19. AALE-Konferenz
Luxemburg, 08.-10. März. 2023

Synthetische Trainingsdatengenerierung und Objekterkennung mit Deep Learning für Mixed Reality-Anwendungen mit Digitalen Zwillingen

Thomas Fettahoglu[1], Jana Hönig[2], Marc Schnierle[3], Sascha Röck[4]

Zusammenfassung

Die Fähigkeit von Maschinen, Gegenstände über Kamerabildern und Videosequenzen wahrzunehmen, ermöglicht neue Formen der Mensch-Maschine-Interaktion. Bislang wird der breite Einsatz von Objekterkennung mit Deep Learning Algorithmen in der Automatisierungstechnik durch die aufwändige Datenakquise und den daraus resultierenden Mangel an Daten für das Training künstlicher neuronaler Netze gehemmt. Insbesondere die Kombination von Methoden der Objekterkennung mit Mixed Reality-Technologien und realdatengetriebenen Digitalen Zwillingen wird in Zukunft zahlreiche Anwendung im industriellen Kontext finden.

Dieser Beitrag stellt ein Konzept für die Generierung synthetischer Trainingsdaten für die Objekterkennung mit Deep Learning auf Basis von Digitalen Modellen über das Training künstlicher neuronaler Netze bis hin zur Mixed Reality-Anwendung vor. Durch die automatisierte Generierung der synthetischen Trainingsdaten soll der Prozess der Trainingsdatenakquise bedeutend beschleunigt und die Qualität der Trainingsergebnisse gegenüber der Verwendung manuell erstellter Trainingsdaten erhöht werden. Die Tragfähigkeit des vorgestellten Konzeptes wird durch eine Realisierung in der am Virtual Automation Lab (VAL) der Hochschule Esslingen entwickelten Digital Twin as a Service Plattform (DTaaSP) sowie anhand von zwei Anwendungsbeispielen aufgezeigt.

Stichwörter

Künstliche Intelligenz, Synthetische Trainingsdaten, Objekterkennung, Digitaler Zwilling, Mixed Reality

1 Motivation

Methoden der Künstlichen Intelligenz bieten für kleine und mittelständische Unternehmen (KMU) in der Automatisierungstechnik vielfältige Potenziale und werden in Zukunft zur nachhaltigen Wettbewerbsfähigkeit des Wirtschaftsstandorts Deutschland beitragen. Ein Einsatzgebiet dieser Methoden ist die automatische Objekterkennung zur kontextsensitiven Informationsdarstellung mittels Mixed Reality-Technologien. Beispielhafte Anwendungen sind die Visualisierung von Prozessdaten der realen Anlage über ein Augmented Reality-Endgerät an der betrachteten Komponente oder die Unterstützung von manuellen Montageprozesse durch eine automatisierte Teile- und Werkzeugerkennung.

Die Basis dieser Anwendungen bildet eine Objekterkennungssoftware basierend auf Deep Learning Algorithmen, die über einer auf dem Mixed Reality-Endgerät (bspw. Microsoft HoloLens 2 oder Apple iPad) verfügbaren Kamera reale Objekte automatisiert erkennt und dazu bedarfsgerecht Kontextinformationen einblendet (siehe Abbildung 1). Eine große Herausforderung stellt in diesem Kontext die Trainingsdatenakquise dar: Deep Learning Algorithmen zur Objekterkennung in Bildern oder Videosequenzen erfordern große Trainingsdatensätze mit hoher Varianz in denen die Daten mit Objektklassen annotiert sein müssen.

[1] Thomas Fettahoglu, M.Sc., Virtual Automation Lab, Hochschule Esslingen
[2] Jana Hönig, M.Eng., Virtual Automation Lab, Hochschule Esslingen
[3] Marc Schnierle, M.Sc., Virtual Automation Lab, Hochschule Esslingen
[4] Prof. Dr.-Ing. Sascha Röck, Virtual Automation Lab, Hochschule Esslingen

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +

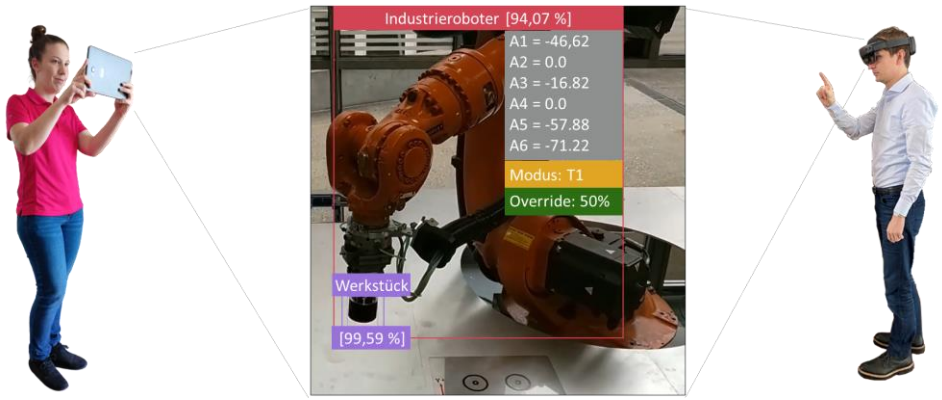

Abbildung 1: Einblendung bedarfsgerechter Kontextinformationen über Mixed Reality-Endgeräte

2 Stand der Technik und Forschung

Derzeit werden die Trainingsdatensätze für die Objekterkennung durch manuelle Annotationsschritte auf Basis von realen Bildern aufgebaut. Durch die manuelle Datenakquise und Datenvorverarbeitung ist dieser Prozess sowohl fehleranfällig als auch kosten- und zeitintensiv [1]. Die Trainingsdaten werden zudem für ausgewählte Anwendungen erstellt und sind nur unter großem Aufwand auf weitere Anwendungsfälle übertragbar. Daraus resultiert häufig eine begrenzte Datenbasis, insbesondere bei einer großen Produktvariantenvielfalt.

An diesen Herausforderungen greifen aktuelle Forschungsaktivitäten an. Ein großer Forschungsbereich ist die synthetische Trainingsdatengenerierung [2] anhand von virtuellen 3D-Objekten, die die Notwendigkeit von Bildern realer Objekte reduzieren und ggf. sogar eliminieren sollen. In einer virtuellen Umgebung können die Perspektive, Beleuchtung und Position des Objekts flexibel variiert werden, wodurch die Qualität und die Varianz der erstellten Bilder systematisch sichergestellt werden kann. Durch den Einsatz leistungsstarker Rechen-Hardware werden die benötigten Trainingsdaten deutlich schneller mit größerer Varianz im Vergleich zur manuellen Generierung erstellt. Synthetische Trainingsdaten werden bereits in verschiedenen Anwendungen eingesetzt. Diese reichen von der synthetischen Erzeugung zufälliger Fehlerbilder für die Erkennung und Klassifikation von Produktionsfehlern in der Blechverarbeitung [3] über die synthetischen Trainingsdatengenerierung und Objekterkennung für Handhabungssysteme in komplexen Fertigungshallen in der Automobilindustrie [4] bis hin zur synthetischen Erzeugung simulierter Verkehrsszenarien für das Autonome Fahren [5][6].

Bestehende Ansätze befassen sich mit anwendungsspezifischen Realisierungen der synthetischen Trainingsdatengenerierung für die Objekterkennung. Für die weitere Verbreitung dieser Technologie im industriellen Umfeld, insbesondere in kleinen und mittleren Unternehmen (KMU), muss die synthetische Trainingsdatengenerierung mit bereits vorhandenen Daten aus dem mechatronischen Entwicklungsprozess erfolgen. Damit sind insbesondere Digitale Modelle, die im digitalen Entwicklungsprozess (CAD/CAE) entstehen, gemeint auf deren Basis durchgängige Konzepte für die synthetische Trainingsdatengenerierung für die Objekterkennung der modellierten Baugruppen generiert werden können. Für einen niederschwelligen Zugang zu diesen Technologien sollte neben der Trainingsdatengenerierung auch die Anwendung in das Konzept integriert werden.

3 Lösungsansatz

Zur Bewältigung der dargestellten Hemmnisse wird in diesem Beitrag die im Forschungsprojekt accelerateKI[5] entworfene Methode zur automatisierten Generierung synthetischer Trainingsdaten für die Objekterkennung mit Deep Learning Algorithmen unter Einsatz Digitaler Modelle von Maschinen und Anlagen vorgestellt. Der Ansatz umfasst ein durchgängiges Konzept von der Anbindung Digitaler Modelle über die Generierung der Datensätze und dem Training der neuronalen Netze bis hin zur Objekterkennung in Mixed Reality (siehe Abbildung 2). Durch diesen Ansatz soll der Prozess der

[5] Forschungsprojekt „Servicebasierte KI-Konfigurationsunterstützung als Accelerator für KI-Anwendungen in KMUs – accelerateKI“ gefördert vom Ministerium für Wissenschaft, Forschung und Kunst Baden-Württemberg

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +

Trainingsdatengenerierung bedeutend beschleunigt und die Qualität der Trainingsdaten systematisch sichergestellt werden.

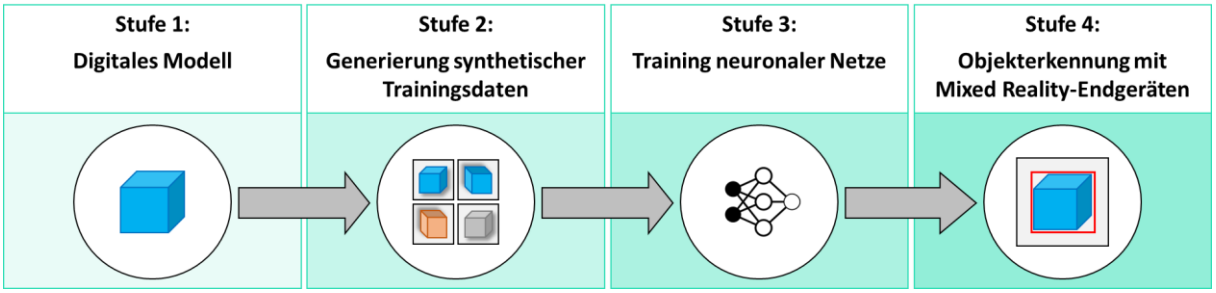

Abbildung 2: Konzept für die Generierung synthetischer Trainingsdaten

Nachfolgend werden die vier Stufen des Konzeptes detaillierter vorgestellt.

Stufe 1: Digitales Modell

Die Bereits aus dem Engineering vorhandenen Digitalen Modelle der Anlage bilden mit ihrem Geometrie- und Verhaltensmodell die Grundlage für eine automatisierte Trainingsdatengenerierung. Durch die Integration der Digitalen Modelle in den Generierungsprozess und deren Anbindung an eine Laufzeitumgebung können die virtuellen Anlagen zur Laufzeit des Generierungsprozesses über das Verhaltensmodell manipuliert werden, so dass beispielsweise verschiedene Roboterposen in den Generierungsprozess integriert und somit eine hohe Varianz in den Bildern erzielt werden kann.

Stufe 2: Generierung synthetischer Trainingsdaten

Die Generierung der synthetischen Trainingsdaten erfolgt in einer virtuellen Szene, die schrittweise automatisch verändert wird. Durch die Automatisierung können eine Vielzahl an verschiedenen randomisierten Bildern nacheinander gerendert werden. Der Ablauf ist in Abbildung 3 dargestellt.

In der Initialisierungsphase werden die Modelle der Objekte und Hintergründe für die virtuelle Szene importiert. Zur Datengenerierung kommt der Ansatz der Domain Randomization zum Einsatz. Bei diesem Ansatz wird eine Vielzahl von virtuellen Szenen anhand randomisierter Parameter automatisiert erstellt. Dabei werden sowohl Objektparameter wie Anzahl, Position und Orientierung als auch das Hintergrundbild, Störobjekte im Vorder- oder Hintergrund, Lichtquellen sowie Kameraperspektiven randomisiert erzeugt und gerendert. Danach erfolgt die automatisierte Annotation der synthetischen Bilder, um die Position der zu erkennenden Objekte im Bild zu extrahieren und damit den Erkennungsalgorithmus anlernen zu können. Hierzu werden die betrachteten Objekte isoliert und jeder Objektinstanz ein eindeutiger Farbcode zugewiesen. In der daraus resultierenden Segmentationsmaske werden über eine Schwellwertbetrachtung die Bounding Boxes der Objektinstanzen berechnet. Dabei sind die Bounding Boxes das kleinstmögliche achsenparallele Rechteck, das die entsprechende Objektinstanz umschließt. Abschließend werden bei der Augmentation die generierten Bilder durch Filter wie Gaußsche Unschärfe und Farboperationen (z.B. Invertierung oder Graustufe) randomisiert manipuliert, um den Datensatz zu erweitern und die Varianz zu erhöhen.

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +

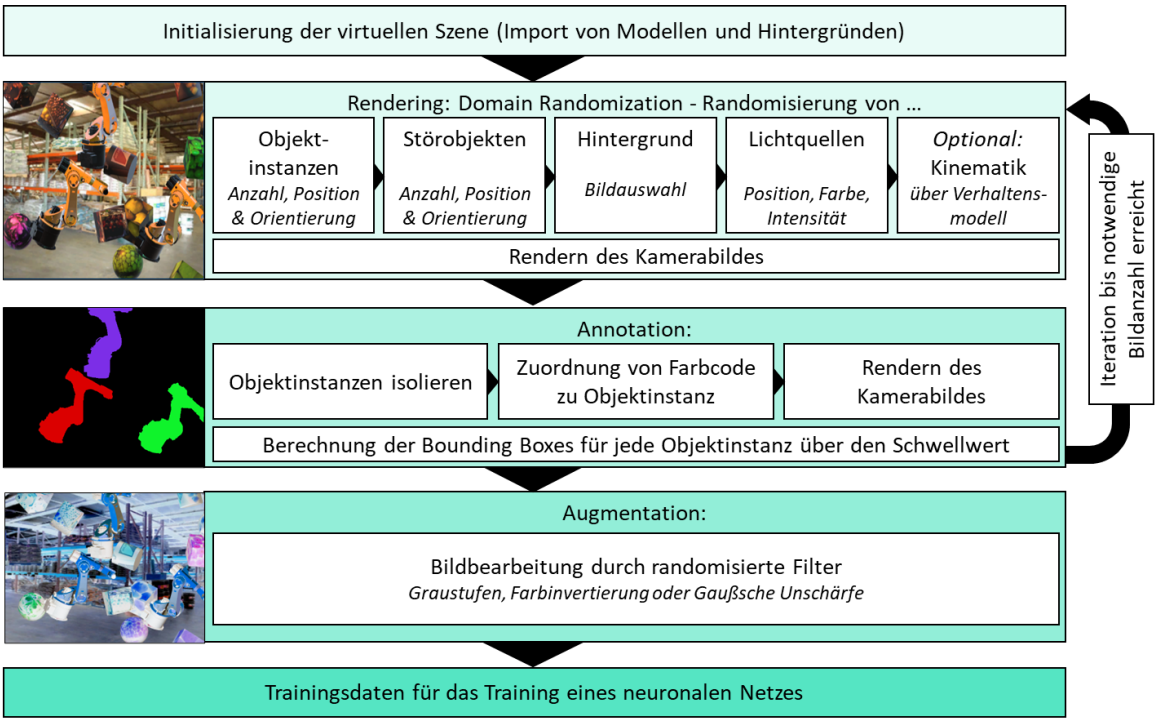

Abbildung 3: Ablauf der automatisierten Generierung der Trainingsdaten

Stufe 3: Training der neuronalen Netze

Der generierte Datensatz wird anschließend zum Training eines mehrschichtigen neuronalen Netzes eingesetzt. Für die Objekterkennung durch Merkmalextraktion in Bildern eignen sich insbesondere Convolutional Neural Networks (CNN). Die externen Daten (Bilder) werden in der Input-Schicht eingespeist und in der Output-Schicht werden die Klassifikation und Position der erkannten Objekte im Bild als Ergebnis ausgegeben. In mehreren dazwischenliegenden Schichten, den so genannten Hidden Layers, erfolgt die Verarbeitung und Auswertung der Daten. Bei Convolutional Neural Networks werden in den Convolutional Layers (Faltungsebenen) Merkmale (z.B. Kanten, Formen, Muster) durch den Einsatz von Filtern mit definierter Pixelgröße erkannt und im Bild extrahiert. Die Gewichtungen der Matrixelemente bei der Faltungsoperation werden während dem Trainingsprozess dynamisch angepasst. Während frühe Convolutional Layers Basismerkmale wie zum Beispiel Farben, Kanten und Linien analysieren, werden in nachfolgenden Schichten komplexere Merkmale wie zum Beispiel Muster erkannt. Die Ergebnisse werden in einer Ergebnismatrix abgespeichert. In der darauffolgenden Pooling Layer werden die Information über erkannte Merkmale verdichtet, indem jeweils die stärksten Merkmale in einer Matrix herausgefiltert werden. Dadurch erfolgt eine Trennung von überflüssigen und relevanten Informationen. Die Convolutional Layers und Pooling Layers wechseln sich in einer definierten Reihenfolge ab. Nach mehrfachen Iterationen von Merkmalextraktion und Infomationsverdichtung wird der Zusammenhang von mehreren Merkmalen über die abschließenden Fully Connected Layers abgebildet. Jeder dieser vollständig verknüpften Schichten besteht aus Neuronen (Datenpunkten), die über Gewichte mit den Neuronen der vorangegangenen Schicht sowie denen der nachfolgenden Schicht verbunden sind. Die Gewichte sind Zahlenwerte, die bestimmen welchen Einfluss eine Ausgabe eines Neurons auf die Eingabe der nachfolgenden Neuronen in der nächsten Schicht hat. Die Gewichte werden zu Beginn des Trainings mit zufälligen Werten belegt und während des Trainingsprozesses optimiert. Neben den Gewichten kann jedem Neuron ein Bias zugewiesen werden, der die Verschiebung des Ergebnisses der Schicht in eine vorgegebene Richtung ermöglicht. Die Kontrolle des Ergebnisses erfolgt beim überwachten Lernen anhand von bereits vorab annotierten Referenz-Bildern. Das neuronale Netz berechnet in jeder Trainingsiteration unter Berücksichtigung aller Gewichte die Abweichung der eigenen Vorhersage zu dem zur Verfügung gestellten annotierten Bild. Der resultierende Fehler wird zurück in das neuronale Netz geleitet (Backward-Propagation). Auf Basis des erzielten Zwischenergebnisses werden für die nächste Iteration die Gewichte und Bias-Terme so angepasst, dass der Fehler minimiert wird. Die Größe der Anpassung wird durch den Anteil, den ein bestimmtes Neuronen-Gewicht am Ergebnis hatte und die vorgegebene Lernrate bestimmt. Die benötigte Rechenleistung für das Training des neuronalen Netzes ist unter anderem abhängig von dessen Netzarchitektur und hängt maßgeblich von der Anzahl der zu trainierenden Parameter (Gewichte und Bias) eines Netzes ab.

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +

Das neuronale Netz wird so trainiert, dass es sich nur auf Merkmale des Objekts konzentriert und den Hintergrund und Störobjekte ausblendet und somit in allen Umgebungen funktionieren soll. Durch diesen Ansatz ist es wahrscheinlich, dass sich das neuronale Netz auch an die reale Umgebung anpassen kann, da die reale Umgebung nur eine Stichprobe dieser vielfältigen Trainingsvariationen sein wird [1].

Stufe 4: Objekterkennung mit Mixed Reality-Endgeräten

Nach Abschluss des Trainings wird das neuronale Netz zur Objekterkennung im Live-Kamerabild des Mixed Reality-Endgerätes eingesetzt. Das trainierte neuronale Netz kann die zuvor eingelernten Merkmale in den Kamerabildern des Mixed Reality-Endgeräts erkennen und die Position des erkannten Objekts im Kamerabild bestimmen. Zusätzliche Informationen über das erkannte Objekt können nun auf dem Mixed Reality-Endgerät in das Sichtfeld des Nutzers eingeblendet werden. Durch die Nutzung von Digitalen Zwillingen der erkannten Objekte (z.B. Maschinenkomponenten, Werkstücke) können statische und dynamische Zustandsdaten (z.B. Position, Geschwindigkeit, sonstige Sensor- und Aktorwerte) den entsprechenden Objekten zugeordnet und aus dem Digitalen Zwilling abgerufen werden.

4 Realisierung

Am Virtual Automation Lab (VAL) der Hochschule Esslingen wird eine Digital Twin as a Service Plattform (DTaaSP) entwickelt, die die flexible Kopplung von Digitalen Zwillingen mit Realdaten-Quellen wie industrielle Steuerungssysteme, Sensor-Gateways sowie Programmier- und Simulationswerkzeuge (Assets) über Kommunikationsstandards wie OPC UA oder MQTT ermöglicht. Die DTaaSP bindet dabei webbrowserbasierte User-Interfaces sowie Visualisierungs- und Interaktionstechnologien der Mixed Reality (Clients) ein (vgl. [7]). Die Server-Client Architektur der DTaaSP ist in Abbildung 4 (links) dargestellt.

Zur Erstellung eines Digitalen Zwillings können CAD-Daten der Objekte über das Webfrontend VAL 3D-Webstudio auf die DTaaSP hochgeladen werden. Die CAD-Daten werden dafür beispielsweise in einer CAD-Software erstellt oder mit Hilfe eines 3D Scanners hochauflösend und mit Texturen eingescannt. Das logische und kinematische Verhalten des Modells kann über einen 3D-Editor und einen Objektbaum in Verbindung mit einer JavaScript-Programmierschnittstelle modelliert und an reale Assets (z.B. industrielle Steuerung) oder virtuelle Assets (z.B. Offline-Programmiersystem) angebunden werden. Die Ausführung des Digitalen Zwillings erfolgt in der DT Runtime der DTaaSP. Durch die strikte Trennung des Digitalen Zwillings von den Nutzer-Endgeräten über eine Client-Schnittstelle kann der Digitale Zwilling mit seinem Verhaltensmodell endgeräte- und standortunabhängig genutzt werden. Die DTaaSP ermöglicht dadurch die flexible und endgeräteunabhängige Bereitstellung des Digitalen Zwillings auf verschiedenen Mixed Reality-Endgeräten, wobei sich auch mehrere Nutzer mit ihrem Mixed Reality-Endgerät auf dieselbe Szene verbinden können.

Die DTaaSP kann durch die Micro-Service-Architektur flexibel um zusätzliche Mehrwertdienste erweitert werden, wie beispielsweise einem Generierungsservice für synthetische Trainingsdaten und einem Service für Objekterkennung auf Basis von Deep Learning Algorithmen (siehe Abbildung 4 rechts). Durch die Umsetzung der Mehrwertdienste als Webservices können diese flexibel an die bestehende Infrastruktur der DTaaSP angebunden und mit anderen Mehrwertdiensten der DTaaSP (z.B. Sprachverarbeitung oder Zeitreihendatenbank) gekoppelt werden.

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +

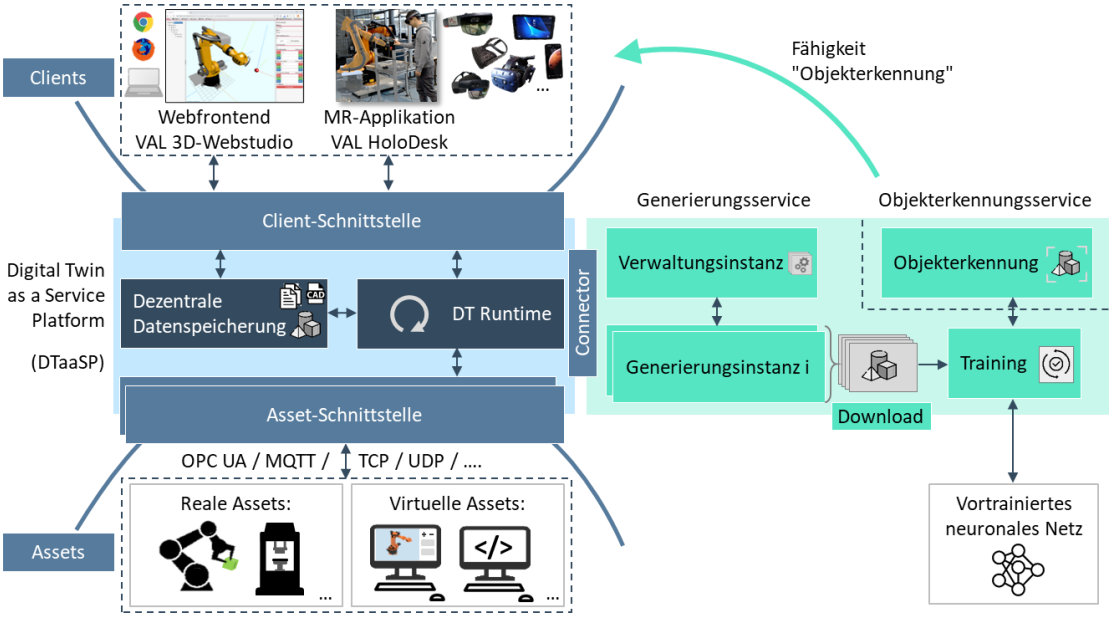

Abbildung 4: Architektur der am Virtual Automation Lab der Hochschule Esslingen entwickelten Digital Twin as a Service Plattform (DTaaSP) mit Erweiterung um den Generierungsservice und Objekterkennungsservice

Für die Generierung der synthetischen Trainingsdaten im Generierungsservice werden die bereits in der DTaaSP vorhandenen Modelle der Digitalen Zwillinge eingesetzt. Dabei können nicht nur die Geometriemodelle der betrachteten Objekte, sondern auch deren Verhaltensmodell genutzt werden. Das Verhaltensmodell wird zur Laufzeit der Trainingsdatengenerierung für die automatisierte Manipulation des Geometriemodells eingesetzt, wodurch eine Vielzahl verschiedener Trainingsdaten mit einer realistischen Zustandsveränderung (z.B. verschiedene Roboterposen) generiert werden können.

Der implementierte Webservice für die Generierung synthetischer Trainingsdaten umfasst eine Verwaltungsinstanz und mehrere Generierungsinstanzen, wobei jeder auszuführende Generierungsauftrag eine eigene Generierungsinstanz erstellt. In der Verwaltungsinstanz werden die Geometriemodelle und die erzeugten Datensätze gespeichert und über den Fortschritt laufender Generierungsprozesse informiert. Im Generierungsauftrag kann der Nutzer die grundlegende Konfiguration der Generierung, wie beispielsweise die Anzahl der Bilder, das Bildformat, die Bildauflösung und den Anteil der manipulierten Bilder für die drei Filteroperationen Graustufe, Invertierung und Gaußsche Unschärfe festlegen. Für die Festlegung der auszuwählenden Parameter wurden in Vorarbeiten der Einfluss der unterschiedlichen Parameter auf die Genauigkeit der Erkennung untersucht (vgl. [8]). Basierend auf der ausgewählten Konfiguration baut sich die Bildgenerierungsszene der entsprechenden Generierungsinstanz automatisiert auf.

Der Nutzer kann zudem festlegen, auf welcher Hardware (z.B. CPU oder GPU) die Trainingsdaten erstellt werden. Da das Rendering der Bilder sehr ressourcenintensiv ist, sollte die Trainingsdatengenerierung bevorzugt auf der GPU ausgeführt werden, da dies signifikant schneller zu einem Ergebnis führt als auf der CPU. Durch die offene Schnittstelle ist auch eine Anbindung an externe Rechen- und Speicherressourcen, wie beispielsweise an das High Performance Computing (HPC) Cluster bwUniCluster2.0[6], eine landesweite Hochleistungs- und Speicherinfrastruktur in Baden-Württemberg, möglich. Während der Generierung der Trainingsdaten kommunizieren Verwaltungs- und Generierungsinstanz bidirektional, um den aktuellen Status des Prozesses auszutauschen und den Nutzer über den Generierungsfortschritt zu informieren. Abschließend steht der generierte Trainingsdatensatz per Schnittstelle zum Export sowie zur weiteren Verarbeitung in der DTaaSP zum Beispiel für das Training eines neuronalen Netzes zur Verfügung.

Für die Anwendung der Objekterkennung in Mixed Reality wird der generierte Trainingsdatensatz in der DTaaSP zum Training eines neuronalen Netzes eingesetzt. Dabei eignen sich State-of-the-Art Netzwerke (wie z.B. EfficientDet [9], YOLOv4 [10] oder YOLOv7 [11]), die einen Fokus auf geringer Rechenzeit bei bestmöglicher Präzision legen, sodass die Latenz durch die Objekterkennung in Einzelbildern minimal ist.

[6] bwHPC: High Performance Computing, Data Intensive Computing and Large Scale Scientific Data Management in Baden-Württemberg. https://bwhpc.de

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +

Nach dem abgeschlossenen Training können die neuronalen Netze im Objekterkennungsservice innerhalb der DTaaSP über offene Schnittstellen angesprochen werden und erweitern somit die Endgeräte um die Fähigkeit der Objekterkennung. Die zyklisch übermittelten Live-Kamerabilder der Mixed Reality-Endgeräte werden im Objekterkennungsservice verarbeitet und die erkannten Objekte sowie deren Positionen an die Mixed Reality-Endgeräte übertragen und dem Nutzer auf dem Display angezeigt. So können dem Nutzer zudem hinterlegte Informationen des erkannten Objekts aus dem angebundenen Digitalen Zwilling kontextsensitiv auf dem Mixed Reality-Endgerät angezeigt und damit neue Anwendungen in der Automatisierungstechnik ermöglicht werden.

5 Anwendungsbeispiele

Mit der zuvor vorgestellten DTaaSP Plattform und deren Generierungsservice wurden bereits verschiedene Trainingsdatensätze für die Objekterkennung mit Deep Learning Algorithmen für Mixed Reality-Anwendungen umgesetzt. Nachfolgend werden zwei ausgewählte Anwendungen vorgestellt, um die Tragfähigkeit des zuvor beschriebenen Ansatzes zu evaluieren.

Anwendungsbeispiel 1:

Kontextsensitive Informationsdarstellung von Achsdaten eines Industrieroboters KUKA KR 16 auf Mixed Reality-Endgeräten

Für die Trainingsdatengenerierung wird das bereits auf der DTaaSP vorhandene Digitale Modell eines Industrieroboters KUKA KR 16 eingesetzt. Das Digitale Modell umfasst neben dem Geometriemodell auch das modellierte Roboterverhalten bei der Handhabungsaufgabe. Abbildung 5 zeigt den Ablauf der Anwendungserstellung.

Für die synthetische Trainingsdatengenerierung wird das Geometriemodell des Industrieroboters sowie des Werkstückes der Handhabungsaufgabe aus der DTaaSP an den Generierungsservice gekoppelt. Im Generierungsservice wird eine Datensatzkonfiguration für 10.000 Bilder im PNG-Bildformat mit einer Auflösung von jeweils 512x512 Pixel angelegt. Die Bilder werden mit dem zuvor beschriebenen Ansatz zur Trainingsdatengenerierung mittels Domain Randomization erzeugt, wobei je 1.000 Bilder durch Invertierung, Graustufen und Gaußschen Unschärfe-Filter augmentiert werden. Die Generierung der Datensätze der neuronalen Netze mit den dafür notwendigen Prozessschritten Rendering, Annotation und Augmentation erfolgt aufgrund der hohen Rechenintensivität sowie der Möglichkeit der Parallelisierbarkeit der Prozesse ausgelagert auf den Rechen- und Speicherressourcen des HPC Cluster bwUniCluster2.0. Als neuronales Netz wird das open source Netz YOLOv4 [10] eingesetzt, das bereits mit dem Microsoft COCO Datensatz [12] vortrainiert ist. Mittels Transfer Learning wird das bereits vortrainierte neuronale Netz mit Hilfe der generierten synthetischen Trainingsdaten an die spezielle Aufgabenstellung angepasst (Fine-Tuning). Insbesondere die Gewichte der Verbindungen der letzten Schichten werden für die spezielle Aufgabe neu initialisiert und im Trainingsverlauf optimiert. Die restlichen Gewichte werden anfänglich eingefroren und im Verlauf des Trainings schrittweise für die Optimierung freigegeben. Während die bisherigen Prozessschritte offline erfolgen, bedarf die Mixed Reality-Anwendung eine Online-Anbindung an den Objekterkennungsservice. Die angebundenen Mixed Reality-Endgeräte senden ihre Kamerabilder zyklisch per HTTP-Request (HTTP: Hypertext Transfer Protocol) an die Schnittstelle im Objekterkennungsservice der DTaaSP und erhalten die Information über die erkannten Objekte sowie deren Position innerhalb des Kamerabildes zurück. Während die Objekterkennung auf dem Server gerechnet wird, erfolgt die Kopplung der über die Client-Schnittstelle bereitgestellten Informationen der DTaaSP (z.B. Achswerte) sowie die Visualisierung der virtuell bereitgestellten Informationen auf dem Endgerät selbst.

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +

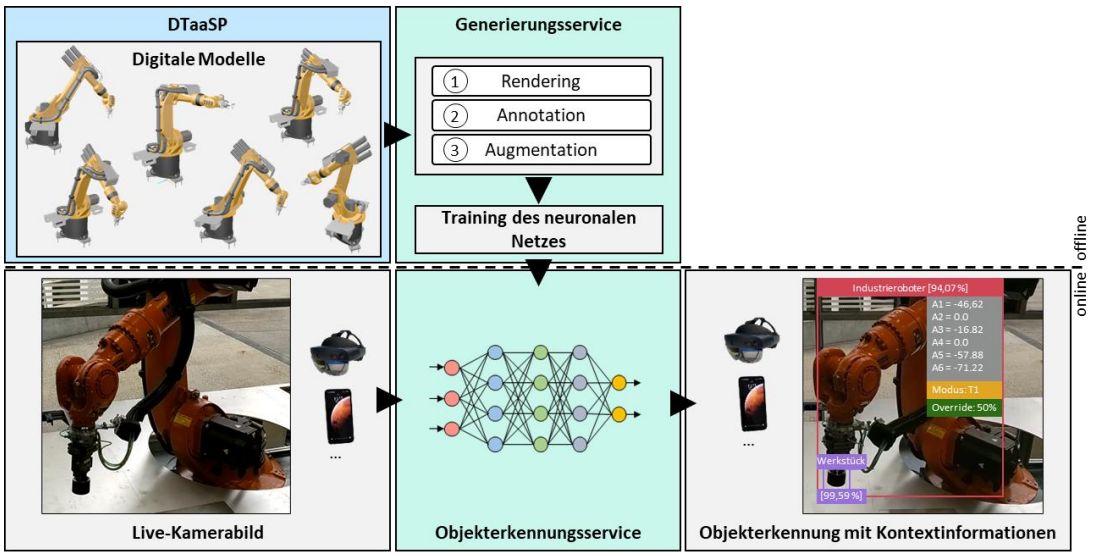

Abbildung 5: Anwendungsbeispiel 1 - Kontextsensitive Informationsdarstellung von Achsdaten eines Industrieroboters über Mixed Reality-Endgeräte

In diesem Anwendungsbeispiel werden nach erfolgreicher Erkennung des Industrieroboters KUKA KR 16 dessen Achswerte, Betriebsmodus und Override als Text eingeblendet. Darüber hinaus wird parallel zur Erkennung des Industrieroboters auch das vom Roboter gegriffene Werkstück erkannt und dessen Bezeichnung angezeigt. Damit kann gezeigt werden, dass eine Multi-Objekterkennung mit dem vorgestellten Ansatz möglich ist. Durch die kontextsensitive Einblendung der Achs- und Werkstückinformationen über das Mixed Reality-Endgerät kann der Handhabungsprozess immersiv analysiert und die Informationen intuitiv mit der Anlage und dem Prozess in Verbindung gebracht werden. Dies erweitert die bisherige Mensch-Maschine-Schnittstelle und ermöglicht neue Formen der Mensch-Maschine-Interaktion.

Anwendungsbeispiel 2:

Synthetische Trainingsdatengenerierung mit Hilfe eines 3D Scanners für die automatisierte Werkzeugerkennung über Mixed Reality-Endgeräte

Die Erstellung von Digitalen Modellen komplexer Objekte und Umgebungen wird durch das Fehlen von geometrischen Daten erschwert. Die Rekonstruktion der Modelle in CAD-Programmen ist nur unter großem manuellem Aufwand möglich. Der Einsatz von 3D Scannern ermöglicht bei fehlenden Modelldaten das dreidimensionale Scannen und digitale Nachbilden von einzelnen Objekten bis hin zu kompletten Umgebungen. Je nach Fähigkeit des 3D Scanners können neben der Geometrie auch Texturen erfasst und so ein hochaufgelöstes digitales Abbild des betrachteten Objekts rekonstruiert werden.

Dieses Anwendungsbeispiel soll zeigen, wie synthetische Trainingsdaten für die automatisierte Werkzeugerkennung mit Hilfe eines 3D Scanners generiert werden können. Als 3D Scanner wird der Einscan HX der Firma Shining 3D eingesetzt, der neben der Geometrie auch die Oberfläche des Objekts erfasst und in Form von Texturen auf der Geometrie abbildet. Nach erfolgter Scan-Operation wird das erzeugte Geometriemodell inklusive Texturen aus der Scan-Software exportiert und in die DTaaSP für den Generierungsservice importiert (siehe Abbildung 6 oben links). Bei diesem Anwendungsbeispiel werden sechs verschiedene Werkzeugklassen eingescannt: Schraubenzieher, Schraubenschlüssel, Hammer, Zange, Teppichmesser und Fuchsschwanzsäge.

Für die Generierung der synthetischen Trainingsdaten wird eine Kombination aus bereits bestehenden und eingescannten Geometriemodellen verwendet. Der Datensatz umfasst pro Objektklasse 2.000 individuelle Bilder und 8.000 gemischte Bilder in denen mehr als eine Objektklasse vorkommt. Die 20.000 Bildern werden durch Domain Randomization erstellt, wobei je 1.000 Bilder durch Invertierung und Graustufen sowie 4.000 Bilder durch Anwendung des Gaußschen Unschärfe-Filters augmentiert werden. Die Konfiguration für Bildformat und Bildauflösung wird analog zum ersten Anwendungsbeispiel mit PNG und jeweils 512x512 Pixeln gewählt. Für die betrachtete Aufgabenstellung wird analog zum ersten Anwendungsbeispiel das bereits mit dem Microsoft COCO Datensatz [12] vortrainiere neuronale Netz YOLOv4 [10]

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +

mittels Transfer Learning an das spezielle Anwendungsgebiet angepasst. Aufgrund der hohen Rechenintensivität werden der Generierungs- und Trainingsprozess an die Ressourcen des HPC Cluster bwUniCluster2.0 ausgelagert.

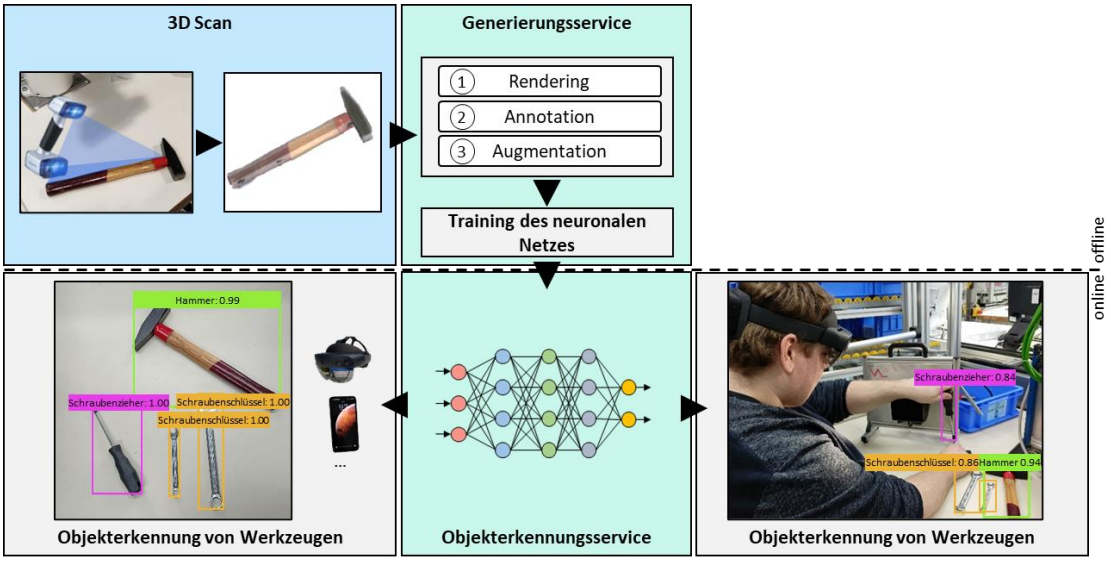

Abbildung 6: Anwendungsbeispiel 2 – Synthetische Trainingsdatengenerierung mit Hilfe eines 3D Scanners für die automatisierte Werkzeugerkennung über Mixed Reality-Endgeräte

Nach vollendetem Training ist das trainierte neuronale Netz über die Schnittstelle des Objekterkennungsservice der DTaaSP verfügbar und kann von den angebundenen Mixed Reality-Endgeräten genutzt werden (siehe Abbildung 6 unten). Die Objekterkennung von Werkzeugen über Mixed Reality-Endgeräte kann beispielsweise für die Erkennung von manuellen Montageschritten anhand der verwendeten Werkzeuge eingesetzt werden.

6 Zusammenfassung und Ausblick

In diesem Beitrag wurde eine Methode für die synthetische Generierung von Trainingsdaten und Anwendungen zur Objekterkennung mit Deep Learning Algorithmen auf Mixed Reality-Endgeräten vorgestellt. Die Methode umfasst den gesamten Prozess von der Generierung synthetischer Trainingsdaten auf Basis Digitaler Modelle der zu erkennenden Objekte, über das Training des künstlichen neuronalen Netzes bis hin zur Bereitstellung und Anwendung der Objekterkennung auf Mixed Reality-Endgeräten. Die automatisierte Generierung der synthetischen Trainingsdaten soll den Prozess der Trainingsdatenakquise bedeutend beschleunigen und die Qualität der Trainingsergebnisse gegenüber der Verwendung manuell erstellter Trainingsdaten erhöhen. Die Funktionsfähigkeit der vorgestellten Methode wurde durch die Realisierung des Konzepts in der am Virtual Automation Lab (VAL) der Hochschule Esslingen entwickelten Digital Twin as a Service Plattform (DTaaSP) sowie anhand von zwei ausgewählten Anwendungsbeispielen aufgezeigt. Insbesondere die Kombination von Methoden der Objekterkennung mit Mixed Reality-Technologien und realdatengetriebenen Digitalen Zwillingen wird in Zukunft zahlreiche weitere Anwendung im industriellen Kontext finden.

Aufbauend auf diesen Ergebnissen soll in einem nächsten Schritt der Einsatz der vorgestellten Methode für die Mitarbeiterunterstützung bei der manuellen Montage komplexer Baugruppen erforscht werden. Die Monteure sollen mittels Mixed Reality-Endgerät kontextsensitive Inhalte wie beispielsweise Montagefortschritt, Montagefehler und Handlungsanweisungen in der realen Arbeitsumgebung visualisiert bekommen. Die für den Montageprozess relevanten Daten sollen dabei teilautomatisiert aus den Entwicklungsdaten abgeleitet und in der vorgestellten DTaaSP zur Verfügung gestellt werden. Die über das Mixed Reality-Endgerät abgefragten und eingebrachten Informationen sollen in einer Datenbank angereichert und für weitere Monteure zur Verfügung gestellt werden. Durch diesen Ansatz soll der kontinuierliche mitarbeiterübergreifende Austausch von Fehlerbehebungsstrategien sichergestellt werden.

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +

7 Danksagung

Die Autoren der Veröffentlichung danken dem Ministerium für Wissenschaft, Forschung und Kunst (MWK) des Landes Baden-Württemberg für die Förderung des Projektes „accelerate KI“ und die Förderung der Hochleistungs- und Speicherinfrastruktur HPC Cluster bwUniCluster2.0.

8 Literatur

[1] Hinterstoisser, S.; Pauly, O.; Heibel, H.; Marek, M.; Bokeloh, M.: An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Detection. ICCV Workshops 2019: 2787-2796. Seoul. 2019. https://doi.org/10.1109/ICCVW.2019.00340

[2] de Melo, C.M.; Torralba, A.; Guibas, L.; DiCarlo, J.; Chellappa, R.; Hodgins, J.: Next-generation deep learning based on simulators and synthetic data. Trends Cogn Sci. 2022 Feb;26(2):174-187. https://doi.org/10.1016/j.tics.2021.11.008

[3] Boikov, A.; Payor, V.; Savelev, R.; Kolesnikov, A.: Synthetic Data Generation for Steel Defect Detection and Classification Using Deep Learning. Symmetry 2021, 13, 1176. https://doi.org/10.3390/sym13071176

[4] Nvidia Corporation: NVIDIA, BMW Blend Reality, Virtual Worlds to Demonstrate Factory of the Future. 2021 April 13. https://blogs.nvidia.com/blog/2021/04/13/nvidia-bmw-factory-future

[5] Tremblay, J.; Prakash, A.; Acuna, D.; Brophy, M.; Jampani, V.; Anil, C.; To, T.; Cameracci, E.; Boochoon, S.; Birchfield, S.: Training deep networks with synthetic data: Bridging the reality gap by domain randomization. Proceedings of the IEEE conference on computer vision and pattern recognition workshops 2018 (pp. 969-977). https://doi.org/10.1109/CVPRW.2018.00143

[6] Prakash, A.; Boochoon, S.; Brophy, M.; Acuna, D.; Cameracci, E.; State, G.; Shapira, O.; Birchfield, S.: Structured domain randomization: Bridging the reality gap by context-aware synthetic data. 2019 International Conference on Robotics and Automation (ICRA) (pp. 7249-7255). IEEE. https://doi.org/10.1109/ICRA.2019.8794443

[7] Schnierle, M.; Röck, S.: Plattform für die Mixed-Reality-in-the-Loop-Simulation. wt-online. Internet: www.werkstattstechnik.de. Düsseldorf: Springer-VDI-Verlag, 2018.

[8] Kloos, Y.: Generierung synthetischer Trainingsdaten für die Objekterkennung mit Deep Learning in industriellen Anwendungen unter Verwendung Digitaler Zwillinge. 2021. Unveröffentlichte Bachelorthesis. Hochschule Esslingen.

[9] Tan, M.; Pang, R.; Le, Q.V.: EfficientDet: Scalable and Efficient Object Detection. CoRR, abs/1911.09070. 2019. https://doi.org/10.48550/arXiv.1911.09070

[10] Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M.: YOLOv4: Optimal Speed and Accuracy of Object Detection. CoRR, abs/2004.10934. 2020. https://doi.org/10.48550/arXiv.2004.10934

[11] Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M.: YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. CoRR, abs/2207.02696. 2022. https://doi.org/10.48550/arXiv.2207.02696

[12] Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Fleet, D. et al. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8693. Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48

Open Access. © 2023 Thomas Fettahoglu, Jana Hönig, Marc Schnierle, Sascha Röck
This work is licensed under the Creative Commons Attribution 4.0 License. +