DSGVO-konforme Personendetektion in 3D-LiDAR-Daten mittels Deep Learning Verfahren

Dennis Sprute, Tim Westerhold, Florian Hufen, Holger Flatt und Florian Gellert

Zusammenfassung

Im Fabrikkontext spielt die Detektion von Personen eine wichtige Rolle bei Maßnahmen zur Erhöhung der Sicherheit von Arbeitern oder zur Optimierung des Fabriklayouts. Kamerasensoren bilden die Grundlage für robuste bildbasierte Personendetektionsverfahren, werden aber aufgrund von Datenschutzaspekten häufig kritisch gesehen. Diese Bedenken können durch die Lokalisation von IoT-Devices, die von Personen getragen werden, adressiert werden, jedoch muss eine Person stets mit einem entsprechendem IoT-Device ausgerüstet sein. In diesem Beitrag wird ein alternativer Ansatz zur Adressierung der Problematik vorgeschlagen, der auf etablierten Bildverarbeitungsverfahren beruht, jedoch inhärent DSGVO-konform ist. Hierzu wird distanzmessende 3D-LiDAR-Sensorik genutzt, um 3D-Punktwolken der Umgebung aufzunehmen. Diese ermöglichen eine Detektion von Personen (Klassifikation und Lokalisation), jedoch keine Identifizierung der Personen. Hierfür wird ein Verfahren vorgestellt, das einzelne Objekte in einer Punktwolke zunächst in ein Tiefenbild umwandelt, um auf diesem anschließend robuste Bildverarbeitungsverfahren basierend

D. Sprute (✉) · T. Westerhold · F. Hufen · H. Flatt · F. Gellert
Fraunhofer IOSB, Institutsteil für industrielle Automation (IOSB-INA), Lemgo, Deutschland
E-Mail: dennis.sprute@iosb-ina.fraunhofer.de

T. Westerhold
E-Mail: tim.westerhold@iosb-ina.fraunhofer.de

F. Hufen
E-Mail: florian.hufen@iosb-ina.fraunhofer.de

H. Flatt
E-Mail: holger.flatt@iosb-ina.fraunhofer.de

F. Gellert
E-Mail: florian.gellert@iosb-ina.fraunhofer.de

V. Lohweg (Hrsg.), *Bildverarbeitung in der Automation*, Technologien für die intelligente Automation 17, https://doi.org/10.1007/978-3-662-66769-9_3

auf Deep Learning einzusetzen. Die Evaluation des Verfahrens zeigt eine Genauigkeit (Accuracy) von 98%, um zwischen Personen und anderen Objekten zu unterscheiden, und ist somit für darauf aufbauende Anwendungen gut geeignet.

Schlüsselwörter

Personendetektion · Deep Learning · 3D-LiDAR · 3D-Punktwolke · DSGVO-Konformität

1 Motivation

Die Erfassung und genaue Lokalisation von Personen im Fabrikkontext kann einen wichtigen Beitrag zur Erhöhung der Sicherheit von Arbeitern und zur Optimierung des Fabriklayouts leisten. Beispielsweise kann durch die Detektion von unautorisierten Personen in Sicherheitsbereichen oder die Erkennung von Notfallsituationen die Sicherheit erhöht werden. Zudem kann die Analyse von Personenbewegungen oder die Erkennung von Gruppenbildungen am Fließband zur Optimierung des Fabriklayouts genutzt werden. Bei der Erfassung von Personen spielt insbesondere in Deutschland und der EU der Datenschutz eine wichtige Rolle, bei dem es um den Schutz von personenbezogenen Daten von Personen geht[1]. Es sollte also nicht möglich sein, eine erfasste und lokalisierte Person zu *identifizieren*, sodass z. B. erzeugte Bewegungsprofile einer konkreten Person zugeordnet werden können. Kamerasensoren, die an geeigneten Stellen in einer Fabrik installiert sind, können sehr gut mittels heutigen Verfahren des Stands der Technik für die Detektion von Personen in Bildern genutzt werden [1, 2]. Jedoch erlauben die Bilder bei hohen Auflösungen durch die Bestimmung von Merkmalen wie Gesicht, Hautfarbe u. ä. oftmals auch eine Identifikation der entsprechenden Personen, sodass dieser Ansatz kritisch zu sehen ist. Weitere Ansätze für diese Problemstellung umfassen Systeme zur Indoor-Lokalisierung, die Personen mittels funkbasierten IoT-Devices lokalisieren können [3]. Hierbei muss jedoch jeder Arbeiter stets mit einem entsprechenden IoT-Tag ausgerüstet sein.

In diesem Beitrag wird ein alternativer Ansatz vorgeschlagen, der die Vorteile kamerabasierter Sensoren und bildbasierter Verarbeitung nutzt, aber inhärent DSGVO-konform ist. Dieser Ansatz verwendet distanzmessende 3D-LiDAR-Technologie in Kombination mit bildbasierten Deep Learning Verfahren zur Detektion (Lokalisation und Klassifikation) von Personen in 3D-Punktwolken. Im Gegensatz zu Kameras entstehen auf diese Weise keine 2D-Bilder mit visuellen Informationen, sondern 3D-Punktwolken der räumlichen Umgebung, die bei heutigen Auflösungen keine Identifikation von Personen zulassen (s. Abb. 1).

Die folgenden Abschnitte dieses Beitrags sind wie folgt gegliedert. Im nächsten Abschnitt wird ein Überblick über heutige Ansätze zur Objektdetektion in 2D-Bildern und

[1] https://www.datenschutz-grundverordnung.eu/ (abgerufen am 15.09.2022)

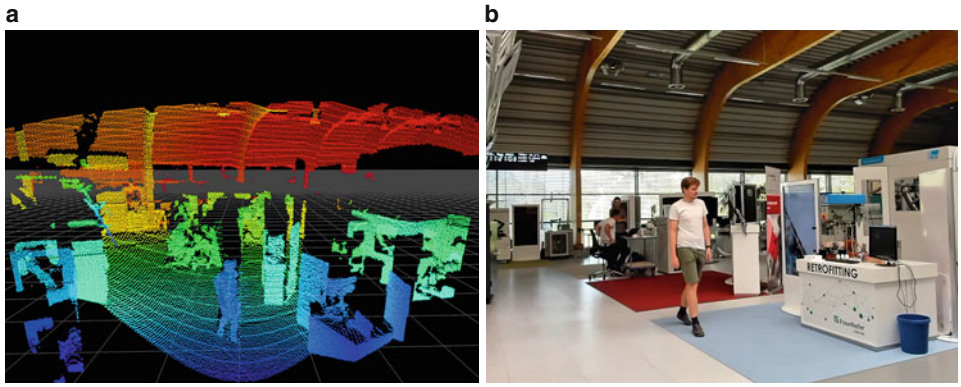

Abb. 1 3D-Punktwolke (**a**) und 2D-Farbbild (**b**) einer Szene

3D-Punktwolken gegeben, bevor das entwickelte DSGVO-konforme Personendetektionsverfahren im darauffolgenden Abschnitt im Detail vorgestellt wird. Dieses wird anschließend im folgenden Abschnitt bzgl. verschiedener Evaluationsmetriken bewertet. Abgeschlossen wird dieser Beitrag mit einem gesamtheitlichen Fazit und einem Ausblick auf weitere Folgeaktivitäten.

2 Stand der Technik

Da bei dem vorgeschlagenen Ansatz 3D-Punktwolken mittels bildbasierten Deep Learning Verfahren verarbeitet werden, wird in den folgenden Unterabschnitten auf den Stand der Technik bei Objektdetektionsverfahren für 2D-Bilder und 3D-Punktwolken eingegangen.

2.1 Objektdetektion in 2D-Bildern

Die Detektion von Objekten in 2D-Bildern hat sich in den letzten 10 Jahren stark weiterentwickelt und hat einen robusten Stand erreicht, der auch in Produkten eingesetzt wird. Der Grund für diesen großen Fortschritt begann mit dem Aufkommen der ersten tiefen neuronalen Netze zur Klassifikation von Bildern zu Beginn des vergangenen Jahrzehnts [4]. Auch wenn die Idee von neuronalen Netzen zur direkten Verarbeitung von Bilddaten (*Convolutional Neural Network*, CNN) zu diesem Zeitpunkt nicht mehr neu war [5], so führten die Verbesserung der verfügbaren (GPU)-Rechenkapazität und die Verfügbarkeit großer annotierter Datensätze [6] zu einem Durchbruch von tiefen neuronalen Netzen. Dies zeigt sich vor allen Dingen durch immer neuere und komplexere Architekturen der neuronalen Netze für die Bildklassifikation, die zu einer stetigen Verbesserung der Erkennungsleistung auf komplexen und herausfordernden Bilddatensätzen führten. Be-

kannte grundlegende Architekturen sind hierbei u. a. VGG [7], GoogleNet/Inception [8], ResNet [9] oder DenseNet [10]. Zudem gab es viele inkrementelle Weiterentwicklungen dieser Architekturen [11, 12], sodass es heutzutage eine sehr gute und robuste Basis für die Bildklassifikation gibt.

Dieser Fortschritt im Bereich der Bildklassifikation mittels Verfahren des Deep Learning führte auch zu einem signifikanten Fortschritt im Bereich der Objektdetektion, bei dem mehrere Objekte in einem Bild sowohl in Form von *Bounding Boxes* lokalisiert als auch klassifiziert werden. Bei diesen Verfahren unterscheidet man generell zwischen zweistufigen und einstufigen Verfahren. Die Familie von R-CNN Architekturen [2, 13, 14] ist die wohl bekannteste Vertreterin von zweistufigen Objektdetektionsverfahren, bei denen zunächst Kandidaten für Objekte auf unterschiedliche Weise generiert und anschließend einzeln klassifiziert werden. Im Gegensatz dazu bestimmen einstufige Verfahren die Bounding Boxes und Klassenzugehörigkeit von Objekten in einem Schritt, ohne dass explizit Kandidatenregionen generiert werden müssen. Hierbei wird ein Backbone-Netz zur Extraktion von *Feature Maps* genutzt (ähnlich wie bei einer Bildklassifikation) und anschließend weitere Schichten zur Bestimmung der Bounding Boxes und Klassenzugehörigkeit angefügt. Bekannte Architekturen dieser Kategorie sind YOLO (und dessen Weiterentwicklungen) [1, 15], SSD [16] und RetinaNet [17]. Heutzutage bilden bildbasierte Objektdetektionsverfahren die Basis für viele verschiedene Anwendungen, u. a. bei der Verkehrsüberwachung, in der Robotik oder in der Industrie. Dies zeigt, dass neuronale Netze zur Objektdetektion auf 2D-Bildern einen hohen Reifegrad erreicht haben.

2.2 Objektdetektion in 3D-Punktwolken

Im Vergleich zur Objektdetektion auf 2D-Bildern ist die Detektion von Objekten in 3D-Punktwolken komplexer und bringt zusätzliche Herausforderungen mit sich. So sind 3D-Punktwolken inhärent unsortiert, nur spärlich besetzt und die Punktdichten unterscheiden sich stark. Diese Effekte entstehen z. B. durch Verdeckungen, Scanmuster oder die effektive Reichweite des Sensors, wobei Punkte in der Entfernung eine geringere Dichte aufweisen als in der Nähe. Ähnlich wie im Bereich der bildbasierten Objektdetektion können hier klassische Ansätze genutzt werden, bei denen Merkmale von Objekten manuell entwickelt werden und für eine anschließende Klassifikation dienen.

Mit dem Aufkommen von Deep Learning Ansätzen im Gebiet der Bildverarbeitung können diese auch für eine 3D-Objektdetektion unter Anpassungen genutzt werden. Derartige Verfahren erfordern strukturierte Daten/Tensoren, z. B. Bilder oder Videos, was jedoch nicht zu den Eigenheiten von Punktwolken gehört, sodass die Verfahren entsprechend adaptiert werden müssen. Qian et al. unterscheiden generell zwischen zwei unterschiedlichen Ansätzen (und einer Kombination aus beiden Ansätzen), um diese Herausforderung zu adressieren [18]: Voxel-basierte Ansätze wandeln die irregulären Punktwolken in reguläre Strukturen um, auf denen dann CNN angewandt werden können. Ein wichtiger Vertreter dieser Kategorie ist das 3D-Detektionsframework VoxelNet, das die Punktwolke

in gleich große Voxel aufteilt, die durch eine einheitliche Merkmalsrepräsentation beschrieben werden und als Basis für eine Objektdetektion dienen [19]. Weitere Vertreter dieser Kategorie sind PointPillars [20], wo eine Punktwolke zunächst in der x-y-Ebene diskretisiert wird und in eine Menge von *Pillars* resultiert, und CenterPoint [21], bei dem auf Basis einer erstellten *top-view* Karte die Objektzentren bestimmt werden. In der zweiten Kategorie von Ansätzen werden Punktwolken direkt verarbeitet, wie es z. B. bei PointNet [22] der Fall ist, das die Basis für weitere Verfahren bildet [23, 24]. Die aktuelle Forschung im Bereich der Objektdetektion in 3D-Punktwolken zeigt, dass diese Verfahren einen enormen Fortschritt machen und hohes Potenzial aufweisen, aber noch nicht den Reifegrad von bildbasierten Objektdetektionsverfahren erreicht haben, insbesondere auch im Hinblick auf Verfügbarkeit von entsprechenden Algorithmen in Open-Source-Software-Bibliotheken oder Unterstützung durch die Community.

Weitere Ansätze kombinieren für bessere Ergebnisse Tiefeninformationen mit Farbbildern [25, 26], jedoch widerspricht dies dem Ziel einer DSGVO-konformen Lösung, die im Fokus dieses Beitrags steht.

3 DSGVO-konforme Personendetektion

Um von der Robustheit bildbasierter Personendetektionsverfahren mittels Deep Learning zu profitieren und gleichzeitig inhärent Aspekte des Datenschutzes zu berücksichtigen, wird in diesem Beitrag ein Ansatz basierend auf distanzmessender 3D-LiDAR-Sensorik in Kombination mit etablierten bildbasierten Deep Learning Verfahren vorgeschlagen. Dies lässt sich als zweistufiges Objektdetektionsverfahren einordnen, bei dem zunächst Objektregionen in der 3D-Punktwolke generiert und anschließend im 2D-Bild klassifiziert werden. Ein Überblick über das entwickelte Personendetektionsverfahren ist in Abb. 2 dargestellt.

3.1 Datenerfassung & Hintergrundentfernung

Zur Datenerfassung wird 3D-LiDAR-Sensorik eingesetzt, die statisch in die Umgebung installiert wird und die Umgebung in Form einer 3D-Punktwolke abbildet. Diese ermöglicht die Detektion von Personen (und anderen Objekten), jedoch keine Identifikation der Personen, wie es z. B. mit hochauflösenden Kameras möglich ist. Daher werden auf diese Weise keine personenbezogenen Daten aufgenommen. Um die zu verarbeitende Datenmenge zu reduzieren, wird das statische Setup des Sensors ausgenutzt, denn die relevanten Objekte (Personen) sind nicht Teil der statischen Umgebung. Hierzu wird der Hintergrund der aktuellen Punktwolke mit Hilfe eines zuvor erstellten Hintergrundmodells abgeglichen und entfernt. Das Hintergrundmodell wird initial über mehrere Frames bei einem statischen Hintergrund erstellt und umfasst somit die statischen Messpunkte der Umgebung, z. B. nicht-bewegliche Maschinen oder statische Strukturen der Fabrikumgebung. Das Er-

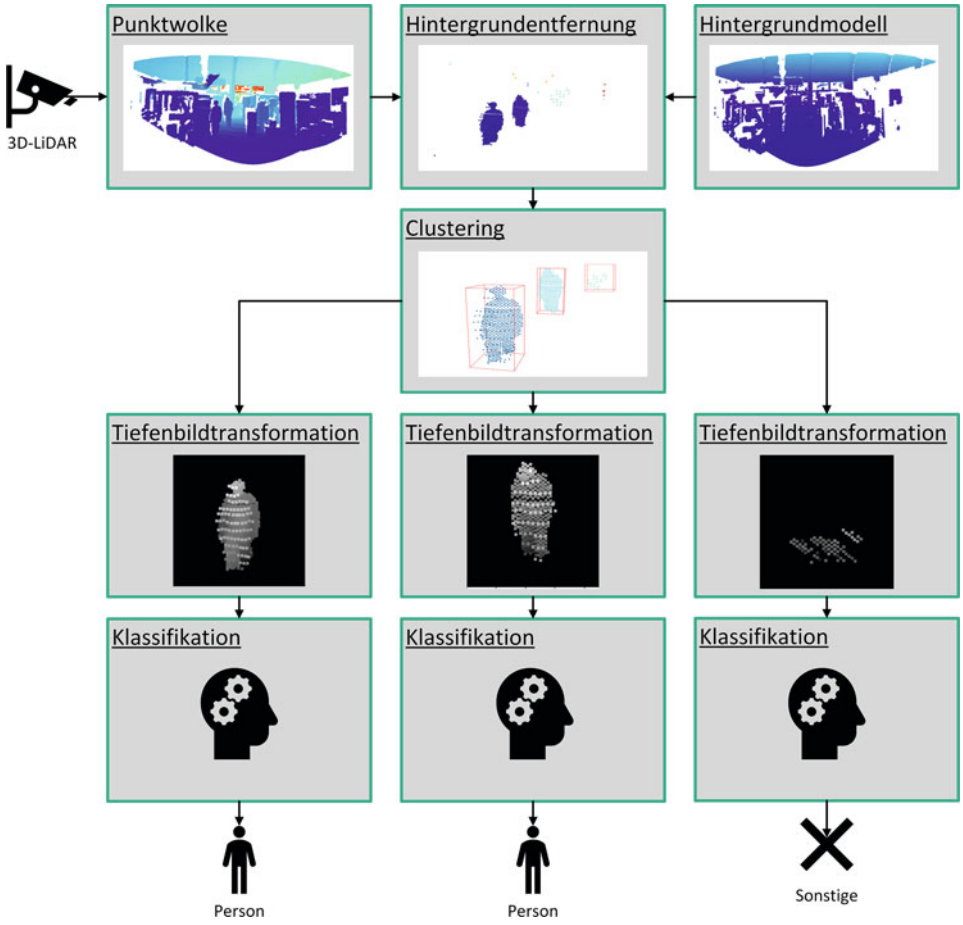

Abb. 2 Visualisierung der Verarbeitungspipeline

gebnis der Hintergrundentfernung ist eine Punktwolke, die nur noch Punkte enthält, die sich nicht im Hintergrundmodell befinden. Dies sind im Fabrikkontext typischerweise sich bewegende Objekte, wie z. B. Personen, Roboterarme oder Fahrerlose Transportfahrzeuge (FTF). Da diese Menge an Punkten im Verhältnis zur gesamten Anzahl an Punkten in der Punktwolke in den meisten Fällen kleiner ist, kann so auch die zu verarbeitende Datenmenge abhängig von der Umgebung signifikant reduziert werden. Dies erleichtert eine spätere Umsetzung des Verfahrens auf einer Embedded Hardware mit limitierten Ressourcen.

3.2 Clustering

Das Ziel des nächsten Verfahrensschritts ist die Generierung von Objektregionen in der 3D-Punktwolke, die anschließend in 2D-Bilder transformiert werden. Dazu werden räumlich naheliegende Punkte mit Hilfe des dichtebasierten Clusteringverfahrens DBSCAN [27] zu Objekten zusammengefasst. Dieser Algorithmus wird genutzt, da er Cluster in beliebiger Form finden kann, die Anzahl der Cluster nicht von vornherein bekannt sein muss und Rauschobjekte erkannt werden können. Das Resultat des Clustering ist eine Menge von Objekten, wobei jedes Objekt wiederum aus einer Menge von Punkten besteht.

3.3 Tiefenbildtransformation & Klassifikation

Aufgrund des Messprinzips eines LiDAR-Sensors lassen sich die Objekte direkt im 3D-Raum lokalisieren, jedoch sind die Objekte noch nicht klassifiziert. Um hier auf etablierte Verfahren des bildbasierten Deep Learning zurückzugreifen, werden die Punktwolken der Objekte mit entsprechenden Eigenschaften im darauffolgenden Schritt in ein 2D-Tiefenbild transformiert, wobei der Grauwert die Distanz zum Sensor angibt. Hierbei wird eine Frontalansicht gewählt, um den charakterisierenden Umriss einer Person bestmöglich zu erfassen. Dabei wird die Punktwolke aus der Sicht des 3D-LiDAR-Sensors betrachtet und der entsprechende Tiefenwert in ein 2D-Bild projiziert. Jedes Tiefenbild, das mit einem Objekt korrespondiert, wird abschließend mittels eines speziell angepassten tiefen neuronalen Netzes klassifiziert. Hierfür wird ein zuvor eigener aufgebauter Datensatz mit Annotationen für das Training des neuronalen Netzes genutzt, das die beiden Klassen *Person* und *Sonstige* berücksichtigt.

4 Evaluation

Zur Evaluation des entwickelten Verfahrens wurde eine mobile Messeinrichtung entworfen, um an verschiedenen Standorten 3D-Punktwolken aufzunehmen. Anschließend wurde ein Bilddatensatz aufgebaut und entsprechend annotiert, sodass dieser für das Training des neuronalen Netzes zur Objektklassifikation genutzt werden konnte. Die Details zu der Evaluation werden in den folgenden Unterabschnitten genauer erläutert.

4.1 Hardware

Die entwickelte Messeinrichtung umfasst u. a. eine Recheneinheit inklusive Datenspeicher, einen WLAN Access Point und einen Akku, um eine mobile und temporäre Datenaufzeichnung zu ermöglichen. Als 3D-LiDAR-Sensor, der an die Messeinrichtung

angeschlossen wird, wird ein Blickfeld Cube 1[2] verwendet. Dieser LiDAR hat eine typische Reichweite von 1,5 m bis 75 m mit einem maximalen Öffnungswinkel vom $72° \times 30°$, sodass ein weites Sichtfeld auf diese Weise abgedeckt werden kann und für einen Fabrikkontext geeignet ist. Die Auflösung der Punktwolke und die Bildwiederholrate lässt sich in Abhängigkeit voneinander konfigurieren. Mit dem Ziel, Personen zu detektieren, wurde der Fokus bei der Parametrisierung des Sensors vornehmlich auf eine hohe Auflösung der Punktwolke und weniger auf eine hohe Bildwiederholrate gelegt. Hierbei wurde eine vertikale Auflösung von 230 Scanlinien und eine horizontale Auflösung von 0,4° bei einer Bildwiederholrate von 2,4 Hz gewählt. Dadurch sollen Details zur Klassifikation eines Objekts als Person erkennbar werden, während die relativ geringe Bildwiederholrate für die Erfassung von Personen bei typischen Geschwindigkeiten von etwa $1{,}5\,\mathrm{m\,s^{-1}}$ ausreichend ist.

4.2 Datenaufnahme & Datensatz

Mittels dieser Messeinrichtung mit angeschlossenem 3D-LiDAR-Sensor wurde ein Datensatz von Punktwolken aufgenommen. Hierzu wurde der Sensor auf einem Stativ in einer Höhe von etwa 4 m mit einer Neigung von 16° an unterschiedlichen Standorten installiert, um einem beispielhaften Aufbau in einer Fabrikumgebung nahezukommen. Diese extrinsischen Kalibrierungsparameter wurden gespeichert und bei der Tiefenbildtransformation genutzt. Anschließend wurden gezielt Punktwolken mit Objekten, insbesondere Personen, aufgezeichnet, wobei auch auf eine möglichst hohe Varianz geachtet wurde. Dies sind Varianten, die auch in (größeren) Fabriken auftauchen können. So wurden u. a. allein gehende Personen, Personengruppen, Personen mit Koffern, Fahrrädern oder Schiebewagen mit aufgezeichnet. Die Personen hatten bei der Datenaufnahme eine Distanz von bis zu 25 m zum Sensor. Ein Betreuer des Messaufbaus war während der Datenakquise permanent anwesend und hat sich zu den aufgezeichneten Punktwolken die entsprechenden Objektklassen notiert.

Jede der aufgezeichneten Punktwolken wurde anschließend mit Hilfe des in Kap. 3 beschriebenen Verfahrens bis zur Erstellung eines Tiefenbildes pro Objekt verarbeitet. Auf diese Weise ist ein Bilddatensatz mit etwa 30K Bildern mit einer Auflösung von 224×224 Pixel entstanden[3]. Jedes dieser Bilder wurde manuell mit einer der beiden zu berücksichtigen Klassen annotiert: *Person* und *Sonstige*. Ein Überblick über Beispielbilder der beiden Klassen ist in Abb. 3 visualisiert. Während die Klasse *Person* alle Bilder mit Personen enthält, umfasst die Klasse *Sonstige* alle vom Hintergrund extrahierten Objekte, die keine Person darstellen, z. B. Teile von sich bewegenden Objekten.

[2] https://www.blickfeld.com/lidar-sensor-products/cube-1/ (abgerufen am 15.09.2022)

[3] Eine Bildauflösung von 224×224 Pixel ist typisch für CNN-basierte Klassifikationsnetze und historisch bedingt [4, 7, 8].

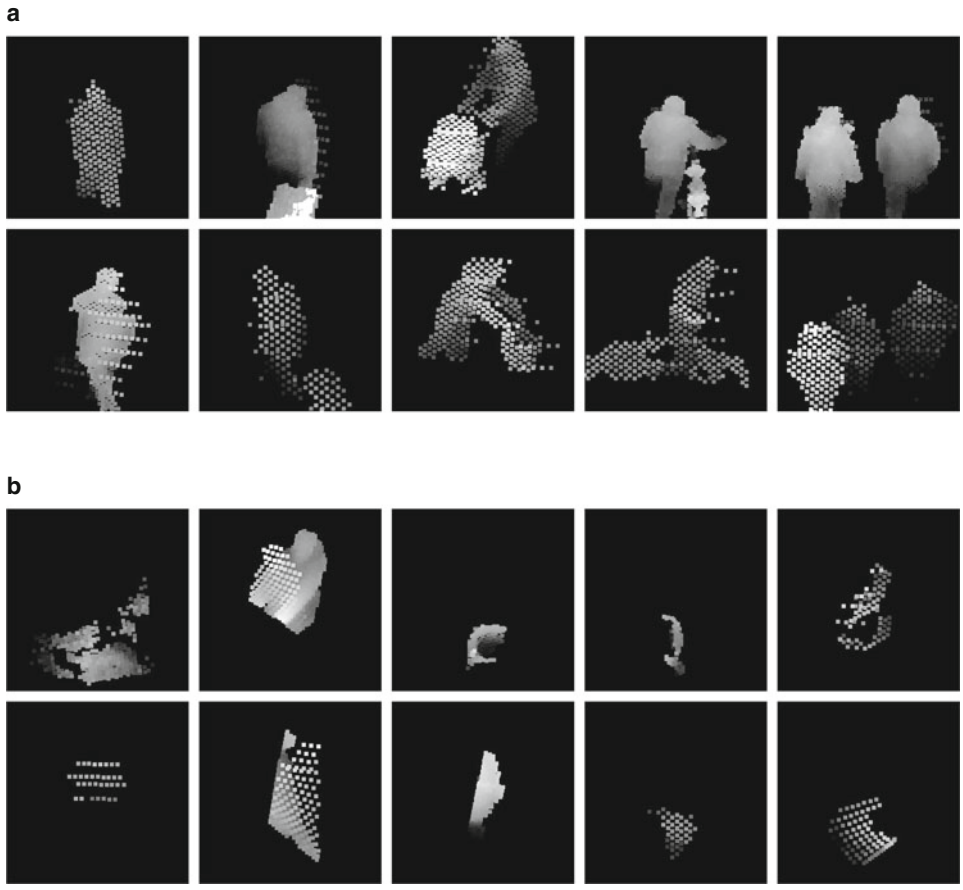

Abb. 3 Beispielbilder aus dem erstellten Bilddatensatz mit den beiden Klassen *Person* (**a**) und *Sonstige* (**b**) für das Training des neuronalen Netzes

4.3 Training

Dieser annotierte Bilddatensatz wurde anschließend genutzt, um ein tiefes neuronales Klassifikationsnetz zu trainieren. Hierzu wurde der Datensatz zunächst in einem Verhältnis von 70:10:20 in einen Trainings-, Validierungs- und Testdatensatz aufgeteilt. Um das neuronale Netz robuster gegenüber bestimmten Transformationen zu machen, wurden die Bilddaten zufällig augmentiert (Flip, Translation, Rotation). Als Architektur des neuronalen Netzes wurde ein ResNet-50 [9] verwendet, das eine state-of-the-art Architektur für neuronale Netze zur Bildklassifikation ist. Dies besitzt zwar weniger Parameter als andere Architekturen, z. B. ResNet-101 [9] aus derselben Familie, ist jedoch für das vorliegende binäre Problem ausreichend komplex und hat zudem positive Effekte auf Speicherbedarf und Inferenzzeit. Als Loss-Funktion wurde die Kreuzentropie zwischen den tatsächlichen

Klassen des Bilddatensatzes und der Ausgabe des neuronalen Netzes berechnet. Zur Optimierung der Parameter des neuronalen Netzes während des Trainingsprozesses wurde der Adam-Optimierer [28] mit einer Lernrate von 0.001 genutzt. Die Eingabebilder hatten eine Auflösung von 224×224 Pixel und wurden in Batchgrößen von 64 Bildern bereitgestellt. Diese entstammten dem Trainingsdatensatz mit etwa 21K Bildern für die Optimierung der Zielfunktion und dem Validierungsdatensatz mit etwa 3K Bildern zur Bestimmung des Loss am Ende jeder Epoche. Der Trainingsprozess fand über 100 Epochen auf zwei NVIDIA Tesla V100 Tensor-Recheneinheiten statt.

4.4 Ergebnisse

Nach dem Training des neuronalen Netzes wurde dessen Güte auf dem Testdatensatz mit etwa 6K Bildern (20 % des Bilddatensatzes) evaluiert. Diese Bilddaten waren nicht Teil des Trainingsprozesses, sodass diese für das neuronale Netz neu waren. Die quantitativen Ergebnisse dieser Auswertung sind in Tab. 1 zusammengefasst. Insgesamt erreicht das neuronale Netz für dieses binäre Klassifikationsproblem eine Accuracy von 98 %, was auf eine sehr gute Güte hinweist. Fehlklassifikationen können entstehen, wenn sich die Bilder beider Klassen stark ähneln, z. B. bei weit entfernten Objekten, die aufgrund der geringeren Punktdichte in der Entfernung keine entsprechenden Merkmale mehr aufweisen.

Das gute Ergebnis zeigt sich auch in der Visualisierung der Personendetektion in einer 3D-Punktwolke, die in Abb. 4 dargestellt ist. In dieser sind zwei Personen zu sehen, die

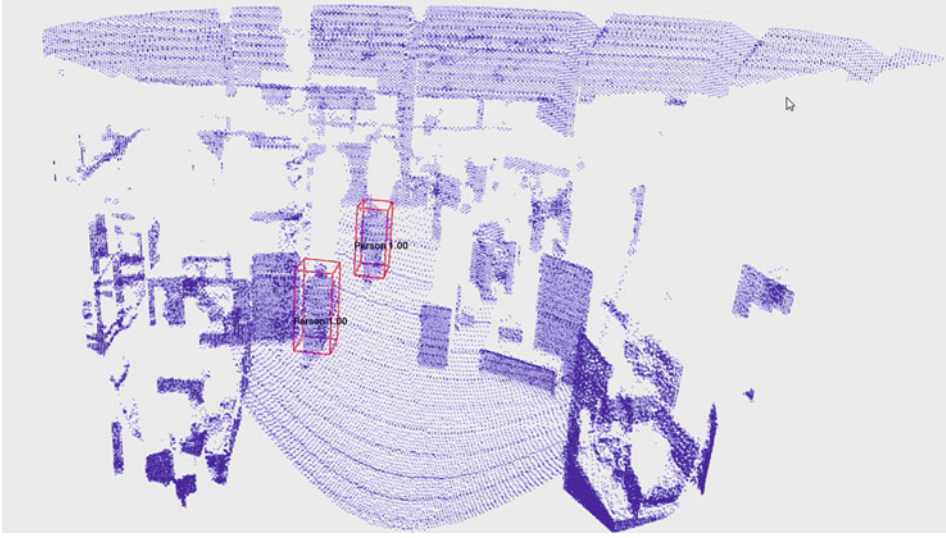

Abb. 4 Visualisierung der Personendetektion in einer 3D-Punktwolke

Tab. 1 Evaluationsmetriken des neuronalen Netzes auf dem Testdatensatz

	Precision	Recall	F1-Score	Anzahl
Sonstige	0,97	0,99	0,98	2594
Person	1,00	0,98	0,99	3469
Gewichtetes Mittel	0,98	0,98	0,98	6063
Accuracy			0,98	6063

detektiert und korrekterweise als Personen klassifiziert werden. Die restlichen Punkte der Umgebung werden richtigerweise nicht als Objekte bestimmt.

5 Fazit und Ausblick

Die Ergebnisse dieses Beitrags zeigen, dass es mit Hilfe des entwickelten Verfahrens möglich ist, Personen robust in 3D-Punktwolken zu detektieren und von anderen Objekten zu unterscheiden. Im Gegensatz zu etablierten Objektdetektionsverfahren basierend auf (hochaufgelösten) Farbbildern werden bei dem vorgestellten Ansatz aufgrund seines Messprinzips keine personenbezogenen Daten verarbeitet, sodass dieser Ansatz an sich alleine in Bezug auf den Datenschutz unkritisch zu sehen ist. Zudem liefert dieser Ansatz im Vergleich zu kamerabasierten Ansätzen inhärent Tiefeninformationen der Szene, die für eine 3D-Positionsbestimmung der Objekte direkt genutzt werden können. Weiterhin müssen Personen keine Gegenstände, wie z. B. drahtlose IoT-Devices, bei sich tragen, um in der Umgebung lokalisiert zu werden. Solch ein Ansatz ist gut dafür geeignet, im Rahmen einer Fabrikumgebung eingesetzt zu werden, um u. a. die Sicherheit von Arbeitern zu erhöhen oder das Fabriklayout zu optimieren.

Zukünftig soll das entwickelte Verfahren auf einer Embedded Hardware umgesetzt und als prototypisches System in der SmartFactoryOWL[4] evaluiert werden. Hierbei bietet es sich zur Optimierung der Verarbeitung an, vorverarbeitende Schritte des Verfahrens, wie z. B. die Hintergrundentfernung, direkt in den 3D-LiDAR-Sensor auszulagern. Zusätzlich sollen die Detektionen im 3D-Raum zeitlich verfolgt werden (*Tracking*), um Bewegungsmuster von Personen zu bestimmen und höherwertige Informationen abzuleiten. Zudem ist dieser Ansatz nicht nur auf einen Fabrikkontext beschränkt, sondern kann auch in anderen Domänen genutzt werden, um DSGVO-konform Personen zu detektieren. Beispielsweise wurde im Rahmen des Projekts „KI4PED“ der vorgeschlagene Ansatz zur Erfassung von Personen im Straßenverkehr mit dem Ziel einer Optimierung der Fußgängerüberquerungszeiten an Lichtsignalanlagen erprobt.

Danksagung Dieser Beitrag entstand im Rahmen des Projekts „KI4PED“ (FKZ: 19F1090A), das im Rahmen der Innovationsinitiative mFUND durch das Bundesministerium für Digitales und Verkehr (BMDV) gefördert wurde.

[4] https://smartfactory-owl.de/ (abgerufen am 15.09.2022)

Literatur

1. Bochkovskiy A, Wang CY, Liao HYM (2020) YOLOv4: Optimal speed and accuracy of object detection (arXiv preprint arXiv:2004.10934)
2. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: towards real-time object detection with region proposal networks. Adv Neural Inf Process Syst 28:91–99
3. Silva B, Pang Z, Akerberg J, Neander J, Hancke G (2014) Experimental study of UWB-based high precision localization for industrial applications. In: IEEE International Conference on Ultra-WideBand (ICUWB), S 280–285
4. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems (NIPS), S 1097–1105
5. LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324
6. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 248–255
7. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR), S 1–14
8. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 1–9
9. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 770–778
10. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 2261–2269
11. Szegedy C, Ioffe S, Vanhoucke V, Alemi AA (2017) Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI Conference on Artificial Intelligence. AAAI, Palo Alto, S 4278–4284
12. Xie S, Girshick R, Dollár P, Tu Z, He K (2017) Aggregated residual transformations for deep neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 5987–5995
13. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 580–587
14. Girshick R (2015) Fast R-CNN. In: IEEE International Conference on Computer Vision (ICCV), S 1440–1448
15. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 779–788
16. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. In: European Conference on Computer Vision (ECCV), S 21–37
17. Lin TY, Goyal P, Girshick R, He K, Dollár P (2017) Focal loss for dense object detection. In: IEEE International Conference on Computer Vision (ICCV), S 2999–3007
18. Qian R, Lai X, Li X (2022) 3D object detection for autonomous driving: a survey. Pattern Recognit 130:1–19

19. Zhou Y, Tuzel O (2018) VoxelNet: end-to-end learning for point cloud based 3D object detection. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 4490–4499
20. Lang AH, Vora S, Caesar H, Zhou L, Yang J, Beijbom O (2019) PointPillars: fast encoders for object detection from point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 12689–12697
21. Yin T, Zhou X, Krähenbühl P (2021) Center-based 3D object detection and tracking. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 11779–11788
22. Charles RQ, Su H, Kaichun M, Guibas LJ (2017) PointNet: deep learning on point sets for 3D classification and segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), S 77–85
23. Qi CR, Yi L, Su H, Guibas LJ (2017) PointNet++: deep hierarchical feature learning on point sets in a metric space (arXiv preprint arXiv:1706.02413)
24. Shi S, Wang X, Li H (2019) PointRCNN: 3D object proposal generation and detection from point cloud. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), S 770–779
25. Gonzalez A, Villalonga G, Xu J, Vázquez D, Amores J, Lopez AM (2015) Multiview random forest of local experts combining RGB and LIDAR data for pedestrian detection. In: Intelligent Vehicles Symposium (IV), S 356–361
26. Simon M, Amende K, Kraus A, Honer J, Sämann T, Kaulbersch H, Milz S, Gross HM (2019) Complexer-YOLO: real-time 3D object detection and tracking on semantic point clouds. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), S 1190–1199
27. Ester M, Kriegel HP, Sander J, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: International Conference on Knowledge Discovery and Data Mining (KDD). AAAI, Palo Alto, S 226–231
28. Kingma DP, Ba J (2015) Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR), S 1–15