahl an Annotationen reduziert wurde. Die verbleibende Zunahme wird in der ersten Stufe des PPAL-Algorithmus vermutet. Die erste Stufe von PPAL nutz die gewichtete Summe der Klassenentropie als Kriterium. Durch die Wahl der Summe als Aggregationsfunktion entsteht eine Tendenz zu Punktwolken mit vielen Objekten. Hierdurch ist die Auswahl des Core-Set Algorithmus auf verhältnismäßig große Punktwolken beschränkt.

CRB

Der CRB-Algorithmus wurde speziell für die Anwendung in Punktwolken entwickelt. Es ist jedoch festzustellen, dass die angestrebte Reduktion des Datenvolumens nicht in dem erwarteten Maße realisiert wurde. Als potenzielle Ursache hierfür wird die Durchführung des Erkennungsprozesses unter Aktivierung der Dropout-Funktion vermutet, welche zu einer Häufung von Fehlerkennungen führte. Bei ca. 20% der Punktwolken traten signifikante Fehlerkennungen auf, bei denen die generierten BB unverhältnismäßig große Bereiche umfassten. Ein Beispiel hierfür findet sich in Abbildung 42. In dieser Darstellung werden die detektierten BB für fünf Durchgänge mit aktiviertem Dropout mittels farbiger Umrisse visualisiert.

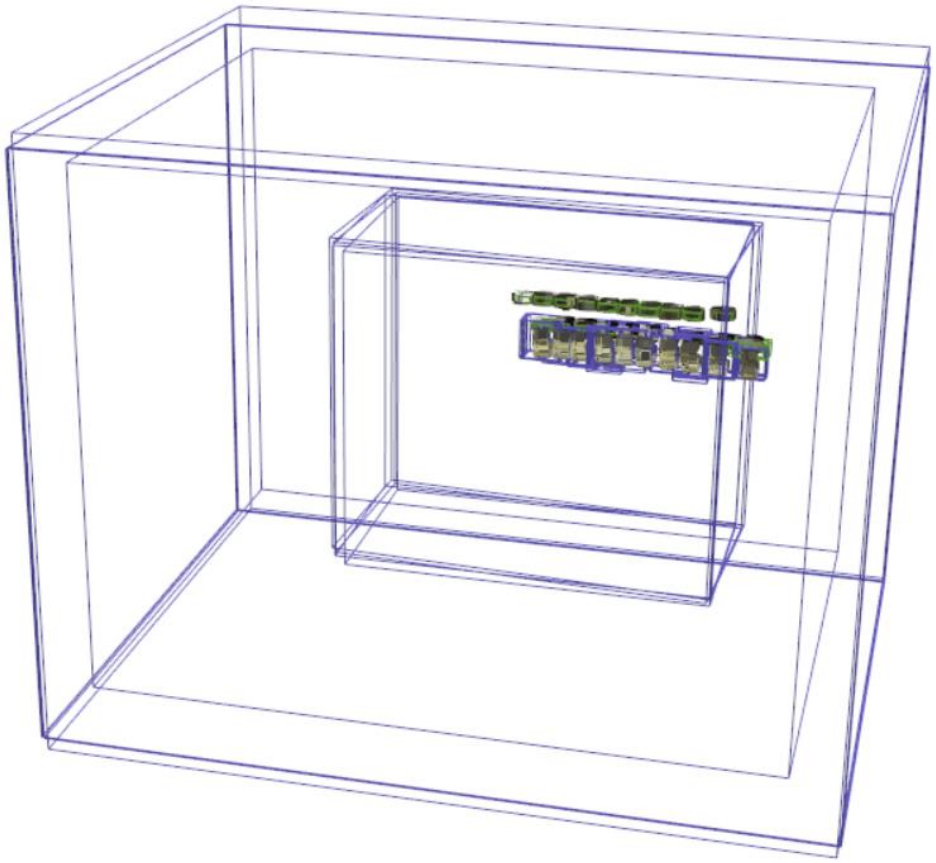


Abbildung 42: Fehlerkennung durch Markov Dropout

Es ist deutlich zu erkennen, dass einige BB alle Objekte der Punktwolke beinhalten. Auch bei der Erkennung ohne aktiviertes Dropout können in nahezu jeder untersuchten Punktwolke solche Phantomerkennungen beobachtet werden. Unter normalen Bedingungen weisen diese Fehlerkennungen jedoch eine Erkennungswahrscheinlichkeit unterhalb des definierten Schwellenwerts auf, wodurch sie im Regelfall verworfen werden. Die Aktivierung des Dropouts führte somit zu einer Erhöhung der Erkennungswahrscheinlichkeit dieser Fehlklassifikationen. Da die generierten Erkennungen als Basis für die Erstellung hypothetischer Annotationen dienen, welche zur Bestimmung der Gradienten für die Diversitätsmaximierung herangezogen werden, lässt sich ein potenziell negativer Einfluss auf den Prozess der Datenauswahl vermuten.

7 Zusammenfassung & Ausblick

Die vorliegende Arbeit widmet sich der Untersuchung der Übertragbarkeit von Active Learning Algorithmen aus dem Bereich der zweidimensionalen Objekterkennung auf die dreidimensionale Domäne. Die Relevanz dieser Untersuchung ergibt sich daraus, dass die Effektivität von AL von der Wahl des eingesetzten Modells sowie der Charakteristik des zugrundeliegenden Datensatzes abhängt. Infolgedessen führt eine Erweiterung des methodischen Repertoires zu einer verbesserten Adaptivität von AL auf neue Datensätze und Modellarchitekturen.

Zu diesem Zweck erfolgte die Identifikation und Analyse der spezifischen Herausforderungen, die bei der Anwendung von AL-Methoden im Kontext von Punktwolken auftreten können. Es konnte festgestellt werden, dass die Hauptproblematik in der Anwendbarkeit von Diversitätsmaßen liegt, welche aufgrund der dünn verteilten Natur der Punktwolken nicht ohne Weiteres einsetzbar sind. Hierdurch weist ein Vergleich der Feature Maps nicht die gewünschte Effektivität auf, da der Großteil des Raums leer ist und somit meist keine korrespondierenden Features zur Verfügung stehen.

Basierend auf diesen Erkenntnissen und bestehenden wissenschaftlichen Arbeiten wurden Lösungsansätze erarbeitet. Diese umfassten ein modellarchitekturbasiertes Vorgehen mit einer Konvertierung von dünnbesetzten Eingabedaten zu dichtbesetzten Features, eine gradienten-basierte Diversitätsanalyse durch das Schätzen von hypothetischen Annotationen sowie die Feature-Aggregation mit einem globalen oder lokalen Kontext.

Zudem wurde eine Taxonomie als Instrument zur Selektion der Algorithmen entwickelt. Der Selektionsprozess basierte dabei auf dem Prinzip, ein möglichst breites Spektrum der Taxonomie abzudecken. Dies ermöglichte ein systematisches Vorgehen bei der Auswahl der Algorithmen. Im Rahmen der praktischen Umsetzung erfolgte die Implementierung der ausgewählten Algorithmen, deren Leistungsfähigkeit und Effizienz anschließend einer empirischen Untersuchung unterzogen wurde. Hierfür wurde das Fully Convolutional Neural Network FCAF3D verwendet. Die Experimente wurden auf einem nicht öffentlichen Datensatz durchgeführt, welcher die Rohrleitungssysteme verschiedener Industrieanlagen abbildet. Bei der Evaluation wurde festgestellt, dass die Taxonomie nicht zur Prognose der Leistungsfähigkeit geeignet ist, da die Algorithmen Consensus Score [35] und Localization Stability [33] dieselbe Klassifikation besitzen, sich aber in ihrer Leistungsfähigkeit erheblich unterscheiden.

Von den neun getesteten Algorithmen konnten die vier Algorithmen Region of Interest Matching [35], Localization Stability [33], PPAL [30] und eine Variation der Localization Stability zu einer konsistenten Reduktion der benötigten Datenmenge führen. Hierbei kam es jeweils nur zu einer Reduktion für eine einzelne Klasse. Die Neigung der Algorithmen PPAL und Localization Stability zu Punktwolken mit vielen Objekten war evident. Aus diesem Grund wurde zur Evaluation die Average Precision im Verhältnis zu der Gesamtanzahl an

annotierten Objekten als Kriterium herangezogen. Eine Betrachtung der Leistungsfähigkeit im Verhältnis zu den genutzten Punktwolken würde zu einer unverhältnismäßigen Bevorzugung der beiden Algorithmen führen. Die größte mittlere Reduktion konnte die Localization Stability [35] mit –28% erzielen. Um die Leistungsfähigkeit der Algorithmen aus dem 2D Bereich in Kontext zu setzen, wurde der CRB-Algorithmus implementiert. Dieser AL-Algorithmus wurde für die Anwendung in Punktwolken entwickelt. Entgegen der Erwartung konnte dieser Ansatz die erforderliche Datenmenge nicht reduzieren. Die Ursache hierfür wird in der verwendeten Modellarchitektur vermutet.

Um die Leistungsfähigkeit von Diversitätsmaßen in Punktwolken zu steigern sind noch weitere Untersuchungen notwendig. Als vielversprechender Ausgangspunkt für weiterführende Forschungsaktivitäten kann der PPAL-Algorithmus herangezogen werden. Die Methodik des lokalen Vergleichs von Features birgt das Potenzial, die inhärenten Herausforderungen bei der Anwendung von Diversitätsmaßen auf Punktwolken zu adressieren. Problematisch hierbei ist jedoch die Bevorzugung von Punktwolken mit hoher Objektdichte, was als signifikante Limitierung zu betrachten ist. Zur Überwindung dieser Problematik bietet sich die Adaption von Techniken an, die im Bereich der 3D semantischen Segmentierung innerhalb von AL-Algorithmen verwendet werden. Ein charakteristisches Merkmal dieser Ansätze ist, dass nur Teilregionen der Punktwolken für Annotations- und Trainingszwecke verwendet werden. Die Integration eines solchen Vorgehens könnte nicht nur für den PPAL-Algorithmus, sondern auch für andere Verfahren, die eine Tendenz zur Bevorzugung objektreicher Punktwolken aufweisen, von Nutzen sein. Die Implementierung eines solchen Ansatzes würde nicht nur dem im Rahmen dieser Arbeit verwendeten Datensatz zugutekommen. Auch andere Datensätze sind von einer inhomogenen Verteilung von Objekten gekennzeichnet. Exemplarisch sei hier auf Datensätze aus dem Bereich des autonomen Fahrens verwiesen, die eine vergleichbare Datenverteilung aufweisen. Hier kann beobachtet werden, dass Aufnahmen aus ländlichen Regionen typischerweise eine geringere Objektdichte aufweisen als solche aus urbanen Gebieten.

Eine weitere Möglichkeit besteht in der Verwendung von Transfer Learning oder Self Supervised Learning. Die Integration dieser Verfahren birgt das Potential zur Effizienzsteigerung diversitätsbasierter Techniken. Der Vorteil besteht in der Fähigkeit, auch bei einer limitierten Anzahl annotierter Objekte repräsentative Features zu extrahieren, die für die Bestimmung von Ähnlichkeiten von essenzieller Bedeutung sind. Hierdurch kann eine Verbesserung der Performanz der AL-Systeme bei kleinen Datensätzen entstehen. Obgleich dies mit einer Zunahme des Rechenaufwands einhergeht, würde dies vor allem in kleinen Datensätzen zu einer Steigerung der Performance führen.

Durch die Abhängigkeit der Performance von AL-Algorithmen von dem verwendeten Datensatz und der Modellarchitektur sind die erzielten Resultate nur für das spezifische Anwendungsszenario gültig. Dies ergibt die Notwendigkeit von weiteren Untersuchungen mit verschiedenen Datensätzen

und Modellarchitekturen. Des Weiteren sind weitere Untersuchungen mit verschiedenen Trainingsparametern notwendig. So wurde durch die Begrenzte Zeit auf eine Überprüfung von verschiedenen initialen Trainingsdaten verzichtet. Außerdem wurden die Hyperparameter des Modells nur für die volle Menge an Trainingsdaten optimiert.

Dennoch konnte gezeigt werden, dass einige Methoden aus der 2D-Domäne erfolgreich auf Punktwolken übertragen werden können. Die erzielten Ergebnisse unterstreichen das Potenzial von Active Learning zur Datenreduktion in dreidimensionalen Daten. Durch die Identifikation, der mit der Anwendung von Diversitätsmaßen in Punktwolken assoziierten Problematik, konnte eine Basis für weitere Forschung geschaffen werden.

8 Quellenverzeichnis

[1] R. Ranjan *u. a.*, „Deep Learning for Understanding Faces: Machines May Be Just as Good, or Better, than Humans“, *IEEE Signal Process. Mag.*, Bd. 35, Nr. 1, S. 66–83, Jan. 2018, doi: 10.1109/MSP.2017.2764116.

[2] H. M. Ahmad und A. Rahimi, „Deep learning methods for object detection in smart manufacturing: A survey“, *J. Manuf. Syst.*, Bd. 64, S. 181–196, Juli 2022, doi: 10.1016/j.jmsy.2022.06.011.

[3] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, und S. Hu, „Traffic-Sign Detection and Classification in the Wild“, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Juni 2016, S. 2110–2118. doi: 10.1109/CVPR.2016.232.

[4] Z. Zou, K. Chen, Z. Shi, Y. Guo, und J. Ye, „Object Detection in 20 Years: A Survey“, *Proc. IEEE*, Bd. 111, Nr. 3, S. 257–276, März 2023, doi: 10.1109/JPROC.2023.3238524.

[5] H. H. Aghdam, A. Gonzalez-Garcia, A. Lopez, und J. Weijer, „Active Learning for Deep Detection Neural Networks“, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South): IEEE, Okt. 2019, S. 3671–3679. doi: 10.1109/ICCV.2019.00377.

[6] B. Settles, „Active Learning Literature Survey“.

[7] D. Garcia, J. Carias, T. Adão, R. Jesus, A. Cunha, und L. G. Magalhães, „Ten Years of Active Learning Techniques and Object Detection: A Systematic Review“, *Appl. Sci.*, Bd. 13, Nr. 19, Art. Nr. 19, Jan. 2023, doi: 10.3390/app131910667.

[8] P. Rubinowicz und K. Czyńska, „Study of City Landscape Heritage Using Lidar Data and 3d-City Models“, *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, Bd. XL-7/W3, S. 1395–1402, Apr. 2015, doi: 10.5194/isprsarchives-XL-7-W3-1395-2015.

[9] Y. Li und J. Ibanez-Guzman, „Lidar for Autonomous Driving: The principles, challenges, and trends for automotive lidar and perception systems“, *IEEE Signal Process. Mag.*, Bd. 37, Nr. 4, S. 50–61, Juli 2020, doi: 10.1109/MSP.2020.2973615.

[10] M. Beland *u. a.*, „On promoting the use of lidar systems in forest ecosystem research“, *For. Ecol. Manag.*, Bd. 450, S. 117484, Okt. 2019, doi: 10.1016/j.foreco.2019.117484.

[11] S. Song, S. P. Lichtenberg, und J. Xiao, „SUN RGB-D: A RGB-D scene understanding benchmark suite“, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Juni 2015, S. 567–576. doi: 10.1109/CVPR.2015.7298655.

[12] C. Shorten und T. M. Khoshgoftaar, „A survey on Image Data Augmentation for Deep Learning“, *J. Big Data*, Bd. 6, Nr. 1, S. 60, Dez. 2019, doi: 10.1186/s40537-019-0197-0.

[13] K. Weiss, T. M. Khoshgoftaar, und D. Wang, „A survey of transfer learning“, *J. Big Data*, Bd. 3, Nr. 1, S. 9, Dez. 2016, doi: 10.1186/s40537-016-0043-6.

[14] G. K. Rajbahadur *u. a.*, „Can I use this publicly available dataset to build commercial AI software? -- A Case Study on Publicly Available Image Datasets“, 11. April 2022, *arXiv*: arXiv:2111.02374. Zugegriffen: 19. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2111.02374

[15] I. Goodfellow, Y. Bengio, und A. Courville, *Deep learning*. in Adaptive computation and machine learning. Cambridge, Massachusetts London, England: The MIT Press, 2016.

[16] M. Mots’oehli und K. Baek, „Deep Active Learning in the Presence of Label Noise: A Survey“, 19. September 2023, *arXiv*: arXiv:2302.11075. Zugegriffen: 16. Juni 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2302.11075

[17] P. Kumar und A. Gupta, „Active Learning Query Strategies for Classification, Regression, and Clustering: A Survey“, *J. Comput. Sci. Technol.*, Bd. 35, Nr. 4, S. 913–945, Juli 2020, doi: 10.1007/s11390-020-9487-4.

[18] F. Carcillo, Y.-A. L. Borgne, O. Caelen, und G. Bontempi, „Streaming Active Learning Strategies for Real-Life Credit Card Fraud Detection: Assessment and Visualization“, *Int. J. Data Sci. Anal.*, Bd. 5, Nr. 4, S. 285–300, Juni 2018, doi: 10.1007/s41060-018-0116-z.

[19] D. Lieber, B. Konrad, J. Deuse, M. Stolpe, und K. Morik, „Sustainable Interlinked Manufacturing Processes through Real-

Time Quality Prediction“, in *Leveraging Technology for a Sustainable World*, D. A. Dornfeld und B. S. Linke, Hrsg., Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, S. 393–398. doi: 10.1007/978-3-642-29069-5_67.

[20] R. Schumann und I. Rehbein, „Active Learning via Membership Query Synthesis for Semi-Supervised Sentence Classification“, in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, Hong Kong, China: Association for Computational Linguistics, 2019, S. 472–481. doi: 10.18653/v1/K19-1044.

[21] A. Tharwat und W. Schenck, „A Survey on Active Learning: State-of-the-Art, Practical Challenges and Research Directions“, *Mathematics*, Bd. 11, Nr. 4, S. 820, Feb. 2023, doi: 10.3390/math11040820.

[22] Y. Fu, X. Zhu, und B. Li, „A survey on instance selection for active learning“, *Knowl. Inf. Syst.*, Bd. 35, Nr. 2, S. 249–283, Mai 2013, doi: 10.1007/s10115-012-0507-8.

[23] S. Dasgupta, „Two faces of active learning“, *Theor. Comput. Sci.*, Bd. 412, Nr. 19, S. 1767–1781, Apr. 2011, doi: 10.1016/j.tcs.2010.12.054.

[24] Y. Geifman und R. El-Yaniv, „Deep Active Learning over the Long Tail“, 2. November 2017, *arXiv*: arXiv:1711.00941. Zugegriffen: 14. Juni 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1711.00941

[25] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford, und A. Agarwal, „Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds“, 23. Februar 2020, *arXiv*: arXiv:1906.03671. Zugegriffen: 20. November 2023. [Online]. Verfügbar unter: http://arxiv.org/abs/1906.03671

[26] S. Mittal, J. Niemeijer, J. P. Schäfer, und T. Brox, „Best Practices in Active Learning for Semantic Segmentation“, 15. März 2023, *arXiv*: arXiv:2302.04075. Zugegriffen: 10. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2302.04075

[27] C.-A. Brust, C. Käding, und J. Denzler, „Active Learning for Deep Object Detection“, 26. September 2018, *arXiv*: arXiv:1809.09875. Zugegriffen: 29. März 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1809.09875

[28] D. Feng, X. Wei, L. Rosenbaum, A. Maki, und K. Dietmayer, „Deep Active Learning for Efficient Training of a LiDAR 3D Object Detector“, in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Juni 2019, S. 667–674. doi: 10.1109/IVS.2019.8814236.

[29] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, und H. Hu, „Propagate Yourself: Exploring Pixel-Level Consistency for Unsupervised Visual Representation Learning“, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Juni 2021, S. 16679–16688. doi: 10.1109/CVPR46437.2021.01641.

[30] C. Yang, L. Huang, und E. J. Crowley, „Plug and Play Active Learning for Object Detection“, 14. März 2024, *arXiv*: arXiv:2211.11612. Zugegriffen: 24. Mai 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2211.11612

[31] Y. Gal und Z. Ghahramani, „Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning“.

[32] D. Wang und Y. Shang, „A new active labeling method for deep learning“, in *2014 International Joint Conference on Neural Networks (IJCNN)*, Juli 2014, S. 112–119. doi: 10.1109/IJCNN.2014.6889457.

[33] C.-C. Kao, T.-Y. Lee, P. Sen, und M.-Y. Liu, „Localization-Aware Active Learning for Object Detection“, 16. Januar 2018, *arXiv*: arXiv:1801.05124. Zugegriffen: 7. Dezember 2023. [Online]. Verfügbar unter: http://arxiv.org/abs/1801.05124

[34] D. Yoo und I. S. Kweon, „Learning Loss for Active Learning“, 9. Mai 2019, *arXiv*: arXiv:1905.03677. Zugegriffen: 6. Juni 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1905.03677

[35] S. Schmidt, Q. Rao, J. Tatsch, und A. Knoll, „Advanced Active Learning Strategies for Object Detection“, in *2020 IEEE Intelligent Vehicles Symposium (IV)*, Las Vegas, NV, USA: IEEE, Okt. 2020, S. 871–876. doi: 10.1109/IV47402.2020.9304565.

[36] W. Yu, S. Zhu, T. Yang, und C. Chen, „Consistency-Based Active Learning for Object Detection“, 2022.

[37] O. Sener und S. Savarese, „Active Learning for Convolutional Neural Networks: A Core-Set Approach“, 1. Juni 2018, *arXiv*: arXiv:1708.00489. doi: 10.48550/arXiv.1708.00489.

[38] A. Moses, S. Jakkampudi, C. Danner, und D. Biega, „Localization-based active learning (LOCAL) for object detection in 3D point clouds“, in *Geospatial Informatics XII*, K. Palaniappan, G. Seetharaman, und J. D. Harguess, Hrsg., Orlando, United States: SPIE, Mai 2022, S. 9. doi: 10.1117/12.2618513.

[39] T.-H. Wu *u. a.*, „ReDAL: Region-based and Diversity-aware Active Learning for Point Cloud Semantic Segmentation“, in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Okt. 2021, S. 15490–15499. doi: 10.1109/ICCV48922.2021.01522.

[40] Z. Liang, X. Xu, S. Deng, L. Cai, T. Jiang, und K. Jia, „Exploring Diversity-based Active Learning for 3D Object Detection in Autonomous Driving“, 16. Mai 2022, *arXiv*: arXiv:2205.07708. doi: 10.48550/arXiv.2205.07708.

[41] Y. Luo, Z. Chen, Z. Wang, X. Yu, Z. Huang, und M. Baktashmotlagh, „Exploring Active 3D Object Detection from a Generalization Perspective“, 8. Februar 2023, *arXiv*: arXiv:2301.09249. Zugegriffen: 20. November 2023. [Online]. Verfügbar unter: http://arxiv.org/abs/2301.09249

[42] S. Hwang, S. Kim, Y. Kim, und D. Kum, „Joint Semi-Supervised and Active Learning via 3D Consistency for 3D Object Detection“, in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom: IEEE, Mai 2023, S. 4819–4825. doi: 10.1109/ICRA48891.2023.10160433.

[43] C. T. Lüth, T. J. Bungert, L. Klein, und P. F. Jaeger, „Navigating the Pitfalls of Active Learning Evaluation: A Systematic Framework for Meaningful Performance Assessment“, 3. November 2023, *arXiv*: arXiv:2301.10625. Zugegriffen: 17. Juni 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2301.10625

[44] R. Qian, X. Lai, und X. Li, „3D Object Detection for Autonomous Driving: A Survey“, *Pattern Recognit.*, Bd. 130, S. 108796, Okt. 2022, doi: 10.1016/j.patcog.2022.108796.

[45] C. R. Qi, L. Yi, H. Su, und L. J. Guibas, „PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space“, 7. Juni 2017, *arXiv*: arXiv:1706.02413. doi: 10.48550/arXiv.1706.02413.

[46] C. Choy, J. Gwak, und S. Savarese, „4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks“, arXiv, Juni 2019. doi: 10.48550/arXiv.1904.08755.

[47] A. Moses, C. Bogart, K. O'Haire, L. Solorzano, und E. Yeo, „Diversity-based active learning: creating a representative object detection dataset in 3D point clouds", in *Geospatial Informatics XIII*, K. Palaniappan, G. Seetharaman, und J. D. Harguess, Hrsg., Orlando, United States: SPIE, Juni 2023, S. 16. doi: 10.1117/12.2663179.

[48] Y. Zhou und O. Tuzel, „VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection", gehalten auf der Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, S. 4490–4499. Zugegriffen: 16. Mai 2023. [Online]. Verfügbar unter: https://openaccess.thecvf.com/content_cvpr_2018/html/Zhou_VoxelNet_End-to-End_Learning_CVPR_2018_paper.html

[49] E. Haussmann *u. a.*, „Scalable Active Learning for Object Detection", 9. April 2020, *arXiv*: arXiv:2004.04699. Zugegriffen: 16. November 2023. [Online]. Verfügbar unter: http://arxiv.org/abs/2004.04699

[50] F. Shao *u. a.*, „Active Learning for Point Cloud Semantic Segmentation via Spatial-Structural Diversity Reasoning", 18. April 2022, *arXiv*: arXiv:2202.12588. doi: 10.48550/arXiv.2202.12588.

[51] X. Shi, X. Xu, K. Chen, L. Cai, C. S. Foo, und K. Jia, „Label-Efficient Point Cloud Semantic Segmentation: An Active Learning Approach", 12. April 2021, *arXiv*: arXiv:2101.06931. doi: 10.48550/arXiv.2101.06931.

[52] D. Feng, L. Rosenbaum, und K. Dietmayer, „Towards Safe Autonomous Driving: Capture Uncertainty in the Deep Neural Network For Lidar 3D Vehicle Detection", 7. September 2018, *arXiv*: arXiv:1804.05132. Zugegriffen: 8. Mai 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1804.05132

[53] D. Lowell, Z. C. Lipton, und B. C. Wallace, „Practical Obstacles to Deploying Active Learning", 1. November 2019, *arXiv*: arXiv:1807.04801. doi: 10.48550/arXiv.1807.04801.

[54] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, und P. S. Yu, „Active Learning: A Survey", *Algorithms Appl.*.

[55] Y. Zhou, A. Renduchintala, X. Li, S. Wang, Y. Mehdad, und A. Ghoshal, „Towards Understanding the Behaviors of Optimal Deep Active Learning Algorithms", 20. Februar 2021, *arXiv*:

arXiv:2101.00977. Zugegriffen: 29. März 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2101.00977

[56] P. Munjal, N. Hayat, M. Hayat, J. Sourati, und S. Khan, „Towards Robust and Reproducible Active Learning Using Neural Networks“, 15. Juni 2022, *arXiv*: arXiv:2002.09564. Zugegriffen: 1. April 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/2002.09564

[57] B. Zhu, Z. Jiang, X. Zhou, Z. Li, und G. Yu, „Class-balanced Grouping and Sampling for Point Cloud 3D Object Detection“, 26. August 2019, *arXiv*: arXiv:1908.09492. Zugegriffen: 4. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1908.09492

[58] Y. Chen, J. Liu, X. Zhang, X. Qi, und J. Jia, „VoxelNeXt: Fully Sparse VoxelNet for 3D Object Detection and Tracking“, 20. März 2023, *arXiv*: arXiv:2303.11301. doi: 10.48550/arXiv.2303.11301.

[59] S. Shi *u. a.*, „PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection“, gehalten auf der Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, S. 10529–10538. Zugegriffen: 10. Dezember 2023. [Online]. Verfügbar unter: https://openaccess.thecvf.com/content_CVPR_2020/html/Shi_PV-RCNN_Point-Voxel_Feature_Set_Abstraction_for_3D_Object_Detection_CVPR_2020_paper.html

[60] S. Shi *u. a.*, „PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection“, 7. November 2022, *arXiv*: arXiv:2102.00463. doi: 10.48550/arXiv.2102.00463.

[61] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, und H. Li, „Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection“, 5. Februar 2021, *arXiv*: arXiv:2012.15712. doi: 10.48550/arXiv.2012.15712.

[62] S. Shi, X. Wang, und H. Li, „PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud“, 16. Mai 2019, *arXiv*: arXiv:1812.04244. doi: 10.48550/arXiv.1812.04244.

[63] Z. Liu, H. Tang, Y. Lin, und S. Han, „Point-Voxel CNN for Efficient 3D Deep Learning“, 9. Dezember 2019, *arXiv*: arXiv:1907.03739. doi: 10.48550/arXiv.1907.03739.

[64] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, und O. Beijbom, „PointPillars: Fast Encoders for Object Detection from Point Clouds“, 6. Mai 2019, *arXiv*: arXiv:1812.05784. Zugegriffen: 16. Mai 2023. [Online]. Verfügbar unter: http://arxiv.org/abs/1812.05784

[65] Y. Yan, Y. Mao, und B. Li, „SECOND: Sparsely Embedded Convolutional Detection“, *Sensors*, Bd. 18, Nr. 10, S. 3337, Okt. 2018, doi: 10.3390/s18103337.

[66] D. Rukhovich, A. Vorontsova, und A. Konushin, „FCAF3D: Fully Convolutional Anchor-Free 3D Object Detection“, 24. März 2022, *arXiv*: arXiv:2112.00322. doi: 10.48550/arXiv.2112.00322.

[67] K. He, X. Zhang, S. Ren, und J. Sun, „Deep Residual Learning for Image Recognition“, 10. Dezember 2015, *arXiv*: arXiv:1512.03385. Zugegriffen: 6. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1512.03385

[68] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, und S. Belongie, „Feature Pyramid Networks for Object Detection“, 19. April 2017, *arXiv*: arXiv:1612.03144. Zugegriffen: 6. August 2024. [Online]. Verfügbar unter: http://arxiv.org/abs/1612.03144

[69] H. Caesar *u. a.*, „nuScenes: A Multimodal Dataset for Autonomous Driving“, gehalten auf der Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, S. 11621–11631. Zugegriffen: 19. Februar 2024. [Online]. Verfügbar unter: https://openaccess.thecvf.com/content_CVPR_2020/html/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.html

[70] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, und A. Zisserman, „The Pascal Visual Object Classes (VOC) Challenge“, *Int. J. Comput. Vis.*, Bd. 88, Nr. 2, S. 303–338, Juni 2010, doi: 10.1007/s11263-009-0275-4.

9 Anhang

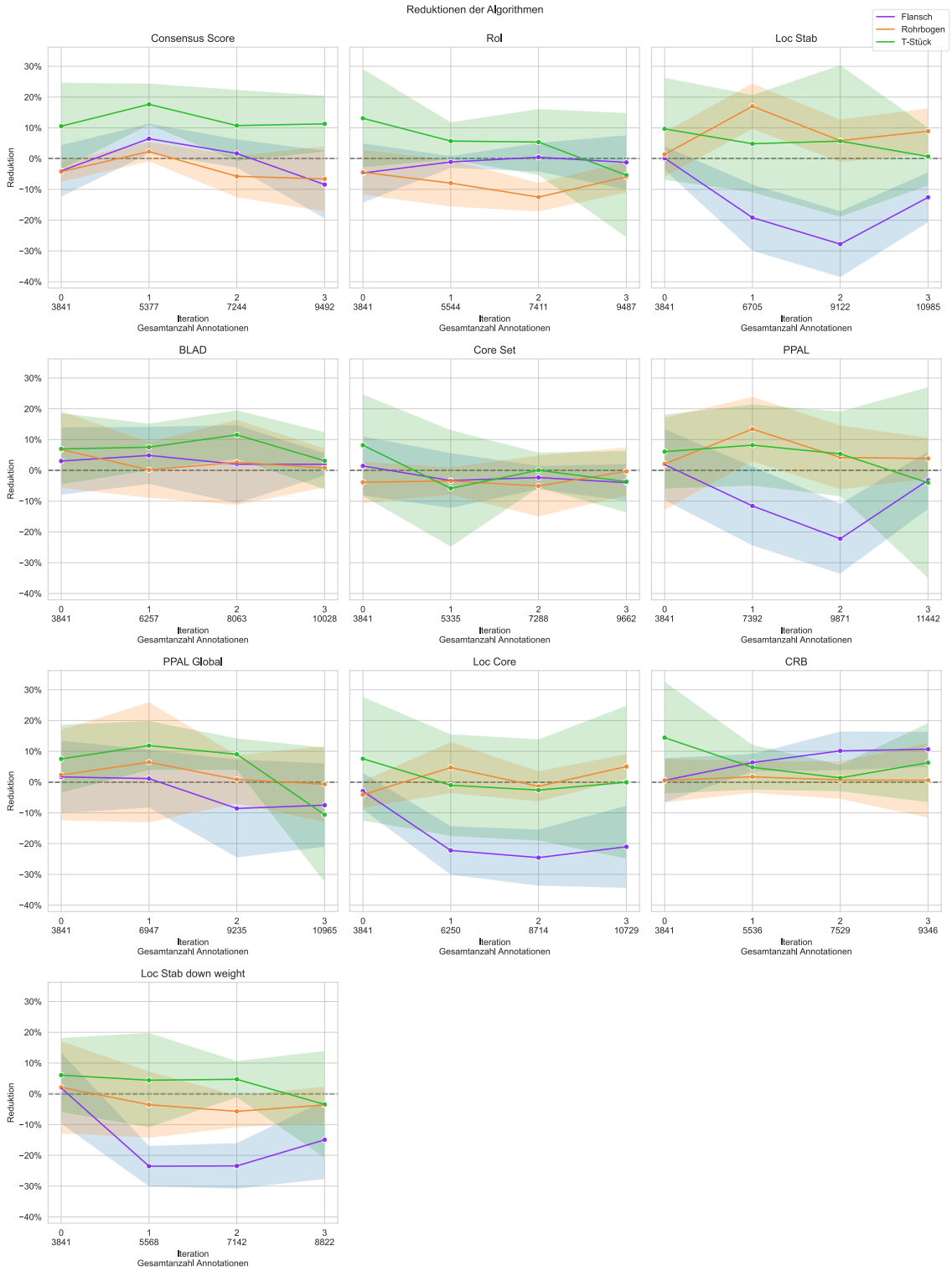


Abbildung 43: Erzielte Reduktion der Gesamtanzahl an Annotationen

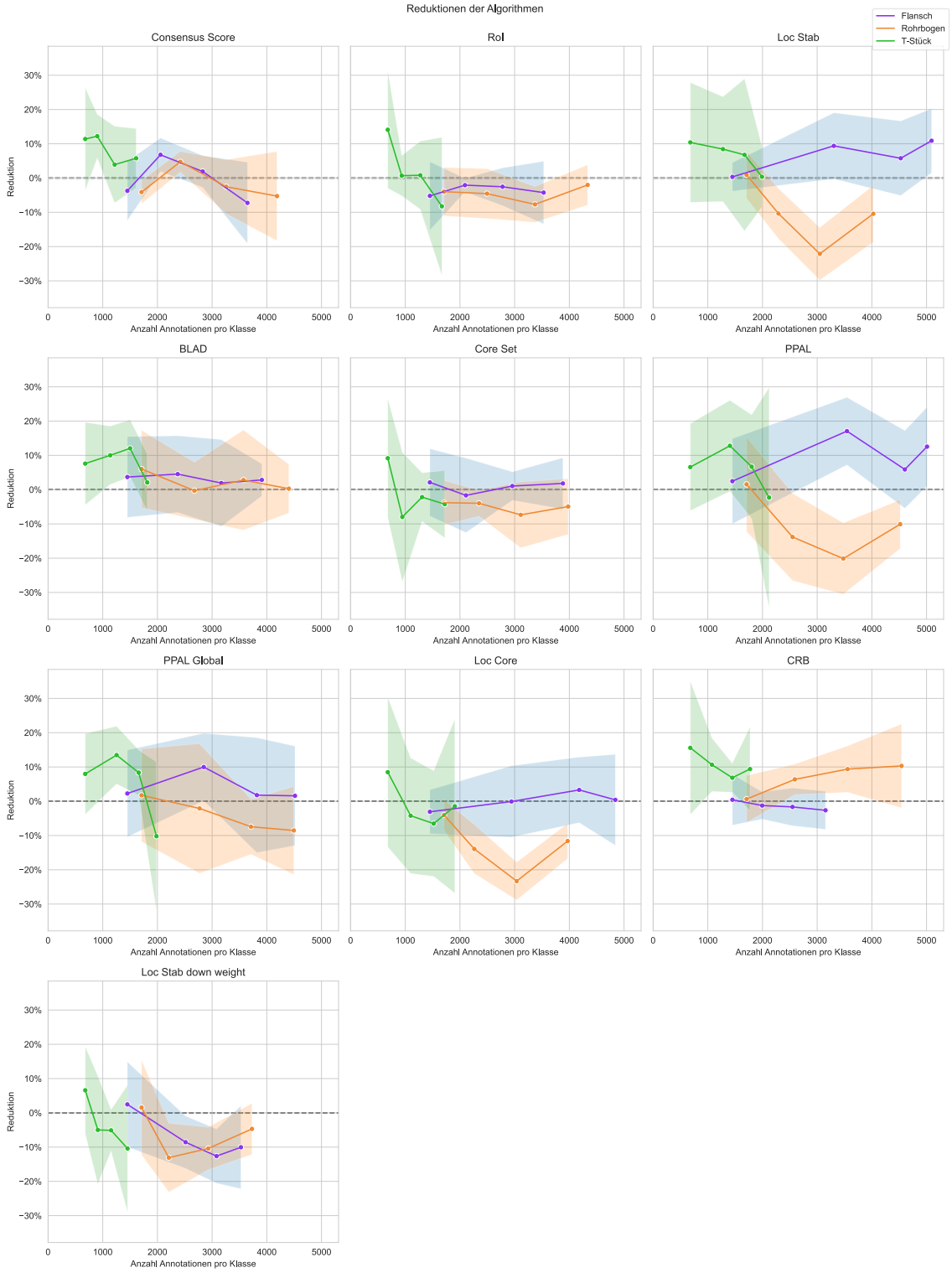
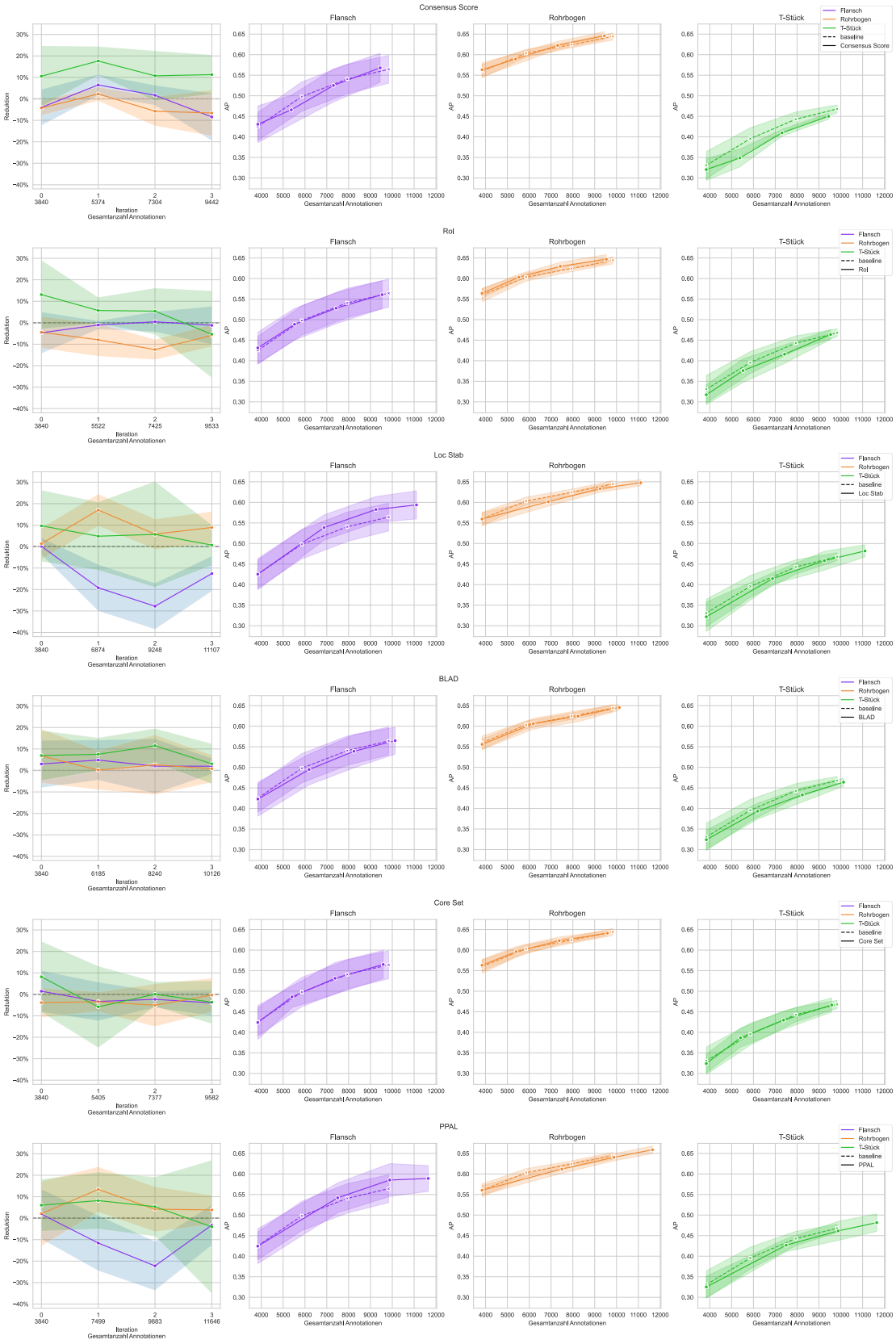


Abbildung 44: Erzielte Reduktion der Annotationen pro Klasse


Consensus Score
Flansch
Rohrbogen
T-Stück
Flansch
Rohrbogen
T-Stück
baseline
Consensus Score
Reduktion
AP
Iteration
Gesamtanzahl Annotationen
RoI
Flansch
Rohrbogen
T-Stück
baseline
RoI
Loc Stab
Flansch
Rohrbogen
T-Stück
baseline
Loc Stab
BLAD
Flansch
Rohrbogen
T-Stück
baseline
BLAD
Core Set
Flansch
Rohrbogen
T-Stück
baseline
Core Set
PPAL
Flansch
Rohrbogen
T-Stück
baseline
PPAL

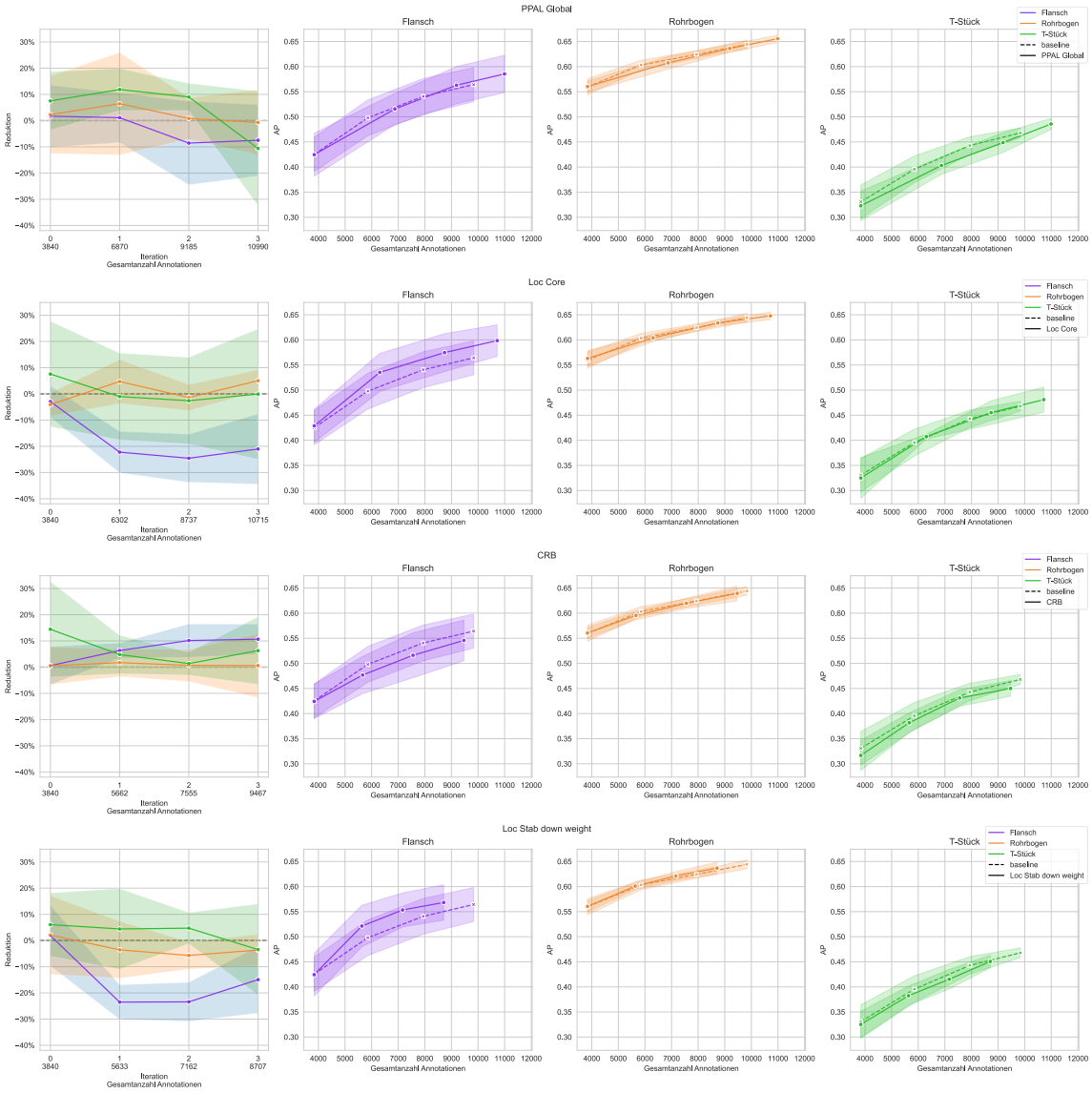
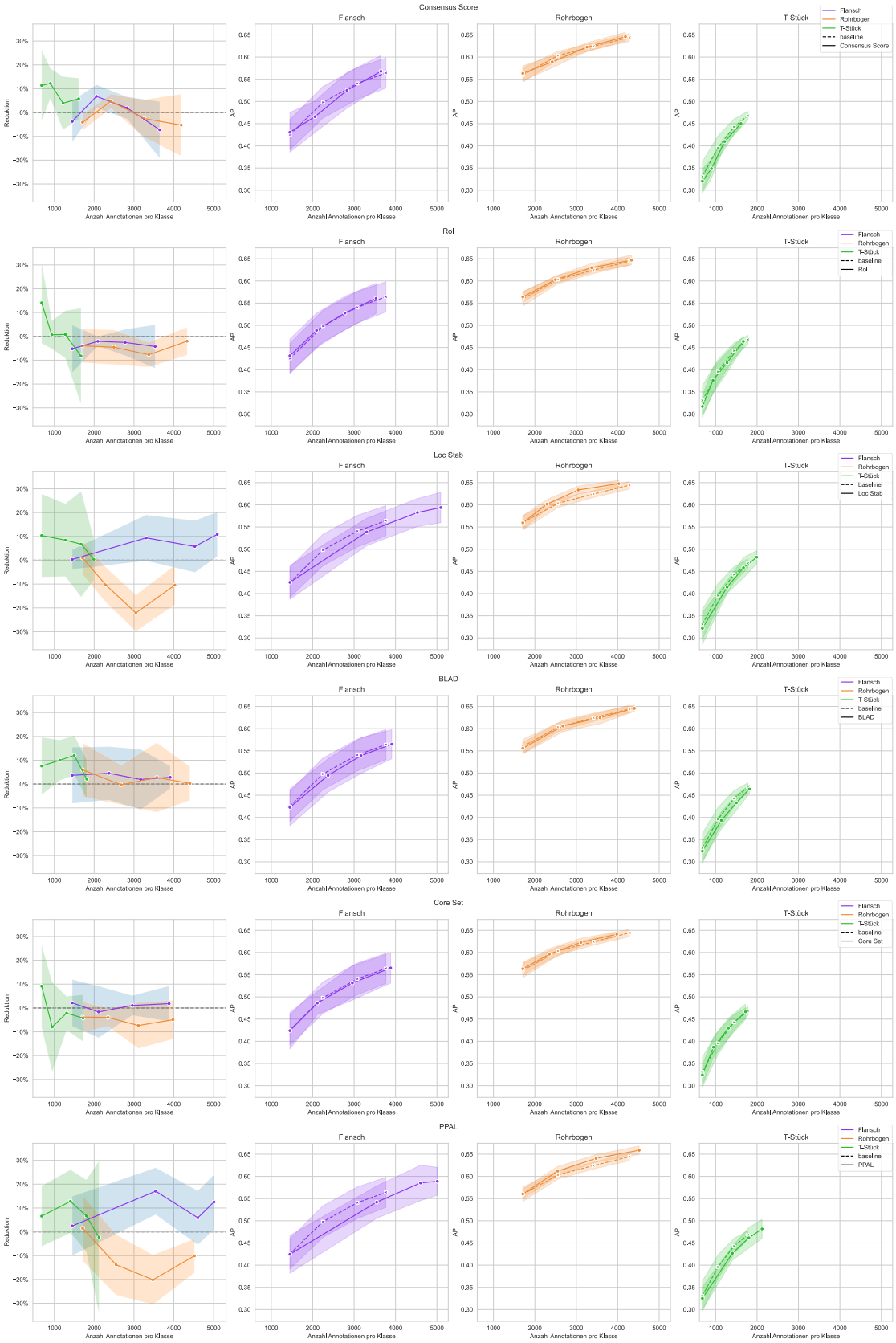


Abbildung 45: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen


Consensus Score
Flansch
Rohrbogen
T-Stück
RoI
Loc Stab
BLAD
Core Set
PPAL
Reduktion
AP
Anzahl Annotationen pro Klasse
baseline

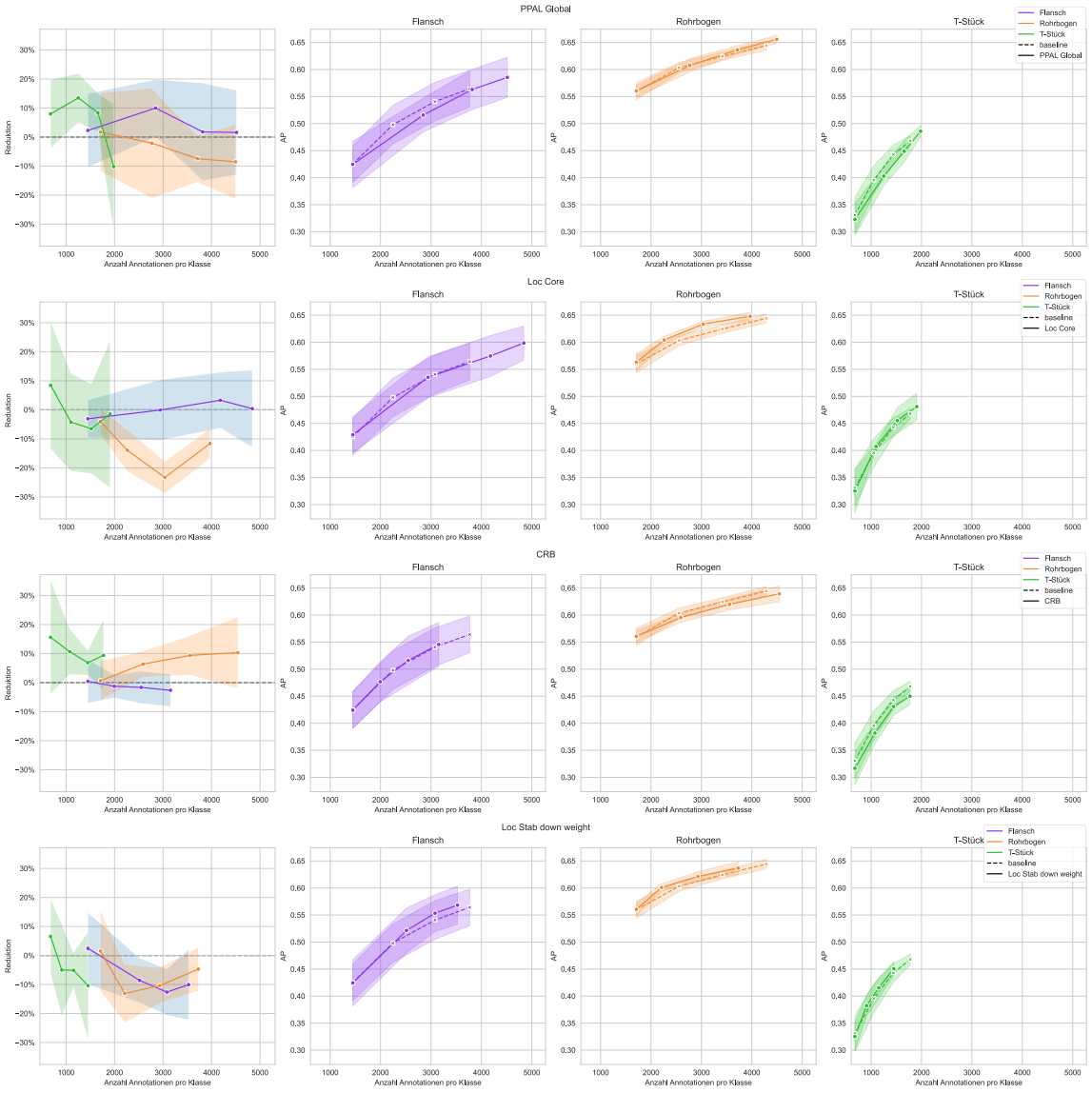


Abbildung 46: Erzielte Reduktion und AP für die Anzahl an Annotationen pro Klasse

Abbildung 47: Anzahl an Annotationen für alle Algorithmen

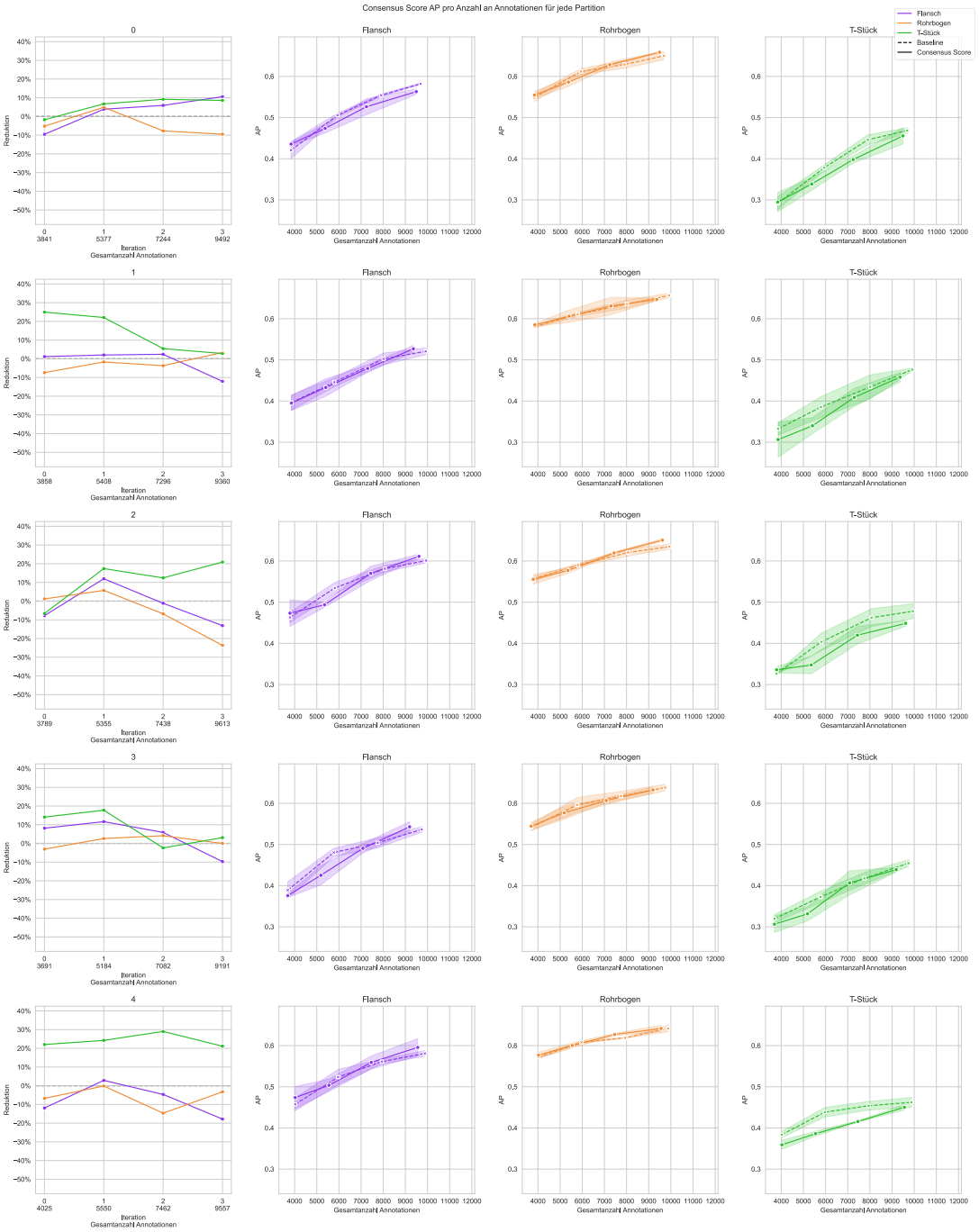


Abbildung 48: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition des Consensus Score

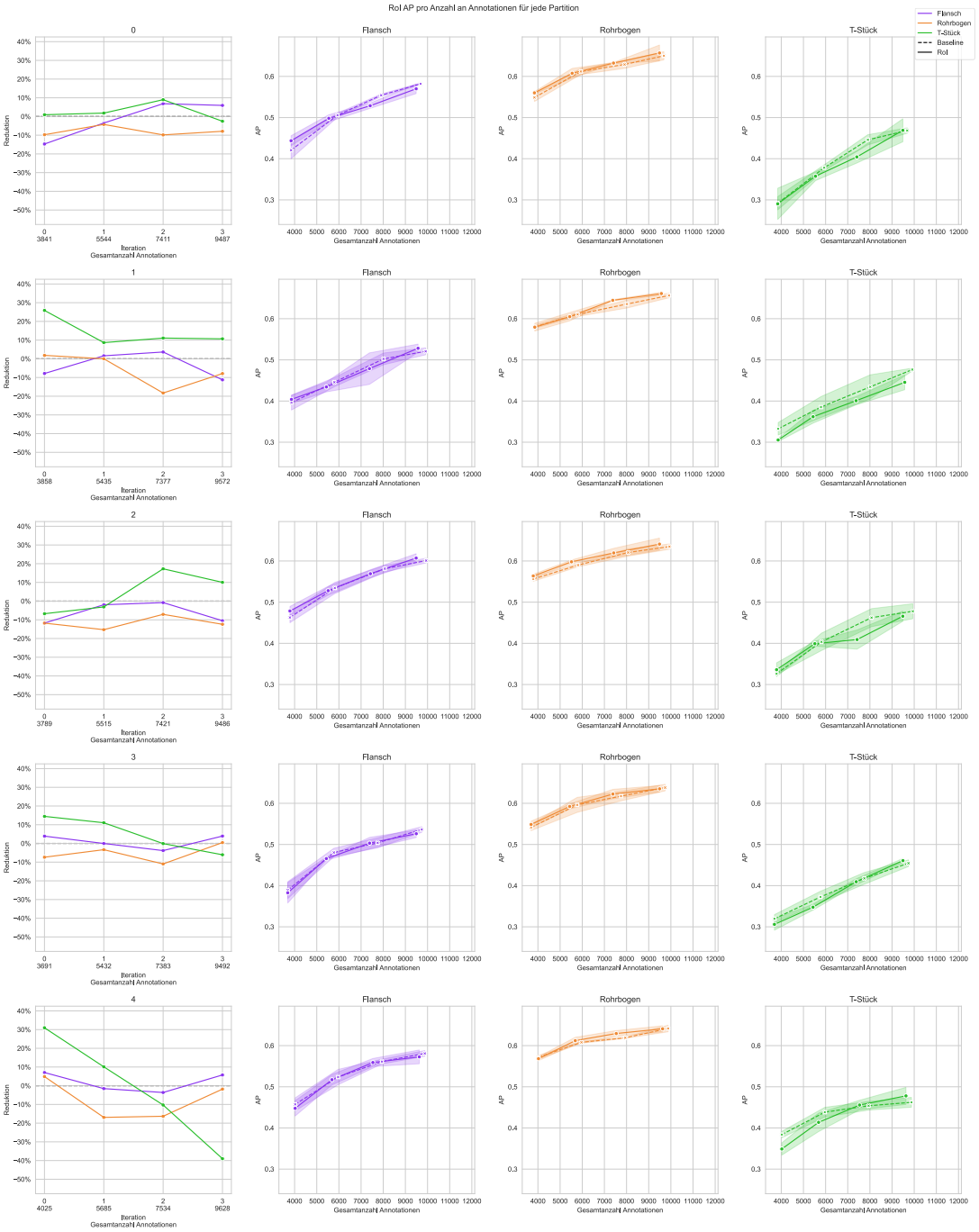


Abbildung 49: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition des RoI

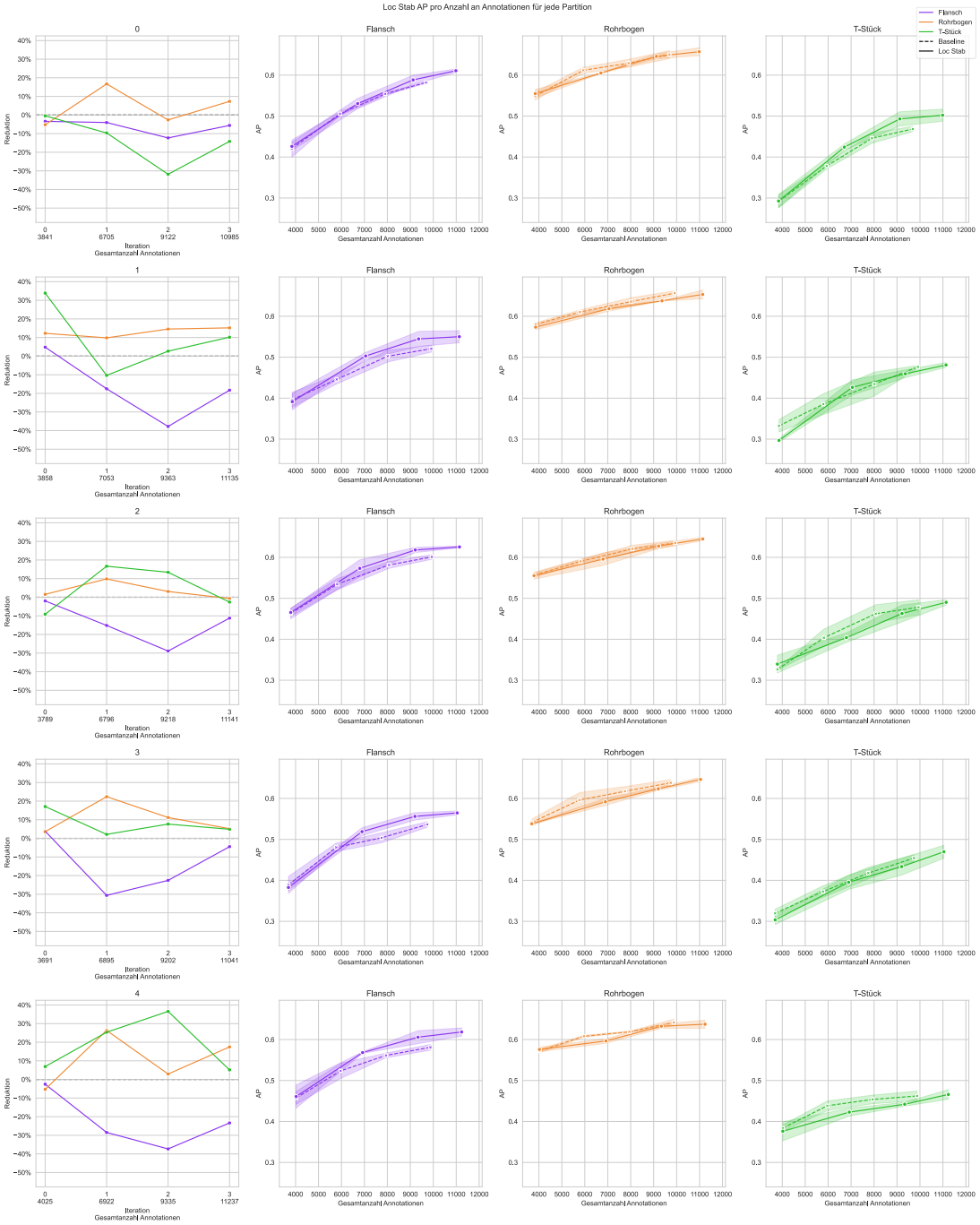


Abbildung 50: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition der Localization Stability

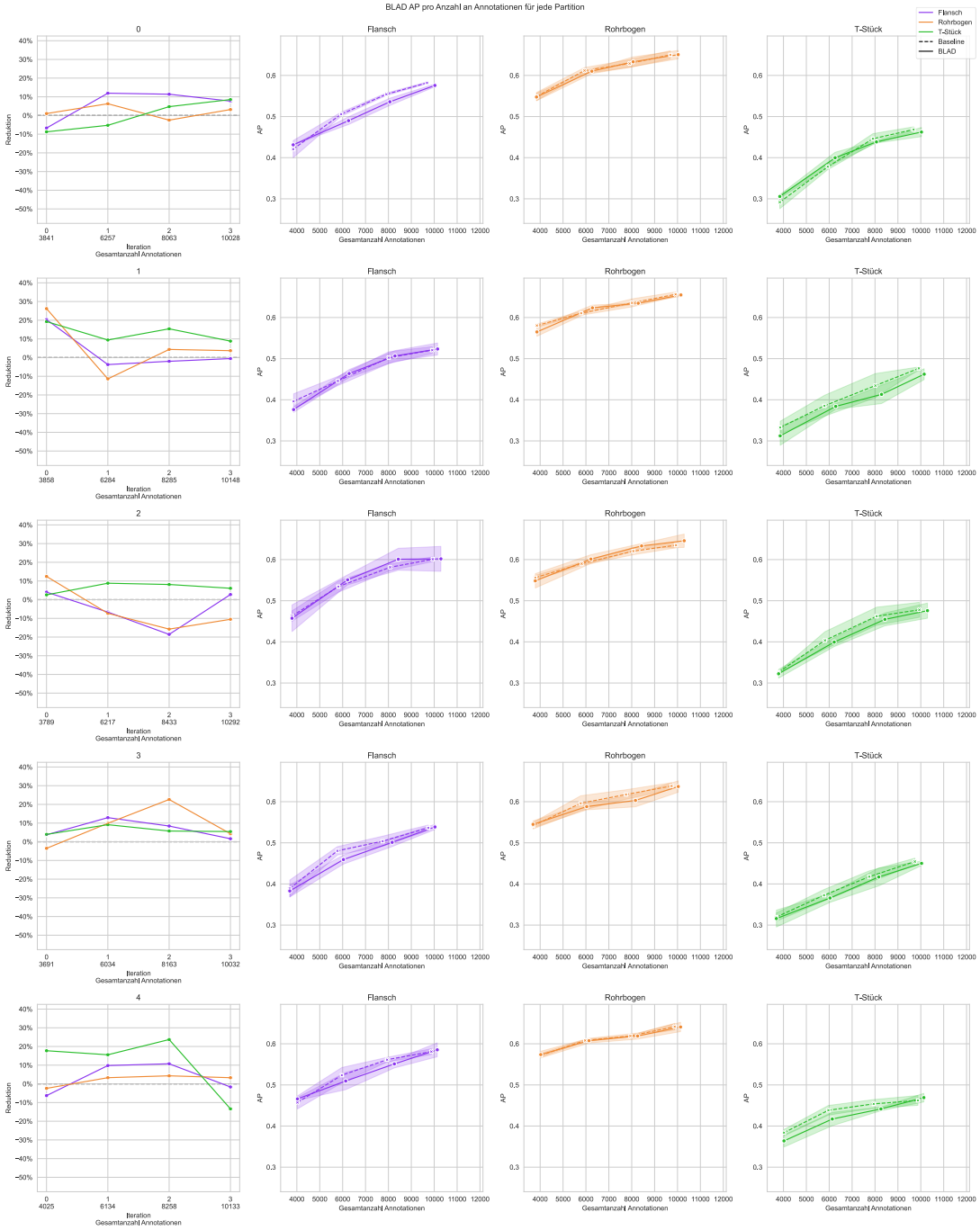


Abbildung 51: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition von BLAD

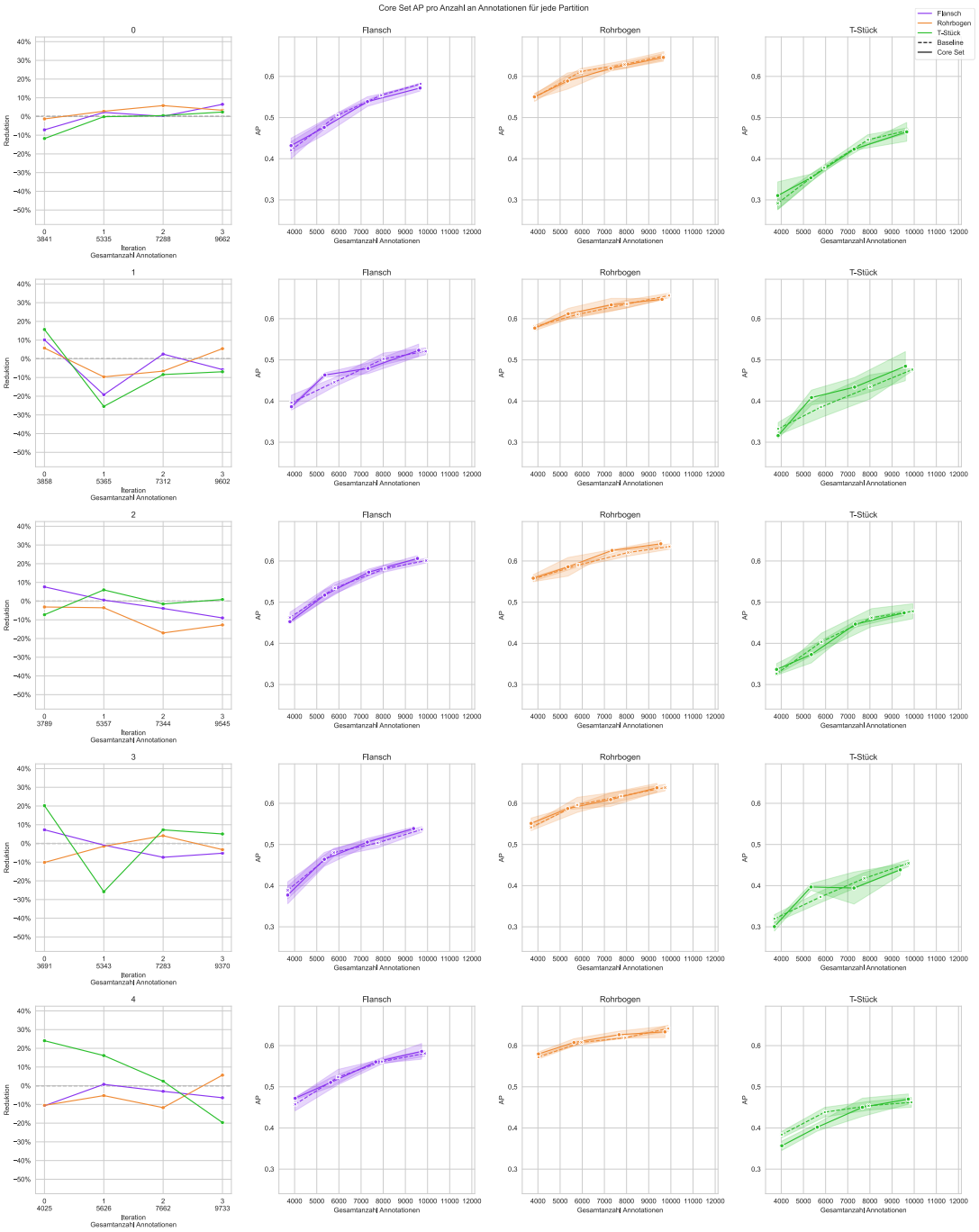


Abbildung 52: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition von Core Set

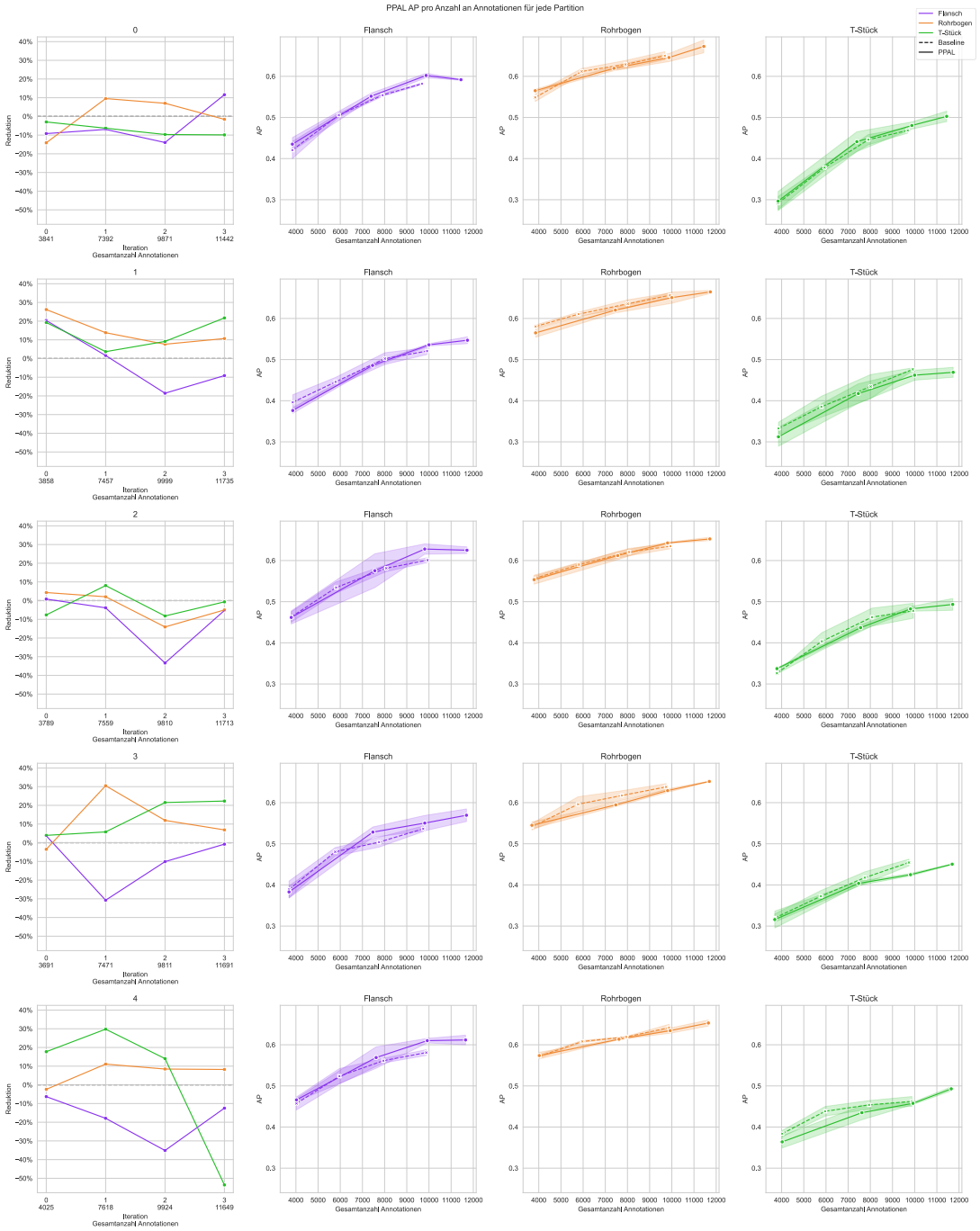


Abbildung 53: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition von PPAL

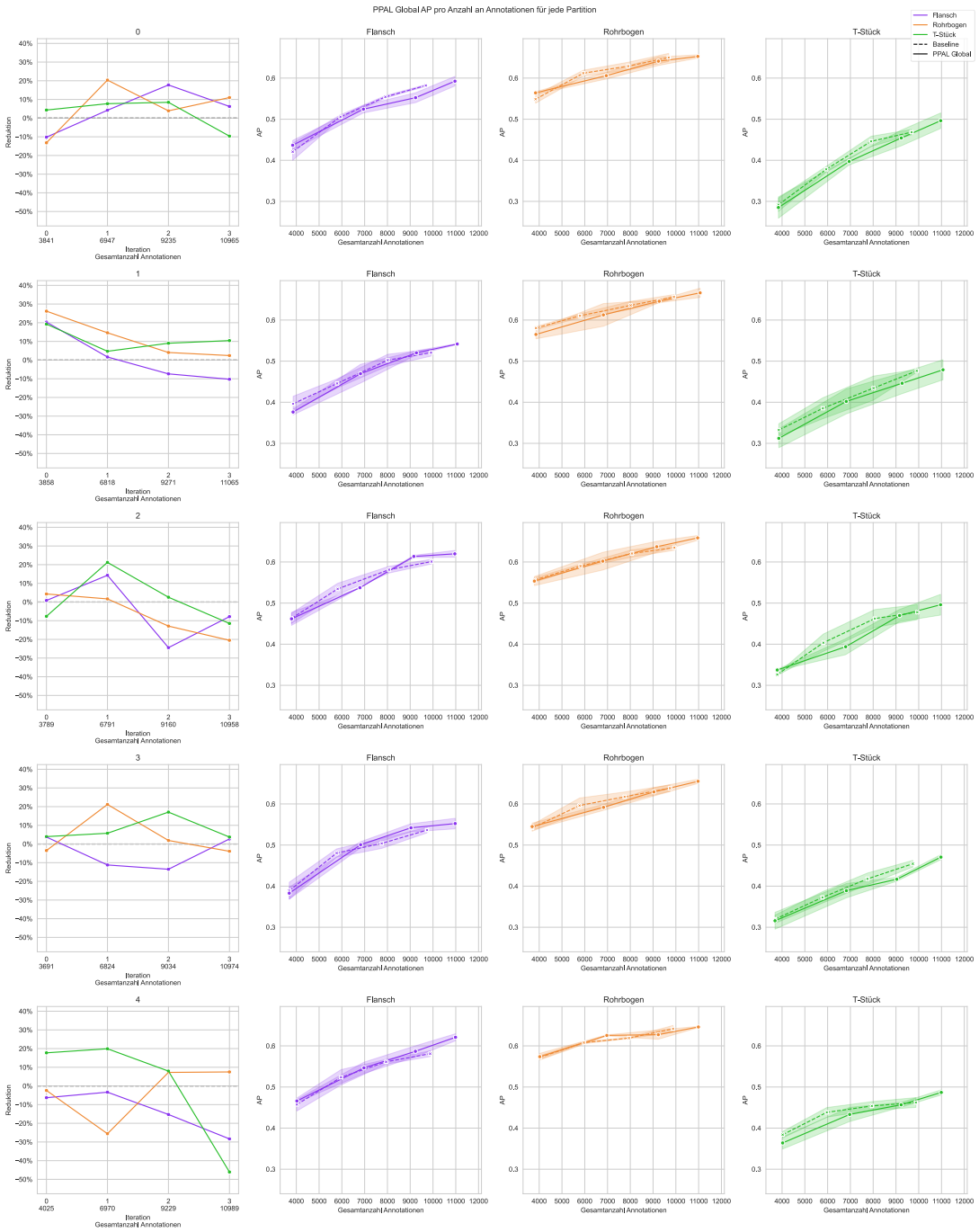


Abbildung 54: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition von PPAL mit globalerem Kontext

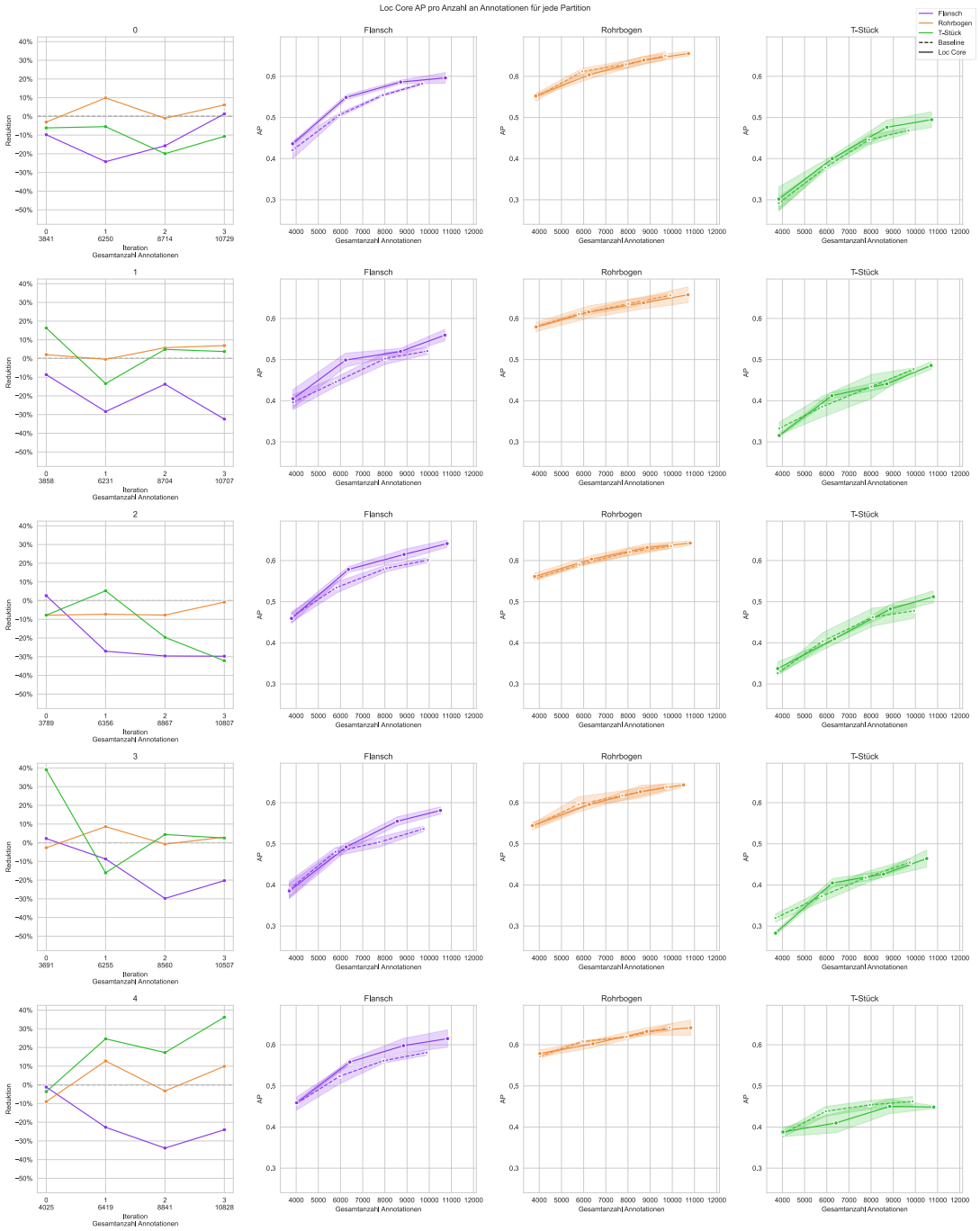


Abbildung 55: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition der Localization Stability mit Core Set

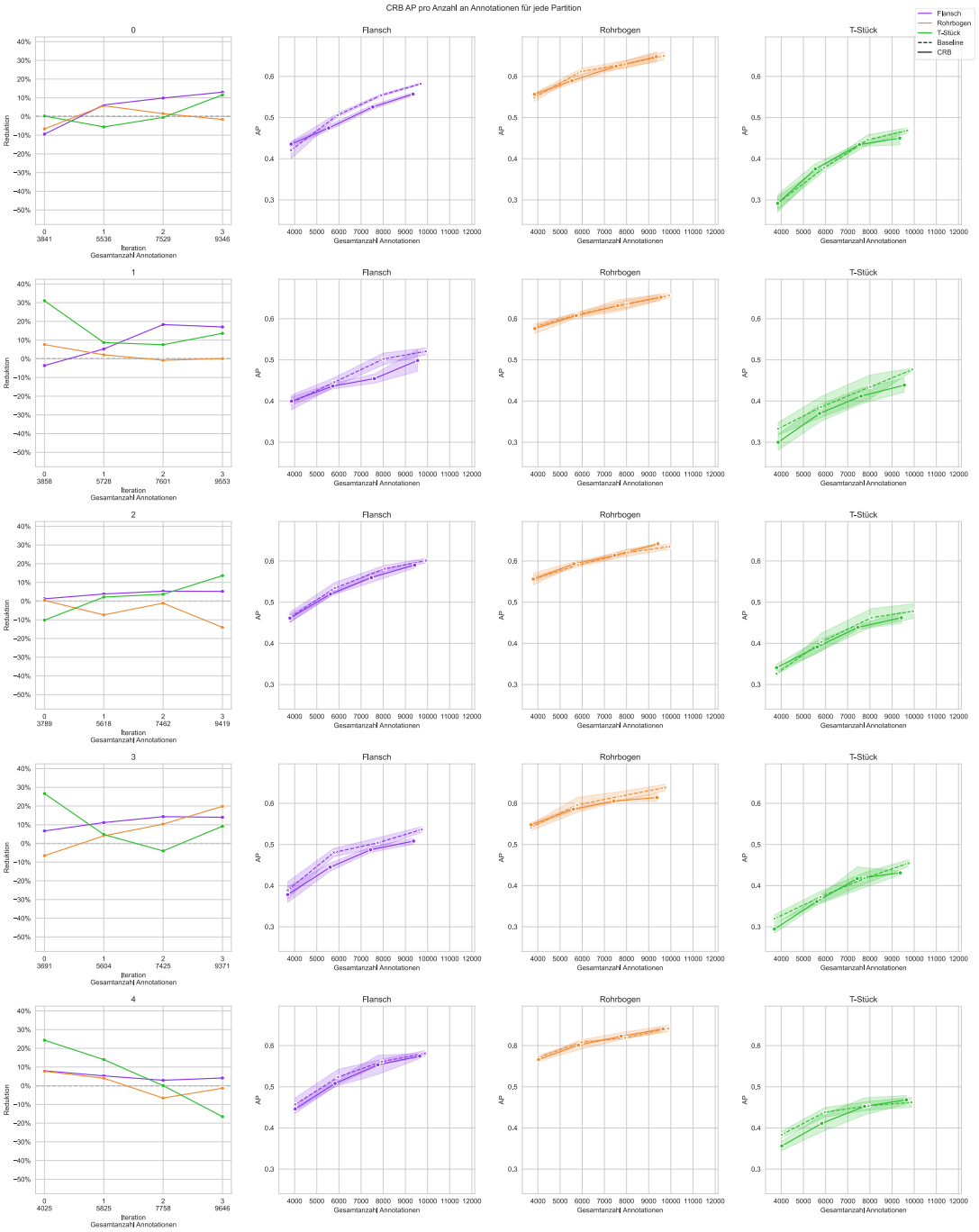


Abbildung 56: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition von CRB

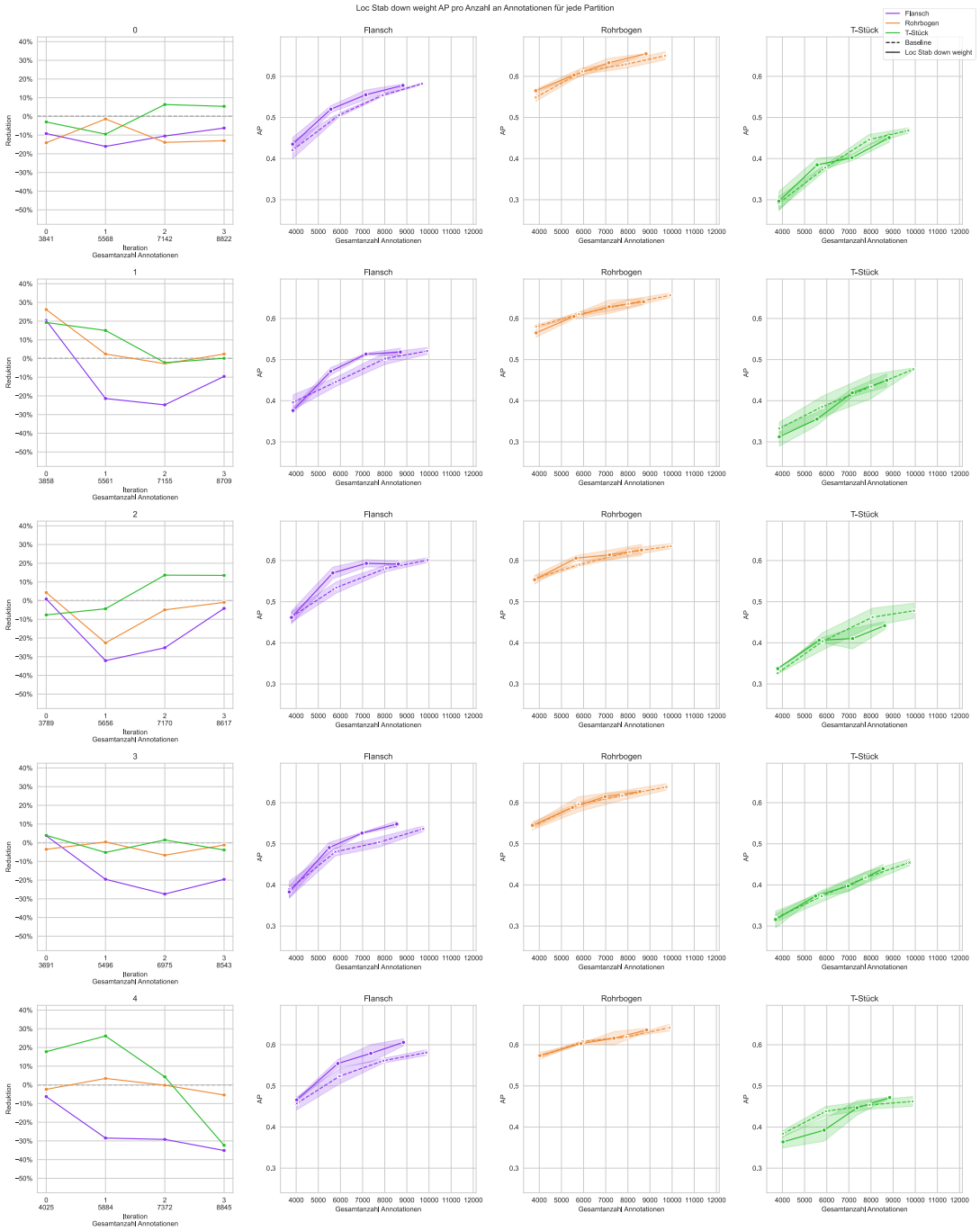


Abbildung 57: Erzielte Reduktion und AP für die Gesamtanzahl an Annotationen für jede Partition der gewichteten Localization Stability