Fraunhofer

Dascha Karelina | Bernd Meese

Transformer in der Bildverarbeitung

Hrsg.: Thomas Bauernhansl, Marco Huber, Werner Kraus

Im Rahmen des

KI Fortschrittszentrum
LERNENDE SYSTEME UND KOGNITIVE ROBOTIK

Vorwort

Künstliche Intelligenz (KI) ist eine der zentralen Technologien für die Zukunft. Ihre Einführung und der Einsatz fordern Unternehmen im besonderen Maß heraus. Es gilt, das Potenzial zu erkennen und dieses wirtschaftlich nutzbar zu machen. Lassen Sie sich dabei von Europas größter Forschungskooperation auf dem Gebiet der KI, Cyber Valley, begleiten.

Mit dem KI-Fortschrittszentrum vom Fraunhofer-Institut für Arbeitswirtschaft und Organisation IAO und Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA und als Teil von S-TEC, dem Stuttgarter Technologie- und Innovationscampus, unterstützen wir Unternehmen dabei, das Potenzial von KI nutzbringend einzusetzen. An der Schnittstelle zwischen anwendungsorientierter Wissenschaft und exzellenter Forschung des Cyber-Valley-Konsortiums entwickeln wir innovative KI-Anwendungen für die Praxis und treiben damit die Kommerzialisierung von KI voran. Erklärtes Ziel ist dabei, menschzentrierte KI-Lösungen zu entwickeln. Denn nur wenn Menschen mit einer neuen Technologie intuitiv interagieren und vertrauensvoll zusammenarbeiten, kann ihr Potenzial optimal ausgeschöpft werden.

Die Studienreihe »Lernende Systeme und Kognitive Robotik« des KI-Fortschrittszentrums gibt Einblick in die Potenziale und die praktischen Einsatzmöglichkeiten von KI. Dabei wurden bereits übergreifende Themen wie Zuverlässigkeit, Erklärbarkeit (xAI), cloudbasierte Plattformen, Technologien und Einführungsstrategien sowie einzelne Anwendungsbereiche vorgestellt. Mit der Fortsetzung der Reihe kommen zahlreiche neue Themen hinzu. Diese reichen von der Bildverarbeitung über die effiziente Software- und Systemintegration in der Robotik, die Gestaltung von KI-Systemen, Sprachmodelle in der Praxis bis hin zu KI-basierten Assistenzsystemen und dem Beitrag von KI zur Fachkräftesicherung.

Im Rahmen der vorliegenden Studie wurde der Einsatz von Transformer-Modellen in der industriellen Bildverarbeitung hinsichtlich unterschiedlicher quantitativer Kriterien untersucht. Die Studie bietet eine Analyse und einen Überblick über verschiedene Transformer-Architekturen und deren Anwendung in der Computer Vision, unterstützt durch Experimente und Benchmarks. Die Ergebnisse der Analyse sowie das kritische Gegenüberstellen der Vor- und Nachteile der Technologie dienen dazu, geeignete Verfahren für spezifische Anwendungsfälle auswählen und kritisch mit aktuellen Entwicklungen umgehen zu können.

Wir wünschen Ihnen eine spannende Lektüre, und freuen uns, wenn wir in Zukunft auch Sie mit unserer Expertise auf Ihrem Weg zur menschzentrierten KI unterstützen dürfen.

Thomas Bauernhansl, Marco Huber, Werner Kraus
Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA

Inhalt

Management Summary

In den letzten Jahren finden Methoden der Künstlichen Intelligenz (KI) vermehrt Anwendung in unterschiedlichsten Bereichen der industriellen Bildverarbeitung. Durch KI können komplexe Probleme vermehrt mit hoher Genauigkeit gelöst werden, was zu gesteigerter Produktqualität, Prozessoptimierung und Kosteneinsparungen führt. Einen neuen Trend bilden hierbei Transformer-Architekturen, die über ihre Verbreitung in der Sprachverarbeitung auch den Weg in die Bildverarbeitung gefunden haben. Die zunehmende Beliebtheit von den Transformer-Modellen, die durch ihren Self-Attention-Mechanismus globale Kontextinformationen erfassen können, hat einen bedeutenden Einfluss auf die Weiterentwicklung von KI-Technologien, auch im Bereich der Bildverarbeitung. Gleichzeitig stellen die neuen Methoden Unternehmen vor neue Herausforderungen und stehen in Konkurrenz mit den herkömmlichen Methoden wie Convolutional Neural Networks (CNNs). Die vorliegende Studie beschäftigt sich daher mit der Entwicklung in der Bildverarbeitung und der kritischen Gegenüberstellung von Transformer-Ansätzen zu traditionellen CNNs.

Hierfür stellt die Studie einen umfassenden Überblick über die Entwicklung von Computer-Vision-Methoden dar, von traditionellen CNNs bis hin zu modernen Transformer-Architekturen. Die Ergebnisse der im Rahmen der Studie durchgeführten Benchmarking-Experimente liefern einen objektiven Qualitätsvergleich zwischen Transformer-Modellen und CNNs. Der Vergleich umfasst verschiedene Anwendungsfälle wie Objekterkennung und Segmentierung im 2D- sowie 3D-Kontext. Weiterhin werden die Potenziale und Grenzen von Transformern diskutiert, insbesondere in Bezug auf Datenbedarf, Rechenaufwand und Inferenzzeiten. All das soll den praktischen Einstieg in den Einsatz von Transformer-Modellen für eigene Anwendungsfälle erleichtern.

1. Einleitung und Motivation

Bildverarbeitung kommt zunehmend in der Fertigungsindustrie zum Einsatz. Dies ist durch mehrere Vorteile begründet. Bildverarbeitung ermöglicht:

- eine genaue und automatische Qualitätsprüfung von Produkten
- Effizienzsteigerungen durch Prozessüberwachung und damit verbundener -optimierung
- Kosteneinsparungen durch frühzeitiges Erkennen von Fehlern und Defekten
- Konsistenz in der Bewertung von Prozessparametern und Produktqualität
- Echtzeitentscheidungen über die Produktqualität, was insbesondere für großvolumige Produktionsumgebungen relevant ist

Insgesamt trägt die industrielle Bildverarbeitung dazu bei, die Produktqualität zu steigern, Prozesse zu optimieren, Kosten zu senken und die Wettbewerbsfähigkeit von Unternehmen zu erhöhen. Der Einsatz Künstlicher Intelligenz (KI) in der Bildverarbeitung spielt dabei in den vergangenen Jahren eine zunehmend wichtige Rolle. Mit Bildverarbeitung, im deutschen Sprachgebrauch oft als Synonym für Computer Vision verwendet, können in Verbindung mit KI komplexe Probleme in bislang nicht gekannter Genauigkeit gelöst werden, sodass sich die oben genannten Vorteile noch verstärken. Wegen des sich daraus ergebenden, enormen Potenzials von Computer Vision wird für den deutschen Computer Vision-Markt allein zwischen 2023 und 2030 mit mehr als einer Verdopplung des generierten Umsatzes von knapp einer halben auf knapp eine Mrd. Euro gerechnet (globaler Anstieg von 11 auf 21 Mrd. Euro, Statista).

Zu den in der Computer Vision verwendeten, tiefen neuronalen Netzen gehören auch sogenannte Transformer-Modelle, die seit der Veröffentlichung des ihnen zugrunde liegenden Attention-Mechanismus [1] in verschiedensten Teildisziplinen immer mehr an Bedeutung gewinnen. In jüngerer Vergangenheit haben Transformer insbesondere wegen ihrer Ursprungsdomäne, dem Natural Language Processing, an Bekanntheit gewonnen. Sprachmodelle wie ChatGPT (OpenAI) oder Bard (Google) demonstrieren einer breiten Öffentlichkeit in beeindruckender Form, dass Transformer-basierte Anwendungen fähig sind, auf komplexe Weise mit Sprache umgehen und menschenähnliche Dialoge führen zu können.

Auch wenn sie ursprünglich für die Anwendung auf sequenzielle Datentypen wie Buchstabenfolgen oder Zeitreihen konzipiert sind, wird das Potenzial von Transformer-Architekturen auch im Kontext anderer Bereiche des maschinellen Lernens untersucht. Die erste Studie, die größere Aufmerksamkeit für die Implementierung eines Transformer-Modells in der Bildverarbeitung erlangt hat, wurde im Jahr 2020 veröffentlicht [2]. In dieser Studie wurde zwar »nur« der bis dato geltende Stand der Technik der Bildklassifizierung erreicht, allerdings schon mit deutlich geringerem Zeitaufwand beim Training der KI. Da das Potenzial von Transformer-Modellen in der Bildverarbeitung wiederholt belegt werden konnte, schreitet die Entwicklung in rasantem Tempo voran, wobei das Spektrum mittlerweile neben verschiedenen 2D-Aufgaben um die Domäne der 3D-Daten erweitert wurde.

Um einen Überblick über diese dynamische Entwicklung zu ermöglichen, beleuchtet diese Studie den Trend der Transformer-Modelle in der Bildverarbeitung. Zunächst wird eine Übersicht über die Entwicklung und Unterschiede verschiedener Modelle in der Computer Vision gegeben. Ergänzend werden Experimente mit verschiedenen Modellen durchgeführt. Die Evaluation ausgewählter Methoden hinsichtlich diverser Bewertungskriterien soll es erleichtern, ein passendes Verfahren für ihren Anwendungsfall auszuwählen und kritisch mit dem Trend umzugehen.

Die Studie ist folgendermaßen aufgebaut: Kapitel 2 bietet einen Überblick über das Forschungsfeld Computer Vision und stellt eine Übersicht über die Entwicklung der Methoden dar. Kapitel 3 bietet zunächst eine Übersicht über die in der Studie betrachteten Benchmark-Datensätze sowie Evaluationsmetriken und beinhaltet die Studienergebnisse sowie deren Diskussion. Zuletzt folgt in Kapitel 4 eine Übersicht relevanter Aspekte in der Praxis und weitere Anwendungsfelder von Transformern in der Computer Vision.

2. Computer Vision, CNN und Transformer

Nach dem Erfolg von Transformer-Architekturen in der Sprachverarbeitung manifestiert sich der Trend hin zu Transformern seit einiger Zeit auch in der Computer Vision. In diesem Kapitel werden sowohl der Forschungsbereich Computer Vision und die beleuchteten Aufgabenfelder als auch der erwähnte Trend und seine Gründe im ersten Unterkapitel vorgestellt und erläutert. Anschließend werden Transformer und ihre Architektur im allgemeinen Kontext der Computer Vision eingeführt und anschließend eine grobe Übersicht über die Methoden des aktuellen Stands der Technik gegeben.

2.1. Einführung in die Computer Vision

Computer Vision ist ein Forschungsgebiet, das sich mit der Entwicklung von Techniken zur Verarbeitung und Analyse visueller Informationen beschäftigt. Das Hauptziel von Computer Vision ist es, die Funktionen des menschlichen visuellen Systems technisch umzusetzen und gegebenenfalls zu verbessern. Es soll computergestützten Anwendungen ermöglichen, visuelle Eingabedaten in Form von Bildern oder Videos zu verstehen und nützliche Informationen zu extrahieren und zu interpretieren [3]. So können im Industrie- oder Produktionskontext Personen in einem Gefahrenbereich oder Qualitätsmängel wie Kratzer auf einem Bauteil erkannt werden. In vielen Computer-Vision-Anwendungen kommt dabei Künstliche Intelligenz zum Einsatz. Zu allgemeinen Begrifflichkeiten sei angemerkt, dass »Image Processing«, im Deutschen »Bildverarbeitung«, nach [4] ein Teilgebiet der »Computer Vision« darstellt, das für die (Vor-)Verarbeitung von 2D-Bildern und 3D-Daten verantwortlich ist. Entgegen der Definition wird der Begriff »Bildverarbeitung« im deutschen Sprachgebrauch oft als Synonym für »Computer Vision« benutzt. In dieser Studie wird meist der weiter gefasste Begriff Computer Vision genutzt.

2.1.1. Computer-Vision-Aufgabenbereiche

Die visuellen Eingabedaten, die verarbeitet werden, können sowohl als Kamerabilder als auch als 3D-Daten in Form von Tiefenbildern oder Punktwolken vorliegen. Die 2D-Bildverarbeitung befasst sich mit der Analyse und Verarbeitung von zweidimensionalen Bildern mit einer Breite b und einer Höhe h, die als geordnetes Array von $b \times h$ Pixeln für jeden der Farbkanäle bestehen. Im Falle von RGB-Farbbildern hat das Array das Format $b \times h \times n$ mit $n = 3$, also einem Kanal für jede Farbe. Bei einem Grauwertbild ist $n = 1$. In der 3D-Bildverarbeitung wird zusätzlich die räumliche Tiefe betrachtet. Die Daten liegen in der Regel nicht in einem vollständigen Array vor, vielmehr entsprechen sie ungeordneten Listen von Punkten, die jeweils durch eine x-, y- und z-Koordinate beschrieben und gegebenenfalls mit einer Farbinformation ergänzt sind.

Unabhängig von der Form und Dimension der Eingangsdaten überschneiden sich die Aufgabenfelder der Computer Vision. Sie reichen von grundlegenden Operationen wie dem Entfernen von Rauschen über das Einteilen eines Bildes in Bereiche/Segmente bis hin zu komplexeren Verfahren: Drei fundamentale Bereiche von Computer Vision sind die Bildklassifikation, Objekterkennung und Segmentierung (Abbildung 1).

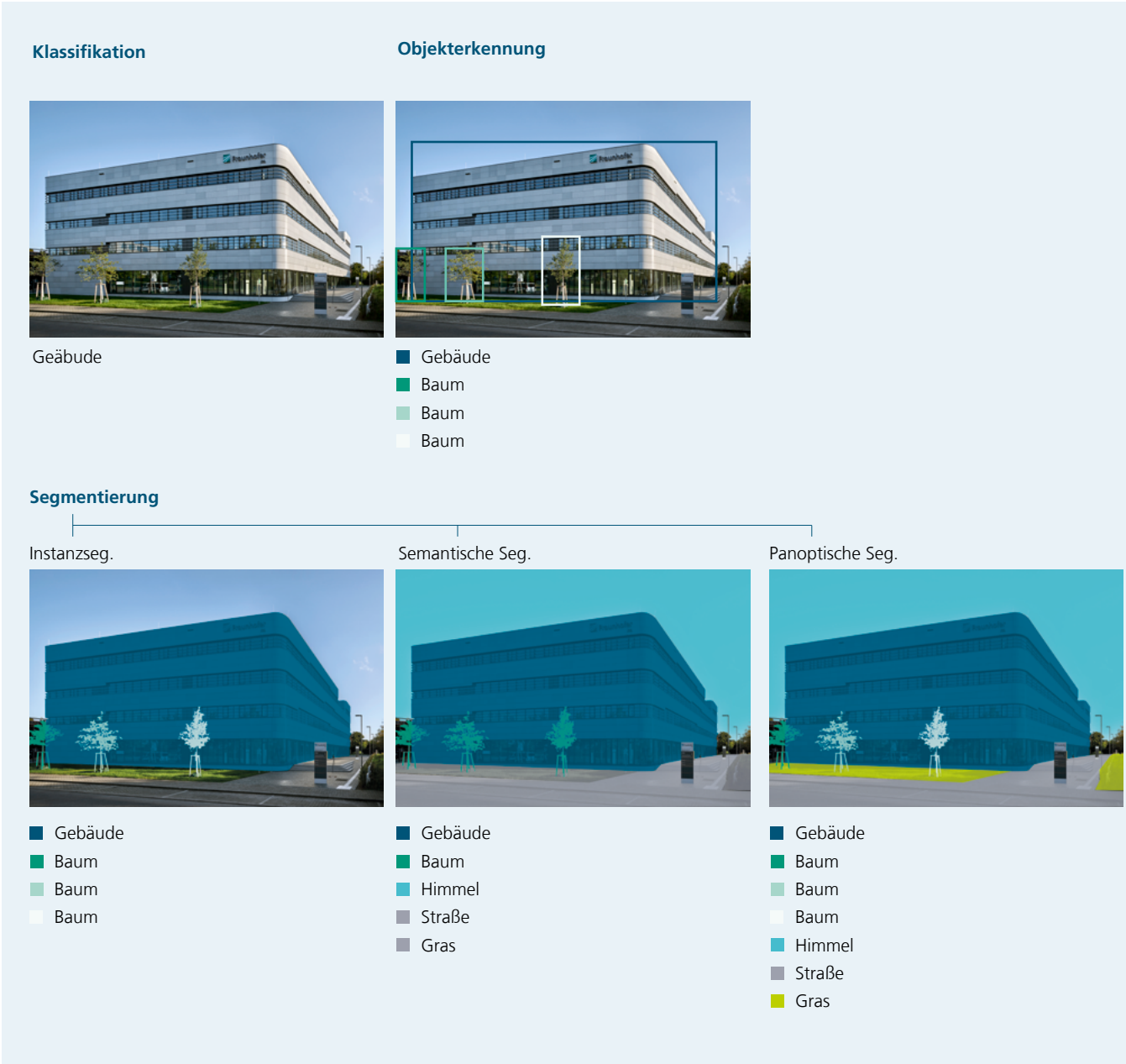


Abbildung 1: Computer-Vision-Aufgabenbereiche anhand eines Beispielbildes. Quelle: Fraunhofer IPA/Foto: Oliver Jaist.

Die **Klassifikation** ist eine der grundlegenden Aufgaben in der Computer Vision und befasst sich mit der Zuordnung einer Klassenbezeichnung zu einem Eingabebild. Sie zielt darauf ab, einem Bild ein eindeutiges Label zuzuordnen. Dabei wird meist das Hauptobjekt des Bildes erkannt. Genauso kann aber auch das Erkennen von Klassen wie »Objekt in Ordnung« und »Objekt defekt« trainiert werden. Der Aufgabenbereich **Objekterkennung** geht einen Schritt weiter und bezieht sich sowohl auf die Identifizierung als auch die Lokalisierung von mehreren Objekten verschiedener Klassen in einem Bild. Im Gegensatz zur Bildklassifikation muss hier

nicht nur das Hauptobjekt benannt, sondern die Positionen und Label aller erkannten Objekte müssen ausgegeben werden. Dies geschieht in der Regel in Form von rechteckigen Umrahmungen der Objekte durch sogenannte Bounding Boxes. Es können je nach Anwendung Objekte wie »Hund«, »Fahrrad« oder ähnliches detektiert werden, aber auch spezifische Fehlerbilder wie »Kratzer« oder »Lufteinschluss«. Der dritte Bereich, die **Segmentierung**, befasst sich mit der Aufteilung eines Bildes in mehrere Segmente, die jeweils einem bestimmten Objekt, Objekttyp oder einer Region im Bild entsprechen. Das Ziel der Bildsegmentierung ist das Zusammenfassen und Trennen von Pixeln basierend auf bestimmten Merkmalen. Man erhält so nicht nur die Positionen, sondern auch genaue Umrisse der Objekte, sogenannte Masken. Die Segmentierung wird noch mal in drei verschiedene Aufgaben untergliedert, je nachdem, ob man einzelne Objekte, Objekttypen oder beides unterscheiden möchte: semantische Segmentierung, Instanzsegmentierung und panoptische Segmentierung [5].

- Bei der **semantischen Segmentierung** werden Bildbereiche, die dem gleichen Objekttyp entsprechen, zusammengefasst, unabhängig von der Lage im Vorder- oder Hintergrund. Es gibt nur so viele semantische Segmente, wie es erkannte Objekttypen gibt. Jedem Pixel im Bild wird einem Segment zugeordnet.
- Bei der **Instanzsegmentierung** geht es darum, einzelne Instanzen von Objekten in einem Bild zu isolieren, der Hintergrund wird ignoriert. Jedes Segment erhält eine eindeutige Kennzeichnung, unabhängig davon, ob es sich um denselben Objekttyp handelt. Damit kann man die Instanzsegmentierung als Präzisierung der Objekterkennung mittels Masken sehen. Es gibt so viele Segmente, wie Objekte im Bild erkannt werden. Nur die Pixel erkannter Objekte werden Segmenten zugeordnet.
- Die Kombination dieser beiden Aufgaben wird **panoptische Segmentierung** genannt. Es werden nicht nur Objektinstanzen im Vordergrund identifiziert, sondern auch Pixel, die zu allgemeinen semantischen Klassen gehören und meist den Hintergrund darstellen. Es gibt so viele Segmente, wie Objekte im Bild erkannt werden. Jedem Pixel im Bild wird einem Segment zugeordnet.

Diese Definitionen haben sich zunächst für 2D-Daten etabliert, wurden aber auch für 3D-Daten übernommen. Das heißt, dass diese Unterscheidungen auch für die Segmentierung von Punktwolken existieren.

In der Produktion geht es oft darum, bildbasierte Entscheidungen zu treffen. Hierfür werden vor allem Objekterkennung, Instanzsegmentierung und semantische Segmentierung eingesetzt. Entsprechend werden die im Rahmen dieser Studie beschriebenen Übersichten und Analysen für diesen spezifischen Anwendungsbereich näher betrachtet.

2.1.2. Backbone-Neck-Head-Architektur

In verschiedenen Aufgabenfeldern der Computer Vision, auch über die genannten hinaus, finden neuronale Netzwerkarchitekturen ihre Anwendung. Zur Beschreibung der verschiedenen Funktionen des Gesamtmodells wird eine Unterteilung in »Backbone«, »Neck« und »Head« benutzt (Abbildung 2). Die metaphorisch eingeführten Begriffe lehnen sich an das menschliche Skelett an, bei dem der Kopf auf dem Rückgrat sitzt und durch den Hals verbunden wird. Analog dazu bildet der Backbone die Grundlage für das Netzwerk, indem er die Merkmale der Eingabedaten extrahiert. Man spricht beim Backbone auch vom Feature Extractor Network, das eine oder mehrere Merkmalskarten des Eingabebilds erzeugt. Oft kann dieser Teil in einer Netzwerkarchitektur ausgetauscht werden. Im Head werden die extrahierten Merkmale verwendet, um die anzugehende spezifische Aufgabe zu lösen und beispielweise eine Klassifizierung durchzuführen. Als Verbindungsteil agiert der Neck. Dort werden die Informationen aus dem Backbone für die spezifische Aufgabe des Heads vorbereitet. Es ist zu beachten, dass diese metaphorische Aufteilung nicht für jedes publizierte, neuronale Netz strikt vorgenommen wird, jedoch oft anzutreffen ist.

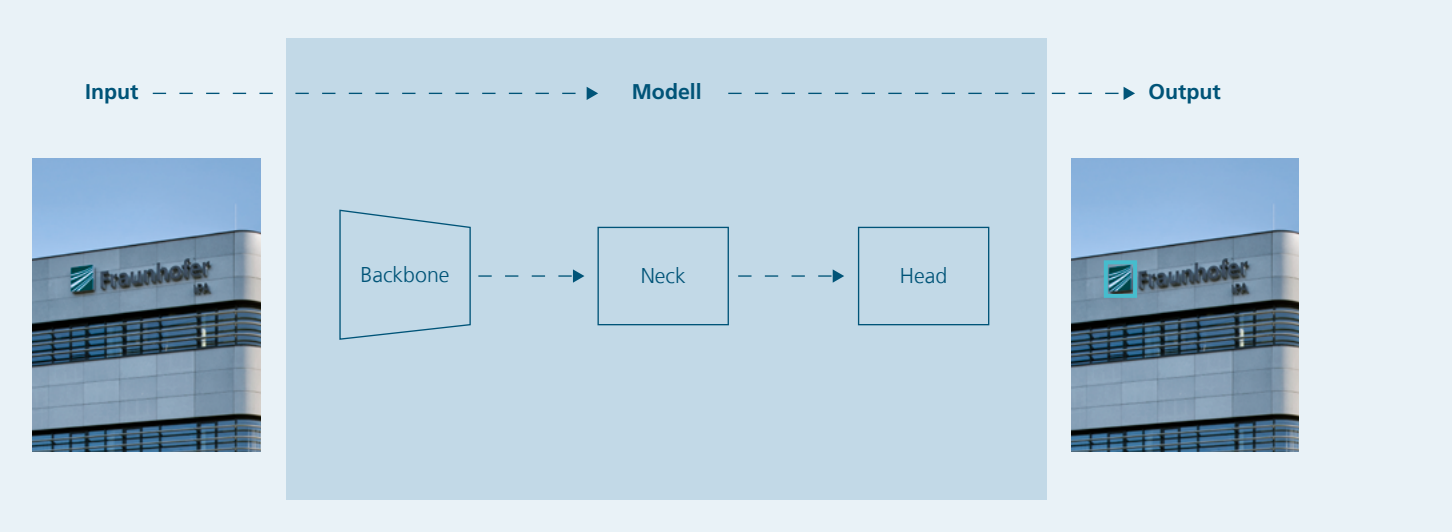


Abbildung 2: Vereinfachte Darstellung der Pipeline von ML-Modellen mit Backbone-Neck-Head-Architektur. Quelle: Fraunhofer IPA/Foto: Oliver Jaist.

2.1.3. Von CNNs zu Transformern

Seit der Veröffentlichung von AlexNet [6] haben sich CNNs als äußerst erfolgreiche Architekturen für die Computer Vision etabliert. Der Kern des Erfolgs von CNNs liegt in der angewendeten mathematischen Operation der Faltung (engl. convolution), die sich im Namen der Architektur widerspiegelt. Sie fungiert als fundamentaler Operationsschritt und grenzt CNNs von anderen neuronalen Netzen ab.

Die Faltungsoperation erfolgt durch das Anwenden eines gewichteten Filters, der auch als Kernel oder Gewichtsmatrix bezeichnet wird. In verschiedenen Schichten des CNNs werden verschiedene Filter angewendet, deren Gewichte während des Trainings iterativ angepasst werden. Bei Eingabedaten in Form von Bildern wird, wie in Abbildung 3 verdeutlicht, der Filter über das Bild geschoben und an jeder Position elementweise mit den Eingabedaten, den Pixelwerten, multipliziert und anschließend aufsummiert. So ergibt sich an jeder Position des Filters ein neuer Wert, der in die Ausgabeschicht, eine sogenannte Feature Map, geschrieben wird. Die Anzahl der berücksichtigten Nachbarpixel hängt von der Größe des Filters ab. Dass dieser immer nur einen Teil des Bildes abdeckt, bedingt den lokalen Charakter der CNNs. So können Faltungen in speziellen Faltungsschichten der CNNs verwendet werden, um lokale Merkmale wie beispielsweise Ecken und Kanten (siehe Abbildung 3) in den Eingabedaten zu identifizieren und zu extrahieren. Diese Merkmale werden anschließend in tieferen Schichten des Netzwerks aggregiert und kombiniert, um sukzessive immer komplexere und abstraktere Merkmalsrepräsentationen zu erzeugen. Je tiefer eine Schicht im Netz liegt, desto komplexer sind die Merkmale, die sie repräsentiert. Die schrittweise Hierarchie dieser Merkmalsdarstellungen ermöglicht dem CNN, Abstraktionsebenen zu erlangen, die schließlich zur Durchführung von komplexen Aufgaben der Computer Vision, wie dem Erkennen von Objekten in einem Bild, verwendet werden können [7].

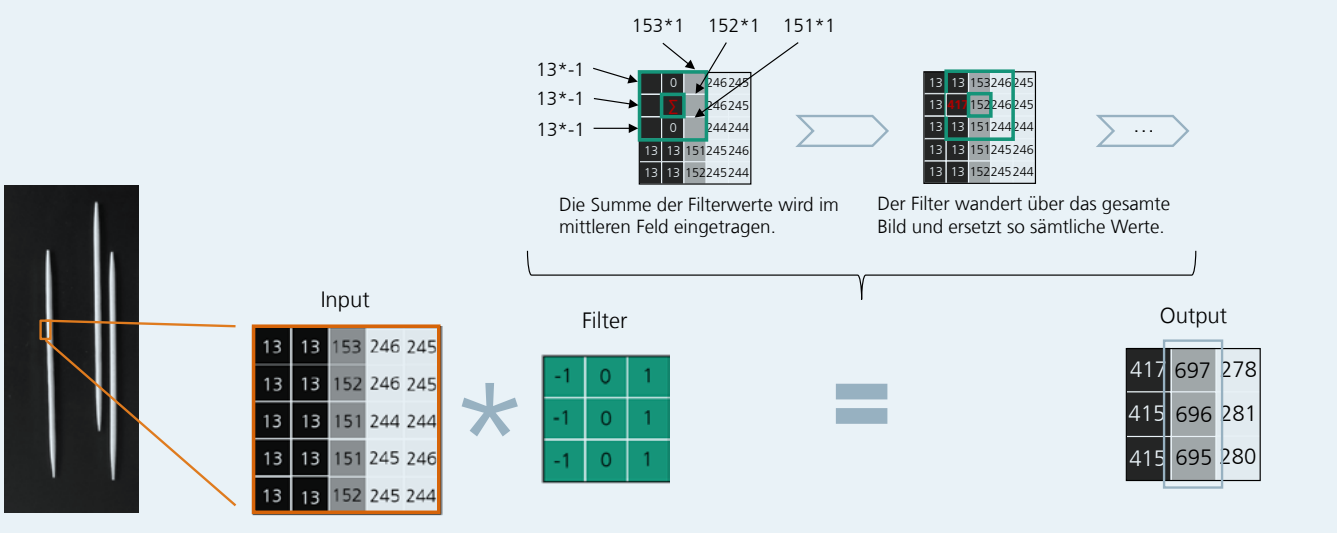


Abbildung 3: Faltungsoperation am Beispiel einer Kante in einem Grauwertbild in Form einer Matrix. Der Filter sorgt dafür, dass sich der Verlauf der vertikalen Kante im Bild durch sehr hohe Werte im Output widerspiegelt. Quelle: Fraunhofer IPA.

Aufgrund des lokalen Charakters von CNNs besteht die Möglichkeit, dass wertvolle Informationen bei der Analyse von Bildern nicht beachtet werden. Diese Annahme der Lokalität wird als Inductive Bias (induktive Voreingenommenheit) der CNNs bezeichnet (z.B. in [4]). Aufgrund dieser Annahme stehen CNNs seit einiger Zeit im Wettbewerb mit Transformern, die in ihrer Struktur Encoder-Decoder-Architekturen sind, die auf dem Attention-Mechanismus basieren [3]. Im Zusammenhang mit Computer Vision spricht man auch von Vision Transformern. Die Verwendung von Transformern im Bereich des maschinellen Sehens ist durch ihre Fähigkeit motiviert, globale Kontextinformation erfassen zu können. Im Gegensatz zu den lokalen Abhängigkeiten der Faltungsoperationen von CNNs können Vision Transformer mittels des Self-Attention-Mechanismus den Kontext der gesamten Eingabedaten analysieren und effektiv Beziehungen zwischen verschiedenen Bildbereichen oder Videosequenzen modellieren [4].

Bei der Auswertung von 3D-Daten können die beschriebenen Faltungsoperationen nicht direkt auf die Eingabedaten angewendet werden, da die Daten nicht als dreidimensionale Arrays der Form *b x h x n*, sondern als Punktwolken in Form ungeordneter Listen vorliegen. Die meisten Verfahren für 3D-Daten nutzen Multi Layer Perceptrons (MLP), die lokale und globale Features von Inputvektoren (bzw. Punkten) extrahieren und diese Features invariant zur Reihenfolge oder Länge der Inputdaten in einen neuen Vektor fester Länge aggregieren (z.B. Pointnet++, [8]). Diese globale Darstellung der Punktwolke kann anschließend zur weiteren Verarbeitung, beispielsweise zur Klassifizierung oder Segmentierung, analog zu den 2D-Daten verwendet werden.

2.2. Transformer-Architektur

2.2.1. Encoder-Decoder-Architektur

Transformer sind in der Regel Encoder-Decoder-Architekturen. Die Eingabedaten gelangen zunächst in den Encoder, der die Daten in einen Vektor kodiert, der die Eingabe möglichst gut repräsentiert. Der von allen Repräsentanzvektoren aufgespannte Vektorraum wird oft als verdeckter Raum (engl. latent feature space oder embedding space) bezeichnet. Die Repräsentation der Eingabe wird anschließend zur Ausgabe-Dekodierung in den Decoder übergeben (Abbildung 4). Je nach Aufgabe und Implementierung kann das Ziel des Decoders sein, den Repräsentanzvektor wieder in die Eingabedaten umzuwandeln oder wie bei der Textübersetzung eine Entsprechung der Eingabe in einer anderen Sprache auszugeben. Beides kann nur gelingen, wenn das Modell lernt, für jede Eingabe möglichst eineindeutige Repräsentanzvektoren zu erzeugen. Wie im Kapitel zum Stand der Technik beschrieben, wird in manchen Anwendungen mit Transformer-Architektur nur der Encoder-Teil des trainierten Netzes verwendet und die Objekterkennung oder Segmentierung bereits anhand des Repräsentanzvektors durchgeführt. Im Gegensatz dazu basiert das Sprachmodell GPT auf einem Decoder ohne Encoder. Aber unabhängig von den Details einer Implementierung bestehen bei Transformern sowohl der Encoder als auch der Decoder aus mehreren Blöcken, die jeweils alle den für Transformer charakteristischen Self-Attention-Mechanismus verwenden.

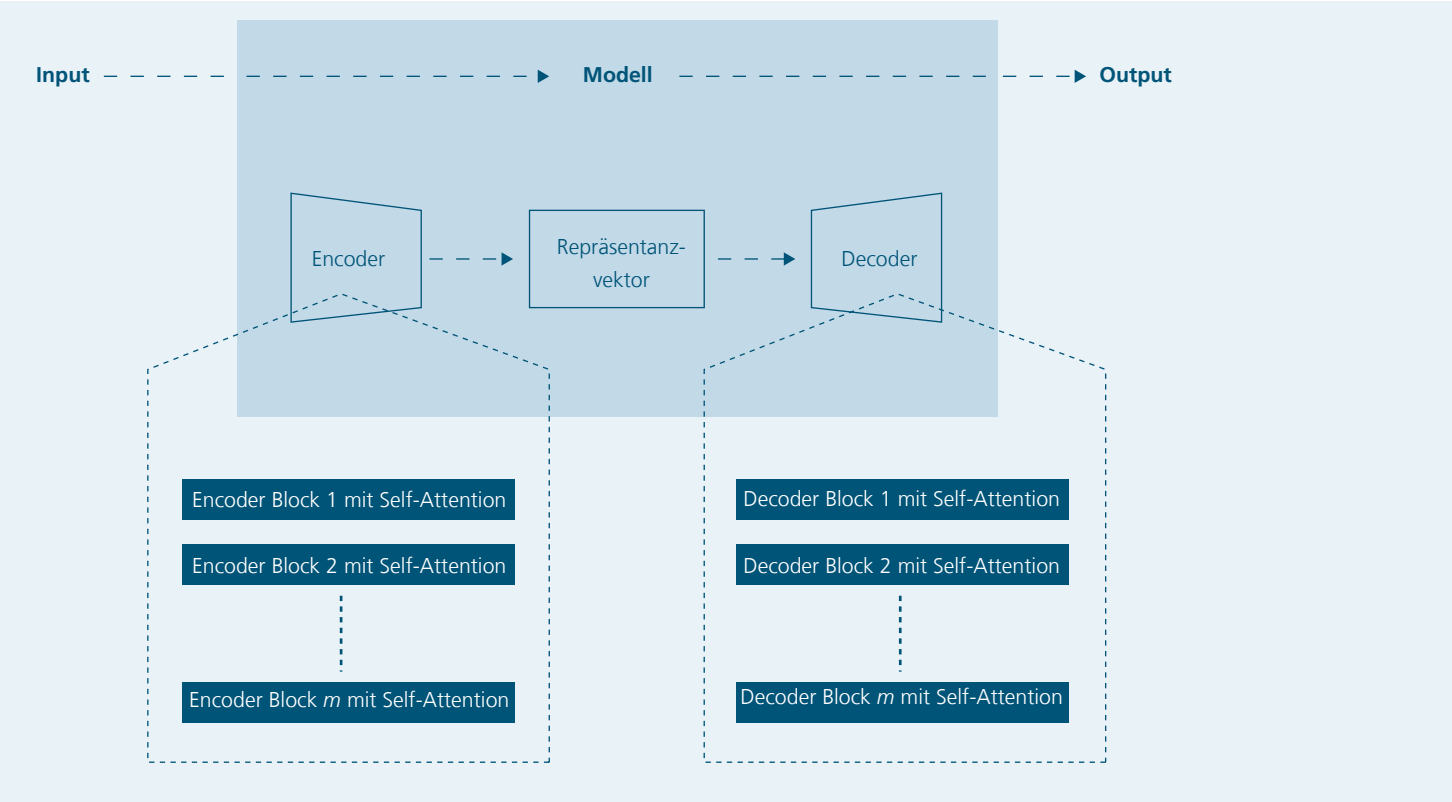


Abbildung 4: Gängige Encoder-Decoder-Architektur mit Self-Attention Mechanismus in Encoder- und Decoder-Blöcken. Quelle: Fraunhofer IPA.

2.2.2. Self-Attention-Mechanismus

Die Komponente, die die Transformer besonders auszeichnet, ist der Self-Attention-Mechanismus. Dieser ermittelt den Zusammenhang zwischen den einzelnen Elementen einer Eingabesequenz wie z.B. den einzelnen Wörtern in einem Satz. Die Besonderheit des Mechanismus ist seine Fähigkeit, globale Abhängigkeiten zu erfassen. Satzübergreifend erfasst der Mechanismus beispielsweise, welche Wörter zusammen verwendet werden und sich auch bei großer Distanz aufeinander beziehen [1]. Der Self-Attention-Mechanismus berechnet eine sogenannte Attention Map zwischen den verschiedenen Eingabeelementen. Es entscheidet sich hier, wie viel Aufmerksamkeit/Gewicht jedes Element bei dem Encoding des aktuell betrachteten Elements erhält [1]. Da der Zusammenhang aller Elemente zueinander betrachtet wird, ist die Rechenkomplexität quadratisch, d. h. bei n Elementen einer Sequenz beträgt die Komplexität $O(n^2)$. In einigen aktuellen Algorithmen werden mehrere Self-Attention-Mechanismen kombiniert, man spricht dann von Multi-Head Self-Attention (z.B. [9]). Sie kommt zum Beispiel beim Multi-modal Self-Attention zum Einsatz, das ermöglicht, mehrere verschiedene Modalitäten (z. B. Text, Bild, Audio) gleichzeitig zu verarbeiten und deren Informationen zu kombinieren. Dabei wird der Self-Attention-Mechanismus so angepasst, dass er die Beziehungen und Abhängigkeiten zwischen den verschiedenen Modalitäten erfassen kann.

2.2.3. Vom Satz zum Bild

In der Computer Vision liegen Eingabedaten in Form von Bildern vor. Zur Verarbeitung mit Self-Attention müssen die Bilder ähnlich wie Sätze eines Texts zunächst in Unterstrukturen aufgebrochen werden. Eine Aufteilung eines Bilds in einzelne Pixel verbietet sich ähnlich wie die Verwendung von Buchstaben eines Satzes wegen der quadratischen Komplexität des Mechanismus. Stattdessen entsprechen die einzelnen Eingabeelemente bei Vision Transformern im einfachsten Fall rechteckigen Teilbereichen des Eingabebilds, in der Fachsprache Patches genannt [2]. Das bedeutet, dass vor der Anwendung eines Transformers im ersten Schritt zunächst eine Aufteilung des Bilds in eine Reihe von kleineren Patches erfolgt, die als Eingabe für den Transformer dienen (z.B. [10]). Analog zu Wörtern eines Satzes in der Sprachverarbeitung entspricht jeder Patch einem Wort. Mitunter werden für Vision Transformer allerdings keine Patches von dem Eingabebild selbst verwendet, sondern Patches der Feature Maps, die zuvor ein Feature Extractor liefert.

2.3. Transformer – Stand der Technik

Seitdem die Forschung versucht, Transformer von NLP auf die Computer Vision zu übertragen, wurden verschiedene Ansätze entwickelt. Neben reinen Transformer-Architekturen gibt es mittlerweile verschiedene Kombinationen von Transformern und CNNs. So sollen die Vorteile beider Domänen verbunden und vor allem die Fähigkeit der CNNs, lokale Informationen zu extrahieren, genutzt werden [3]. Hierbei kann man grundsätzlich zwischen Architekturen mit Transformer Backbone und CNN Head oder andersherum, mit CNN Backbone und Transformer Head, unterscheiden (z.B. [9], [3]). Im Folgenden werden verschiedene Methoden aus diesen Kategorien im 2D- und 3D-Bereich vorgestellt, wobei der Fokus auf historisch relevanten sowie auf in der Studie verwendeten Modellen liegt. Die Auswahl der verwendeten Modelle beruht dabei auf drei Faktoren: erstens der freien Verfügbarkeit der Modelle, zweitens dem Training auf denselben Datensätzen, um eine Vergleichbarkeit der Ergebnisse gewährleisten zu können, sowie drittens der Performance der Modelle.

2.3.1. 2D-Transformer

Obgleich die Veröffentlichung von ViT (kurz für Vision Transformer) von [2] nicht die erste war, die den Ansatz verfolgt, Transformer-Modelle in die 2D-Computer-Vision-Domäne zu übertragen, gilt sie retrospektiv als Meilenstein auf dem Gebiet. Das ViT-Modell beinhaltet nur den Encoder-Teil der Transformer-Architektur und ist ein Transformer Backbone. Das Zerlegen eines Bilds in eine Sequenz von Patches und das anschließende Prozessieren der Sequenz mit dem Self-Attention-Mechanismus, um Repräsentanzen für die Bildklassifizierung zu lernen, ist Grundlage aller in der Folge entwickelten Transformer-Modelle in der Bildverarbeitung. Allerdings werden für das (Vor-)Trainieren von ViT im Vergleich zu klassischen CNNs deutlich größere Datenmengen benötigt, um ähnliche Ergebnisse erreichen zu können. Davon abgesehen erzeugt ViT lediglich eine Feature Map pro Bild, deren Berechnungsaufwand quadratisch mit der Bildgröße ansteigt (siehe Self-Attention-Mechanismus). Während das ideal für die reine Bilderkennung kleiner Bilder ist, müssen im Rahmen der meisten Computer-Vision-Aufgaben Objekte verschiedener Größe in einem Bild erkannt oder segmentiert und Merkmale auf verschiedenen Skalen in sogenannten Multi-Scale-Feature-Maps extrahiert werden. Das Lösen dieser und weiterer Probleme haben die in der Folge veröffentlichten Weiterentwicklungen des ursprünglichen ViT zum Ziel.

Der Swin Transformer (Shifted Windows Transformer) [11] ist ein weiterer, wichtiger Transformer Backbone. Er ist einer der erfolgreichsten Lösungsansätze, der die Verarbeitung und Skalierbarkeit höher aufgelöster Bilder ermöglicht. Analog zum ViT-Modell beinhaltet das Swin Modell auch nur die Encoder-Komponente der Transformer-Architektur. Kern des Modells sind die lokale Aufmerksamkeit und die hierarchische Struktur. Analog zu manchen CNNs ist bei Swin eine Struktur angelegt, in der Swin-Transformer-Blöcke hierarchisch auf verschiedenen Ebenen angewendet werden, um Features unterschiedlicher Ausdehnung im Bild extrahieren zu können. Außerdem wird zur Komplexitätsreduktion lokale Aufmerksamkeit mithilfe eines Fensteransatzes eingesetzt. Das Bild wird in sogenannte lokale Aufmerksamkeitsfenster aufgeteilt, innerhalb derer jeweils Attention für je 16 Patches berechnet wird. Durch die hierarchische Struktur werden in der nächsten Hierarchieebene die Fenster automatisch »verschoben«, damit die Beziehung zu allen benachbarten Regionen ermittelt werden kann. Mit jeder weiteren Ebene werden vor der Anwendung des Swin-Transformer-Blocks je 2x2 benachbarte Patches zusammengeführt und die Tiefe der Schichten entsprechend erhöht. Am Ende entsteht so eine Feature Map analog zu klassischen CNNs wie ResNet (Residual Network) [12] oder VGG (Visual Geometry Group) [13].

Neben Architekturen, die CNN Backbones durch Transformer Backbones ersetzen, gibt es auch Architekturen, die Transformer Heads mit Features aus einem CNN Backbone speisen. Letzteres ist bei dem 2D-Objekterkenner DETR [14], dessen Abkürzung für Detection Transformer steht, der Fall. DETR verwendet ein neuronales Netz (wie beispielsweise ResNet 101 [12]) als Backbone, kommt aber ohne Region Proposals oder Anchor Boxes aus, da der Transformer Head direkt den Output des CNN, ein Set von Feature Maps, als Input verwendet. Der Transformer Head besteht sowohl aus einem Encoder als auch einem Decoder. Ähnlich wie bei ViT werden aus den Feature Maps Patches gebildet und als Input für den Encoder verwendet. Im Encoder werden sämtliche Informationen aus den Feature-Maps kombiniert und im Decoder zusammen mit gelernten Object Queries für die Objekterkennung dekodiert (z.B. [15]). Object Queries können vereinfacht als Informationen zu verschiedenen Zielobjekten betrachtet werden, die der Decoder im Repräsentanzvektor zu identifizieren versucht.

Trotz seiner Leistungsfähigkeit zeigt DETR Schwächen bei der Erkennung kleiner Objekte und weist hohe Rechenkosten auf. Diese Herausforderungen wurden in der Nachfolgearchitektur, bekannt als Deformable DETR, erfolgreich adressiert. Weitere Verbesserungen und die Weiterentwicklung der Architektur wurden in Form von Modellen wie DINO [16] und CO-DETR [17] vorgenommen.

Auch im Bereich der Segmentierung werden CNNs und Transformer kombiniert. In dem Model MaskFormer [18] wird ein CNN Backbone für den Pro-Pixel-Einbettungsvektor (engl. embedding vector) und ein Transformer für das Generieren von Klassenvorhersagen und Einbettungsvektoren der Masken eingesetzt. Die Kombination, genauer gesagt das Kreuzprodukt, dieser Komponenten liefert die Maskenvorhersage. Eine Verbesserung des Modells wurde unter dem Namen Mask2Former [19] veröffentlicht.

2.3.2. 3D-Transformer

Es gibt mittlerweile auch eine Reihe von Transformer-Architekturen, die auf 3D-Punktwolken Objekterkennung durchführen oder diese segmentieren. Oft können mit leichten Anpassungen der Modellarchitektur beide Aufgaben gelöst werden, indem der Merkmale extrahierende Backbone dann zum Beispiel mit entsprechend abgeändertem Head genutzt wird. Ebenso wie im 2D-Bereich ist es verbreitet, Transformer mit CNNs zu kombinieren. PointNet++ [8] stellt in der Verarbeitung von 3D-Daten einen Meilenstein dar und wird von vielen Transformer-Modellen als Backbone genutzt. Im Rahmen der Studie wurden Modelle gewählt, die zum Zeitpunkt der Erstellung dieser Studie zu den besten für ihre jeweilige Aufgabe, also 3D-Objekterkennung oder semantische 3D-Segmentierung, gehören (siehe zum Beispiel [20]).

Group-Free 3D Object Detection via Transformers (kurz GroupFree3D) [21] ist ein Beispiel für Architekturen, die ein PointNet++-CNN als Backbone und den Attention-Mechanismus im Head verwenden. Der Backbone kann bei GroupFree3D theoretisch gegen ein beliebiges Netz zur Merkmalsextraktion (Transformer- oder CNN-basiert) getauscht werden. Nach der Merkmalsextraktion aller Punkte werden zunächst regelbasiert Kandidaten für mögliche Objekte in der Punktwolke bestimmt. Für diese Objektkandidaten werden dann mittels eines sechsfach wiederholt angewendeten Attention-Moduls Objektmerkmale bestimmt. Bei jeder Iteration werden Objektvorhersagen generiert. Aus allen Vorhersagen werden am Ende mittels Non Maximum Suppression die final auszugebenden Objekte bestimmt.

OneFormer3D [22] verwendet ebenfalls ein CNN als Backbone, in diesem Fall ein 3D U-Net [23]. Über einen Pooling Layer werden dann sogenannte Superpoint Features erzeugt, die ihrerseits als Eingangsdaten für einen sechsfachen Transformer Decoder verwendet werden. Die besondere Innovation des OneFormer3D Frameworks ist, dass gleichzeitig Instanz- und semantische Segmentierung durchgeführt werden, wodurch sich unterschiedliche Heads für verschiedene Segmentierungsaufgaben sowie die Objekterkennung erübrigen.

Bei Swin3D [24] wird die Backbone-Methode von Swin Transformer [11] zur Merkmalsextraktion für 3D-Punktwolken von Innenraumszenen adaptiert. Hierfür wird die Eingabepunktwolke zunächst in ein 2cm-Voxel-Raster umgewandelt, bevor analog zu Swin auf fünf verschiedenen Skalen Merkmale extrahiert werden. Auf jeder Skalenebene wird der den Attention-Mechanismus enthaltende Swin3D-Block doppelt angewendet, sowohl auf Fenstern des regulären Grids als auch auf den hier in drei Dimensionen verschobenen Grid-Ausschnitten (engl. *shifted windows*). Der Swin3D-Backbone wurde auf einem sehr großen, synthetischen Datensatz trainiert (Structured3D [25]). Für neue Anwendungen kann er beispielsweise mit dem Head des CAGroup3D-Netzes (Wang et al. 2022) verwendet werden, welches in dieser Studie auch für die Objekterkennung genutzt wird, und muss dann nicht komplett neu trainiert werden. Für diese Studie werden Swin3D-S bzw. Swin3D-L verwendet, die sich nicht wesentlich in der Struktur, sondern vor allem in der Anzahl der (zu trainierenden) Parameter unterscheiden (24 Mio. bzw. 61 Mio.) und entsprechende Vorteile hinsichtlich Performanz oder Inferenzzeit bieten.

Superpoint Transformer (SPT) [26] basiert zunächst auf einem effizienten Algorithmus, der Punktwolken in eine hierarchische Super-Point-Struktur überführt und segmentiert. Anschließend nutzt SPT einen Self-Attention-Mechanismus, um die Beziehungen zwischen Superpunkten auf mehreren Skalen zu erfassen und Elemente in der Punktwolke zu klassifizieren. Der Transformer Head ermöglicht, komplexe Beziehungen und Kontextinformationen zwischen den extrahierten Super-Points zu erfassen, was zu einer präziseren und robusteren Segmentierung führt.

3. Evaluation und Vergleich anhand von Praxisbeispielen

Ziel dieses Kapitels ist, die Performance von Transformer- und Nicht-Transformer-Architekturen darzustellen, um eine erste Einschätzung dieser zwei Architekturtypen zu ermöglichen.

Es werden die im Rahmen dieser Studie durchgeführten Experimente und deren Ergebnisse beschrieben. Zunächst werden dazu Bewertungskriterien und Rahmenbedingungen der Experimente eingeführt. Die Ergebnisse der Experimente werden anschließend vorgestellt und diskutiert.

3.1. Evaluationsmetriken und Benchmark-Datensätze

Um den Vergleich und Fortschritt von KI-Verfahren gewährleisten zu können, sind Benchmark-Datensätze und Evaluationsmetriken zentrale Komponenten. Ein Benchmark-Datensatz ist eine öffentlich zugängliche Sammlung von Daten und dient als Referenzgrundlage für die Evaluierung verschiedener Ansätze in einem bestimmten Anwendungsfall. Solche Datensätze werden unter anderem bei der Organisation von Wettbewerben im Computer-Vision-Forschungsbereich erstellt. Neben den Daten selbst beinhalten die Datensätze Annotationen, das heißt, Informationen zu den in den Daten dargestellten Objekten. Durch den Vergleich der Annotationen und der Ergebnisse eines Modells ermöglichen Evaluationsmetriken, die Stärken und Schwächen von Verfahren nachvollziehbar und vergleichbar zu machen. Die in dieser Studie verwendeten Datensätze sowie die verschiedenen Evaluationsmetriken werden im Folgenden eingeführt.

3.1.1. 2D-Daten

Im 2D-Bereich bestehen Benchmark-Datensätze aus einer Sammlung von Bildern verschiedenster Objekte und den dazugehörigen Informationen in Form von Annotationen. Die Bandbreite der Objekte reicht von Menschen und Tieren bis hin zu Alltagsgegenständen wie Gabeln und Tassen, aber auch spezifischere Datensätze wie z. B. für medizinische Zwecke oder den Bereich des autonomen Fahrens (KITTI-Datensatz [27]) sind öffentlich verfügbar. Der im Rahmen dieser Studie eingesetzte Datensatz und die Bewertungsmetriken im 2D-Bereich werden in den folgenden Abschnitten beschrieben.

MS-COCO

In dieser Studie wird für die 2D-Verfahren der bekannte Datensatz MS-COCO (Abkürzung für Common Objects in Context) [28] als Eingabedaten verwendet, da dieser sowohl für die Objekterkennung als auch -segmentierung eingesetzt werden kann und reale, komplexe Szenen

abbildet. Der COCO-Datensatz besteht aus ungefähr 200 000 Bildern, auf denen 80 verschiedene Objektklassen in unterschiedlichen realen Szenarien dargestellt sind. Vor allem das Vorhandensein von vergleichsweise kleinen und dicht beieinanderliegenden Objekten verleihen dem Datensatz seine Relevanz.

Metriken

Im Bereich der 2D-Objekterkennenung und -Instanzsegmentierung wird die Leistung der Modelle in der Regel über die Metrik **Mean Average Precision** (kurz: mAP) gemessen. Die Metrik mAP bildet, sehr einfach gesprochen, ab, wie gut ein Modell alle unterschiedlichen Klassenkategorien im Mittel vorhersagt. Dabei steht der Begriff »Mean«‘ in der Metrik mAP für die Mittelung über verschiedene Objektkategorien, während »Average Precision« eine Metrik ist, die für jede einzelne Objektklasse berechnet wird. Die Average Precision hängt von den Metriken **Precision**, Recall und Intersection over Union (kurz: IoU) ab. Die Precision gibt den Anteil der richtigen Vorhersagen (true positives) an allen Vorhersagen des Modells an. Der **Recall** misst den Anteil der richtigen Vorhersagen an den tatsächlich vorhandenen Objekten, der Ground Truth. Welche Vorhersagen dabei als richtig identifiziert werden, hängt bei der Objekterkennung und -segmentierung von der Metrik IoU ab, denn neben der Klassifikation ist auch die exakte Lokalisierung der erkannten Objekte in Form von Bounding Boxes oder Masken von großer Bedeutung. **IoU** gibt das Verhältnis der Schnittmenge (engl. *intersection*) der vorhergesagten und tatsächlichen Objektposition und der Vereinigung (engl. *union*) der vorhergesagten und tatsächlichen Objektposition an. Für die Berechnung der Precision und des Recall werden anschließend diejenigen Vorhersagen als korrekt angesehen, deren IoU über einem angegebenen Schwellenwert liegt. Bei einem Schwellenwert von 0,5 werden beispielsweise alle Lokalisierungen und damit die gesamte Vorhersage als korrekt angesehen, wenn der IoU-Wert über dem Schwellenwert von 0,5 liegt (siehe Abbildung 5).

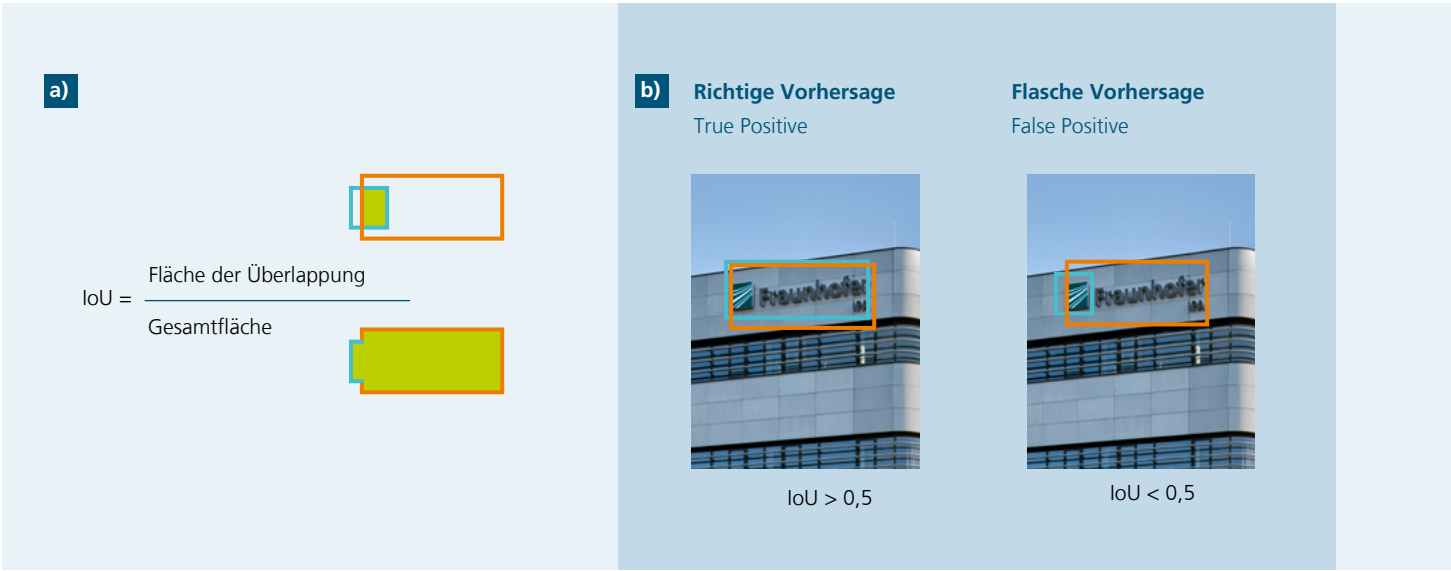


Abbildung 5: a) Veranschaulichung der Berechnung der Metrik IoU b) Identifizierung einer richtigen Vorhersage am Beispiel der Objekterkennung mit Schwellenwert 0,5. Quelle: Fraunhofer IPA/ Foto: Oliver Jaist.

Nach der Bestimmung des Recalls und der Precision kann die Average Precision berechnet werden. Die **Average Precision** stellt die Fläche unter der Precision-Recall-Kurve dar und ist eine Art Mittelung der Precision, bei welcher der Recall die Gewichte der Mittelung bestimmt. Die Average Precision wird für jede Klasse mit einem festen Schwellenwert für die IoU berechnet und für die mAP wird das Mittel aller AP aller Klassen gebildet. In der für 2D-Daten gängigen Performancemetrik mAP[0.5:0.95] wird zusätzlich die Mittelwertbildung über verschiedene Schwellenwerte im Bereich von 0,5 bis 0,95 mit einer Schrittweite von 0,05 vorgenommen.

Diese Metrik berücksichtigt sowohl die Klassifikationsgenauigkeit als auch die Anforderungen an die Genauigkeit der Lokalisierung von Objekten [29].

Um zusätzlich eine detaillierte Analyse der Modellleistung in Bezug auf die Größe der erkannten Objekte zu ermöglichen, wurden bei der COCO-Challenge die Metriken **mAP_small, mAP_medium und mAP_large** (bzw. kurz mAP_s, mAP_m und mAP_l) eingeführt. Bei der Berechnung der jeweiligen Metrik werden nur »kleine«, »mittelgroße« bzw. »große« Objekte beachtet. Die Einordnung eines Objekts in die drei verschiedenen Größenkategorien basiert auf der Fläche. Kleine Objekte nehmen eine Fläche von 32^2 Pixeln ein. Objekte mit einer Fläche über 96^2 Pixeln werden als groß eingestuft. Alle Objekte dazwischen erhalten die Bezeichnung »mittelgroß« [30].

Eine weitere Metrik, die vor allem für den Einsatz der Modelle in der Praxis relevant ist, ist die Inferenzlaufzeit, auch Inferenzzeit genannt. Die Inferenzzeit gibt an, wie lange ein Modell zur Generierung eines Ergebnisses benötigt. Dieser Faktor spielt eine entscheidende Rolle bei der Wahl eines Modells, wenn dieses in Echtzeit Ergebnisse bereitstellen soll. Niedrige Inferenzzeiten tragen zu einem schnellen Feedback bei. Dabei ist immer zu beachten, dass die Inferenzzeit von der verwendeten Hardware und dem Framework abhängt.

3.1.2. 3D-Daten

Analog zu 2D-Daten existieren Benchmark-Datensätze für unterschiedliche 3D-Anwendungen, von annotierten Punktwolken ganzer Straßenszenen, die für die semantische oder panoptische Segmentierung beim autonomen Fahren benötigt werden, über Daten von Innenraumszenen, die beispielsweise für fahrerlose Transportsysteme (FTS) erforderlich sind, bis hin zu Datensätzen von CAD-Objekten, die für die Objekterkennung oder Objektteilsegmentierung verwendet werden. Auf der Internetseite paperswithcode.com werden derzeit 122 verschiedene Datensätze mit dem Stichwort Point Cloud gelistet (Stand 11/2024). Davon sollen 20 für die semantische Segmentierung und 15 für die Objekterkennung mit entsprechenden 3D-Architekturen geeignet sein, den beiden im Rahmen dieser Studie betrachteten Aufgaben. Um einen fairen Vergleich unterschiedlicher Architekturen zu ermöglichen, wurden je Aufgabe nur Modelle verwendet, die mit demselben Datensatz trainiert wurden. Für das Training und die Inferenz der Modelle für die Objekterkennung wurde der ScanNetV2-Datensatz [31] verwendet, die Modelle für die semantische Segmentierung wurden anhand des S3DIS-Datensatzes [32] trainiert und validiert.

ScanNetV2

Bei dem ScanNetV2-Datensatz [31] handelt es sich ebenfalls um eine Sammlung annotierter Innenraumszenen. Im Unterschied zum S3DIS-Datensatz wurde ScanNetV2 nicht von statischen Scanpositionen aufgenommen. Stattdessen wurde eine an einem Tablet montierte RGB-D-Kamera händisch durch die 1513 Szenen geführt, bis die jeweils benötigte Raumabdeckung erreicht wurde. Bei dem Sensor handelte es sich um einen Occipital Structure Sensor mit einem IR-Projektor und einer IR-Kamera, die das Tiefenbild (Depth – D) erzeugen, sowie eine RGB-Kamera. Das Messprinzip ist identisch mit den Kinect- (Microsoft), Realsense- (Intel), und Oak-D-Lite-Sensoren (Luxonis). Aus den Aufnahmen von 1513 Szenen wurden 3D-Punktwolken berechnet und der Datensatz in 20 Klassen annotiert.

S3DIS

Der »Stanford 3D Indoor Scenes« Datensatz [32], kurz S3DIS genannt, besteht aus insgesamt 695.878.620 Punkten mit Farbinformation. Er wurde mit einer Matterport Camera in drei Gebäuden mit mehr als 70.000 Scans aufgenommen und bildet insgesamt 270 verschiedene Räume ab, darunter Büros, Gänge, Lagerräume etc. mit einer Gesamtfläche von 6.020m² (Abbildung 6). Die Daten wurden händisch annotiert und zwölf verschiedenen Klassen zugeordnet (Decke, Boden, Wand, Tisch, Stuhl usw.). Die in der Studie gezeigten Benchmark-Tests wurden, wie in entsprechender Literatur üblich, anhand der Area5-Daten durchgeführt, einem von 6

Teilen des Datensatzes. Der S3DIS-Datensatz gehört neben dem ScanNet-Datensatz aktuell zu den wichtigsten, weil größten und am besten annotierten Datensätzen real existierender Umgebungen, die frei verfügbar sind.

Abbildung 6: Der S3DIS-Datensatz [32] wurde über mehrere Stockwerke von über 70000 Messposen aus aufgenommen. Links unten sind einige Messpunkte hinterlegt, rechts sind die Rohdaten sowie die finale, annotierte Punktwolke zu sehen. Quelle: Armeni et al., 2016 [32].

Metriken

Die zum Vergleich der verschiedenen Modelle verwendeten Metriken für die 3D-Objekterkennung sind an die der 2D-Metriken angelehnt. Die **Intersection over Union** (IoU) wird analog zu den 2D-Daten bestimmt, nur dass hier drei statt nur zwei Dimensionen betrachtet werden. Allerdings wird bei der 3D-Objekterkennung abweichend von den 2D-Metriken kein Bereich von IoU-Schwellenwerten zwischen beispielsweise 0,50 bis 0,95 betrachtet, sondern üblicherweise werden lediglich zwei konkrete Werte als Schwellenwert genutzt: 0,25 und 0,50. Entsprechend werden die Average Precision AP_0,25 sowie AP_0,50 bestimmt. Für die **Mean Average Precision** (mAP_0,25 und mAP_0,50) wird jeweils das arithmetische Mittel der Werte aller Objektklassen gebildet. Analog dazu wird der **Mean Average Recall** mAR_0,25 und mAR_0,50 berechnet. Da entgegen der Metrik für 2D-Daten der Recall bei der mAP-Berechnung für 3D-Daten nicht berücksichtigt wird, ist es sinnvoll, ihn zusätzlich anzugeben.

Im Falle der semantischen Segmentierung wird für die Berechnung der Metriken in der Regel der IoU-Schwellenwert 0,5 verwendet. Für jede Objektklasse wird der IoU-Wert und hier die Accuracy bestimmt. Die Accuracy gibt für jede Klasse separat an, wie viele Punkte der Punktwolke nur in Hinblick auf die Klasse (gehört dazu/gehört nicht dazu) korrekt vorhergesagt wurden. Das arithmetische Mittel über alle Objektklassen ergibt die **Mean Intersection over Union** (mIoU)

sowie die Mean Accuracy (mAcc). Mit All Accuracy (allAcc) wird bezeichnet, wie viele Punkte der gesamten Punktwolke korrekt klassifiziert wurden.

Für den Vergleich unterschiedlicher Architekturen für Objekterkennung und semantische Segmentierung wurden zusätzlich Inferenzzeiten in Anhängigkeit von der Größe der Eingabepunktwolken bestimmt und ausgewertet.

3.1.3. Experiment-Setup

Die im Rahmen dieser Studie durchgeführten Experimente stellen die Performance von Transformer- und Nicht-Transformer-Architekturen dar, um eine erste Einschätzung dieser zwei Architekturtypen zu ermöglichen. Vor allem für die Anwendung der Modelle in Echtzeitanwendungen ist der Vergleich der Inferenzzeit der Modelle in einer festgelegten Hardwareumgebung von Vorteil. Darum wurden alle nachfolgenden Experimente auf derselben Hardware, einer GPU Nvidia A100 40GB auf einem NVIDIA DGX a100, durchgeführt. Die genutzten 2D-Modelle sind alle dem OpenMMLab Framework (https://openmmlab.com/) entnommen, einer Open-Source-Plattform für Computer Vision-Algorithmen. Für die Auswertung von 3D-Daten gibt es derzeit kein entsprechend breit aufgestelltes Framework. Daher wurde für jedes der getesteten Modelle mittels der parallel zur jeweiligen Veröffentlichung zur Verfügung gestellten Gitlab-Archive ein Docker Container gebaut. Lediglich PointNet++ und GroupFree3D sind in der MMDetection3D Bibliothek und dem entsprechenden Docker Container des OpenMMLab Frameworks enthalten. Die vortrainierten Gewichte der auf den verwendeten Datensätzen trainierten Modelle wurden den genannten Frameworks bzw. Gitlab-Archiven entnommen. Die Anzahl der jeweiligen Trainingsepochenanzahl kann von Modell zu Modell variieren und wird in den jeweiligen Modelltabellen (Tabelle 2-3) für die Nachvollziehbarkeit vermerkt.

Die Laufzeit der 2D-Modelle wird (wie in der Dokumentation von MMdetection beschrieben, https://mmdetection.readthedocs.io/en/latest/model_zoo.html) für 2000 Bilder und fünf Durchläufe gemessen und anschließend gemittelt, wodurch pro Modell die durchschnittliche Zeit für das Verarbeiten eines Bildes vorliegt. Bei den Messungen werden pro Bild sowohl das Network Forwarding als auch die Nachbearbeitung (engl. post-processing), jedoch nicht das Laden der Daten einbezogen. Aus der durchschnittlichen Laufzeit pro Bild ergibt sich die Verarbeitungsrate, auch als Framerate oder fps bekannt, die angibt, wie viele Bilder pro Sekunde das Modell verarbeiten kann. Diese wird in den folgenden Experimenten angegeben und verglichen.

Bei der Auswertung der Laufzeiten pro Punktwolke bzw. der Verarbeitungsrate für die 3D-Objektdetektion und die semantische Segmentierung wird nach einem ähnlichen Schema vorgegangen. Der wesentliche Unterschied ist, dass neben dem Network Forwarding und dem Post-Processing auch das Laden der Daten berücksichtigt wird.

Backbone	Datensatz des Vortrainings
ResNet50	ImageNet1K
Swin-S	ImageNet1K
Swin-L	ImageNet21K (+ Finetuning auf ImageNet1K)

Tabelle 1: Übersicht der verwendeten Backbone-Modelle mit den Details zum Vortraining und Modellarchitektur. Quelle: Fraunhofer IPA.

3.2. 2D-Objekterkennung

Für die Experimente zur 2D-Objekterkennung werden Transformer-Modelle der DETR-Familie Deformable DETR, DINO und CO-DETR betrachtet. Bei den Modellen handelt es sich um Transformer Heads, die mit verschiedenen Backbones kombiniert werden können. Als Standard-Backbone wurde das CNN-Modell ResNet50 [12] gewählt. Bei einem Modell, und zwar dem CO-DETR-Modell, wurde für den Vergleich zusätzlich der Transformer Backbone SWIN-L [11] herangezogen und mit dem Zusatz »swin_l« im Namen des Modells kenntlich gemacht. Neben den Transformer-Modellen wurden als 2D-CNN-Modelle zum einen das ältere aber bekannte Modell Faster R-CNN [33] mit einem ResNet50-Backbone und zum anderen neuere Modelle wie RTMDet [34] und ein Repräsentant der YOLO-Familie, YOLOv8 von Ultralytics (https://github.com/ultralytics/ultralytics), verwendet. Bei den zwei letzteren Modellen gibt es abhängig von der Größe des Modells Abstufungen. Diese reichen von »tiny« bzw. » nano« für das kleinste Modell über »small«, »middle« und »large« bis hin zu »extra large« (kurz »x«) für das größte Modell. Die Größe des Modells beeinflusst die Laufzeit des Modells aber auch die Genauigkeit, die in diesen Experimenten über das mAP gemessen wird. Zur Verdeutlichung der Tendenz und Bandbreite der Modelle wurden in den nachfolgenden Experimenten jeweils das kleinste und größte Modell von RTMDet und YOLOv8 ausgewählt. Eine Übersicht über alle Modelle sowie wichtige Eigenschaften und Details sind in Tabelle 2 angegeben.

Modell	Transformeranteil	Trainingsepochen	mAP	mAP_50	mAP_75	mAP_s	mAP_m	mAP_l	fps (Bild/s)	Zeit pro Bild (ms/Bild)	Anmerkungen
Faster_RCNN		24	0,384	0,59	0,42	0,215	0,421	0,503	13,9	71,9	
RTMDet_det_tiny		300	0,411	0,579	0,447	0,21	0,455	0,583	5,4	186,8	
RTMDet_det_x		300	0,528	0,704	0,577	0,361	0,574	0,692	4,2	239,4	
DDETR	Head	50	0,47	0,661	0,509	0,301	0,498	0,619	6,8	147,6	
YOLOv8_n		500	0,373	0,526	0,405	0,185	0,413	0,531	35,8	27,9	
YOLOv8_x		500	0,54	0,71	0,588	0,36	0,594	0,705	21,3	47,2	
DINO_4scale	Head	12	0,49	0,664	0,533	0,314	0,522	0,64	6,4	156,6	
CO_DETR	Head	12	0,52	0,69	0,572	0,348	0,556	0,671	3,9	256,7	LSJ-Augmentierung
CO_DETR_swin_l	Head & Backbone	12	0,593	0,773	0,649	0,434	0,633	0,755	2,3	428,1	LSJ-Augmentierung

Tabelle 2: Übersicht der verwendeten 2D-Modelle für die Objekterkennung mit den Details zur Modellarchitektur und Performance, wobei mAP für mAP[0.5:0.95] steht. Quelle: Fraunhofer IPA.

In Abbildung 7 ist die Performance aller ausgewählten 2D-Architekturen für die Objekterkennung dargestellt. Die beste Performance anhand der Metrik mAP[0.5:0.95] liefert die Transformer-Architektur Co-DETR mit einem Transformer Backbone. An zweiter Stelle steht mit einem Unterschied von 0,05 mAP-Punkten die Nicht-Transformer-Architektur YOLOv8-x. Der große Vorteil der YOLO-Familie wird in Abbildung 8 deutlich. In der Abbildung können die Laufzeiten bzw. die Verarbeitungsrate der Architekturen gemessen an der Anzahl der Bilder, die pro Sekunde durch das Modell verarbeitet werden, auf der x-Achse verglichen werden. Je höher die Verarbeitungsrate ist, desto schneller ist das Modell und desto besser ist es für Echtzeitanwendungen geeignet. Das CNN-Modell YOLOv8-n und YOLOv8-x weisen den höchsten Wert für die Verarbeitungsrate und Co-DETR einen der geringsten auf. Betrachtet man in Abbildung 8 beide Vergleichsmetriken, das heißt sowohl die Verarbeitungsrate als auch mAP, stellen Modelle, die eine Position in der rechten oberen Ecke aufweisen, den besten Kompromiss zwischen den

zwei Metriken dar. Die Architektur Yolov8-x ist in diesen Experimenten solch ein Modell, das die beste Performance liefert, sofern beide Metriken relevant für den betrachteten Anwendungsfall sind.

Die Experimente verdeutlichen zudem, wie die Auswahl des Backbones im Allgemeinen Einfluss auf die Performance, aber auch die Laufzeit hat. Die Verarbeitungsrate des Co-DETR Modells mit dem Swin-L-Backbone hat sich im Gegensatz zum Modell mit dem ResNet50-Backbone um ungefähr 41 Prozent von 3,9 fps auf 2,3 fps verringert (siehe Tabelle 2). Die Performance hingegen wird durch den Swin-L-Backbone um 0,07 mAP erhöht. Dabei muss jedoch angemerkt werden, dass, wie in Tabelle 1 dargestellt, SWIN-L im Gegensatz zu ResNet50 auf einem größeren Datensatz, dem ImageNet21K [35], vortrainiert wurde und dadurch mehr Daten zum Lernen zur Verfügung hatte. Die Experimente aus der Veröffentlichung von Ridnik et al. 2021 weisen nach, dass die Performance von ResNet50 durch das Training auf dem größeren Datensatz verbessert werden kann. Zudem existieren mittlerweile auch Weiterentwicklungen von CNN Backbones wie beispielsweise ResNeXt-101 (Xie et al. 2017). [36] verdeutlicht am Beispiel des Modells Deformable DETR, wie die weiterentwickelten Varianten von ResNet die Performance des Modells verbessern. Diese Studie trifft somit die obigen Aussagen nur für die am häufigsten anzutreffenden, vortrainierten Gewichte mit ResNet und SWIN-L. Durch das häufige Antreffen dieser vortrainierten Modelle haben die Aussagen Relevanz für Machbarkeitsstudien und erste Auswertungen in industriellen Anwendungen.

Die Evaluationsmetriken mAP_s, mAP_m und mAP_l für unterschiedlich große Objekte zeigen in Tabelle 2 beim Vergleich der Modelle untereinander keine Abweichungen zur allgemeinen mAP[0.5:0.95]-Metrik. Sortiert man die Modelle nach ihrer Performance in mAP_s, mAP _m und mAP_l, erhält man die gleiche Reihenfolge wie für die Evaluationsmetrik mAP[0.5:0.95] in Abbildung 7. Diese Beobachtung verdeutlicht, dass die Modelle alle eine gleiche Sensibilität bezüglich unterschiedlicher untersuchter Objektgrößen aufweisen.

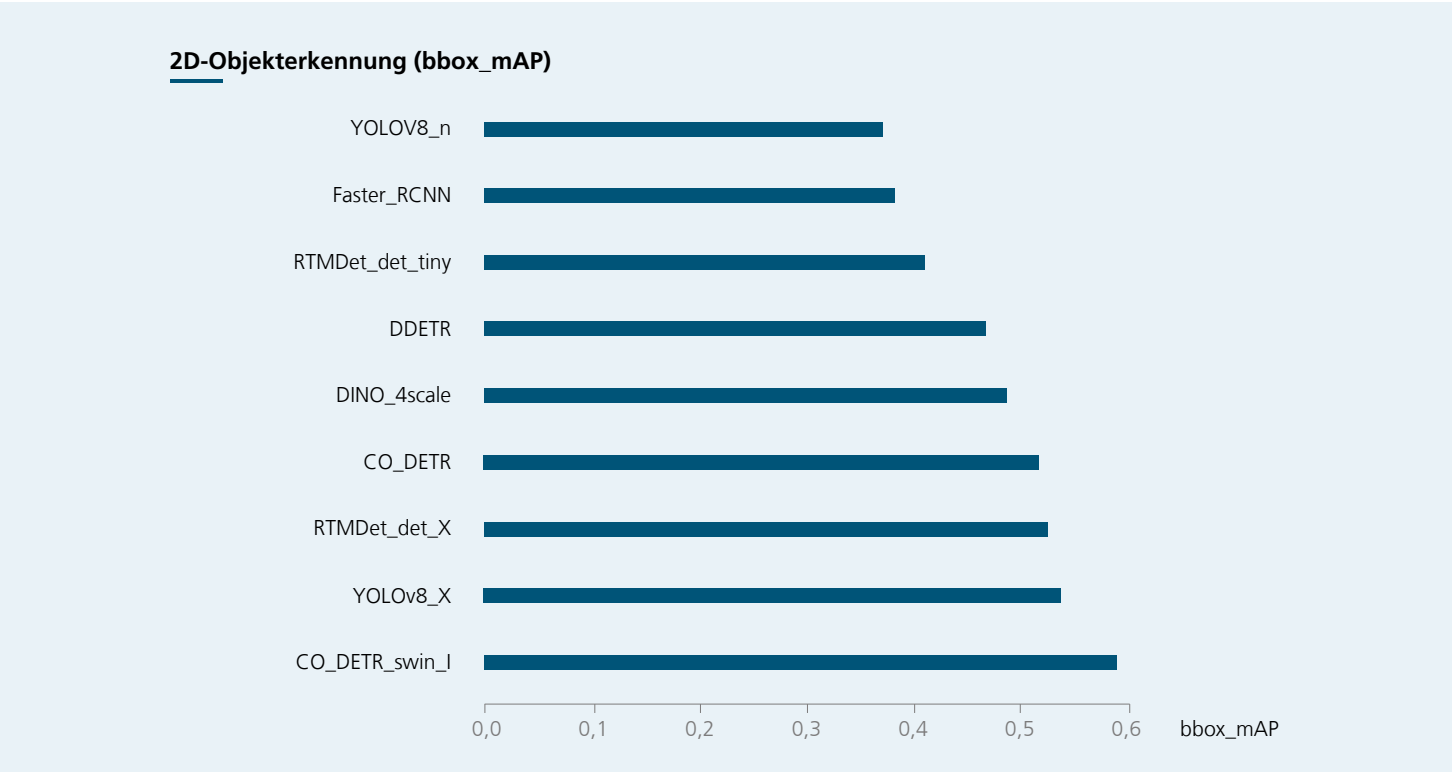


Abbildung 7: mAP gemittelt über IoU-Schwellenwerte von 0,5 bis 0,95 (mAP[0.5:0.95]) für die betrachteten 2D-Modelle. Quelle: Fraunhofer IPA.

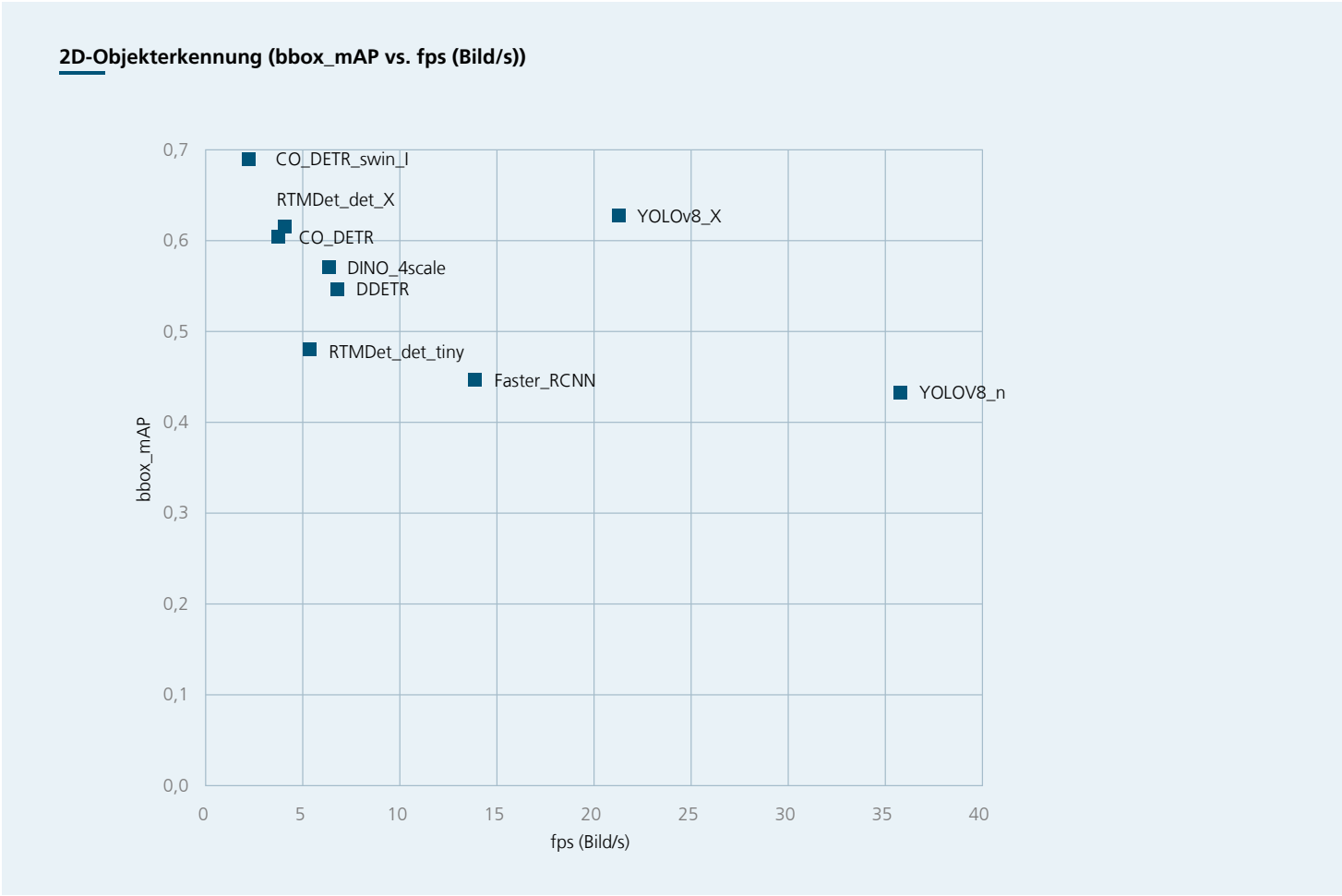


Abbildung 8: Vergleich der Verarbeitungsrate (Bilder/Sekunde) auf der x-Achse und des mAP gemittelt über die Intersection over Union-Schwellenwerte von 0,5 bis 0,95 (mAP[0.5:0.95]) auf der y-Achse für die betrachteten Modelle. Quelle: Fraunhofer IPA.

Modell	Transfor-meranteil	Trainings-epochen	mAP	mAP_50	mAP_75	mAP_s	mAP_m	mAP_l	fps(Bild/s)	Zeit pro Bild (ms/Bild)
Mask_R_CNN		36	0,371	0,584	0,399	0,185	0,398	0,529	8	124,6
YOLACT		55	0,27	0,455	0,275	0,093	0,292	0,431	16,2	64,7
RTMDet_ins_tiny		300	0,354	0,551	0,376	0,13	0,383	0,566	16,7	60,2
RTMDet_ins_x		300	0,446	0,674	0,478	0,222	0,49	0,655	8,8	113,5
Mask2Former	Head	50	0,429	0,653	0,46	0,22	0,463	0,647	4,5	223,4
Mask2Former_ swin_s	Head & Backbone	50	0,461	0,694	0,497	0,253	0,497	0,684	4	253,4

Tabelle 3: Übersicht der verwendeten 2D-Modelle für die Instanzsegmentierung mit den Details zur Modellarchitektur und Leistung, wobei mAP für mAP[0.5:0.95] steht. Quelle: Fraunhofer IPA.

3.3. 2D-Segmentierung

In den Experimenten zur 2D-Instanzsegmentierung werden das Transformer-Modell Mask2Former sowie die CNN-Modelle Mask R-CNN [37], YOLACT [33] und RTMdet-ins [34] betrachtet. Wie im vorangegangenen Experiment wurde RTMdet in seiner kleinsten (markiert mit »_tiny«) und größten Ausführung (markiert mit »_x«) integriert. Der Backbone von YOLACT, Mask R-CNN und Mask2Former ist das Modell ResNet50. Zum Vergleich zwischen einem CNN und einem Transformer Backbone wurde das Experiment bei dem Modell Mask2Former zusätzlich mit dem Transformer Backbone SWIN-S [11] durchgeführt.

Analog zu den Beobachtungen in den Experimenten zur Objekterkennung zeichnet sich bei der Instanzsegmentierung, wie in Abbildung 9 dargestellt, ein vergleichbarer Trend ab: Die höchsten Werte für die Metrik mAP[0.5:0.95] liefern sowohl das Transformer-Modell Mask2Former mit dem SWIN-S-Backbone als auch die Nicht-Transformer-Architektur RTMdet_ins_x. Mask2Former mit SWIN-S benötigt im Vergleich zu RTMDet_ins_x jedoch mit 253 ms pro Bild ungefähr die doppelte Zeit für das Prozessieren eines Bildes. Somit liegt die Verarbeitungsrate des Mask2Former-Modells bei 4 Bildern pro Sekunde und bei RTMDet_ins_x bei 8,8 Bildern pro Sekunde.

Die Experimente in Abbildung 10 verdeutlichen, analog zur Objekterkennung, die Auswirkung der Auswahl des Backbones auf die Performance, aber auch auf die Laufzeit. Die Architektur Mask2Former erhöht seine Leistung durch den Einsatz eines Transformer Backbones (SWIN-S) um zwei bis drei mAP-Punkte, verliert jedoch gleichzeitig an Schnelligkeit im Verarbeiten der Bilder. In diesem Fall sind sowohl der ResNet-Backbone als auch der SWIN-Backbone auf demselben Datensatz (ImageNet1K) vortrainiert worden. Es sei jedoch auch hier angemerkt, dass sich CNN Backbones weiterentwickelt haben und diese Experimente keine pauschale Aussage über die Güte von allen CNN Backbones liefern. Die Evaluationsmetriken mAP_s, mAP_m und mAP_l zeigen in Tabelle 3 beim Vergleich der Segmentierungsmodelle untereinander keine Abweichungen zur allgemeinen mAP[0.5:0.95]-Metrik.

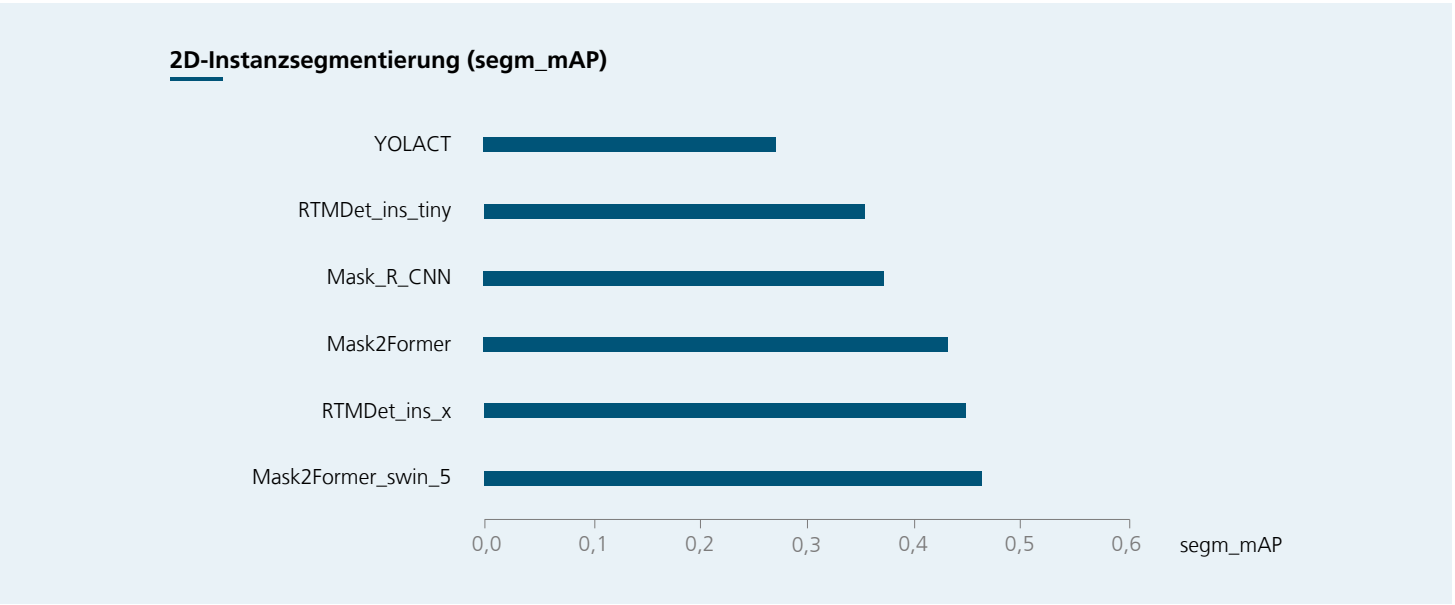


Abbildung 9: mAP über die Intersection-over-Union-Schwellenwerte von 0,5 bis 0,95 (mAP[0.5:0.95]) der Segmentierung für die betrachteten Modelle. Quelle: Fraunhofer IPA.

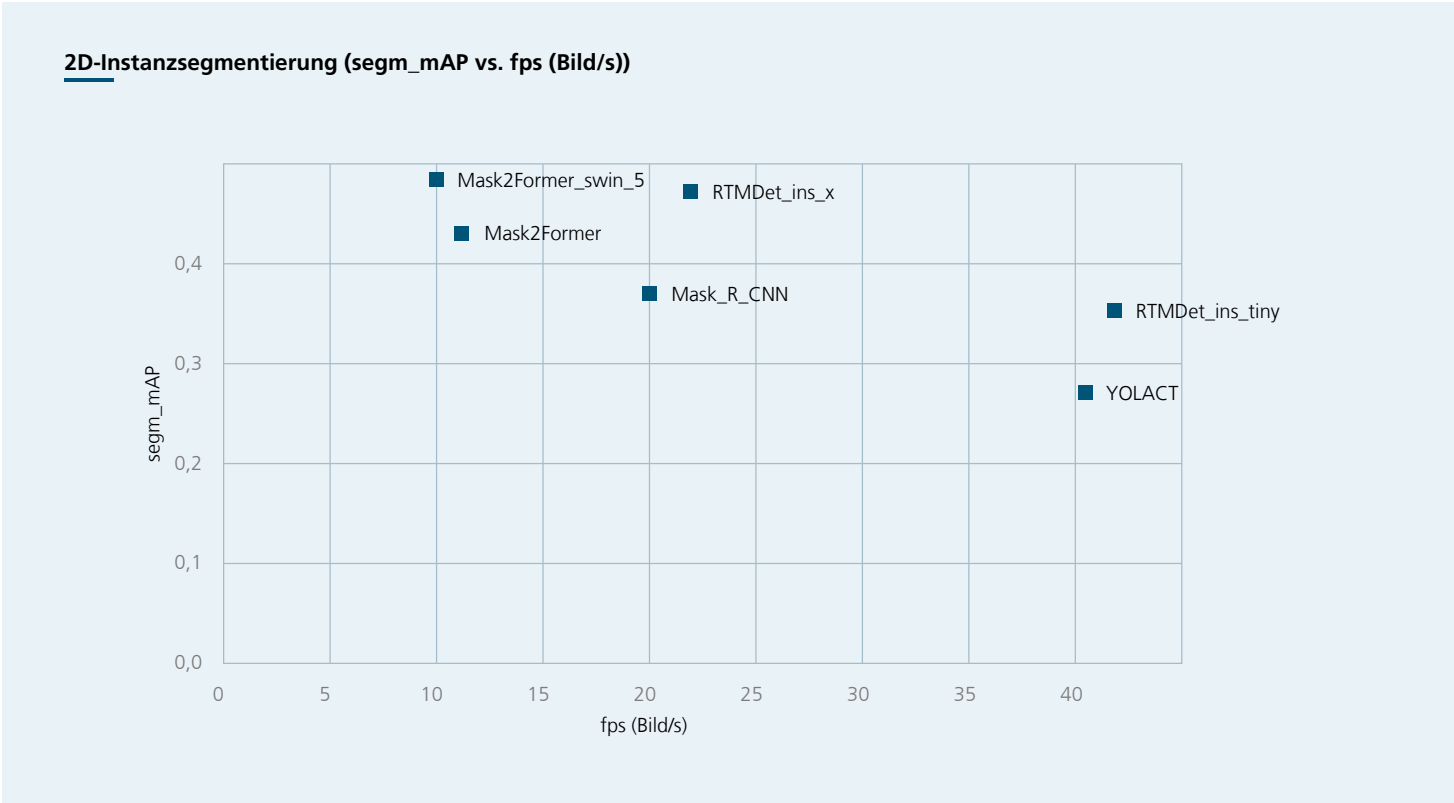


Abbildung 10: Vergleich der Verarbeitungsrate (Bilder/Sekunde) auf der x-Achse und des mAP der Segmentierung gemittelt über die Intersection-over-Union-Schwellenwerte von 0,5 bis 0,95 (mAP[0.5:0.95]) auf der y-Achse für die betrachteten Modelle. Quelle: Fraunhofer IPA.

3.4. 3D-Objekterkennung

Die für die 3D-Objekterkennung verwendeten Netze sind CAGroup3D (CNN) [39], GroupFree3D [21] und OneFormer3D [22], beides Transformer mit CNN Backbone. Die Gewichte der auf dem ScanNetV2-Datensatz vortrainierten Netze wurden von den jeweiligen Git-Repositories heruntergeladen. Zusätzlich wurde der ScanNetV2-Datensatz heruntergeladen. Die angegebenen Statistiken zu den Inferenzen basieren auf Docker-Implementierungen der Netze und dem Originaldatensatz. In Tabelle 4 ist eine Übersicht der getesteten Modelle einschließlich der Anzahl der Trainingsepochen dargestellt.

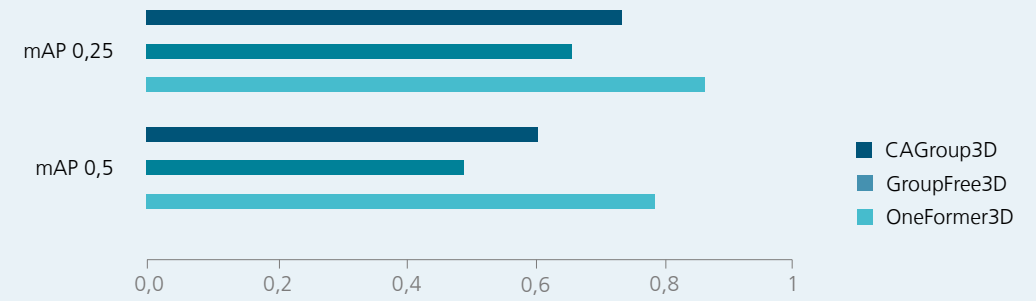


Abbildung 11: mAP 0,25 und mAP 0,5 (Mean Average Precision) für die drei betrachteten Transformer- (GroupFree3D, OneFormer3D) und Nicht-Transformer-Modelle (CAGroup3D). OneFormer3D sticht klar heraus, das klassische CNN CAGroup3D liegt in der Mitte. Quelle: Fraunhofer IPA.

Die Ergebnisse des Vergleichs der drei Modelle für die Objekterkennung sind in Abbildung 11 und Abbildung 12 sowie in Tabelle 4 dargestellt. Bei Betrachtung der Precision ist das OneFormer3D-Modell am besten. Gemittelt über alle Klassen gehören 87 Prozent (0,87 mAP 0,25) bzw. 79 Prozent (0,79 mAP 0,50) der erkannten Objekte tatsächlich zu den bestimmten Klassen. Das zweite Modell mit Transformer Head, GroupFree3D, kommt nur auf 2/3 bis 3/4 der Werte (0,66 mAP 0,25/0,49 mAP 0,50). Etwa in der Mitte landet beim Vergleich der Precision das CNN-basierte Modell CAGroup3D (0,74 mAP 0,25/0,61 mAP 0,50).

Die Auswertung des Recall (Abbildung 12) zeichnet ein etwas anderes Bild als die der Precision. Der mAR 0,25 ist für alle verglichenen Modelle sehr ähnlich: Zwischen 84 Prozent und 90 Prozent aller Objekte des ScanNetV2-Testdatensatzes wurden richtig erkannt und der richtigen Klasse zugeordnet. Den besten Wert erreichte dabei das CNN mit einem mAR 0,25 von 0,90. Beim mAR 0,50 liegen CAGroup3D und OneFormer3D etwa gleich auf (0,77 und 0,78), GroupFree3D liegt wieder abgeschlagen bei 0,64. Insgesamt lässt sich zusammenfassen, dass GroupFree3D von den betrachteten Modellen am ungenauesten ist. Unter den Modellen CAGroup3D und OneFormer3D gibt es hinsichtlich des Recalls keinen klaren Gewinner, beide schneiden je nach betrachtetem IoU knapp besser/schlechter ab.

An der Inferenzlaufzeit (Tabelle 4, Abbildung 13) lässt sich erkennen, dass das verwendete CNN gegenüber den Transformern einen geringen Vorteil hat: Die Transformer-Modelle GroupFree3D und OneFormer3D benötigten mit einer mittleren Verarbeitungsgeschwindigkeit von 4,27 bzw. 5,36 Punktwolken pro Sekunde etwas länger als CAGroup3D (5,77 fps).

GroupFree3D wurde im April 2021 veröffentlicht, CAGroup3D im Oktober 2022 und One-Former3D noch ein Jahr später, im November 2023. In Anbetracht der vergleichsweise kurzen Evolution seit der Veröffentlichung des ersten funktionalen MLP für 3D-Daten (PointNet++, 2017) ist die Entwicklung sowohl von Transformern, als auch von CNNs enorm vorangeschritten. Das vergleichsweise schlechte Abschneiden des GroupFree3D-Transformers bildet jedenfalls auch den jeweiligen Stand der Technik ab. Der gezeigte Vergleich lässt entsprechend auch keine abschließende Aussage zu, ob Transformer nun besser oder schlechter als CNNs für die 3D-Objekterkennung geeignet sind, obgleich aktuelle Transformer tendenziell genauer als moderne CNN-Architekturen arbeiten. In jedem Fall zeigt der Vergleich der beiden neueren Modelle, dass weniger die Architektur eines Modells in Bezug auf eine spezifische Anwendung von Bedeutung ist, sondern vielmehr, ob ein Modell hinsichtlich der jeweils relevanten Metriken (Precision, Recall, Inferenzzeit) gut abschneidet oder nicht. Die Inferenzzeiten aller drei betrachteten Modelle sind vergleichbar, was darauf zurückzuführen ist, dass alle im Kern den gleichen Merkmalsextraktor verwenden (PointNet++) und lediglich der Head variiert.

	mAP 0,25	mAR 0,25	mAP 0,50	mAR 0,50	Inferenz in fps
CAGroup3D	0,74	0,90	0,61	0,76	5,77
GroupFree3D	0,66	0,84	0,49	0,64	4,27
OneFormer3D	0,87	0,84	0,79	0,78	5,36

Tabelle 4: Ermittelte Qualitätsmetriken für die 3D-Objekterkennung mit zwei Transformer-Modellen (GroupFree3D, OneFormer3D) und einem icht-Transformer-Modell (CAGroup3D). Quelle: Fraunhofer IPA.

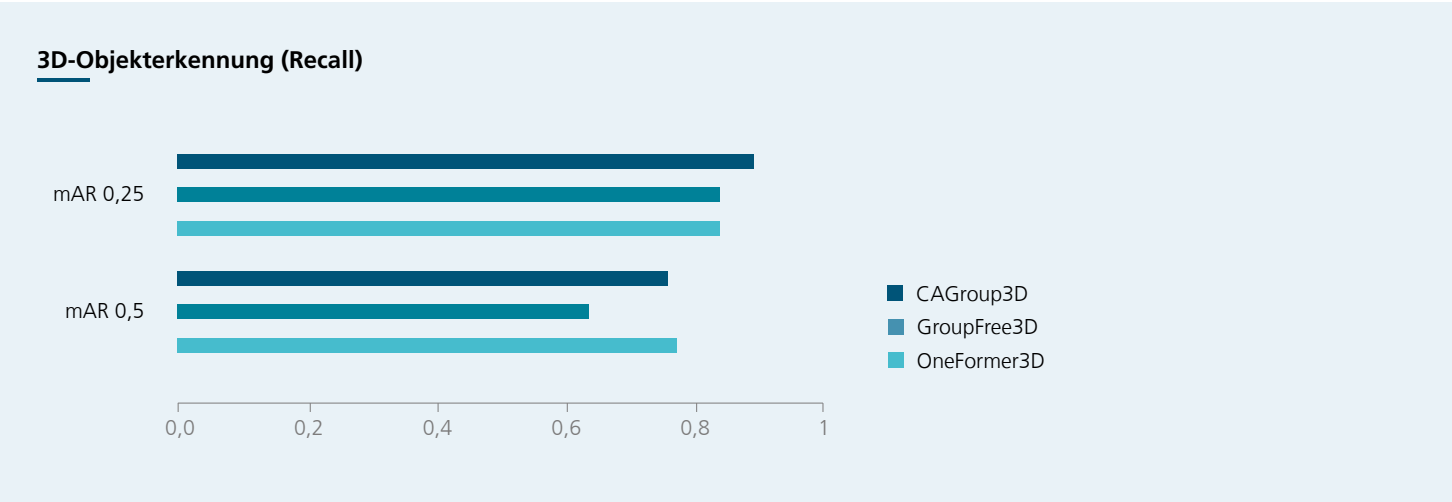


Abbildung 12: Mean Average Recall für Intersection-over-Union-Schwellenwerte von 0,25 und 0,5 für die zwei betrachteten Transformer- (GroupFree3D, OneFormer3D) und Nicht-Transformer-Modelle (CAGroup3D). Quelle: Fraunhofer IPA.

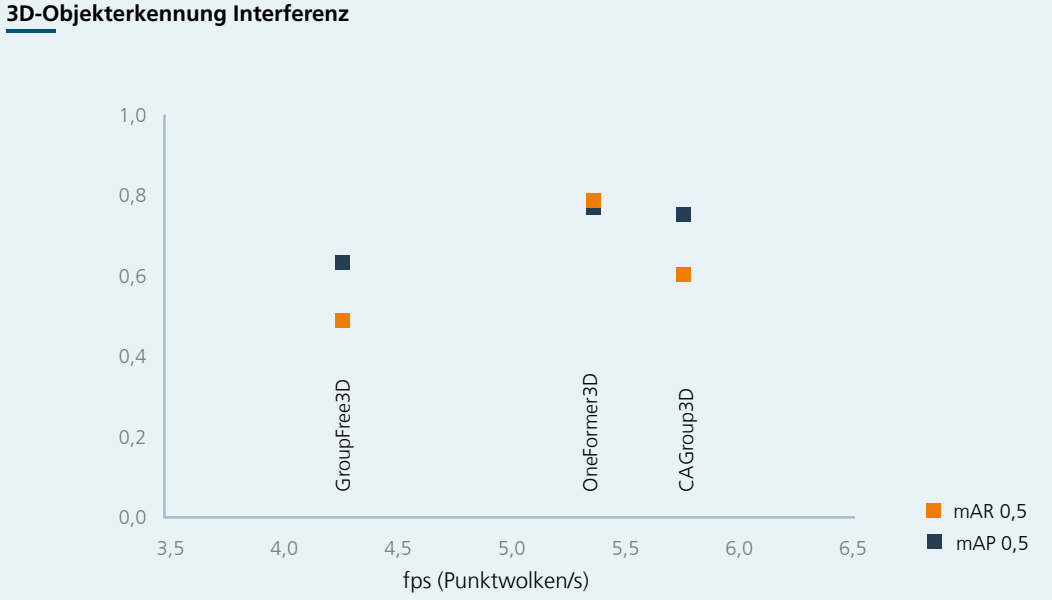


Abbildung 13: Verarbeitungsrate in fps für die 3D-Objekterkennung. Während das CNN (CAGroup3D) am schnellsten ist (5,8 fps), sind die Transformer im direkten Vergleich langsamer. Quelle: Fraunhofer IPA.

3.5. 3D-Segmentierung

Für die Vergleiche zur 3D-Segmentierung wurden PointNet++ (reines CNN) [8], Swin3D (Transformer Backbone und CNN Head) [24] mit den zwei Varianten Swin3D-S und Swin3D-L sowie SPT (CNN Backbone und Transformer Head) [26] verwendet. Die Gewichte der auf S3DIS vortrainierten Netze sowie der S3DIS-Datensatz wurden analog zur 3D-Objekterkennung von den jeweiligen Git-Repositories bzw. der Webpage der Stanford University heruntergeladen. Im Vergleich des aus 2017 stammenden CNN zu den aktuellen Transformer-Architekturen zeigt sich der erwartet große Unterschied: In den 6 Jahren seit der PointNet++-Veröffentlichung hat die Entwicklung und Einführung von Transformern auch bei der semantischen Segmentierung zu einem enormen Qualitätssprung geführt. Die mIoU basierend auf der Area 5 des S3DIS-Datensatzes ist je nach Architektur zwischen 0,12 und 0,17 gestiegen (Abbildung 14, Tabelle 5). Auch hinsichtlich der Gesamt-Accuracy (mAcc) und der gemittelten Accuracy über alle Objektklassen (allAcc) fand ein Anstieg von mindestens 0,13 (mAcc) bzw. 0,05 (allAcc) statt. Bei den Inferenzzeiten bestätigen sich Literaturangaben, laut derer Swin3D mit Verarbeitungsraten zwischen 0,1 und 0,2 fps im Vergleich zu anderen Architekturen zu den langsamsten Transformern und auch Nicht-Transformern gehört (Abbildung 15). SPT ist bei der Segmentierung zwar weniger genau, hat gegenüber Swin3D aber leichte Vorteile bei den Inferenzzeiten.

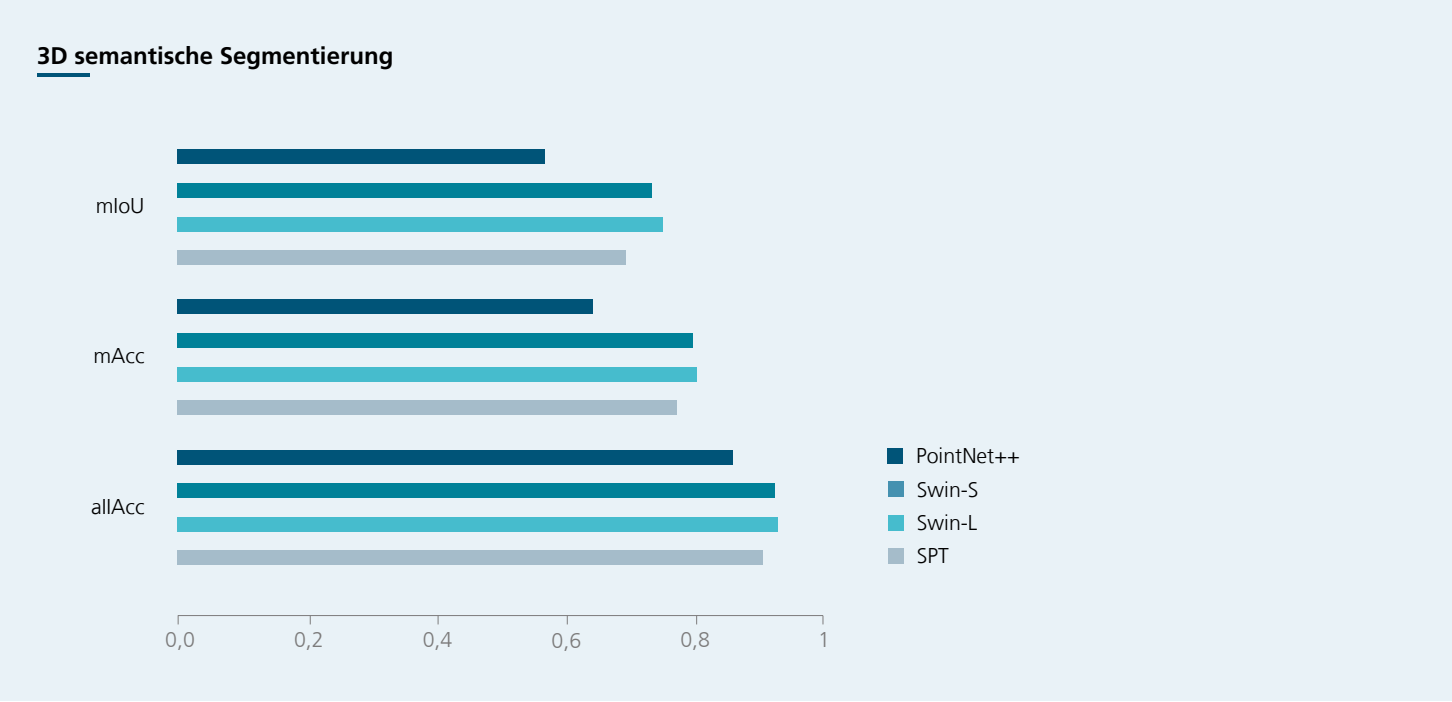


Abbildung 14: Die drei bestimmten Performanz-Metriken für die semantische Segmentierung von 3D-Punktwolken. Quelle: Fraunhofer IPA.

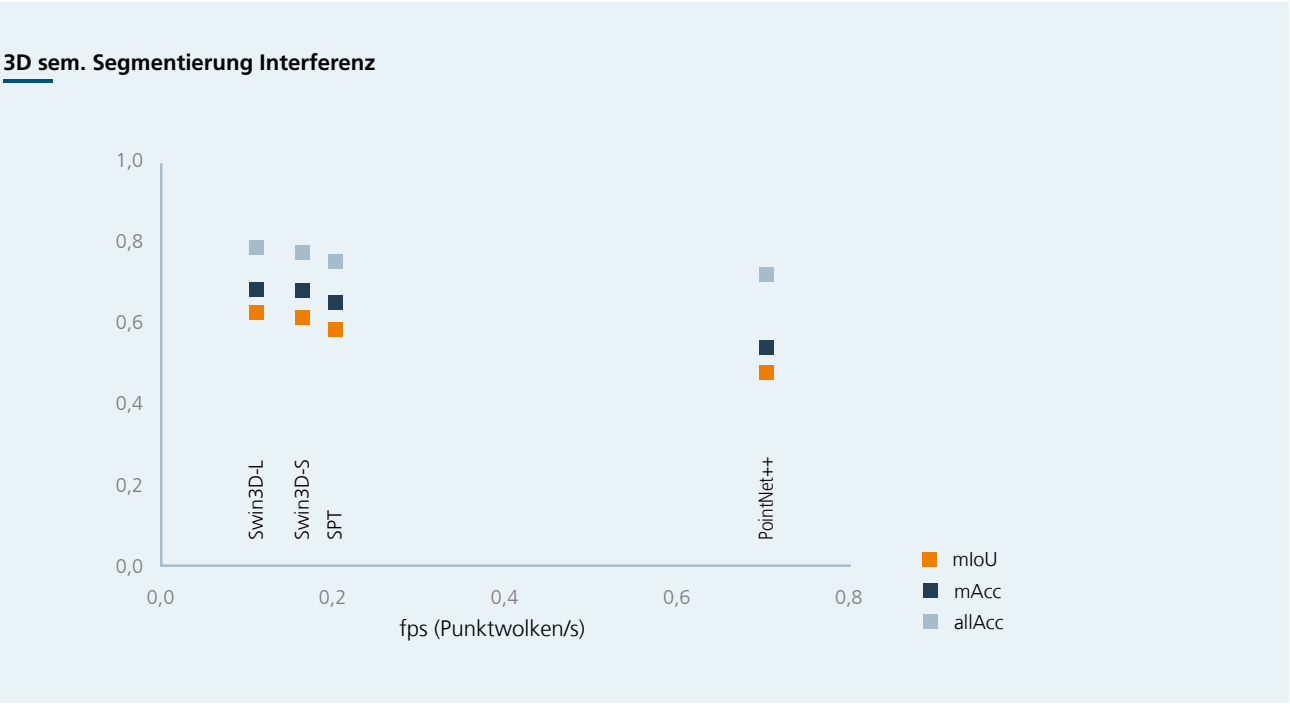


Abbildung 15: Qualitative Metriken der getesteten Modelle für die semantische Segmentierung von 3D-Daten gegenüber der Verarbeitungsrate in fps. Auch hier ist das Nicht-Transformer-Modell, in diesem Fall PointNet++, das schnellste. Quelle: Fraunhofer IPA.

	mIoU	mAcc	allAcc	Inferenz s-1
PointNet++	0,57	0,64	0,86	0,70
Swin-S	0,73	0,80	0,92	0,17
Swin-L	0,74	0,80	0,93	0,11
SPT	0,69	0,77	0,90	0,20

Tabelle 5: Der Vergleich von Transformer- und Nicht-Transformer-Modellen (PointNet++) für semantische Segmentierung von 3D-Punktwolken offenbart zwar Nachteile ersterer hinsichtlich der Verarbeitungsrate, zeigt aber auch ein besseres Abschneiden in Bezug auf die getesteten Qualitätsmetriken. Quelle: Fraunhofer IPA.

Beispielsweise für die autonome Navigation eines FTS in einem Gebäude können die genannten, bereits vortrainierten Netze grundsätzlich genutzt werden. Allerdings ist zu beachten, dass sich die von der Sensorik eines FTS oder eines sonstigen Geräts aufgenommenen Daten immer mindestens geringfügig hinsichtlich Dichte und Vollständigkeit von den ursprünglichen Trainings- und Testdaten unterscheiden.

Im Folgenden wird demonstriert, wie ein auf dem S3DIS-Datensatz vortrainiertes Superpoint-Transformer-Netz [26] auf Daten einer anderen Datenquelle performt. In der Regel ist bei der Segmentierung mit Daten anderer Quellen mit einem schlechteren Ergebnis zu rechnen als beim Test-Set des Trainingsdatensatzes, da sowohl das Rauschen der Daten als auch die Punktdichte deutlich unterschiedlich sein können. Als Anwendungsbeispiel wurde das Vision Lab des Fraunhofer IPA mit einem FARO-Scanner von vier verschiedenen Positionen aus gescannt, eine kumulierte Punktwolke erzeugt und diese auf ein Voxelgrid mit 4 mm Kantenlänge heruntergerechnet. Ein Ausschnitt daraus sowie die darin erkannten Objekte sind in Abbildung 16 dargestellt. Stühle, Tische, Wand und Boden wurden sehr gut segmentiert und zugeordnet. Für Desktop-PCs und Monitore (graue Objekte auf den Tischen) gibt es im S3DIS-Trainingsdatensatz keine eigene Klasse, sodass diese Objekte korrekt als »Sonstiges« klassifiziert wurden. Auch in der Punktwolke vorhandene Objekte, die im Trainingsdatensatz nicht vorkommen, werden dieser Klasse zugeordnet. Dies gilt beispielsweise für das hellgraue, hohe Objekt (rechts in Abbildung 16), das einem in einen Aufsteller integrierten Monitor entspricht. Insgesamt wurde lediglich die Fensterfront zwar gut segmentiert, aber größtenteils falschen Klassen zugeordnet. Hier würde gegebenenfalls ein weiterer Scan für ein besseres Ergebnis sorgen.

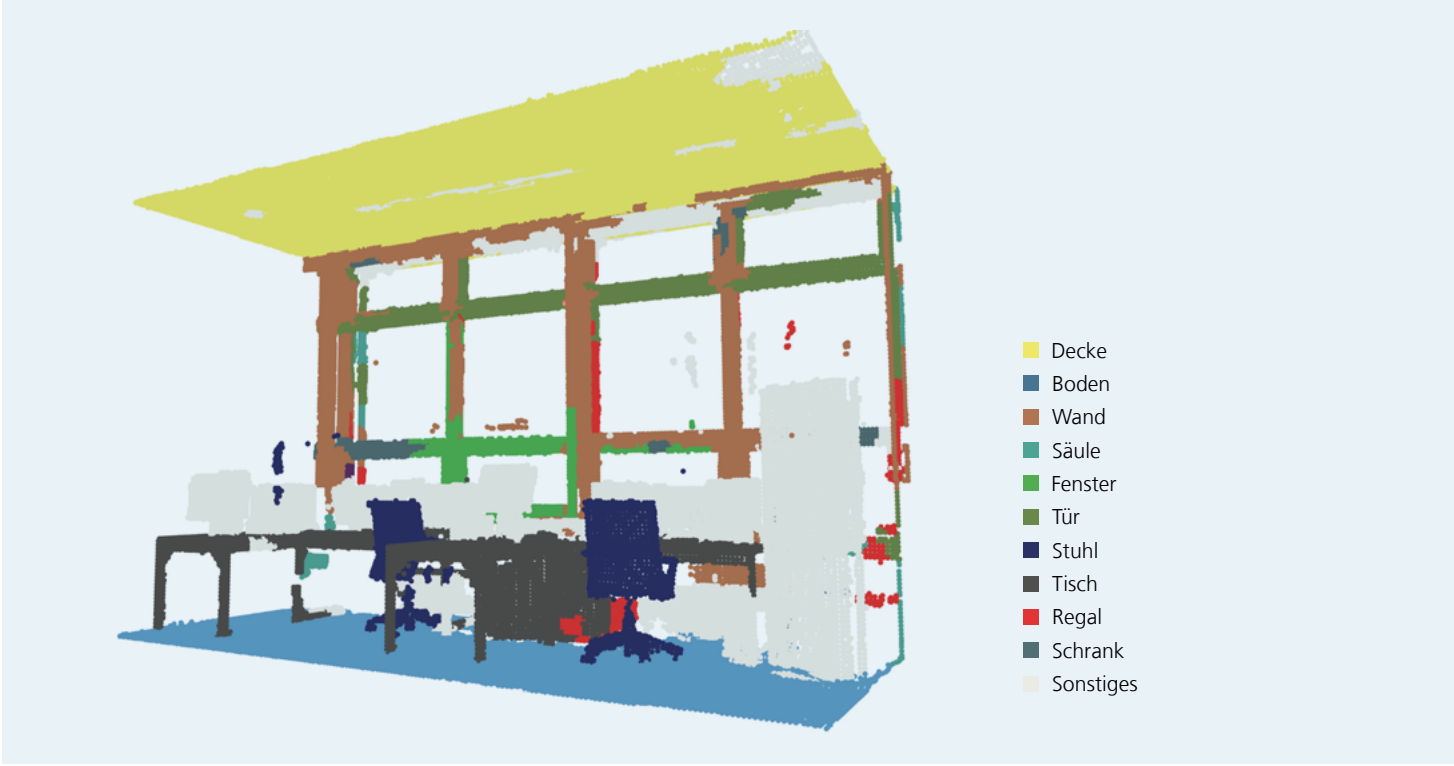


Abbildung 16: Semantische Segmentierung von 3D-Daten aufgenommen im Vision Lab des Fraunhofer IPA. Das verwendete Netz ist ein auf dem S3DIS-Datensatz trainiertes Super Point Model [26]. Trotz verschiedener verwendeter 3D-Sensoren werden wichtige Features wie Stühle, Tische, Decke und Boden gut erkannt. Quelle: Fraunhofer IPA.

4. In der Praxis

Dieses Kapitel diskutiert verschiedene Aspekte von Transformer-Architekturen, die für die Anwendung in der Praxis relevant und zu beachten sind. Insbesondere wird auf die Potenziale und Herausforderungen von Transformer-Modellen, ihre weiteren Anwendungsbereiche und Best Practices für die Anwendung eingegangen.

4.1. Potenziale und Anwendungsmöglichkeiten in der Industrie

Potenziale und Herausforderungen von Transformern
Transformer-Modelle werden durch allgemeine Pretext Tasks vortrainiert und können dann an spezielle Aufgaben angepasst bzw. nachtraininert werden. Diese Eigenschaft erklärt sich durch die wenigen Annahmen der Transformer bei der Struktur des zu lösenden Problems. Zum Beispiel nehmen Transformer im Gegensatz zu CNNs keine lokalen Zusammenhänge in einem Bild an, wie in Kapitel 2 beschrieben. Dadurch könnte sich der Aufwand fürs Annotieren verringern [9]. Gleichzeitig brauchen die Transformer dieses Vortraining, um die gleiche Performance wie CNNs zu liefern, da sie Annahmen wie Translational Equivariance nicht in sich tragen und diese zunächst lernen müssen. Transformer sind somit datenhungrig, da sie erst lernen müssen, wie Eingabedaten, also ein Bild oder eine Punktwolke, strukturiert sind. CNNs erhalten dieses Wissen indirekt über ihre a priori festgelegte Funktionsweise mittels Faltungen [9].

Eine weitere Herausforderung bei Transformern sind ihre hohen Anforderungen an die Rechenleistung (z.B. [9]). Durch die Größe und Komplexität der Transformer-Modelle ergeben sich hohe Trainings- und Inferenzzeiten und -kosten, wie sich sowohl in den 2D- als auch 3D-Tests zeigt. Der Fokus bei der Entwicklung von Transformer-Architekturen liegt aus diesem Grund aktuell auch darauf, die Komplexität zu reduzieren, um effizientere Modelle wie beispielsweise Point Transformer V3 (PTv3) realisieren zu können. Wenn es auf mittlere Sicht gelingt, den Laufzeitnachteil der Transformer-Modelle zu überwinden, steht der breiteren Anwendung in der Praxis nichts im Wege.

In sicherheitskritischen Anwendungsfeldern von KI gewinnt ein anderes Thema immer mehr an Bedeutung: Das Verstehen und Interpretieren von Modellen. Dieses Thema ist im Zusammenhang mit Transformern aktuell Gegenstand der Forschung. Durch die komplexen Zusammenhänge der Attention Layer ist die Interpretierbarkeit von Transformer-Modellen eine erst auf mittlere Sicht zu lösende Herausforderung. Die Erklärbarkeit von CNNs ist bereits etwas weiter fortgeschritten, bedarf aber ebenfalls noch umfangreicher Forschung.

Anwendungsbereiche von Transformern
Transformer finden in der industriellen Bildverarbeitung grundsätzlich Anwendung an den gleichen Stellen wie konventionelle CNNs. Dies betrifft viele Anwendungen, die bereits im Kapitel 1 aufgeführt wurden. Es wurden in dieser Studie vor allem die Objekterkennung und -segmentierung für den 2D- und 3D-Bereich beleuchtet. Transformer finden jedoch in allen Bereichen der Bildverarbeitung Verwendung [9]. Beispielsweise werden Transformer für die Posenschätzung (engl. human pose estimation) eingesetzt. In diesem Gebiet soll anhand von Eingabedaten, in der Regel 2D-Bilder oder 3D-Punktwolken, die Pose eines Menschen geschätzt werden. Das bedeutet, dass die Modelle alle Gelenkpunkte des menschlichen Körpers vorhersagen sollen. Diese Technologie kann beispielswiese für das maschinelle Verstehen von Zeichensprache eingesetzt werden. Eine Transformer-Architektur für diesen Aufgabenbereich ist beispielsweise ViTPose [15]. Ein weiterer Bereich, in

dem Transformer momentan im Fokus der Forschung stehen, ist die Bildgenerierung. Die Aufgabe bei der Bildgenerierung ist es, aus der eingegebenen Beschreibung ein möglichst realistisches Bild (Foto, Zeichnung etc.) zu erschaffen. Hierfür sind beispielsweise die DALL-E-Modelle von OpenAI (Ramesh et al. 2021) prominente Vertreter. Die Bildgenerierung kann bei der Erweiterung von Datensätzen unterstützen, was vor allem bei fehlenden Beispielen wie Bauteilen mit Defekten auf der Oberfläche vorteilhaft sein kann (e.g. [40]).

4.2. Best Practices

Die folgenden Hinweise zur Entwicklung und Implementierung von Transformern in der industriellen Bildverarbeitung haben zwar für KI-Anwendungen allgemein Gültigkeit, sollen aber trotzdem nicht unerwähnt bleiben.

Softwarebibliotheken
Für die KI-gestützte Verarbeitung von 2D- als auch 3D-Daten empfiehlt es sich, Bibliotheken zu verwenden, die mehrere Architekturen für die jeweils zu lösende Aufgabe bereitstellen. Dies reduziert den Implementierungsaufwand, da Bibliotheken wie openCV (2D), open3D (3D) oder das openMMLab-Framework (2D + 3D) tendenziell besser gepflegt werden als primäre Quellen, das heißt die Git-Repositories der jeweiligen Methodenautoren. Letztere werden in der Regel mit deutlich geringerem Aufwand auf verschiedener Hardware mit unterschiedlichen Software-Umgebungen getestet und zumeist einmalig zur Verfügung gestellt, ohne dass eventuell auftretende Fehler verfolgt oder behoben würden. Der zweite Vorteil breit aufgestellter Bibliotheken besteht darin, dass – einmal auf einem Rechner implementiert – verschiedene Methoden mit sehr wenig Aufwand verglichen werden können.

Inferenzzeit vs. Durchsatz
Bei den berichteten Zahlen zur Inferenzlaufzeit ist zu beachten, dass diese mit einer Batchsize > 1 bestimmt wurden. Das heißt, dass immer mehrere Bilder oder Punktwolken gleichzeitig vom jeweiligen Netz verarbeitet wurden. Neben dem eigentlichen Ziel, dass die Modelle so besser und damit schneller lernen, führt dieses gängige Vorgehen dazu, dass die gemittelte Inferenzzeit eines Modells mitunter deutlich sinkt. Da in der finalen Anwendung Bilder oder Punktwolken meist nur einzeln verarbeitet werden, kann die tatsächliche Durchsatzrate entsprechend geringer ausfallen.

Rechtsrahmen
Für alle Datensätze und Softwarebibliotheken gibt es sogenannte End User License Agreements, die den Rahmen der jeweils erlaubten Nutzung genau festlegen. Manche Datensätze und Bibliotheken dürfen kommerziell gar nicht oder nicht bedingungslos, beispielsweise ohne entsprechende Nutzungsgebühren abzuführen, genutzt werden (z. B. ImageNet, S3DIS). Dies ist für jedes Modell und jeden Datensatz einzeln zu prüfen und liegt in der Verantwortung des Anwenders. Weiterhin hat die Europäische Union mittlerweile den European AI Act beschlossen, nach dem für veröffentlichte KI-Anwendungen eine Risikobewertung durchzuführen ist, die gegebenenfalls ein striktes Monitoring der Entwicklung, der Inbetriebnahme und des eigentlichen Betriebs der KI-Anwendung nach sich zieht. Auch wenn der EU AI Act noch in nationales Recht umgesetzt werden muss, empfiehlt es sich, schon zu Beginn einer Entwicklung zu prüfen, was an begleitender Dokumentation erforderlich ist.

Nachhaltigkeit
KI hat generell das Potenzial, Prozesse in der Produktion effizienter zu gestalten und so den Energieverbrauch zu reduzieren. Allerdings steigt mit der Komplexität eines Modells die erforderliche Rechenleistung eines Computers und somit auch dessen Stromverbrauch. Im Hinblick auf die CO2Bilanz und somit die Nachhaltigkeit eines Unternehmens sollte daher möglichst moderne, für KI-Anwendungen optimierte und entsprechend effiziente Hardware eingesetzt werden.

5. Fazit

Bei der Suche nach Lösungen für die industrielle Bildverarbeitung spielen neuronale Netze allgemein und Transformer-Architekturen im Speziellen eine nicht mehr wegzudenkende Rolle. Ein Schwerpunkt des Forschungsbereichs Künstliche Intelligenz und Maschinelles Sehen des Fraunhofer IPA besteht darin, den Transfer der aussichtsreichsten, von der Grundlagenforschung entwickelten Modelle, einschließlich neuester Transformer-Architekturen, in die Industrie voranzutreiben. Diese Studie zeigt zusammenfassend auf, dass der Fokus sowohl auf CNNs als auch Transformern liegen sollte, da in vielen Fällen Ideen aus beiden Bereichen kombiniert werden und (zumindest bisher) in unterschiedlichen Anwendungsfällen unterschiedliche Empfehlungen ausgesprochen werden können. Das geht beispielsweise aus dem Vergleich der Laufzeit und der Genauigkeit in dieser Studie hervor. Es ist somit zu empfehlen, die weitere Entwicklung beider Architekturtypen zu verfolgen.

Glossar

Anchor Box: Eine Anchor Box ist eine vorab definierte rechteckige Umrahmung, die in der Objekterkennung verwendet wird, um verschiedene Größen und Seitenverhältnisse von Objekten in Bildern abzudecken.

Annotationen: Sie sind zusammen mit den betreffenden Bildern oder Punktwolken das Wissen, das für das Training und die Evaluation einer Computer-Vision-Aufgabe benötigt wird. Sie werden oft auch als Ground-Truth (zu Deutsch »Grundwahrheit«) bezeichnet. Annotationen geben beispielsweise in Form von Bounding Boxes oder (Freiform-)Bereichen an, wo sich ein spezifisches Objekt in den jeweiligen 2D- oder 3D-Daten befindet und um welche Objektklasse es sich handelt.

Bildklassifikation: Bei der Bildklassifikation wird einem Bild eine bestimmte Klasse oder Kategorie zugeordnet.

Bounding Box: Rahmen, der die Ausdehnung eines Objektes in 2D-Bildern definiert. Im Fall von 3D-Punktwolken sind Bounding Boxes Quader.

False Positive: Eine Vorhersage des Modells, die mit der Realität bzw. der Ground Truth nicht übereinstimmt.

Feature Extraction: Feature Extraction ist der Prozess, bei dem relevante Merkmale aus Rohdaten wie Bildern extrahiert werden, um sie für das Maschinelle Lernen zu nutzen.

Feature Map: Eine Feature Map ist das Ergebnis eines Layer in einem CNN und enthält Informationen darüber, welche Merkmale in verschiedenen Bereichen des Bilds erkannt wurden.

Instanzsegmentierung/Instance Segmentation: Hierbei wird jedes Objekt in einem Bild/einer Punktwolke nicht nur einer bestimmten Klasse zugeordnet, sondern auch jedes einzelne Objekt dieser Klasse separat segmentiert. Einer Klasse können mehrere Objekte zugeordnet sein.

Network Forwarding: Network Forwarding ist der Prozess, bei dem die Eingabedaten wie beispielweise Bilder durch ein neuronales Netzwerk geleitet werden, um eine Vorhersage zu erzeugen. Im Gegensatz dazu steht die Backpropagation, die eingesetzt wird, um ein Netzwerk bzw. seine Gewichte zu optimieren. Während des Trainings wird bei der Backpropagation der Fehler zwischen der Ausgabe eines Network Forwardings und der Ground Truth rückwärts durch das Netzwerk propagiert und die Gewichte angepasst, sodass der Fehler miniert wird.

Non Maximum Suppression: Ist ein Verfahren, bei dem in der Objekterkennung überlappende oder nahegelegene Objekte anhand von Wahrscheinlichkeiten reduziert werden. Falls beispielsweise ein Objekt mehrfach mit geringem Versatz oder unterschiedlicher Ausdehnung erkannt wurde, wird es so auf eine Bounding Box reduziert.

Object Detection: Identifizieren von Objekten innerhalb eines Bilds/einer Punktwolke und Bestimmung der Position. Mehrere Objekte gleicher Klassen können erkannt werden, Positionsangabe mittels Bounding Boxes.

Object Queries: Das Konzept findet vor allem in transformatorbasierten Architekturen wie DETR (Detection Transformer) Verwendung und dient dazu, spezifische Objekte im Bild zu identifizieren und zu lokalisieren. Dabei wird eine feste Anzahl von Abfragen initialisiert, welche mit den Merkmalen des Bilds durch eine Transformatorstruktur interagiert. Das Ergebnis sind Vorhersagen über die Existenz und Position von Objekten im Bild.

Panoptic Segmentation: Vereint semantische und Instanzsegmentierung: Hierbei wird jeder Pixel/Punkt eines Bilds/einer Punktwolke einer Objektklasse zugeordnet, jedes einzelne Objekt dieser Klasse wird separat segmentiert. Zu einer Klasse können mehrere Objektinstanzen gehören.

Patch: Bild- oder Datenausschnitt, im Falle von Swin Transformer werden Bilder in 4x4 Pixel große rechteckige Patches unterteilt.

Pretraining: Siehe Vortraining

Region Proposal: Region Proposal ist der Vorschlag eines Verfahrens, das potenzielle Bildbereiche identifiziert, in denen sich ein relevantes Objekt befindet.

Semantische Segmentierung / Semantic Segmentation: Hierbei wird jeder Pixel/Punkt eines Bilds/einer Punktwolke einer Objektklasse zugeordnet. Innerhalb einer Klasse wird nicht zwischen einzelnen Objekten unterschieden.

Translational Equivariance: Translational Equivariance ist eine Eigenschaft von CNNs, bei der eine Verschiebung des Eingabebilds zu einer Verschiebung der Feature Map führt, ohne dass die relevanten Merkmale sich ändern.

True Positive: Eine Vorhersage des Modells, die mit der Realität bzw. der Ground Truth übereinstimmt.

Vortraining: Training auf einer großen und allgemeinen Datenmenge, bevor die KI für eine spezifische Aufgabe feinabgestimmt (fine-tuned) wird. Dieser Ansatz wird häufig verwendet, um die Effizienz und Genauigkeit von CNNs zu verbessern, insbesondere wenn die verfügbare Datenmenge für die spezifische Aufgabe begrenzt ist.

Voxel-Raster: Eine Punktwolke kann in ein beliebig definiertes, regelmäßiges 3D-Raster übertragen werden. Dabei wird für jeden Würfel oder Voxel (Volumenpixel) des Rasters geprüft, ob ein Punkt darin liegt oder nicht. Der Würfel bekommt entsprechend den Wert eins oder null. Voxel-Raser werden genutzt, um ungeordnete Punktwolken als geordnete, komprimierte 3D-Arrays darzustellen.

References

[1] A. Vaswani et al., "Attention Is All You Need," 2017.

[2] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 2020.

[3] K. Han et al., "A Survey on Vision Transformer," IEEE transactions on pattern analysis and machine intelligence, vol. 45, no. 1, pp. 87–110, 2023, doi: 10.1109/TPAMI.2022.3152247.

[4] R. Szeliski, Computer vision: Algorithms and applications. Cham: Springer, 2022. [Online]. Available: https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=6839940

[5] X. Li et al., "Transformer-Based Visual Segmentation: A Survey," Apr. 2023. [Online]. Available: http://arxiv.org/pdf/2304.09854.pdf

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in Advances in Neural Information Processing Systems, 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[7] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in Computer vision - ECCV 2014: 13th European conference, Zurich, Switzerland, September 6 - 12, 2014; proceedings, 2014, pp. 818–833. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53

[8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," Jun. 2017. [Online]. Available: http://arxiv.org/pdf/1706.02413.pdf

[9] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in Vision: A Survey," ACM Comput. Surv., vol. 54, 10s, pp. 1–41, 2022, doi: 10.1145/3505244.

[10] S. Jamil, M. Jalil Piran, and O.-J. Kwon, "A Comprehensive Survey of Transformers for Computer Vision," Drones, vol. 7, no. 5, p. 287, 2023, doi: 10.3390/drones7050287.

[11] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," 2021.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," in Computer vision - ECCV 2016: 14th European conference, Amsterdam, The Netherlands, October 11-14, 2016 : proceedings, 2016, pp. 630–645. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-46493-0_38

[13] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014. [Online]. Available: http://arxiv.org/pdf/1409.1556

[14] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in Springer eBook Collection, vol. 12346, Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds., 1st ed., Cham: Springer International Publishing; Imprint Springer, 2020, pp. 213–229.

[15] Y. Xu et al., "Transformers in computational visual media: A survey," Comp. Visual Media, vol. 8, no. 1, pp. 33–62, 2022, doi: 10.1007/s41095-021-0247-3.

[16] H. Zhang et al., "DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection," arXiv e-prints, arXiv:2203.03605, 2022, doi: 10.48550/arXiv.2203.03605.

[17] Z. Zong, G. Song, and Y. Liu, "DETRs with Collaborative Hybrid Assignments Training," in 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 2023.

[18] B. Cheng, A. G. Schwing, and A. Kirillov, Per-Pixel Classification is Not All You Need for Semantic Segmentation, 2021.

[19] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar, "Masked-attention Mask Transformer for Universal Image Segmentation," in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

[20] Papers With Code, https://paperswithcode.com/sota/semantic-segmentation-on-s3dis (accessed: Jul. 2 2024).

[21] Z. Liu, Z. Zhang, Y. Cao, H. Hu, and X. Tong, "Group-Free 3D Object Detection via Transformers," in 2021 IEEE/CVF International Conference on Computer Vision: ICCV 2021 : 11-17 October 2021, virtual event : proceedings, Montreal, QC, Canada, 2021, pp. 2929–2938.

[22] M. Kolodiazhnyi, A. Vorontsova, A. Konushin, and D. Rukhovich, "OneFormer3D: One Transformer for Unified Point Cloud Segmentation," Nov. 2023. [Online]. Available: http://arxiv.org/pdf/2311.14405.pdf

[23] C. Choy, J. Gwak, and S. Savarese, "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[24] Y.-Q. Yang et al., "Swin3D: A Pretrained Transformer Backbone for 3D Indoor Scene Understanding," Apr. 2023. [Online]. Available: http://arxiv.org/pdf/2304.06906.pdf

[25] J. Zheng, J. Zhang, J. Li, R. Tang, S. Gao, and Z. Zhou, "Structured3D: A Large Photo-Realistic Dataset for Structured 3D Modeling," in Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX, 2020, pp. 519–535. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-58545-7_30

[26] D. Robert, H. Raguet, and L. Landrieu, "Efficient 3D Semantic Segmentation with Superpoint Transformer," Jun. 2023. [Online]. Available: http://arxiv.org/pdf/2306.08045

[27] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[28] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in Lecture notes in computer science, vol. 8693, Computer vision - ECCV 2014: 13th European conference, Zurich, Switzerland, September 6 - 12, 2014; proceedings, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., Cham: Springer, 2014, pp. 740–755.

[29] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object Detection in 20 Years: A Survey," Proc. IEEE, vol. 111, no. 3, pp. 257–276, 2023, doi: 10.1109/jproc.2023.3238524.

[30] COCO: Offizielle Seite COCO. [Online]. Available: https://cocodataset.org/#detection-eval

[31] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Niessner, "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes," in 30th IEEE Conference on Computer Vision and Pattern Recognition: CVPR 2017 : 21-26 July 2016, Honolulu, Hawaii : proceedings, Honolulu, HI, 2017, pp. 2432–2443.

[32] I. Armeni et al., "3D Semantic Parsing of Large-Scale Indoor Spaces," in 29th IEEE Conference on Computer Vision and Pattern Recognition: CVPR 2016 : proceedings : 26 June-1 July 2016, Las Vegas, Nevada, Las Vegas, NV, USA, 2016, pp. 1534–1543.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE transactions on pattern analysis and machine intelligence, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[34] C. Lyu et al., "RTMDet: An Empirical Study of Designing Real-Time Object Detectors," Dec. 2022. [Online]. Available: http://arxiv.org/pdf/2212.07784.pdf

[35] T. Ridnik, E. Ben-Baruch, A. Noy, and L. Zelnik-Manor, "ImageNet-21K Pretraining for the Masses," Apr. 2021. [Online]. Available: http://arxiv.org/pdf/2104.10972.pdf

[36] Y. Yang et al., "Transformers Meet Visual Learning Understanding: A Comprehensive Review," Mar. 2022. [Online]. Available: http://arxiv.org/pdf/2203.12944.pdf

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," Mar. 2017. [Online]. Available: http://arxiv.org/pdf/1703.06870.pdf

[38] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-Time Instance Segmentation," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[39] H. Wang et al., "CAGroup3D: Class-Aware Grouping for 3D Object Detection on Point Clouds," Oct. 2022. [Online]. Available: http://arxiv.org/pdf/2210.04264.pdf

[40] O. de Mitri, A. Frommknecht, M. F. Huber, F. Müller-Graf, and C. Distante, "Synthetic data generation for improvement of machine learning-based optical quality control: a practical approach in the welding context," in Multimodal Sensing and Artificial Intelligence: Technologies and Applications III: 27-29 June 2023, Munich, Germany, Munich, Germany, 2023, p. 46. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12621/2682047/Synthetic-data-generation-for-improvement-of-machine-learning-based-optical/10.1117/12.2682047.full

KI-Fortschrittszentrum

Das KI-Fortschrittszentrum »Lernende Systeme und Kognitive Robotik« unterstützt Firmen dabei, die wirtschaftlichen Chancen der Künstlichen Intelligenz und insbesondere des Maschinellen Lernens für sich zu nutzen. In anwendungsnahen Forschungsprojekten und in direkter Kooperation mit Industrieunternehmen arbeiten die Stuttgarter Fraunhofer-Institute für Produktionstechnik und Automatisierung IPA sowie für Arbeitswirtschaft und Organisation IAO daran, Technologien aus der KI-Spitzenforschung in die breite Anwendung der produzierenden Industrie und der Dienstleistungswirtschaft zu bringen. Unterstützt werden sie dabei vom Institut für Arbeitswissenschaft und Technologiemanagement IAT der Universität Stuttgart. Finanzielle Förderung erhält das Zentrum vom Ministerium für Wirtschaft, Arbeit und Tourismus Baden-Württemberg.

Mission

Das KI-Fortschrittszentrum ist der anwendungsorientierte Zweig von Cyber Valley, Europas größter Forschungskooperation im Bereich der Künstlichen Intelligenz. Darüber hinaus ist das KI-Fortschrittszentrum Bestandteil von S-TEC, dem Stuttgarter Technologie- und Innovationscampus: www.s-tec.de

Das KI-Fortschrittszentrum schlägt die Brücke von der KI-Spitzenforschung in den Mittelstand und macht KI-Technologien für die Wirtschaft in Baden-Württemberg und darüber hinaus nutzbar. Als führender Innovationspartner für den Mittelstand arbeitet das Zentrum an Themen, die für den Einsatz von KI und Robotik branchenübergreifend von zentraler Bedeutung sind, beispielsweise Autonomie, Effizienz und Nachhaltigkeit, Mensch-Maschine-Interaktion sowie Vertrauen. Das KI-Fortschrittszentrum informiert Unternehmen über Technologietrends und deren Einsatzpotenziale und unterstützt sie bedarfsgerecht und niedrigschwellig bei der Entwicklung und Umsetzung von ambitionierten KI-Innovationen, damit sie die wirtschaftlichen Chancen der KI künftig noch besser nutzen können.

Vision

Das KI-Fortschrittszentrum ist ein Leuchtturm für erfolgreichen Technologietransfer in den Mittelstand und ermöglicht Unternehmen einen wirtschaftlichen und verantwortungsvollen Einsatz von Künstlicher Intelligenz und Robotik für unternehmerischen Erfolg sowie individuellen und gesellschaftlichen Nutzen.

Studienreihe »Lernende Systeme und Kognitive Robotik«

Die Studienreihe »Lernende Systeme und Kognitive Robotik« gibt Einblick in die Potenziale und die praktischen Einsatzmöglichkeiten von KI. Nähere Informationen und die aktuellen Versionen der Studien finden Sie unter: www.ki-fortschrittszentrum.de/de/themen/studien.html

Fraunhofer

Forschen für die Praxis ist die zentrale Aufgabe der Fraunhofer-Gesellschaft. Die 1949 gegründete Forschungsorganisation betreibt anwendungsorientierte Forschung zum Nutzen der Wirtschaft und zum Vorteil der Gesellschaft. Vertragspartner und Auftraggeber sind Industrie- und Dienstleistungsunternehmen sowie die öffentliche Hand.

Die Fraunhofer-Gesellschaft betreibt in Deutschland derzeit 74 Institute und Forschungseinrichtungen. Mehr als 28 000 Mitarbeiterinnen und Mitarbeiter, überwiegend mit natur- oder ingenieurwissenschaftlicher Ausbildung, erarbeiten das jährliche Forschungsvolumen von mehr als 2,8 Milliarden Euro. Davon entfallen mehr als 2,3 Milliarden Euro auf den Leistungsbereich Vertragsforschung. Rund 70 Prozent dieses Leistungsbereichs erwirtschaftet die Fraunhofer-Gesellschaft mit Aufträgen aus der Industrie und mit öffentlich finanzierten Forschungsprojekten. Rund 30 Prozent werden von Bund und Ländern als Grundfinanzierung beigesteuert, damit die Institute Problemlösungen entwickeln können, die erst in fünf oder zehn Jahren für Wirtschaft und Gesellschaft aktuell werden.

Internationale Kooperationen mit exzellenten Forschungspartnern und innovativen Unternehmen weltweit sorgen für einen direkten Zugang zu den wichtigsten gegenwärtigen und zukünftigen Wissenschafts- und Wirtschaftsräumen.

Mit ihrer klaren Ausrichtung auf die angewandte Forschung und ihrer Fokussierung auf zukunftsrelevante Schlüsseltechnologien spielt die Fraunhofer-Gesellschaft eine zentrale Rolle im Innovationsprozess Deutschlands und Europas. Die Wirkung der angewandten Forschung geht über den direkten Nutzen für die Kund*innen hinaus: Mit ihrer Forschungs- und Entwicklungsarbeit tragen die Fraunhofer-Institute zur Wettbewerbsfähigkeit der Region, Deutschlands und Europas bei. Sie fördern Innovationen, stärken die technologische Leistungsfähigkeit, verbessern die Akzeptanz moderner Technik und sorgen für die Aus- und Weiterbildung des dringend benötigten wissenschaftlich-technischen Nachwuchses.

Ihren Mitarbeiterinnen und Mitarbeitern bietet die Fraunhofer-Gesellschaft die Möglichkeit zur fachlichen und persönlichen Entwicklung für anspruchsvolle Positionen in ihren Instituten, an Hochschulen, in Wirtschaft und Gesellschaft. Studierenden eröffnen sich aufgrund der praxisnahen Ausbildung und Erfahrung an Fraunhofer-Instituten hervorragende Einstiegs- und Entwicklungschancen in Unternehmen.

Namensgeber der als gemeinnützig anerkannten Fraunhofer-Gesellschaft ist der Münchner Gelehrte Joseph von Fraunhofer (1787–1826). Er war als Forscher, Erfinder und Unternehmer gleichermaßen erfolgreich.

Impressum

Kontaktadresse
Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA
Nobelstraße 12, 70569 Stuttgart

Dascha Karelina
Telefon +49 711 970-1861
dascha.karelina@ipa.fraunhofer.de

Dr. Bernd Meese
Telefon +49 711 970-1943
bernd.meese@ipa.fraunhofer.de

Fraunhofer-Publica
http://dx.doi.org/10.24406/publica-3790

Titelbild
© NicoElNino – stock.adobe.com, Oliver Jaist

Satz und Gestaltung
Franz Schneider, Fraunhofer IAO

Gefördert durch das Ministerium für Wirtschaft, Arbeit
und Tourismus Baden-Württemberg

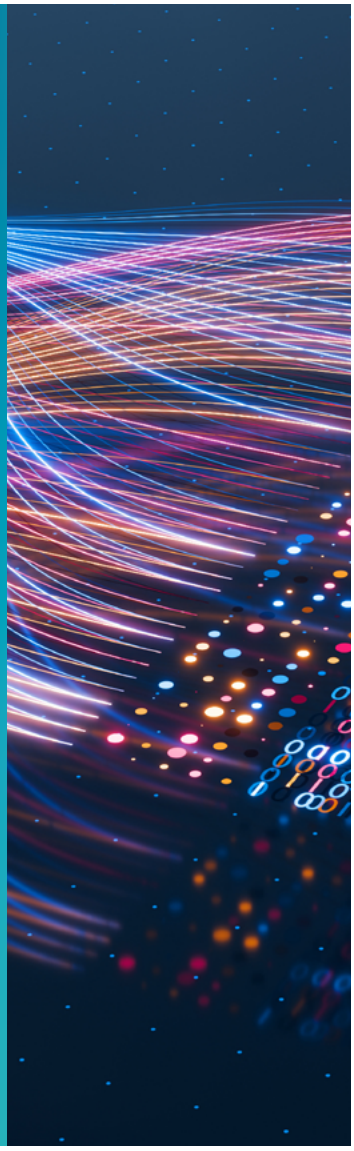
Kontakt

KI-Fortschrittszentrum
Fraunhofer IAO und Fraunhofer IPA
Nobelstraße 12
70569 Stuttgart
www.ki-fortschrittszentrum.de

Dascha Karelina
Telefon +49 711 970-1861
dascha.karelina@ipa.fraunhofer.de

Dr. Bernd Meese
Telefon +49 711 970-1943
bernd.meese@ipa.fraunhofer.de

Fraunhofer IPA
Nobelstraße 12
70569 Stuttgart
www.ipa.fraunhofer.de



Gefördert durch

Baden-Württemberg
Ministerium für Wirtschaft,
Arbeit und Tourismus

Partner

Cyber Valley

S·TEC

Universität Stuttgart
Institut für Arbeitswissenschaft und
Technologiemanagement IAT