

KI-basierte Detektion von Komponenten und Segmenten in Leitungssätzen

L. Zeh, L. Steinle, A. Lechler, A. Verl, ISW, Universität Stuttgart

Abstract

Wiring harnesses are branched deformable linear objects with complex structures that pose challenges for robotic automation, particularly in determining suitable gripping points for path planning. This article investigates AI-based detection of wiring harness components and segments to address these challenges. A custom wiring harness, modeled after a door wiring harness, is designed to generate datasets for training neural networks. Object detection is used to identify branches and connector types, segment detection involves solving the sub-tasks of segment segmentation, classification into two classes, and topological assignment. Several established frameworks are used for these tasks. Optimal training parameters are determined through grid search.

1 Einleitung

Biegeschlaffe Objekte wie Textilien, Kabel oder Seile [1] sind sowohl im Alltag als auch im Produktionsumfeld nahezu überall anzutreffen. Sie gehören zur Klasse der deformierbaren Objekte [2] und zeichnen sich durch ihre Flexibilität aus. Eine Unterkategorie sind verzweigte biegeschlaffe Objekte (VBOs), die aus mehreren Segmenten bestehen, beispielsweise Leitungssätze. Diese verbinden in der Fahrzeugtechnik flexibel verschiedene Komponenten und bestehen sowohl aus starren (z. B. Stecker, Clips) als auch biegeschlaffen Elementen (z.B. Kabel, Bänder, Schläuche). Durch die wachsende Zahl elektrischer Funktionen in modernen Fahrzeugen sind Leitungssätze in den letzten Jahren größer und komplexer geworden [3, 4].

In der Robotik wird beim Greifen und Manipulieren von Objekten meist von starren Objekten ausgegangen. Solange die Verformung der Objekte vernachlässigbar ist, ist dies eine gültige Vereinfachung. Für biegeschlaffe Objekte ist diese Vereinfachung nicht zulässig. Die automatisierte Handhabung biegeschlaffer Objekte durch Roboter ist ein noch nicht gelöstes Forschungsproblem [5, 6]. Die grundlegende Herausforderung ist, dass eine Krafteinwirkung von außen sowohl eine Bewegung als auch eine Formänderung bewirkt. Aufgrund der unendlichen Freiheitsgrade biegeschlaffer Objekte ist die Modellierung dieser Nichtlinearitäten bei der Verformung komplex. Um diese Herausforderungen zu bewältigen, müssen Roboter autonomer werden. Durch optische Sensoren wie Kameras und intelligente Entscheidungsmechanismen können sie auf Änderungen der geometrischen Konfiguration reagieren. Dies ermöglicht eine flexiblere Produktion und ist vor dem Hintergrund des Fachkräftemangels und steigender Produktionskosten besonders erstrebenswert [5].

Zur Erfassung der Verformung werden im Stand der Technik verschiedene Sensoren eingesetzt, darunter taktile, kraftmessende und visuelle Sensoren (Kameras). Während Taktile- und Kraftsensoren lokale Informationen liefern, liefern visuelle Sensoren globale Informationen. Ein zentrales Problem ist die Bestimmung geeigneter Greifpunkte, insbesondere wenn das Objekt ungeordnet vorliegt. Zudem fehlt es an zuverlässigen Sensorfeedback-Lösungen, da die Zuordnung von Kamerasensordaten zum unendlich großen Konfigurationsraum biegeschlaffer Objekte eine offene Forschungsfrage bleibt. Diese sogenannte Korrespondenzschätzung ordnet jedem Bildpixel einen Teil des Objekts zu [3, 5, 6, 7, 8, 9].

Für die automatisierte Handhabung von Leitungssätzen ist darüber hinaus eine genaue Kenntnis ihrer Topologie essenziell. Die Bahnplanung erfordert Informationen über die Position einzelner Segmente sowie über relevante Komponenten wie Stecker, Clips oder Verzweigungen. Eine mögliche Lösung ist die Bildsegmentierung zur Detektion der Segmente. Methoden des maschinellen Lernens, insbesondere neuronale Netze, können eine schnellere und robustere Verarbeitung der Sensordaten

ermöglichen. So lassen sich sowohl Komponenten als auch Einzelsegmente direkt aus Kameradaten erlernen und zuordnen.

2 Stand der Technik

In diesem Abschnitt wird zunächst auf den Stand der Technik zur KI-basierten Detektion von Komponenten und anschließend auf die Segmentierung von Einzelleitungen und Kabeln sowie Leitungssätzen Segmenten in Leitungssätzen eingegangen.

2.1 Detektion von Komponenten

Für die Detektion von Komponenten in Kabeln und Leitungssätzen existieren bereits einige Lösungsansätze. Zhou et al. [6] stellen eine Methode zur Erkennung von Steckern vor. Dabei werden mit einer Kamera insgesamt 550 Bilder eines Leitungssatzes aufgenommen. In jedem Bild wird der Stecker manuell annotiert. Anschließend wird ein neuronales Netz, basierend auf der Architektur MobileNet-SSD [10], mit 420 Bildern trainiert. Die verbleibenden 130 Bilder dienen als Testdatensatz. Das trainierte Netzwerk detektiert 98,5 % der Stecker korrekt und benötigt durchschnittlich 0,3 Sekunden für die Detektion der Stecker.

Nguyen et al. präsentieren in [4] und [11] zwei Ansätze zur Inspektion eines Türleitungssatzes. Bei dieser Inspektion sollen unter anderem vier verschiedene Clip-Typen sowie die im Leitungssatz verbauten Stecker detektiert und klassifiziert werden.

In [11] werden für die Komponentendetektion 2D-RGB Bilder verwendet während in [4] 3D-Punktwolken zum Einsatz kommen. In [11] werden 407 Bilder aufgenommen welche für das Training der untersuchten Modelle in 260 Bilder für das Training, 65 Bilder für die Validierung und 82 Bilder für den Test unterteilt werden. Nguyen et al. evaluieren verschiedene Deep-Learning-Architekturen und Backbone-Modelle. Das beste Ergebnis erzielt ein Feature Pyramid Network [12] mit Se-ResNeXt50 als Backbone, das einen Intersection-over-Union (IoU)-Wert von 86,2 % erreicht. Die IoU sagt dabei aus, wie gut die vorhergesagte Bounding Box um das detektierte Objekt mit der in der Ground Truth annotierten Bounding Box übereinstimmt.

In [4] werden neben realen 3D-Punktwolken auch synthetische Punktwolken verwendet. Es werden 150 Bilder mit einer Zivid2 3D-Kamera aufgenommen sowie 325 synthetischen Daten aus einem 3D-CAD-Modell des Leitungssatzes generiert. Aus diesen insgesamt 475 Punktwolken werden sieben Datensätze mit unterschiedlicher Größe und unterschiedlicher Zusammensetzung generiert. Für die Detektion der Komponenten wird PointNet++ [13], zur Segmentierung der Punktwolken verwendet. In der Arbeit werden Experimente mit ausschließlich realen, ausschließlich synthetischen und gemischten Datensätzen durchgeführt. Die besten Ergebnisse werden mit den gemischten Datensätzen erzielt, was auf die darin enthaltene größte Datengrundlage zurückzuführen ist. Die Stecker werden mit einer Genauigkeit von 94,1 % korrekt detektiert, die vier unterschiedlichen Clip-Typen erreichen eine Detektionsgenauigkeit zwischen 79,9 % und 99,6 %.

Die in diesem Abschnitt vorgestellten Arbeiten zeigen zwar, dass es grundsätzlich möglich ist, Stecker an Kabeln beziehungsweise in Leitungssätzen zu detektieren, eine genaue Klassifizierung der einzelnen Steckertypen findet jedoch nicht statt, sofern unterschiedliche Stecker verbaut sind (vgl. [4, 11]).

2.2 Segmentierung von Leitungssatzsegmenten

Auch für die Segmentierung Einzelleitungen oder Kabeln sowie ganze Abschnitte in Leitungssätzen existieren bereits einige Lösungsansätze.

Caporali et al. [14] stellen mit FASTDLO eine effiziente Methode zur Segmentierung von deformierbaren linearen Objekten (z. B. Kabeln) vor. FASTDLO nutzt eine Kombination aus Hintergrundsegmentierung, Skelettgenerierung und neuronalen Netzwerken, um effizient und präzise Einzelleitungen zu segmentieren. Hierfür wird zunächst eine Hintergrundsegmentierung durchgeführt, die eine Binärmaske des Bildes erzeugt. Diese Maske dient als Grundlage für die weitere Verarbeitung. Anschließend wird ein Skelett des Kabels generiert und in die Binärmaske integriert. Die Pixel des Skeletts werden anhand der Anzahl ihrer Nachbarn in drei Klassen unterteilt: Endpunkte, Abschnitte und Schnittpunkte. Diese Klassifizierung ermöglicht es, die Struktur der Einzelleitungen präzise zu erfassen. Um die Verbindungen zwischen den Endpunkten zu bestimmen, kommt ein neuronales Netzwerk zum Einsatz. Dieses Netzwerk berechnet die Wahrscheinlichkeit, dass zwei Endpunkte miteinander verbunden sind. Basierend auf diesen Vorhersagen werden weitere Algorithmen angewendet, um den genauen Verlauf der einzelnen Drahtleitungen zu bestimmen. Dieser Ansatz ist besonders effektiv bei der Verarbeitung von Überschneidungen und komplexen Kabelanordnungen. Ein wesentlicher Vorteil von FASTDLO ist seine Segmentierungszeit von ca. 45 ms. In der Arbeit werden allerdings nur Einzeladern und keine verzweigten Strukturen wie Leitungssätze betrachtet.

Neben der Detektion von Komponenten im Leitungssatz befassen sich Nguyen et al. in [4] und [11] ebenfalls mit der Detektion und Segmentierung von Leitungssatzbereichen. Hierbei werden drei unterschiedliche Bereiche im Leitungssatz unterschieden. Diese sind vollständig umwickelte Leitungsbündel, spiralumwickelte Leitungsbündel und nicht umwickelte Leitungsbündel. Diese beiden Arbeiten zeigen, dass sich sowohl 2D- als auch 3D-Daten für die Detektion von Bereichen im Leitungssatz eignen. In diesen beiden Arbeiten werden zwar Teile des Leitungssatzes segmentiert, bei den segmentierten Abschnitten handelt es sich allerdings nicht um die einzelnen Segmente des Leitungssatzes, sondern um Bereiche des Leitungssatzes, welche sich durch unterschiedliche Ummantelungsarten unterscheiden.

Im Folgenden wird auf Basis der vorgestellten Arbeiten ein Konzept zur Detektion von Leitungssatzkomponenten sowie zur Segmentierung der Leitungssatzsegmente vorgestellt.

3 Konzept

Der in dieser Arbeit vorgestellte Lösungsansatz für die KI-basierte Detektion von Komponenten und Segmenten in Leitungssätzen lässt sich in vier Arbeitspakete gliedern. Zunächst muss ein geeigneter Leitungssatz ausgewählt werden. Im Anschluss erfolgt der Aufbau des Learning-Frameworks zur Datenaufnahme, Annotierung der Bilddaten und dem Training der Modelle. Im nächsten Schritt erfolgt die Detektion von Leitungssatzsegmenten und abschließend werden die Leitungssatzsegmente von den trainierten Modellen segmentiert. In der nachfolgenden **Abbildung 1** ist der Lösungsansatz dargestellt.

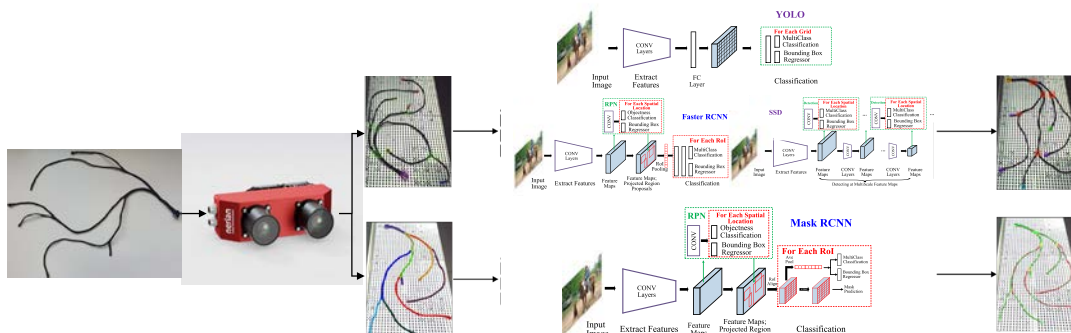


Abbildung 1: Konzept für die KI-basierte Detektion von Komponenten und Segmenten im Leitungssatz

3.1 Auswahl eines geeigneten Leitungssatz

Für die Auswahl eines geeigneten Leitungssatzes wird zunächst eine Recherche zu verschiedenen Modellen von Tür- und Heckklappenleitungssätzen durchgeführt. Diese beiden Leitungssatztypen eignen sich aufgrund ihrer Größe und Komplexität gut als Beispiel. Basierend auf dieser Recherche werden verschiedene Merkmale definiert, die ein Leitungssatz aufweisen kann. Diese sind unterschiedliche Steckertypen und Cliptypen. Leitungssätze bestehen aus einzelnen Segmenten. Diese Segmente haben unterschiedliche Querschnitte und Längen. Die einzelnen Segmente sind durch Verzweigestellen miteinander verbunden. Die einzelnen Segmente können in zwei Klassen unterteilt werden. Segmente zwischen zwei Verzweigestellen sowie Segmente zwischen Verzweigestelle und Stecker. Innerhalb des Leitungssatzes können verschiedene Arten von Ummantelungen verwendet werden. Es können Tüllen zum Einsatz kommen und verschiedene Bereiche können unterschiedliche Farben haben [15].

Da die in der Automobilindustrie verwendeten Leitungssätze groß und komplex sind, am vorhandenen Prüfstand aber nur begrenzt Platz zur Verfügung steht, wird für diesen Beitrag ein beispielhafter Leitungssatz hergestellt, dies hat jedoch keine Auswirkung auf die Praxistauglichkeit des vorgestellten Ansatzes, da sich die Ergebnisse auch auf andere Leitungssätze übertragen lassen. Der Leitungssatz orientiert sich an den Funktionalitäten eines Türleitungssatzes und ist in der nachfolgenden **Abbildung 2** dargestellt.

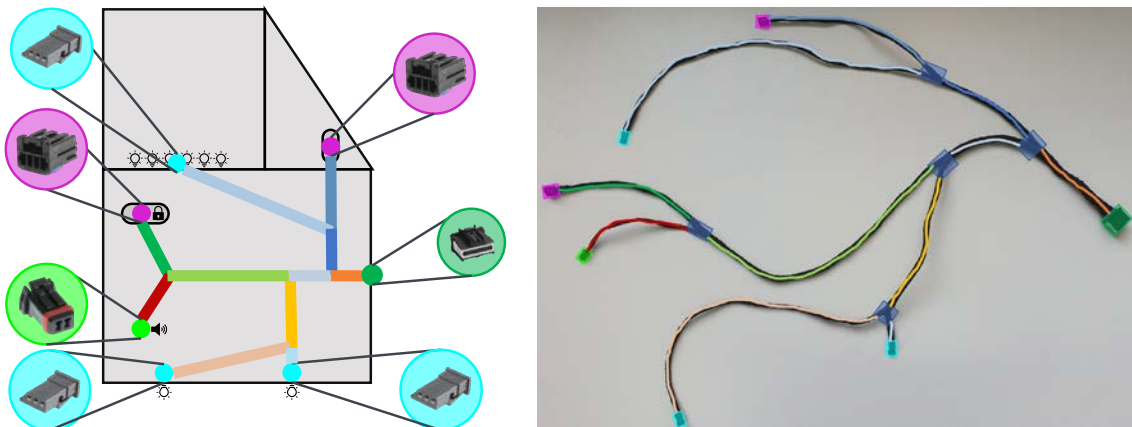


Abbildung 2: *Verwendeter Leitungssatz – links schematischer Aufbau des an einem Türleitungssatz orientierter Leitungssatz, rechts realer Leitungssatz mit farblich hervorgehobenen Komponenten und Segmenten*

Der Leitungssatz besteht aus elf Segmenten, fünf Verzweigestellen und sieben Steckern, wobei insgesamt vier verschiedene Steckertypen verbaut sind. Die einzelnen Segmente des Leitungssatzes haben unterschiedliche Längen und Querschnitte.

3.2 Datengenerierung

Die für das Training der neuronalen Netze benötigten Daten werden am in **Abbildung 3** abgebildeten Versuchstand aufgenommen. Hierfür wird der Leitungssatz in unterschiedlichen Konfigurationen auf der Lochrasterplatte positioniert. Die Aufnahme der Bilder erfolgt mit der am Versuchstand installierten Nerian Scarlet 3D Kamera.

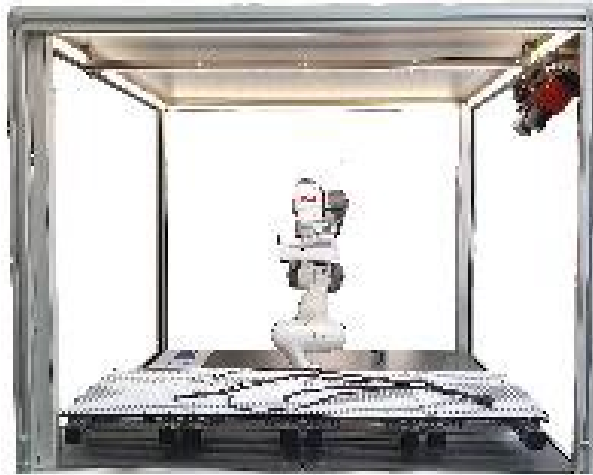


Abbildung 3: Versuchsstand zur Datengenerierung

Für das Training der verschiedenen neuronalen Netze werden zweimal 60 RGB-Bilder aufgenommen, aus denen insgesamt vier Datensätze¹ für das Training erstellt werden. Diese unterteilen sich in einen Datensatz für die Klassifizierung der Komponenten sowie drei Datensätze für die Segmentierung und sind in der nachfolgenden **Tabelle 1** aufgelistet. Jeder Datensatz umfasst 60 Bilder. Für die Klassifizierung werden alle 60 Bilder annotiert. In diesem Datensatz sind die vier verschiedenen Steckertypen (S2, S3, S4 und S20) sowie die Verzweigstellen (VS) mit Bounding Boxen annotiert. In den Segmentierungsdatensätzen sind aufgrund des großen Zeitaufwands bei der Annotierung der einzelnen Segmente durch Polygonmasken nur 20 Bilder annotiert.

Tabelle 1: Übersicht Datensätze

Bezeichnung	Anzahl Bilder	Annotierte Bilder
LS Klassifizierung	60	60
LS Segmentierung 1	60	20
LS Segmentierung 2	60	20
LS Segmentierung 3	60	20

In den nachfolgenden beiden Kapiteln wird auf das Training sowie die Ergebnisse der Komponentendetektion und Segmentierung eingegangen.

4 Detektion von Leitungssatzkomponenten

Ziel ist die Detektion und Klassifizierung von Verzweigstellen sowie der verschiedenen Steckertypen. Für die Detektion und Klassifizierung der einzelnen Leitungssatzkomponenten eignen sich Objekterkennungsnetzwerke. Im Rahmen dieses Artikels werden fünf verschiedene Netzwerkarchitekturen untersucht. Diese sind das Netzwerk YOLO v3 [16] mit dem Backbone DarkNet-53, das Netzwerk YOLOX s [17], das Netzwerk SSD-Lite [10] mit dem Backbone

¹ Die erstellten Datensätze mit Annotierung können unter <https://doi.org/10.18419/DARUS-4778> abgerufen werden

MobileNetV2 sowie das Netzwerk Faster R-CNN [18]. Beim Netzwerk Faster R-CNN werden die beiden Backbones R-50-FPN und X-101-32x4d-FPN verwendet.

Für das Training der unterschiedlichen Modelle kommt Grid Search zum Einsatz. Hierfür werden die Trainingsparameter Batchsize, Lernrate, sowie die Anzahl der trainierten Epochen variiert.

Die fünf untersuchten Netzwerke unterscheiden sich erheblich in ihrer Accuracy (das Netzwerk detektiert alle annotierten Objekte) und ihrer Auswertezeit. Die schnellste Auswertezeit, das heißt die schnellsten Detektionsergebnisse, erreicht das Netzwerk YOLOX mit 13 ms, Faster R-CNN mit dem Backbone X-101-32x4d-FPN benötigt für die Auswertung mit 94 ms am längsten. Trotz der längeren Auswertezeit erzielt Faster R-CNN mit dem Backbone X-101-32x4d-FPN die höchste Genauigkeit von 97,4 %. YOLOX folgt mit einer ebenfalls sehr guten Accuracy von 96,0 %. Die übrigen Netzwerke liegen in ihrer Genauigkeit deutlich darunter, mit Werten zwischen 60 % und 80 %. In der nachfolgenden **Abbildung 4** sind die Detektionsergebnisse von YOLOX und Faster R-CNN dargestellt anhand ihrer Confusion Matrizen dargestellt. Diese bilden die Detektionsergebnisse der beiden Netzwerke in Bezug auf korrekt sowie falsch oder nicht detektierte Komponenten ab.

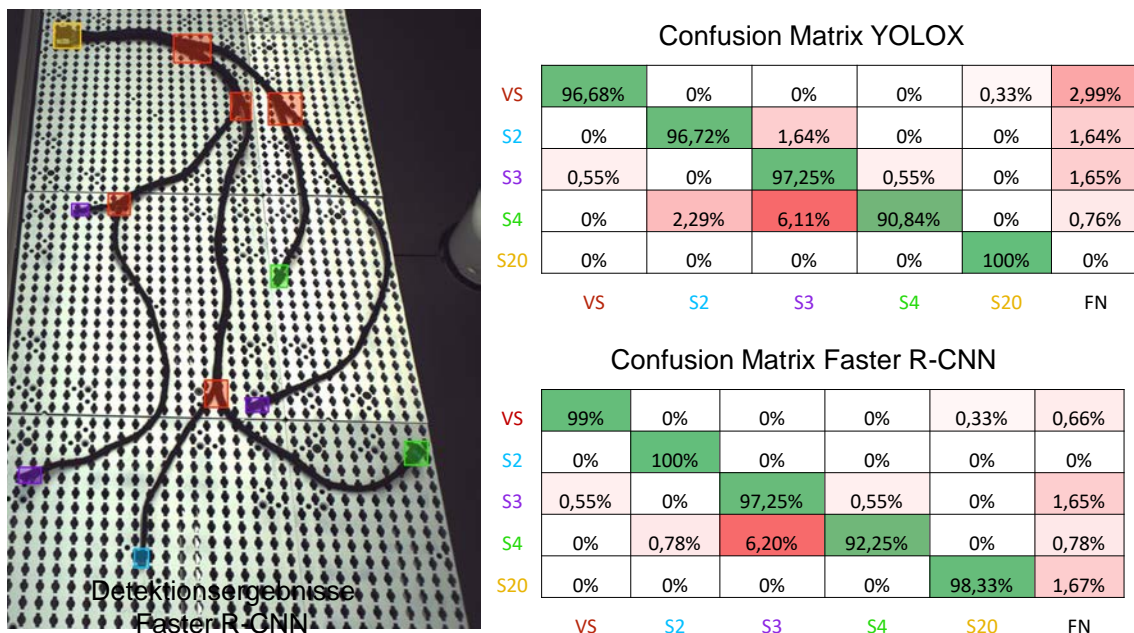


Abbildung 4: Vergleich der Detektionsergebnisse zwischen YOLOX und Faster RCNN

5 Detektion von Leitungssatzsegmenten

Bei der Detektion der einzelnen Leitungssatzsegmente sollen die elf Segmente des Leitungssatzes mit Segmentierungsmasken detektiert werden. Für die Detektion und Segmentierung der Leitungssatzsegmente wird das Netzwerk Mask R-CNN [19] mit dem Backbone R-101-FPN verwendet.

Für die Segmentierung werden drei unterschiedliche Ansätze mit steigender Komplexität untersucht. Diese sind in der **Abbildung 5** visualisiert. Zunächst die reine Segmentierung der elf Leitungssatzsegmente und Detektion als Leitungssatzsegment. Für diesen Ansatz sind im Datensatz LS Segmentierung 1 alle Segmente gleich annotiert. Der zweite Ansatz ist die Segmentierung der Leitungssatzsegmente und Detektion der beiden Segmenttypen. Im Datensatz LS Segmentierung 2 werden daher die beiden Segmenttypen unterschiedlich annotiert. Der dritte Ansatz ist die Segmentierung der Leitungssatzsegmente sowie die eindeutige Zuweisung der Segmente zur Leitungssatztopologie. Hierfür sind im Datensatz LS Segmentierung 3 alle Segmente als einzelne

Klasse annotiert. Dieser dritte Ansatz ist der für die Praxis relevanteste Ansatz, da aus diesem die Topologie des Leitungssatzes direkt abgeleitet werden kann.

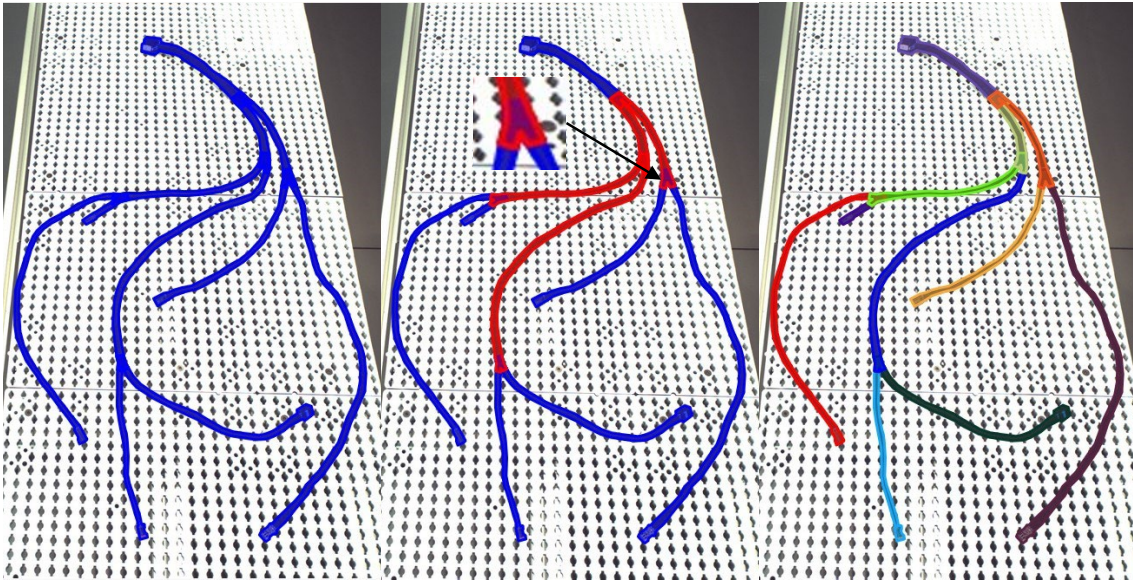


Abbildung 5: Visualisierung der unterschiedlichen Ansätze zur Detektion der Leitungssatz-segmente: Links - Detektion der einzelnen Segmente (Datensatz LS Segmentierung 1), mitte Detektion der einzelnen Segmente und Unterscheidung der beiden Segmenttypen (Datensatz LS Segmentierung 2), rechts - Detektion der Segmente und eindeutige Zuweisung der einzelnen Segmente (Datensatz LS Segmentierung 3)

Werden keine Segmente unterschieden, so erreicht das Netzwerk eine mittlere Intersection over Union (IoU) von 45,7 %. Der Ansatz alle Segmente mit demselben Label zu annotieren ist also in der Lage Teile des Leitungssatzes zu segmentieren. Der relativ kleine Wert der IoU ist vermutlich auf die großen Unterschiede zwischen den einzelnen Segmenten zurückzuführen. So ist bei den Segmenten, welche zwischen zwei Verzweigestellen liegen an beiden Seiten des Segments ein charakteristisches Y im annotierten Polygon eingeschlossen (vgl. Zoom in **Abbildung 5**), während dieses Y bei Segmenten zwischen Verzweigestelle und Stecker nur einmal vorhanden ist.

Die besten Ergebnisse werden bei der Unterscheidung der beiden Segmenttypen erreicht. Hier erreicht das Netzwerk bei Segmenten zwischen zwei Verzweigestellen eine mittlere IoU von 67,7 %. Segmente zwischen Verzweigestelle und Stecker werden mit einer mittleren IoU von 61,9 % segmentiert. In der nachfolgenden **Abbildung 6** ist ein Segmentierungsergebnis dieses Ansatzes dargestellt.

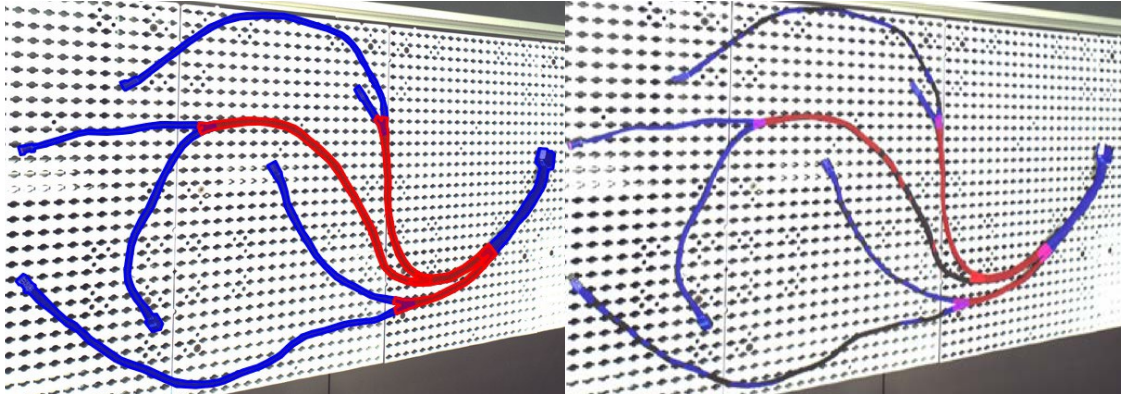


Abbildung 6: Segmentierungsergebnis Unterscheidung der beiden Segmenttypen, links Ground Truth, rechts Prediction

Wie in der Abbildung erkennbar ist, sinkt die Qualität der Segmentierung mit zunehmender Länge der Segmente. Dies ist auf die größere Varianz bei den langen Segmenten zurückzuführen.

Die schlechtesten Ergebnisse werden bei der Unterscheidung aller Segmente erzielt. Während kurze Segmente mit einer mittleren IoU zwischen 80,1 % und 52,3 % segmentiert werden, können die drei längsten Segmente im Leitungssatz, im Gegensatz zu den beiden anderen Ansätzen, bei diesem Ansatz überhaupt nicht segmentiert werden. Dies ist auf die, in diesem Ansatz, noch geringere Datengrundlage zurückzuführen.

6 Zusammenfassung und Ausblick

In diesem Artikel wird verglichen, inwieweit sich etablierte Modelle für die Detektion von Komponenten und Segmenten in Leitungssätzen eignen. Im Bereich der Komponentendetektion können Verzweigestellen erkannt sowie verschiedene Steckertypen detektiert und unterschieden werden. Bei der Segmentierung liefert der Ansatz, bei dem zwei Segmenttypen unterschieden werden, die besten Ergebnisse. Allerdings zeigt sich, dass lange Segmente schlechter segmentiert werden als kurze.

Ein limitierender Faktor ist die Verwendung eines kleinen Datensatzes, insbesondere für die Segmentierung von Leitungssatzsegmenten. Um die Trainingsgrundlage zu erweitern und potenziell bessere Ergebnisse zu erzielen, könnte die analytische Annotierung untersucht werden. Diese Methode ermöglicht es, den Datensatz mit überschaubarem Zeitaufwand (ca. 5 Minuten pro Bild) zu vergrößern. Zusätzlich könnte eine Kombination aus KI- und analytischen Ansätzen (z.B. Kantendetektion) die Ergebnisse bei der Leitungssatzsegmentierung weiter verbessern.

Literatur

- [1] T. Matsuno, D. Tamaki, F. Arai, und T. Fukuda, „Manipulation of deformable linear objects using knot invariants to classify the object condition based on image sensor information“, *IEEEASME Trans. Mechatron.*, Bd. 11, Nr. 4, S. 401–408, Aug. 2006, doi: 10.1109/tmech.2006.878557.
- [2] A. Keipour, M. Bandari, und S. Schaal, „Deformable One-Dimensional Object Detection for Routing and Manipulation“, *IEEE Robot. Autom. Lett.*, Bd. 7, Nr. 2, S. 4329–4336, Apr. 2022, doi: 10.1109/lra.2022.3146920.
- [3] M. Zürn, M. Wnuk, A. Lechler, und A. Verl, „Topology Matching of Branched Deformable Linear Objects“, 2023 *IEEE Int. Conf. Robot. Autom. ICRA*, 2023.
- [4] H. G. Nguyen, R. Habiboglu, und J. Franke, „Enabling deep learning using synthetic data: A case study for the automotive wiring harness manufacturing“, *Procedia CIRP*, Bd. 107, S. 1263–1268, 2022, doi: 10.1016/j.procir.2022.05.142.
- [5] J. Zhu u. a., „Challenges and Outlook in Robotic Manipulation of Deformable Objects“, *IEEE Robot. Autom. Mag.*, Bd. 29, Nr. 3, S. 67–77, Sep. 2022, doi: 10.1109/MRA.2022.3147415.
- [6] H. Zhou, S. Li, Q. Lu, und J. Qian, „A Practical Solution to Deformable Linear Object Manipulation: A Case Study on Cable Harness Connection“, in *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*, Shenzhen, China: IEEE, Dez. 2020, S. 329–333. doi: 10.1109/ICARM49381.2020.9195380.
- [7] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, und Y. Mezouar, „Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey“, *Int. J. Robot. Res.*, Bd. 37, Nr. 7, S. 688–716, Juni 2018, doi: 10.1177/0278364918779698.
- [8] D. Henrich, T. Ogasawara, und H. Worn, „Manipulating deformable linear objects - contact states and point contacts“, in *Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning (ISATP'99)* (Cat. No.99TH8470), IEEE, 1999. doi: 10.1109/isatp.1999.782959.
- [9] V. E. Arriola-Rios, P. Guler, F. Ficuciello, D. Kragic, B. Siciliano, und J. L. Wyatt, „Modeling of Deformable Objects for Robotic Manipulation: A Tutorial and Review“, *Front. Robot. AI*, Bd. 7, S. 82, Sep. 2020, doi: 10.3389/frobt.2020.00082.
- [10] W. Liu u. a., „SSD: Single Shot MultiBox Detector“, in *Computer Vision ECCV 2016*, Springer International Publishing, 2016, S. 21–37. doi: 10.1007/978-3-319-46448-0_2.
- [11] H. G. Nguyen und J. Franke, „Deep learning-based optical inspection of rigid and deformable linear objects in wiring harnesses“, *Procedia CIRP*, Bd. 104, S. 1765–1770, 2021, doi: 10.1016/j.procir.2021.11.297.
- [12] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, und S. Belongie, „Feature Pyramid Networks for Object Detection“, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juli 2017. doi: 10.1109/cvpr.2017.106.
- [13] C. R. Qi, L. Yi, H. Su, und L. J. Guibas, „PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space“, in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, und R. Garnett, Hrsg., Curran Associates, Inc., 2017. [Online]. Verfügbar unter: https://proceedings.neurips.cc/paper_files/paper/2017/file/d8bf84be3800d12f74d8b05e9b89836f-Paper.pdf.

- [14] A. Caporali, K. Galassi, R. Zanella, und G. Palli, „FASTDLO: Fast Deformable Linear Objects Instance Segmentation“, IEEE Robot. Autom. Lett., Bd. 7, Nr. 4, S. 9075–9082, Okt. 2022, doi: 10.1109/ira.2022.3189791.
- [15] J. Trommnau, J. Kühnle, J. Siegert, R. Inderka, und T. Bauernhansl, „Overview of the State of the Art in the Production Process of Automotive Wire Harnesses, Current Research and Future Trends“, Procedia CIRP, Bd. 81, S. 387–392, 2019, doi: 10.1016/j.procir.2019.03.067.
- [16] J. Redmon und A. Farhadi, „YOLO9000: Better, Faster, Stronger“, in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Juli 2017. doi: 10.1109/cvpr.2017.690.
- [17] Z. Ge, S. Liu, F. Wang, Z. Li, und J. Sun, „YOLOX: Exceeding YOLO Series in 2021“, ArXiv Prepr. ArXiv210708430, 2021, doi: 10.48550/ARXIV.2107.08430.
- [18] S. Ren, K. He, R. Girshick, und J. Sun, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks“, IEEE Trans. Pattern Anal. Mach. Intell., Bd. 39, Nr. 6, S. 1137–1149, Juni 2017, doi: 10.1109/tpami.2016.2577031.
- [19] K. He, G. Gkioxari, P. Dollar, und R. Girshick, „Mask R-CNN“, in 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Okt. 2017. doi: 10.1109/iccv.2017.322.

