Bin Picking für das Greifen unbekannter Teile mithilfe Machine-Learning Verfahren auf Basis neuraler Netze

STUDIENARBEIT T3100

des Studienganges Elektrotechnik (Automation)

an der Dualen Hochschule Baden-Württemberg Campus Friedrichshafen

von

Alexander Deutsch, 9301310

Tarek Zwick, 1857672

16.01.2023

Bearbeitungszeitraum	14 Wochen
Kurs	TEA20
Gutachter der Dualen Hochschule	Prof. Dr. Thomas Dietmüller

Erklärung

Gemäß Ziffer 1.1.13 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 25.07.2018:

Wir versichern hiermit, dass wir unsere Studienarbeit T3100 mit dem Thema:

Produktionsnetzwerk: Bin Picking für das Greifen unbekannter Teile mithilfe Machine-Learning Verfahren auf Basis neuraler Netze

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Friedrichshafen, den 16.01.2023

Alexander Deutsch	Tarek Zwick
Unterschrift	Unterschrift

Inhaltsverzeichnis

1. Einleitung 1
1.1. Vorstellung des Produktionsnetzwerkes 1
1.2. Bin Picking 2
1.3. Die Bin-Picking-Station 3
2. Aufgabenstellung 4
3. Grundlagen 5
3.1. Aufbau der Bin-Picking-Station 5
3.2. Cobot 6
3.3. Mikado Control 7
3.3.1. Aufbau eines Programms 8
3.3.2. Bin-Picking in Mikado 8
3.4. Machine Learning 11
3.4.1. Supervised learning 12
3.4.2. Unsupervised learning 12
3.4.3. Semi-supervised learning 13
3.4.4. Reinforcement learning 13
3.4.5. Deep learning 14
3.4.6. Automatisiertes Machine Learning 15
3.4.7. Under- und Overfitting 16
4. Projektplanung 17
5. Umbau 18
5.1. Anforderungen 18
5.2. Planung 18
5.3. Umsetzung 24

6. Prozessintegration 29
6.1. Anforderungen 29
6.2. Geplante Vorgehensweise 30
6.3. Probleme 31
6.3.1. Antransport des Werkstücks 31
6.3.2. Keine Griffe möglich 33
6.3.3. Position des Werkstücks 33
7. Machine Learning 34
7.1. Anforderungen 34
7.2. Mögliche Vorgehensweise 34
8. Fazit und Ausblick 36
9. Literaturverzeichnis 38

Abkürzungsverzeichnis

2D ..... zweidimensional
3D ..... dreidimensional
DHBW ..... Duale Hochschule Baden-Württemberg
HMI ..... Human Machine Interface
IWT ..... Institut für Weiterbildung, Wissens- und Technologietransfer
PTZ ..... Produktionstechnisches Zentrum
ROI ..... Region of Interest
UR3e ..... Universal Robots 3e

Abbildungsverzeichnis

Abbildung 1: Bin Picking Symbolbild 2
Abbildung 2: Bin-Picking-Station 3
Abbildung 3: Strebenprofile 5
Abbildung 4: UR3e 6
Abbildung 5: Fernsteuerung über das HMI des UR3e 7
Abbildung 6: Einlagiges neuronales Netzwerk 14
Abbildung 7: Mehrlagiges neuronales Netzwerk 14
Abbildung 8: Gantt-Diagramm zur Projektplanung 17
Abbildung 9: Bin-Picking-Station von vorne 19
Abbildung 10: Anpassung der Kamerahalterung 19
Abbildung 11: Aufnahme für die Bin-Picking-Station am Förderband 19
Abbildung 12: Bin-Picking-Station von hinten 20
Abbildung 13: 90mm Abstandshalter von hinten 20
Abbildung 14: 90mm Abstandshalter von rechts 20
Abbildung 15: Spannverschluss 21
Abbildung 16: Verwendung Spannverschluss am Abstandshalter unten rechts 21
Abbildung 17: Bin-Picking-Station festgespannt in der Aufnahme 22
Abbildung 18: Bin-Picking-Station und ihre Aufnahme im „MTpro Online Designer“ 23
Abbildung 19: Befestigung der Kamera an dem Profil A 24
Abbildung 20: Nötige Bearbeitung der Arbeitsplatte 24
Abbildung 21: Aufnahme für die Bin-Picking- Station am Förderband 25
Abbildung 22: Vakuum-Endeffektor 27
Abbildung 23: Venturi Effekt 27
Abbildung 24: Elektromagnetventil zur Ansteuerung des Endeffektors 27
Abbildung 25: Bin-Picking-Station nach vollendetem Umbau 28
Abbildung 26: Programmfluss 29
Abbildung 27: 3D-Kamera Kalibrierplatte 36
Abbildung 28: Digitaler Zwilling 36
Abbildung 29: Visualisierung der Objekterkennung 37
Abbildung 30: Ausschnitt in der Arbeitsplatte 37

1. Einleitung

1.1. Vorstellung des Produktionsnetzwerkes

Das Produktionsnetzwerk Fallenbrunnen entsteht in Zusammenarbeit zwischen der DHBW Ravensburg am Campus Friedrichshafen und dem Institut für Weiterbildung, Wissens- und Technologietransfer (IWT), das ebenfalls am Standort Fallenbrunnen vertreten ist.

Das produktionstechnische Zentrum (PTZ) an der DHBW stellt ein Labor für Industrielle Messtechnik und für Automation dar. Hier finden diverse Laboraktivitäten zu Datenaufnahme und -verarbeitung sowie kollaborativer Robotik statt. Die Lernfabrik des IWT Bodensee verfügt über eine vielfältige Ausrüstung in Bezug auf Roboter, Bandsysteme und 3D-Drucker. Gemeinsam wird ein Anwendungsfall geschaffen, in dem spannende Themen der Industrie 4.0-Technologie gelehrt, geforscht und umgesetzt werden können.

Ziel des Produktionsnetzwerkes ist die Abbildung einer realen Wertschöpfungskette mit vernetzten, adaptiven und selbstlernenden Systemen. Im Produktionsnetzwerk soll die Produktion eines Produktes in hoher Variantenvielfalt stattfinden, anschließend soll zur Abbildung einer Kreislaufwirtschaft ebenfalls eine vollständige Demontage ermöglicht werden.

Das Projektteam umfasst 23 Studierende, vier Professoren, vier Laboringenieure, drei wissenschaftliche Mitarbeiter und drei Doktoranden, die das Produktionsnetzwerk in zwölf Teilprojekten entwickeln.

Im PTZ werden zwei kollaborative Roboter in Kombination mit einem manuellen Handarbeitsplatz über ein Transportsystem verbunden. Nach finaler Montage des Vorproduktes und einer Endprüfung wird das Vorprodukt über ein Transportsystem über das Campusgelände zur Lernfabrik transportiert, wo die Endmontage über manuelle und automatisierte Arbeitsschritte erfolgt. Nach bestandener Endprüfung wird das Produkt eingelagert, im selben Kreislauf kann dann die Demontage erfolgen.

Das Großprojekt startet mit dem Beginn der Studienarbeiten im Oktober 2022 und soll im Juli 2023 vollständig produktionsfähig sein. [1]

1.2. Bin Picking

Der englische Begriff „Bin Picking“ kann zu dem „Griff in die Kiste“ übersetzt werden. In den Anwendungsfällen des Bin Pickings liegt ein Behälter mit verschiedenen ungeordneten Einzelteilen vor.

Über eine installierte Kamera erkennt ein Roboter bestimmte, vordefinierte Teile in einem Behälter, aus dem die völlig zufällig angeordneten Werkstücke für weitere Bearbeitungsschritte einzeln herausgegriffen werden.

Ein Bin Picking System wird durch die folgenden Arbeitsschritte charakterisiert:

- Objekterkennung
- Objektlokalisation
- Roboterführung
- (Objektinspektion)

Die Objekterkennung beschreibt die Erkennung der Einzelteile anhand bauteilspezifischer Merkmale. Hierüber lässt sich beispielsweise aus einer Auswahl unterschiedlicher Einzelteile ein entsprechendes finden.

Die Objektlokalisation beschreibt die dreidimensionale Bestimmung der Position und Lage des Einzelteils über mathematische Verfahren. Zusätzlich werden die Koordinaten an den Roboter beziehungsweise dessen Steuerung übertragen.

Unter der Roboterführung wird die Routenplanung des Roboters vom Startpunkt zum Zielpunkt verstanden, die unter anderem die Kollisionserkennung beinhaltet.

Über die Kamera oder externe optische Messmittel ist die Implementierung einer Objektinspektion zur Vermessung der zu greifenden Einzelteile möglich. Diese können dann mit Sollmerkmalen und -toleranzen verglichen werden und entsprechend den Kriterien auch aussortiert werden. [2]



Abbildung 1: Bin Picking Symbolbild [3]

1.3. Die Bin-Picking-Station

Die Bin-Picking-Station ist zentraler Gegenstand dieser Studienarbeit. Es handelt sich hierbei um eine Station, die bisher in Laborversuchen und auf Ausstellungen genutzt wurde. Das Hauptelement dieser Station ist der kollaborative Roboter UR3e. Er führt den Prozess des Bin-Pickings über einen Endeffektor mit zwei Greifbacken aus. Hierfür ist an der Station eine Kamera mit mehreren Linsen angebracht, um so über eine Tiefenwahrnehmung zu verfügen („3D-Kamera“). Das Bin-Picking wird durch die Software Mikado Control gesteuert. Zusätzlich hierzu ist ein optisches 2D-Mikrometer auf der Arbeitsplatte befestigt, das die gegriffenen Einzelteile mit einer Präzision von wenigen Mikrometern vermessen kann.

In bisherigen Laborversuchen griff der Roboter Messingdrehteile in verschiedenen Konfigurationen aus einem Gefäß, diese waren entweder aufrechtstehend palettiert oder chaotisch in einem Behälter verstreut und liegen dabei teilweise auch aufeinander. Diese Drehteile wurden in das optische Mikrometer geführt, wo diese auf diverse Maße und Durchmesser geprüft wurden. Lagen diese im Toleranzbereich wurden sie in einen grünen Behälter abgelegt, andernfalls in einen roten Behälter.

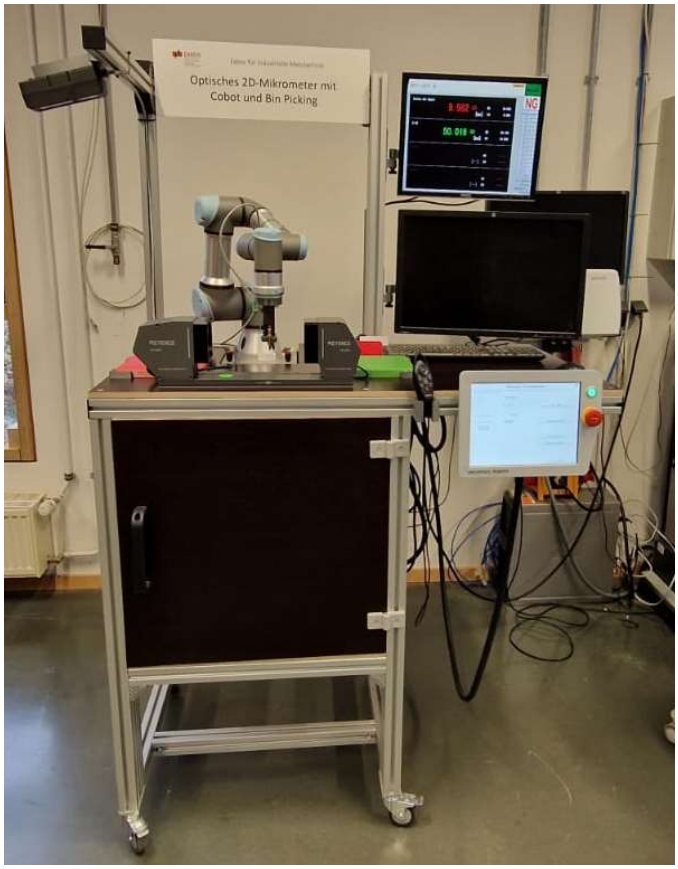


Abbildung 2: Bin-Picking-Station

2. Aufgabenstellung

Bisher wurde die Bin-Picking-Station nur für Laborversuche genutzt. Dies soll nun geändert werden. Die Bin-Picking-Station soll in den Gesamtprozess des PTZs integriert werden und soll dafür an einem Förderband als Transfersystem platziert werden. Auf diesem Förderband laufen in regelmäßigen Abständen Werkstücke entlang, und halten für eine kurze Zeit bei der Station an. Jedes dieser Werkstücke ist durch einen QR-Code gekennzeichnet. Die Aufgabe des Cobots ist es, aus einem Behälter einen Bolzen zu greifen und diesen im Werkstück zu montieren. Je nach QR-Code muss der Roboter keinen, einen oder zwei Bolzen einsetzen.

Um dies zu ermöglichen, muss die Bin-Picking Station zunächst umgebaut werden. Es muss möglich sein, die Bin-Picking Station an ihren festen Platz am Förderband zu manövrieren und sie dort zu befestigen. Auch muss die Station so konstruiert sein, dass der Roboter problemlos auf Kiste und Förderband zugreifen kann. Es soll ohne großen Aufwand möglich sein, die Station vom Förderband zu entfernen und für die bisherigen Laborversuche zu nutzen.

Danach muss die Bin-Picking Station in den Prozessablauf integriert werden. Hierzu muss ein Mikado-Programm erstellt werden, welches den geforderten Produktionsablauf durchführen kann.

Anschließend wird das Entwickeln eines Machine Learning Algorithmus gefordert, mit dessen Hilfe nicht nur bekannte Bolzen, sondern auch unbekannte Bauteile erkannt sowie gegriffen werden können. Der Algorithmus wird von der 3D-Kamera mit einem Bild des Bin-Picking-Behälters, in der sich eine unbekannte Menge unbekannter Bolzen befinden, gefüttert. Der Algorithmus soll dann eine Liste erstellen, in der die Maße aller zu sehenden Bolzen aufgelistet sind sowie einen Punkt finden, an dem der Bolzen gegriffen werden kann.

3. Grundlagen

Im Folgenden sollen einige für diese Arbeit relevante Grundlagen näher erklärt werden.

3.1. Aufbau der Bin-Picking-Station

Die Bin-Picking-Station wurde für die bisherigen Laborversuche auf Basis von Aluminium-Strebenprofilen der Bosch Rexroth-AG aufgebaut. Hierbei wurden Strebenprofile des Rasters 45x45mm und 90x45mm verwendet. Der Aufbau der Strebenprofile ist in nebenstehender Abbildung dargestellt. Mithilfe von Schnellspannverbindern und 90° Winkeln lassen sich die einzelnen Strebenprofile senkrecht zueinander verbinden.

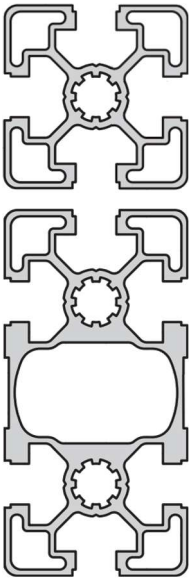


Abbildung 3: Strebenprofile [4]

Zur Vereinfachung wird das 45x45mm Strebenprofil fortan als „Profil“ und das 90x45mm Strebenprofil als „Doppelprofil“ bezeichnet. Vier Doppelprofile bilden das Grundgerüst der Bin-Picking-Station. Diese werden mit Profilen als Querstreben zusammengehalten. Der Rumpf der Station wird so zu den Maßen 70cm x 70cm in der Länge und Breite gebildet. An der Unterseite jedes der vier Doppelprofile sind Rollen befestigt, um die Station zu mobilisieren.

Mit fünf Holzplatten wird ein Schrank gebildet, in dem unter anderem die Steuersysteme für Roboterarm und 2D-Mikrometer untergebracht werden. Die Bodenplatte dieses Schrankes befindet sich in etwa 50cm Höhe. Aus der in Abbildung 2 gewählten Perspektive ergibt sich hierfür ein Volumen von gerundet 60cm in der Breite, 50cm in der Tiefe und 60cm in der Höhe.

Auf dem Rumpf der Station ist in 112cm Höhe über dem Boden über weitere Profile eine Arbeitsplatte der Größe 120cm x 70cm befestigt. Zwei von der Arbeitsplatte in die Höhe ragende, 80cm lange Profile bilden die Befestigungsmöglichkeiten für die beiden Monitore sowie die 3D-Kamera „N45“ der Marke „ensenso“ des Herstellers Optonic GmbH.

Auf der Arbeitsplatte befinden sich der Roboterarm UR3e, ein optisches 2D-Mikrometer der Modellreihe „TM-3000“ des Herstellers Keyence Deutschland GmbH, sowie Aufnahmen für die verschiedenen Behälter aus denen Einzelteile gegriffen beziehungsweise abgelegt werden. Als Peripheriegeräte sind neben Maus und Tastatur auf der Arbeitsplatte noch Fernbedienungsgeräte für den UR3e sowie das 2D-Mikrometer an der Station befestigt.

3.2. Cobot

Das Wort Cobot setzt sich aus den englischen Begriffen „Collaboration“ und „Robot“ zusammen. Es handelt sich um einen kollaborativen Roboter. Bei Robotern unterscheidet man zwischen verschiedenen Stufen der Kollaboration.

Klassischerweise arbeiten Industriebroboter in Zellen, und sind physisch durch einen Zaun o.ä. vom Rest der Welt und vor allem den Arbeitern abgeschnitten. So können die Roboter mit hohen Geschwindigkeiten arbeiten, ohne Menschen zu gefährden.

Die nächste Stufe ist Koexistenz. Der Schutzzaun entfällt, Roboter und Mensch arbeiten aber in getrennten Bereichen.

Bei der Kooperation teilen sich Mensch und Roboter den Arbeitsraum, arbeiten aber nicht gleichzeitig am selben Produkt.

Die höchste Stufe ist die Kollaboration. Mensch und Roboter arbeiten an einem Bauteil Hand in Hand. Dazu sind Cobots in der Lage [5].

Um das zu ermöglichen, muss der Roboter eingeschränkt werden. Es gibt Leistungs-, Kraft- und Geschwindigkeitsbegrenzungen. So wird die Verletzung von Menschen verhindert. Der Roboter braucht Sensorik und Software, um Kollisionen zu bemerken und darauf zu reagieren. Viele Cobots überwachen konstant ihre Umgebung, und verlangsamen ihre Arbeit, wenn Menschen in ihre Nähe kommen.

Der von uns genutzte Roboter ist der UR3e der Firma Universal Robots. Die 3 entspricht hierbei der maximalen Tragkraft von 3 kg. So kann eine sichere Kollaboration ermöglicht werden.



Abbildung 4: UR3e [6]

Der Arbeitsbereich des UR3e hat einen Radius von 0,5 m und er wiegt ca. 11kg, es handelt sich also um einen eher kleinen Roboter. Kraft-, Geschwindigkeits-, Drehmoments-, Energie- und Nachlaufbegrenzungen sorgen für erhöhte Sicherheit [6]. An den Roboter können verschiedene Endeffektoren wie beispielsweise Vakuumgreifer angeschlossen werden. Es stehen mehrere analoge und digitale Ein- und Ausgänge zur Verfügung.

Typischerweise erfolgt die Programmierung eines UR auf dem mitgelieferten HMI (Human Machine Interface). Es handelt sich um einen Touch-Bildschirm mit einigen Knöpfen und Not-Aus-Schalter. An diesem Bildschirm wird der Roboter konfiguriert, d.h. es werden die Gelenke kalibriert und die Koordinatensysteme eingerichtet. Auch kann der Roboter über Steuerschaltflächen ferngesteuert werden. Das HMI besitzt verschiedene free drive Knöpfe. Werden diese gedrückt, so kann der Roboter händisch bewegt werden. Die Programmierung in der Software PolyScope erfolgt vor allem durch das einteachen (= beibringen) von Posen. Der Roboter wird also manuell in eine Pose gebracht, diese wird dann eingespeichert und kann vom Roboter angefahren werden. Es stehen verschiedene Bewegungsarten und Steuerbausteine zur Verfügung.

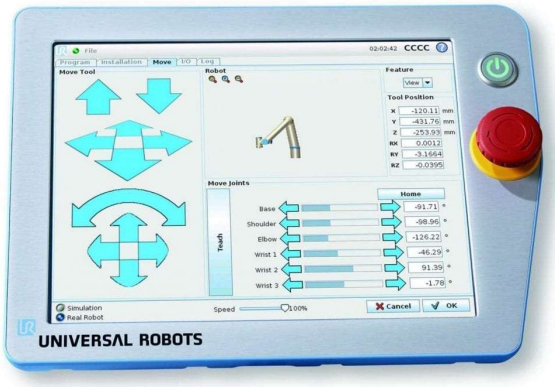


Abbildung 5: Fernsteuerung über das HMI des UR3e [7]

An der Bin-Picking-Station kommt jedoch nicht PolyScope zum Einsatz, sondern die Robotersteuerung Mikado Control auf einem speziellen Computersystem.

3.3. Mikado Control

Wie bereits erwähnt, wird die Robotersteuerung und das Bin-Picking durch die Software Mikado Control ermöglicht. Hierbei handelt es sich um eine speziell für den Bin-Picking-Prozess gefertigte Lösung der Optonic GmbH. Sie erfordert wenige Programmierkenntnisse, um komplexe Aufgaben zu erfüllen.

Zurzeit wird Mikado für den Laborversuch genutzt. Es wird ein Bolzen aus einer Kiste gegriffen und in den Messbereich eines optischen Prüfapparats gehalten. Dann wird er abhängig von seiner Qualität in Kisten einsortiert. In einer anderen Version werden die Kisten durch Steckplätze für die Bolzen ersetzt. Hierbei kommt das Bin-Picking-System zur Anwendung, welches das größte Alleinstellungsmerkmal von Mikado Control ist.

3.3.1. Aufbau eines Programms

In Mikado wird ein digitaler Zwilling des Aufbaus erstellt, so dass der Systemzustand immer verfügbar ist. Der Roboter kann über Mikado manuell angesteuert oder programmiert werden.

Ein Mikado-Programm besteht aus einer oder mehreren Sequenzen. Es gibt immer eine aktive Sequenz, welche in Dauerschleife abgearbeitet wird. Eine Sequenz besteht aus einer oder mehreren Tasks. Tasks sind die Codebausteine des Systems. Durch Tasks können die Sequenzen gewechselt, Bewegungen realisiert, mathematische und logische Operationen durchgeführt und Kameradaten analysiert werden. In Mikado können auch Variablen genutzt werden, wobei hier zum einen die klassischen Datentypen zur Verfügung stehen, zum anderen aber auch Listen, Posen oder Systemzustände gespeichert werden können.

In Mikado können Szenenobjekte definiert werden. Hierbei kann es sich zum einen um simple ROIs (Regions of Interest) handeln. Dies sind Bereiche, auf die sich die Kameraerfassung zukünftig beschränkt. Es können auch Hindernisse definiert werden. Der Roboter vermeidet Kollisionen mit diesen. Zu guter Letzt können Kisten definiert werden, also Behälter aus denen Objekte beim Bin-Picking entnommen werden. Zu jeder Kiste gehört eine ROI und eine sogenannte Übergangspose, die der Roboter beim Greifprozess anfährt. Die genaue Position von Kisten und Hindernissen kann das System per 2D Suche herausfinden. [8]

3.3.2. Bin-Picking in Mikado

Um Bin-Picking zu ermöglichen, müssen im Mikado-Projekt zunächst Produkte gespeichert werden. Beim Produkt handelt es sich um das physische Objekt, das durch die Kamera erkannt und später gegriffen werden soll. Es wird ein Name benötigt, und ein CAD-Modell des Produkts. Des Weiteren müssen verschiedene Erkennungsparameter definiert werden, wie beispielsweise die zu erwartende Produktzahl oder der Mindestabstand zwischen zwei Produkten. Zu guter Letzt müssen Greifposen definiert werden. Jede Greifpose wird in einem eigenen Posenmultiplikator eingespeichert. Dieser enthält Position und Orientierung des Endeffektors (des Greifers) beim Griff, Informationen zur Griffsymmetrie und Griffart, und einen Multiplikator mithilfe dessen die Griffe bewertet werden.

Es gibt immer exakt ein aktives Produkt. Werden produktbezogene Tasks ausgeführt, so wird, sofern nicht explizit anders angegeben, das aktive Produkt für diese genutzt. Wird beispielsweise die Griffsuche gestartet, so sucht diese nur nach Griffen an einem Objekt, das dem aktiven Produkt entspricht. Das aktive Produkt lässt sich per Task ändern.

Zur Objekterkennung erstellt der Roboter mithilfe der 3D-Kamera eine Punktwolke. Die 3D-Kamera muss hierfür im Voraus kalibriert werden. Dann folgen drei Schritte:

1. Näherungsweises Matching
   Das Kamerabild sowie das CAD-Modell des Produkts werden auf wenige Schlüsselpunkte reduziert. Sind zwei Punkte aus Kamera- und CAD-Modell näher beieinander als ein zu definierender „Scoring-Abstand“, so gehen sie positiv in den Erkennungs-Score ein. Ist der Erkennungs-Score höher als der im Produkt definierte minimale Erkennungs-Score, so wird ein Objekt erkannt.
2. Verfeinerung
   Die im ersten Schritt erkannten Objekte werden nun erneut untersucht, diesmal werden genauere Punktwolken verglichen und eine genaue Position und Orientierung der Produkte erstellt.
3. Speichern der Matches
   Ist die Option „Wiederverwendung vorheriger Matches“ aktiviert, so werden zu guter Letzt alle Matches abgespeichert. Ihre Position wird gespeichert, und bei der nächsten Objektsuche muss nur überprüft werden, ob die Matches noch vorhanden sind. Der vorherige Prozess muss unter Umständen nicht wieder vollständig durchlaufen.

Wurden Objekte gefunden, so setzt die Griffsuche an. Sie inspiziert alle Griffe, die an den erkannten Objekten ansetzen. Die möglichen Griffposen wurden zuvor im entsprechenden Produkt definiert. Ein Griff besteht grundsätzlich aus 4 Schritten:

1. Anfahren: Der Roboter fährt von der Ausgangspose, welche der in der Kiste definierten Übergabepose entspricht, zur Anfahrtspose. Die Anfahrtspose wird hierbei selbstständig aus der Greifpose berechnet.
2. Greifen: Der Roboter fährt von der Anfahrtspose linear zur Greifpose
3. Entnahme: Der Roboter fährt von der Greifpose linear zur Entnahmepose. Die Entnahmepose wird von der Greifpose ausgehend berechnet.

4. Zurückfahren: Der Roboter fährt zur Endpose, welche der Übergabepose und damit auch der Ausgangspose entspricht.

Zunächst wird die Greifpose vor und nach dem Greifen auf Kollisionen überprüft. Dann werden verschiedene Trajektorien auf Kollision überprüft. Für das An- und Zurückfahren wird durch die freie Bahnplanung eine kollisionsfreie Bahn berechnet, sofern möglich. Beim Greifen und Entnehmen werden mehrere lineare Bahnen ausprobiert. Der Weg in Greifrichtung, nach unten/oben, aus/in Richtung der Übergabepose sowie mit verschiedenen Winkeln aus/in Richtung Kistenmitte werden in dieser Reihenfolge überprüft, und ein kollisionsfreier Weg gesucht.

Es ist möglich zwei Kollisionsarten zu erlauben: Kollisionen mit anderen Produkt-Objekten und Überlappungskollisionen. Überlappungskollisionen sind Überlappungen des Objekts mit anderen Gegenständen zum Zeitpunkt des Griffs. Sie sind in der Regel Messungenauigkeiten zurückzuführen, da zwei physische Objekte typischerweise nicht den gleichen Platz beanspruchen können. Das Verbieten von Überlappungskollisionen ergibt nur Sinn, wenn in der Kiste ein Objekt liegt, welches wie mehrere verschmolzene Produkte geformt ist und nicht (in diesem Griff) gegriffen werden soll.

Hiernach wird ein einziger Griff herausgesucht, und falls gewünscht auch durchgeführt. Es gibt mehrere Auswahlkriterien für den besten Griff. Am wichtigsten ist die Kollisionsfreiheit. Auch die Fahrtstrecke der Griffe wird miteinbezogen sowie die im Produkt vorgenommene manuelle Gewichtung dieses speziellen Griffs. Auch Faktoren wie die Höhe des Objektmittelpunkts oder die Qualität der Objekterkennung können miteinbezogen werden.

Das Aufnehmen der Punktwolke, die Objekt- und Griffsuche sowie das Greifen des Teils sind einzelne Tasks und müssen separat aufgerufen werden. Da die Griffsuche einige Ressourcen in Anspruch nimmt, empfiehlt es sich, das Aufnehmen der Punktwolke und die Griffsuche so früh wie möglich im Programmablauf zu starten. So können Wartezeiten vor dem tatsächlichen Greifprozess minimiert werden. [8]

3.4. Machine Learning

Das Gebiet des Machine Learnings beschäftigt sich, wie der Name schon sagt, mit lernenden Maschinen, bzw. mit lernenden Algorithmen.

Der Algorithmus muss eine Aufgabe erfüllen, wobei der Erfüllungsgrad oder Qualität der Erfüllung nach einem zu definierenden System bewertet wird. „Lernend“ ist der Algorithmus, wenn die Durchschnittsbewertung selbstständig besser wird, je öfter der Algorithmus die Aufgabe ausführt. Der Algorithmus muss Erfahrungen sammeln und sich selbst dadurch verbessern. Den Prozess der Erfahrungssammelns nennt man Training, die dafür genutzten Daten Trainingsdaten. Der Algorithmus muss vor seinem ersten Einsatz trainiert werden. Es ist möglich, in während seiner Nutzung weiterlernen zu lassen, beispielsweise durch unsupervised learning. [9]

Das Machine Learning findet heutzutage breite Anwendung. Spamfilter arbeiten mit Machine Learning, um Spam-E-Mails zu erkennen. Die Algorithmen der meisten sozialen Netzwerke, die den Nutzern Inhalte empfehlen, sind Machine Learning Algorithmen. Personalisierte Werbung wird durch Machine Learning ermöglicht. Auch sämtliche Sprachassistenten wie Alexa oder Siri nutzen Machine Learning, um gesprochene Befehle zu interpretieren.

Wie zu sehen ist, ist ein wichtiges Anwendungsgebiet des Machine Learnings Kategorisierung (man spricht meist von „labeling“). E-Mails, Nutzer sozialer Medien oder Suchmaschinen, sogar gesprochene Worte werden kategorisiert und mit anderen Dingen verknüpft. Tatsächlich ist das auch alles, was ein fertiger Machine Learning Algorithmus macht. Er erkennt die Ähnlichkeit einer Situation mit Teilen seiner Trainingsdaten, ordnet sie der entsprechenden Kategorie zu, und reagiert wie im Training gelernt. Eine Kategorie könnte also auch sein „Situationen, in denen das selbstfahrende Auto eine Vollbremsung machen und hupen soll“.

Es gibt viele verschiedene Machine Learning Algorithmen, wobei die meisten sich in die Kategorien supervised, unsupervised, semi-supervised und reinforcement learning einteilen lassen.

3.4.1. Supervised learning

Supervised learning bedeutet übersetzt so viel wie „überwachtes Lernen“. Die Trainingsdaten bestehen hier aus Paaren von Inputs und Outputs („Kategorien“). Anhand dieser lernt der Algorithmus, einem unbekannten Input einen korrekten Output zuzuweisen. [9]

Will man beispielsweise einem Algorithmus beibringen, Katzen und Hunde zu erkennen, so füttert man ihn beim supervised learning mit Bildern von Katzen und Hunden sowie mit der Information, um welches Tier es sich handelt. Gibt man dem Algorithmus später ein Bild einer Katze, so gibt er die Information „Katze“ zurück.

Praktisch find das supervised learning beispielsweise bei Bilderkennung oder Spamfiltern Anwendung.

3.4.2. Unsupervised learning

Unsupervised learning bedeutet „unüberwachtes lernen“. Im Gegensatz zum supervised learning bestehen hier die Trainingsdaten nur aus Input. Das macht das unsupervised learning deutlich Ressourceneffizienter, da ein großer Teil der Datenaufbereitung wegfällt. Unsupervised learning wird meist für sogenanntes „clustering“ eingesetzt. Hierbei werden die verschiedenen Inputelemente in Gruppen sortiert. [9]

Will man mithilfe des unsupervised learning einem Algorithmus beibringen, Katzen und Hunde zu erkennen, so füttert man ihn mit Bildern von Katzen und Hunden. Zeigt man dem Algorithmus später ein Bild einer Katze, so erkennt er die Ähnlichkeit zu den restlichen Katzenbildern, kennt aber nicht den Begriff „Katze“. Unsupervised learning eignet sich allerdings ziemlich schlecht zur Bilderkennung. Statt die Bilder nach Tier zu sortierend, würde ein Algorithmus die Bilder wahrscheinlich eher nach Helligkeit oder Farbe sortieren.

Unsupervised learning ist nur für einfache Probleme wirklich geeignet. Praktische Anwendung findet dies beispielsweise in social Media Algorithmen. Diese speichern die Nutzeraktivitäten. Nutzer werden dann mit anderen Nutzern, die ähnliche Inhalte anschauen, zu Gruppen zusammengefasst. Den einzelnen Nutzern können nun Inhalte vorgeschlagen werden, die andere Mitglieder der Gruppe angeschaut haben. Die Wahrscheinlichkeit, dass diese Inhalte den Geschmack der Nutzer treffen, ist sehr hoch.

3.4.3. Semi-supervised learning

Semi-supervised learning kombiniert supervised und unsupervised learning. Supervised learning ist zuverlässiger und besser steuerbar als unsupervised, aber auch aufwendiger. Beim semi-supervised learning erhält der Algorithmus sowohl Input-Output-Paare als auch einzelne Inputs. Mithilfe der Input-Output Paare können die gewünschten Outputs etabliert werden und mithilfe der einzelnen Inputs, die deutlich einfacher zu beschaffen sind, verbessert er die Genauigkeit. [9]

Will man also das semi-supervised learning nutzen, um Katzen und Hunde zu unterscheiden, so füttert man den Algorithmus erst mit einigen Katzen- und Hundebildern, gemeinsam mit der Information um was es sich handelt. So baut der Algorithmus die Kategorien „Katze“ und „Hund“ auf und lernt sie grob zu erkennen. Danach füttert man den Algorithmus mit Katzen- und Hundebildern. Der Algorithmus entscheidet dann, um was es sich handelt, und fügt die Bilder der entsprechenden Datenbank hinzu. Er überwacht sich quasi selbst, und kann so sein Wissen vertiefen. Das semi-supervised learning hat ähnliche Anwendungen wie das supervised learning und ähnelt diesem auch in Bezug auf die erzielten Ergebnisse.

3.4.4. Reinforcement learning

Reinforcement learning, zu Deutsch bestärkendes Lernen, arbeitet über ein Belohnungs- und Bestrafungssystem. Der lernende Algorithmus wird hier meist als „Agent“ bezeichnet. Ziel dieses Agenten ist es, eine Belohnung zu maximieren. Er interagiert mit seiner Umgebung, und erhält dafür eine Belohnung. Der Agent passt sein Verhalten zufällig an, behält hierbei aber eher Strategien, die in der Vergangenheit für eine hohe Belohnung sorgten. Es handelt sich um einen trial-and-error Ansatz. Die Schwierigkeit liegt darin, die Experimentierfreudigkeit des Agenten optimal einzustellen. Er muss viele Möglichkeiten ausprobieren, soll aber irgendwann bei einer Lösung anlangen.

Reinforcement learning ist gut für komplexere Probleme geeignet, welche nicht mit den anderen Methoden gelöst werden können. Auch wenn es kein wirkliches „Richtig“ oder „Falsch“ gibt, ist Reinforcement learning die richtige Wahl. Beispiele hierfür wären das Spielen von Go oder die Steuerung eines selbstfahrenden Autos. [9][10]

Will man einem Algorithmus mit dieser Methode beibringen, Katzen und Hunde zu unterscheiden, so zeigt man ihm eine Serie zufälliger Katzen- und Hundebilder, und gibt für jede richtige Kategorisierung Punkte. Danach verändert der Algorithmus sich, und wiederholt das Ganze.

3.4.5. Deep Learning

Deep Learning ist ein Machine Learning Ansatz, der auf neuronalen Netzwerken basiert. Hierbei handelt es sich um eine bestimmte Art von Lernalgorithmus. Ein neuronales Netzwerk besteht aus mehreren Knoten, sogenannten Neuronen, die über gewichtete Verbindungen miteinander verknüpft sind.

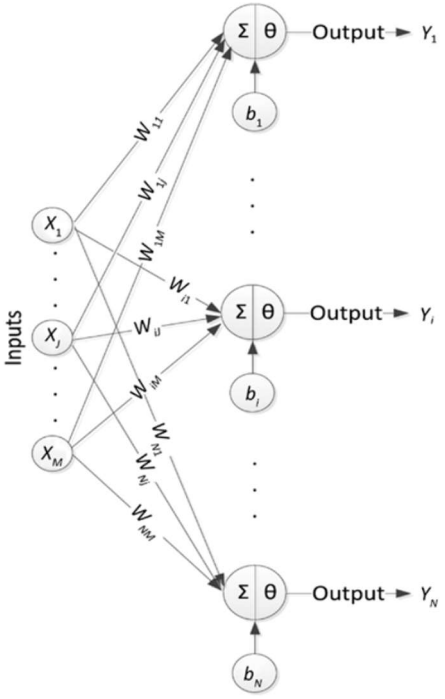


Abbildung 6: Einlagiges neuronales Netzwerk [10]

Das einfachste neuronale Netzwerk ist ein einlagiges, welches in Abbildung 6 dargestellt wird. Auf der linken Seite befindet sich die sogenannte input layer. Sie wird durch die als X_1 bis X_M bezeichneten Neuronen gebildet. In diesen Neuronen werden die einzelnen Eingangswerte abgebildet. Jedes Neuron ist mit jedem Neuron der nächsten Schicht über eine gewichtete Verbindung W verbunden, wobei die Übertragung nur in eine Richtung erfolgt. Die Werte der Eingangsneuronen werden also mit dem Gewicht multipliziert und gelangen dann am nächsten Neuron an. Alle Neuronen, die sich nicht in der Input-Layer befinden, werden hierbei auch als „computing units“ bezeichnet. Die computing units erhalten über verschiedene Verbindungen Werte. Diese werden dann addiert (∑). Auch ein Offset b wird addiert. Zu guter Letzt läuft die erhaltene Summe durch die Aktivierungsfunktion(θ). Diese errechnet den Output. Bei einem einlagigen neuronalen Netzwerk ist dies der Netzwerkoutput.

Bei einem mehrlagigen neuronalen Netzwerk folgen mehrere Schichten von computing units, siehe Abbildung 7. Die letzte Schicht trägt hierbei den Namen output layer, die Schichten zwischen input und output layer sind die hidden layers. Die Eignung eines neuronalen Netzwerks für ein Problem hängt vor allem von der Zahl seiner Schichten und der Anzahl der Neuronen pro Schicht ab. Grundsätzlich gilt: je komplexer das Problem, desto mehr Schichten werden benötigt.

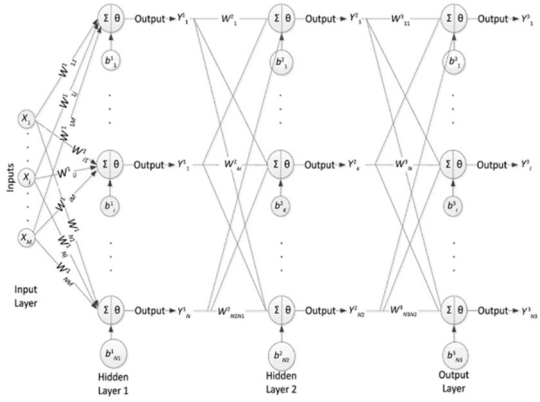


Abbildung 7: Mehrlagiges neuronales Netzwerk [10]

Neuronale Netzwerke können auf unterschiedliche Arten trainiert werden. Grundsätzlich lassen sich supervised, semi-supervised und reinforcement Methoden anwenden. Sie sind sehr beliebt, da sie auch für hochkomplexe Probleme geeignet sind. Nachteil ist, dass die Ausgabe des neuronalen Netzes nicht sehr vorhersehbar ist. Der Entscheidungsprozess des Netzes ist nicht nachvollziehbar und kann zu unerwarteten Ereignissen führen. Auch ist das Training sehr zeit- und ressourcenaufwändig, auch der Betrieb erfordert hohe Rechenleistungen.

Um ein neuronales Netzwerk zu trainieren wird zunächst eine sogenannte „loss function“ benötigt. Diese evaluiert den Unterschied zwischen gewünschtem und tatsächlichem Output. In der Regel wird hierfür der quadratische Mittelwert der Differenz von gewünschtem und tatsächlichem Output genutzt.

Eine Methodik zum Trainieren von neuronalen Netzwerken ist der „gradient descent Algorithmus“. Ein solcher Algorithmus ist in der Lage, die „Richtung“ des schnellsten Abstiegs zu finden. Das bedeutet, dass er berechnet, wie genau einzelne Variablen verändert werden müssen, um den Ausgangswert einer Funktion abzusenken und dabei möglichst wenig an den Variablen zu ändern. Dieser gradient descent Algorithmus wird auf die loss function angewendet, und die Gewichtungen unter Nutzung des gefundenen negativen Gradienten so angepasst, dass die loss function null wird. Dies wird wiederholt für alle Trainingsdaten durchgeführt, bis ein zufriedenstellendes Ergebnis vorliegt. [9][10]

3.4.6. Automatisiertes Machine Learning

Automatisiertes Machine Learning, auch AutoML, bezeichnet Software, die den Erstellungsprozess eines Machine Learning Algorithmus automatisiert. Hierzu zählen Datenvorbereitung, Modellauswahl und Optimierung des Algorithmus. Die Nutzung von AutoML vereinfacht den Erstellungsprozess eines Machine Learning Algorithmus und reduziert den Zeitaufwand erheblich.

3.4.7. Under- und Overfitting

Under- und Overfitting sind die zwei häufigsten Probleme, die bei Machine Learning Algorithmen auftreten.

Underfitting tritt auf, wenn die Trainingsdaten zu breit gestreut und das Training zu kurz sind, oder wenn der Machine Learning Algorithmus zu simpel ist. Will man einen Algorithmus trainieren, Katzen und Hunde zu unterscheiden, trainiert ihn aber nur kurz mit einigen wenigen Bildern, so wird er später sehr unzuverlässig, was das Erkennen der Tiere angeht, sein. Das gleiche tritt auf, wenn man dem Algorithmus als Eingangswerte nur 4 Pixel gibt (zu einfacher Algorithmus). Trainiert man den Algorithmus mit Daten aller Tiere, so wird seine Fähigkeit, Katzen und Hunde zu erkennen bei gleichem Trainingsaufwand geringer (zu breit gestreute Trainingsdaten). Auch das ist Underfitting für die Aufgabe des Unterscheidens von Hunden und Katzen.

Overfitting hingegen tritt auf, wenn der Machine Learning Algorithmus zu komplex ist, oder die Trainingsdaten nicht breit genug gestreut sind und das Training zu lang ist. Soll ein Algorithmus Katzen und Hunde erkennen, und wird dann lange mit einigen wenigen, wiederholt auftretenden Bildern trainiert, kommt es zu Overfitting. Der Algorithmus lernt nicht, Katzen und Hunde zu erkennen, er lernt nur die im Training genutzten Bilder zu erkennen. Dies kann auch passieren, wenn der Algorithmus zu komplex ist und zu schnell lernt, beispielsweise bei einem neuronalen Netzwerk mit zu vielen Schichten. [9]

4. Projektplanung

Zur Projektüberwachung wird ein Gantt-Diagramm erstellt, anhand dessen wichtige Termine sowie die benötigte Zeitdauer der einzelnen Arbeitsschritte im Laufe der Bearbeitung übersichtlich wahrgenommen werden können.

Nach einem Anstoßmeeting am 12.10.22 mit umfassender Vorstellung des PTZ, der Lernfabrik sowie der Bin-Picking-Station beginnt die Bearbeitung der Studienarbeit.

Zunächst gilt es, sich mit dem Bin-Picking-Prozess sowie der Bin-Picking-Station vertraut zu machen. Hierfür erfolgt zu Beginn der Studienarbeit viel Recherche zum UR3e sowie der Mikado Steuerung.

Im nächsten Schritt wird sich der Recherche zu Machine Learning sowie der Planung des Umbaus der Bin-Picking-Station gewidmet. Diese Aufgaben werden aufgeteilt, sodass diese parallel erfolgen können.

Kurz nach Beginn wird bekannt gegeben, dass die Bin-Picking-Station bis zum 30.11.22 für Laborversuche verwendet wird und der Umbau erst im Dezember begonnen werden kann.

Aufgrund eines engen Zeitfensters im Dezember, wird der 30.11.22 als Frist für die abgeschlossene Planung sowie Absprache des Umbaus mit dem Projektbetreuer und zuständigen Laboringenieur gesetzt. So kann bereits am 01.12.22 mit dem Umbau begonnen werden. Um ausreichend Vorbereitungszeit für die anstehende Klausurenphase zu erhalten, wird der 09.12.22 als Frist für den Abschluss des Umbaus gesetzt.

Zwischen dem 26.12.22 und dem 07.04.23 finden die Praxisphasen an anderen Standorten statt, weshalb in diesem Zeitraum kaum eine Bearbeitung erfolgen kann.

Aufgabe	Deadline
Umbau	13.12.2022
Integration per Mikado	15.04.2022
Projektbericht	16.01.2023
Recherche und Auswahl des KI -Algorithmus	10.04.2023
Erstellung der Datenbank	30.04.2023
Erste Implementierung	31.05.2023
Finale Implementierung	15.06.2023
Projektbericht	17.07.2023

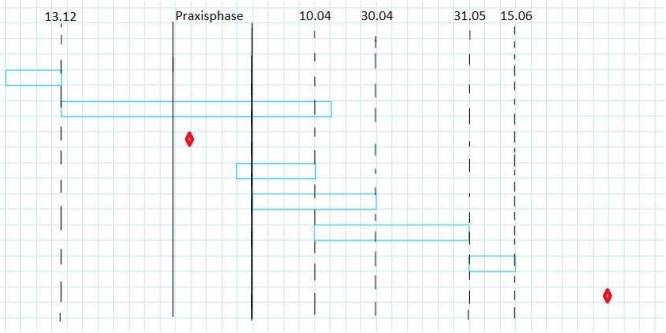


Abbildung 8: Gantt-Diagramm zur Projektplanung

5. Umbau

5.1. Anforderungen

Um die Bin-Picking-Station in die Prozesse des Produktionsnetzwerks integrieren zu können, muss die Station umgebaut werden. An den Umbau werden dabei die folgenden Anforderungen gestellt.

Derzeit schränkt die Befestigung der Kamera die Bewegungsmöglichkeiten des UR3e deutlich ein, da das vertikal verlaufende Profil der Halterung im bisherigen Zustand genau auf der schnellsten Route zwischen Behälter und Förderband liegt, wodurch der Roboterarm für die Bewegung zwischen Behälter und Förderband keine 90° Drehung durchführen kann, sondern diese durch eine Drehung im Winkel von 270° in die entgegengesetzte Richtung ersetzen muss. Um dies zu vermeiden, muss eine andere Befestigungsmöglichkeit für die Kamera gefunden werden. Die Position der Kamera selbst sowie des Behälters, aus dem Objekte gegriffen werden, darf hierbei nicht angepasst werden, da die Bin-Picking-Station nach wie vor für Laborversuche verwendet werden soll. Für Laborversuche soll diese möglichst einfach von dem Förderband des Produktionsnetzwerks getrennt und später in genau dieselbe Position eingebracht werden können. Um die Prozesssicherheit gewährleisten zu können, ist es zwingend erforderlich, dass die Station immer exakt gleich an das Förderband positioniert wird. Für eine verbesserte Handhabung der Station ist zusätzlich ein Austausch der Rollen vorzunehmen.

5.2. Planung

Aufgrund fehlender Dokumentation wird der gesamte bisherige Aufbau aus Strebenprofilen und Holzplatten vermessen, um den Umbau der Bin-Picking-Station möglichst gut planen zu können. Aus den aufgenommenen Messungen wird die Bin-Picking-Station in dem „MTpro Online Designer“ [11] der Bosch Rexroth AG nachgebaut. Für eine bessere Übersichtlichkeit werden der UR3e, das optische 2D-Mikrometer und die gesamte Peripherie des PC aus Monitoren, Maus und Tastatur in den Skizzen vernachlässigt. Zunächst wird sich der Kamera und ihrer Halterung gewidmet. Die Halterung ist derzeit in der hinteren linken Ecke über zwei 90° Winkel an zwei weiteren Strebenprofilen unter der Arbeitsplatte befestigt. Um die möglichen Bewegungsabläufe des UR3e nicht so sehr einzuschränken, wird die Befestigung der Halterung angepasst, anstelle der hinteren linken Ecke soll diese in der vorderen linken Ecke

befestigt werden. Die neue Halterung soll hierfür spiegelverkehrt aufgebaut werden. Das Schild „Optisches 2D-Mikrometer mit Cobot und Bin Picking“ soll vorerst entfallen. Um die exakte Position der Kamera beizubehalten, wird die Befestigung auf dem Strebenprofil verschoben werden müssen.

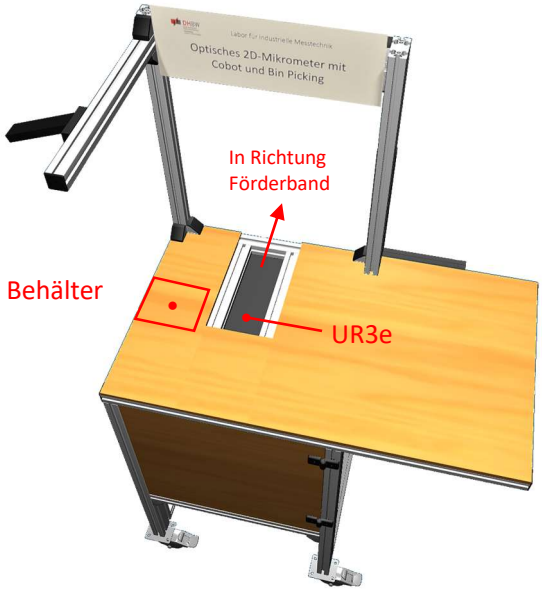


Abbildung 9: Bin-Picking-Station von vorne



Abbildung 10: Anpassung der Kamerahalterung

Um die Schnittstelle zwischen Bin-Picking-Station und Förderband zu planen, wird der benötigte Teil des Aufbaus des Förderbands ausgemessen und ebenfalls in der Online-Umgebung „MTpro Online Designer“ nachgebildet. Der Aufbau wird mit zwei horizontal herausragenden Strebenprofilen ergänzt, die über 90° Winkel an die Stützstruktur des Förderbandes befestigt werden. Der Abstand der beiden Profile beträgt dabei die Breite des Rumpfes der Bin-Picking-Station in Addition mit einer kleinen Toleranz, damit die Station problemlos hineingeschoben werden kann. Um das Einschieben der Station zu vereinfachen, werden 90° Winkel an die Enden der herausstehenden Profile befestigt und Abdeckkappen aus Kunststoff montiert. Hierdurch verbreitert sich die Aufnahme um 45mm sowohl auf der linken als auch auf der rechten Seite und verschmälert sich letztlich auf die Breite des Rumpfes der Bin-Picking-Station.



Abbildung 11: Aufnahme für die Bin-Picking-Station am Förderband

Auf der Rückseite der Bin-Picking-Station befinden sich drei Kabelkanäle für die ordentliche Führung der Verkabelung für die zwei Monitore sowie für die weitere Peripherie des PCs. Diese Kabelkanäle weisen wie die Strebenprofile eine Breite von 45mm auf. Die Kabelkanäle stellen hierbei eine große Behinderung dar, da sie wie in Abbildung 12 zu sehen, auf dem von hinten gesehen linken vertikalen Strebenprofil angebracht sind und somit an die Aufnahme des Förderbandes anstoßen, wobei auf dem von hinten gesehen rechten Strebenprofil kein Kabelkanal angebracht ist. Somit steht eine Seite der Hinterseite um 45mm weiter heraus als die andere Seite, wodurch die Bin-Picking-Station nicht präzise in die Aufnahme eingeschoben werden kann.



Abbildung 12: Bin-Picking-Station von hinten

Um dieser Problematik entgegenzuwirken, werden, wie in Abbildung 13 und Abbildung 14 dargestellt, unten auf der Hinterseite zwei kurze Doppelprofile als Abstandshalter befestigt, die um 45mm weiter herausragen, als die Kabelkanäle.



Abbildung 13: 90mm Abstandshalter von hinten

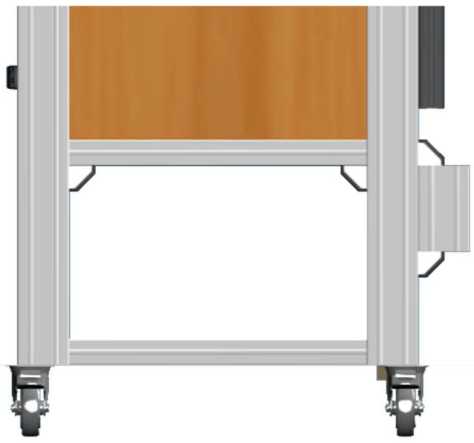


Abbildung 14: 90mm Abstandshalter von rechts

Somit kann die Bin-Picking-Station problemlos an das Förderband geschoben werden und hat in der Aufnahme nur wenige Millimeter Bewegungsmöglichkeit zwischen den beiden herausstehenden Strebenprofilen.

Nun gilt es eine Möglichkeit zu finden, die Bin-Picking-Station dauerhaft an das Förderband befestigen zu können, sodass diese nicht mehr herausgezogen werden kann. Das Anziehen der Bremsen der vier Rollen ist hierbei nicht ausreichend. Die Verbindung soll für den Einsatz zu Laborversuchen allerdings möglichst leicht einfach werden können.

Um dies zu erreichen, werden die in dargestellten Abbildung 15 Spannverschlüsse verwendet. Solche Spannverschlüsse gehören nicht zum Produktportfolio der Bosch Rexroth AG, sondern werden in zahlreicher Ausführung frei verkäuflich im Internet angeboten und sind universal einsetzbar. Diese bestehen aus zwei Teilen, die mit den jeweiligen Gegenständen, die aneinander befestigt werden sollen, verschraubt werden. Die Abbildung stellt einen solchen Spannverschluss in geschlossener, gespannter Form dar. Der linke Teil stellt einen Haken dar, der rechte Teil eine dreieckige Öse, die über Betätigung des roten Hebels über den Haken gespannt und wieder von diesem gelöst werden kann. Die dreieckige Öse ist über ein Gewinde in der Halterung verschraubt, wodurch der Abstand zwischen der Öse und ihrer Halterung um einige Zentimeter angepasst werden kann und so für viele Anwendungsfälle eine Lösung bietet.



Abbildung 15: Spannverschluss

Diese Spannverschlüsse werden bei den zuvor erläuterten, als Abstandshalter verwendeten Doppelprofilen eingesetzt, da diese direkt an die Stützstruktur des Förderbandes geschoben werden. Die Verwendung der Spannverbinder auf der rechten Seite der Bin-Picking-Station ist in Abbildung 16 dargestellt. An dem Abstandshalter wird der Haken befestigt, an der Stützstruktur des Förderbands wird auf selber Höhe die Öse mit Klemmhebel befestigt. Auf der gegenüberliegenden Seite wird ein weiterer Spannverschluss sinngemäß verwendet.

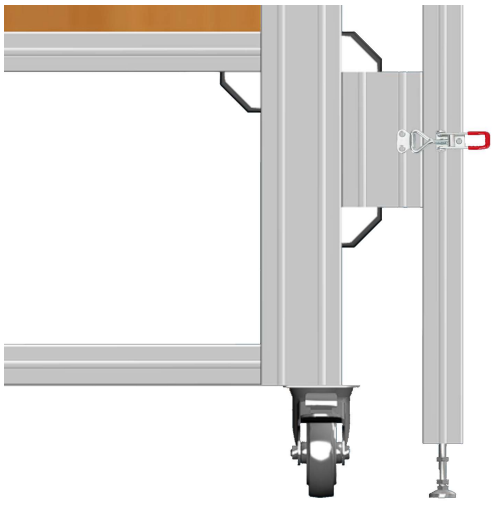


Abbildung 16: Verwendung Spannverschluss am Abstandshalter unten rechts

Um eine exakte Positionierung in der Schnittstelle Bin-Picking-Station ↔ Förderband zu gewährleisten, sollen weitere Spannverschlüsse an den herausstehenden Strebenprofilen der Aufnahme des Förderbandes verwendet werden.

In der folgenden Abbildung 17 ist die Bin-Picking-Station vollkommen eingeschoben in der Aufnahme des Förderbandes dargestellt. Zu sehen ist die Bin-Picking-Station von der linken Seite. Die herausstehenden Strebenprofile als Aufnahme sollen über die Spannverschlüsse mit der Bin-Picking-Station verbunden werden. Dies ist bei den hier verwendeten Spannverschlüssen nur bei zwei parallel verlaufenden Strebenprofilen möglich. Daher wird 45mm unter der Höhe des herausstehenden Profils der Aufnahme ein sehr kurzes Profil mithilfe eines Winkels an der Bin-Picking-Station befestigt. Somit steht nun ein zur Aufnahme paralleles Profil zur Befestigung per Spannverschluss zur Verfügung.



Abbildung 17: Bin-Picking-Station festgespannt in der Aufnahme

Somit ist bei festgespannten Spannverschlüssen jegliche Bewegung der Bin-Picking-Station in der Aufnahme ausgeschlossen.

In der folgenden Abbildung 18 ist der komplette Nachbau der Bin-Picking-Station sowie der Aufnahme des Förderbandes im „MTpro Online Designer“ dargestellt.

Die Kennzeichnungen A bis G der Strebenprofile dienen zur Vereinfachung der weiteren Erläuterung des Umbaus in Kapitel 5.3 Umsetzung. Für die Visualisierung der Profile C und D wurde in der Abbildung die Holzplatte auf der rechten Seite des Rumpfes ausgeblendet.

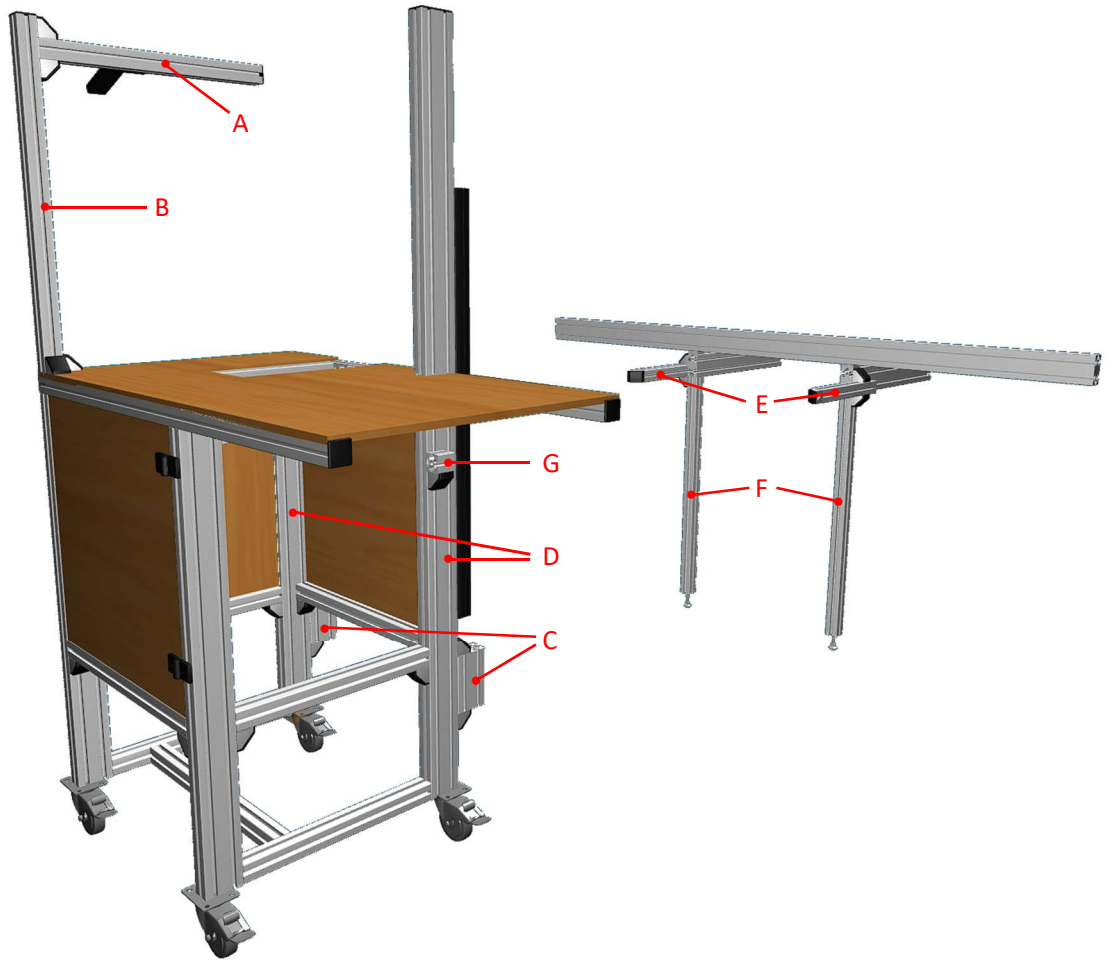


Abbildung 18: Bin-Picking-Station und ihre Aufnahme im „MTpro Online Designer“

5.3. Umsetzung

Um mit dem Umbau beginnen zu können, wird zuerst die Holzarbeitsplatte demontiert. Diese ist über M6 Senkkopfschrauben und Hammermuttern an den darunterliegenden horizontal verlaufenden Strebenprofilen befestigt.

Nun können die Abdeckkappen der 90° Winkel, die die Halterung der Kamera an dem Rumpf der Bin-Picking-Station befestigen, abgenommen werden und die beiden Winkel abgeschraubt werden. Somit ist die gesamte Kamerahalterung aus den beiden Profilen A und B nun demontiert. Diese wird spiegelverkehrt auf der von oben aus gesehen vorderen linken Ecke mithilfe der 90° Winkel sowie M6 Zylinderkopfschrauben und Hammermuttern wieder befestigt. Durch das spiegelverkehrte Befestigen der beiden Profile ist die Kamera nun falsch positioniert, die Linse ist nun nicht mehr auf die Bin-Picking-Station gerichtet, sondern schaut von dieser weg. Die Position der Kamera auf dem Profil A wird ausgemessen, anschließend wird die Kamera demontiert. Das Profil A wird mithilfe einer Metallbandsäge auf eine Länge von 28cm zugesägt, da dieses ungebräuchlich lang war. Das Profil wird über zwei Winkel mit Profil B verbunden und die Kamera wird anschließend auf der gegenüberliegenden Seite des Profils mithilfe einer von bisherigen Studenten entworfenen Konstruktion wieder montiert. Trotz der Bemühungen, die Position der Kamera beim Umbau in Bezug auf die Bin-Picking-Station exakt zu erhalten, ist es empfehlenswert diese im Anschluss neu zu kalibrieren.



Abbildung 19: Befestigung der Kamera an dem Profil A

Um die Holzarbeitsplatte nun wieder befestigen zu können, muss aus dieser ein Ausschnitt für die umpositionierte Kamerahalterung ausgesägt werden. Wird die Arbeitsplatte vorne von oben betrachtet, muss in der linken unteren Ecke ein Ausschnitt von 45x45mm für Profil B vorgenommen werden. Um auch Platz für die beiden Winkel zu bieten, wird der Ausschnitt nach hinten und nach rechts um jeweils 45x45mm erweitert. Dies ist beispielhaft in nebenstehender Abbildung dargestellt. Zu beachten ist, dass die Abbildung nicht maßstabsgetreu ist, sondern lediglich der Veranschaulichung dient. Anschließend wird die Arbeitsplatte wieder über Senkkopfschrauben und Hammermuttern befestigt.



Abbildung 20: Nötige Bearbeitung der Arbeitsplatte

Als nächstes werden die Abstandshalter zur Stützstruktur des Förderbandes befestigt. Hierfür werden mithilfe der Metallbandsäge aus einem Doppelprofil zwei Doppelprofile der Länge 15cm abgesägt. Diese Abstandshalter (Profile C) sollen mit Winkeln an die Rückseite der Bin-Picking-Station an den Profilen des Typs D befestigt werden. Hierfür müssen Gewinde in die Abstandshalter geschnitten werden. In die Profilkernzüge werden M12 Innengewinde geschnitten. So wird je ein Winkel auf beide Enden der Abstandshalter mit einer M12 Zylinderkopfschraube befestigt, abschließend wird jeder der Winkel über jeweils eine M6 Zylinderkopfschraube und Hammermutter an beiden Profilen des Typs D auf der Rückseite der Bin-Picking-Station befestigt.

Anschließend wird die Aufnahme für die Bin-Picking-Station an der Stützstruktur des Förderbandes aufgebaut. Für die zwei herausstehenden Profile des Typs E wird mithilfe der Bandsäge ein Strebenprofil zu zwei Profilen der Länge 35cm zugesägt. Mithilfe von je zwei 90° Winkeln werden diese zwei Profile an jeweils einem parallel verlaufenden Doppelprofil unter dem Förderband befestigt. Von den 35cm Gesamtlänge befinden sich 9cm unter dem Förderband, sodass die Profile des Typs E insgesamt 29cm hervorstehen. In die Profilkernzüge der Profile wird wie bereits bei den Abstandshaltern der Bin-Picking-Station ein M12 Innengewinde geschnitten, sodass 90° Winkel mithilfe von M12 Zylinder-kopfschrauben an den Enden befestigt werden können, um den Einschub der Bin-Picking-Station zu erleichtern.

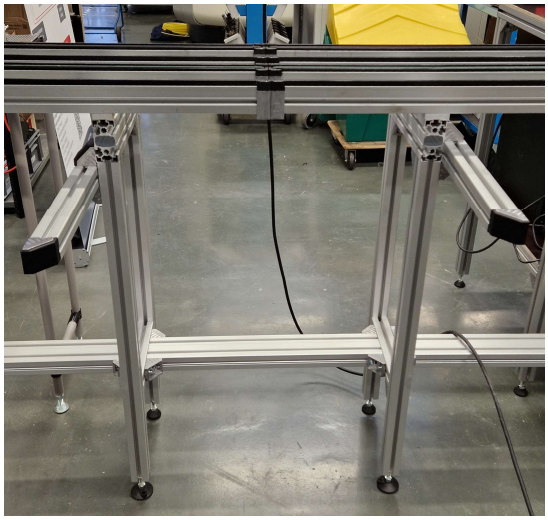


Abbildung 21: Aufnahme für die Bin-Picking-Station am Förderband

Daraufhin werden die sehr kurzen Profile des Typs G befestigt, die benötigt werden, um die Bin-Picking-Station an den Profilen des Typs E festspannen zu können. Zunächst werden mit der Metallbandsäge aus einem Strebenprofil zwei 45mm lange Profile gesägt. Die Unterkante der Profile E befindet sich auf einer Höhe von genau 100cm über dem Boden, daher werden die 45mm langen Profile mithilfe je eines 90° Winkels, einer M6 Zylinderkopfschraube und einer Hammermutter so an den Doppelprofilen des Typs D befestigt, dass sich deren Oberkante wenige Millimeter unter der Höhe von 100cm über dem Boden befindet.

Da nun alle benötigten Profile an der Bin-Picking-Station und am Förderband angebracht sind, kann die Bin-Picking-Station in die Aufnahme geschoben werden. Als nächstes können die Spannverschlüsse montiert werden.

Für die Montage der Spannverschlüsse müssen zunächst die Bohrlöcher der Spannverschlüsse vergrößert werden. Deren Durchmesser beträgt 4mm, sodass M4 Schrauben und Hammermuttern verwendet werden könnten, allerdings bieten diese keinen sicheren Halt in den Nuten der verwendeten Profile. Daher muss auf M5 Befestigungsmaterial zurückgegriffen werden, für deren Einsatz die Bohrlöcher der Spannverschlüsse auf 5mm aufgebohrt werden müssen.

Die Haken der Spannverschlüsse werden auf den Außenseiten der Profile des Typs C sowie auf den Profilen des Typs G montiert. Die vier Haken werden über jeweils zwei M5 Flachkopfschrauben und Hammermuttern in den Nuten der jeweiligen Profile befestigt. Die Gegenstücke der Spannverschlüsse, bestehend aus dreieckiger Öse und rotem Hebel, werden mithilfe jeweils einem Paar aus M5 Sechskantschraube und Hammermutter an den Außenseiten der Profile des Typs E sowie der Profile des Typs F in den Nuten der Profile befestigt. Die Entfernung der dreieckigen Öse zu ihrer Halterung lässt sich über das Herein- oder Herausdrehen der Öse aus dem Innengewinde in der Halterung anpassen. Dieser Abstand wird an jeder Öse so eingestellt, dass die Öse den Haken mit hoher Spannkraft greifen kann, das Lösen der Spannverbindung über den roten Hebel hingegen mit wenig Kraft erfolgen kann. Über zwei M5 Muttern wird der Abstand der Ösen letztlich fixiert.

Daraufhin wird die gesamte Bin-Picking-Station aus der Aufnahme des Förderbandes gezogen und angehoben, um die vier Rollen austauschen zu können. Die neuen Rollen werden mit M12 Zylinderkopfschrauben in bereits geschnittene M12 Innengewinde in den vier Doppelprofilen (unter anderem Typ D) geschraubt.

Um das Greifen kleinerer Bauteile prozesssicherer zu gestalten, wird der Endeffektor des UR3e ausgetauscht. Anstelle der bisher in den Laborversuchen verwendeten Endeffektoren mit unterschiedlichen Greifarmen wird ein Vakuum-Endeffektor der J. Schmalz GmbH verwendet. Somit werden die Bauteile nicht mehr aus dem Behälter gegriffen, sondern angesaugt.



Abbildung 22: Vakuum-Endeffektor

Der Vakuum-Endeffektor nutzt den Venturi-Effekt, hierbei wird Druckluft durch eine Düse in einen offenen Zylinder größeren Durchmessers (Abbildung 22, schwarzes Rohr) befördert. Am Austrittspunkt der Düse entsteht hierbei ein Unterdruck beziehungsweise ein Vakuum, das in diesem Fall zum Ansaugen der Einzelteile verwendet wird. In Abbildung 23 ist der hier verwendete Venturi Effekt dargestellt.

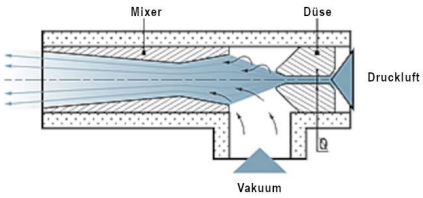


Abbildung 23: Venturi Effekt [12]

Um das Ansaugen zu ermöglichen, wird der Vakuum-Erzeuger, der sich zwischen Endeffektor und Werkzeugaufnahme des UR3e befindet, pneumatisch angesteuert. Hierfür wird das Elektromagnetventil „EMVP-5-24V-DC-3/2-NC“ der J. Schmalz GmbH verwendet, das die im PTZ vorhandene Druckluft bei Anliegen eines Steuersignals dem Endeffektor über Druckluftschläuche bereitstellt. Das Elektromagnetventil verfügt über einen Eingang, an dem die dauerhafte Druckluft angeschlossen ist und zwei Ausgänge, wovon einer einen Öffner und der andere einen Schließer darstellt. Hier wird der Ausgang des Schließers per Druckluftschlauch mit dem Endeffektor verbunden, somit werden Bauteile angesaugt, wenn am Elektromagnetventil eine Spannung anliegt. Der Ausgang des Öffners wird mit einem 1/8“ Blindstopfen verschlossen. Das Elektromagnetventil wird über ein dreiadriges Kabel für Steuersignal, Masseverbindung und Erdung mit der Mikado Steuerung verbunden.

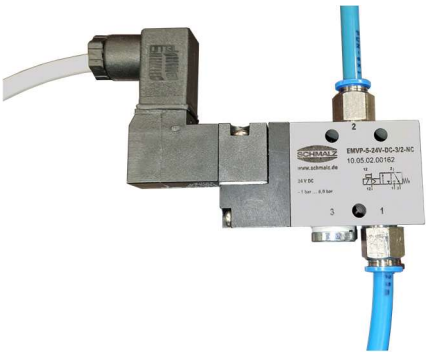


Abbildung 24: Elektromagnetventil zur Ansteuerung des Endeffektors

Damit ist der Umbau der Bin-Picking-Station abgeschlossen. In der nachfolgenden Abbildung ist die Bin-Picking-Station nach vollendetem Umbau im eingeschalteten Zustand aus der Frontperspektive zu sehen.



Abbildung 25: Bin-Picking-Station nach vollendetem Umbau

6. Prozessintegration

Nach dem erfolgten Umbau muss nun eine Software erstellt werden, mit deren Hilfe der Cobot die geforderte Aufgabe erfüllen kann. Er muss einen QR-Code auslesen, dann abhängig vom QR-Code Bolzen aus der Kiste greifen und einbauen.

6.1. Anforderungen

Im Folgenden sollen einige genauere Anforderungen definiert werden.

1. Die Software soll in Mikado erstellt werden.
2. Die Software soll so ressourceneffizient und schnell wie möglich arbeiten.
3. Der Programmfluss soll so aussehen:

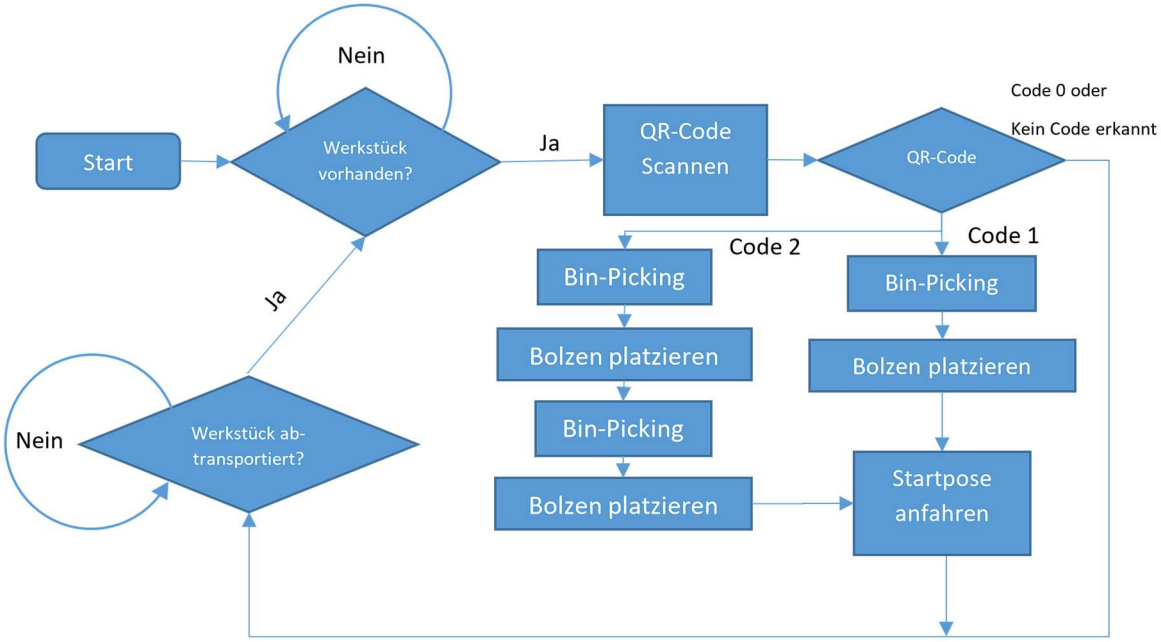


Abbildung 26: Programmfluss

3.1. Das Programm muss erkennen wenn ein Werkstück vorliegt, und auch wenn es abtransportiert wird.
3.2. Der QR-Code muss gescannt werden und auf ihn reagiert werden.
3.3. Bin-Picking für den Bolzen muss möglich sein
3.4. Der Bolzen muss an verschiedenen Positionen platziert werden können

6.2. Geplante Vorgehensweise

Bei der Implementierung ergibt sich das Problem, dass noch keine Kamera zur Erkennung des QR-Codes vorhanden ist. Auch ist nicht bekannt, wie das Auslesen des QR-Codes erfolgt bzw. wie diese Informationen an das System weitergegeben werden. Dies fällt nicht in den Zuständigkeitsbereich dieser Studienarbeit.

Es wird im Folgenden davon ausgegangen, dass der QR-Code einen Zahlenwert Namens „boltnumber“ zurückgibt, welcher der gewünschten Bolzenzahl entspricht. Ist dies nicht der Fall, so muss eine Funktion erstellt werden, die diesen Zahlenwert erstellt. Falls kein QR-Code erkannt werden kann, wird die boltnumber auf null gesetzt.

Es müssen mehrere Sequenzen erstellt werden:

1. startSequence:
   Die 3D-Kamera nimmt eine Punktwolke auf.
   Die Griffsuche wird gestartet.
   Der Roboter bewegt sich in seine Grundpose.
   Es wird zur Sequenz „baseState“ gewechselt.

2. baseState
   Das System überprüft, ob ein Werkstück auf dem Förderband eintrifft.
   Trifft eines ein, so wird zur Sequenz „qrScan“ gewechselt.

3. qrScan:
   Der QR-Code wird gescannt.
   Wenn die boltnumber Null ist, wird zur Sequenz „baseState“ gesprungen.
   Wenn die boltnumber Eins oder Zwei ist wird zur Sequenz „firstBolt“ gewechselt.
4. firstBolt
   Ein Bolzen wird gegriffen.
   Die 3D-Kamera nimmt eine Punktwolke auf.
   Die Griffsuche wird gestartet.
   Der Roboter platziert den Bolzen.
   Der Roboter bewegt sich in die Grundpose.
   Wenn boltnumber Zwei ist wird zur Sequenz „secondBolt“ gesprungen.
   Sonst wird zur Sequenz „baseState“ gesprungen.

5. secondBolt
   Ein Bolzen wird gegriffen.
   Die 3D-Kamera nimmt eine Punktwolke auf.
   Die Griffsuche wird gestartet.
   Der Roboter platziert den Bolzen (an einer anderen Stelle als den ersten).
   Der Roboter bewegt sich in die Grundpose.
   Es wird zur Sequenz „baseState“ gesprungen.

6.3. Probleme

In der zuvor detaillierten Umsetzung bleiben einige Probleme jedoch unbehandelt. Auf sie soll im Folgenden eingegangen werden

Zum einen existiert das bereits erwähnte Problem des Einlesens der QR-Codes. Die Lösung hierfür liegt nur teilweise in unserem Zuständigkeitsbereich. Die Montage der Kamera wird nicht von uns vorgenommen. Da (noch) nicht bekannt ist, wie die QR-Code-Daten an unser System weitergeleitet werden, kann hier nicht sehr exakt geplant werden. Die Zahl der Bolzen aus den QR-Code-Daten zu extrahieren, sollte jedoch relativ einfach mittels einiger Verzweigungen möglich sein. Sobald die Form der Eingangsdaten bekannt ist, kann dieses Problem gelöst werden.

6.3.1. Antransport des Werkstücks

Ein weiteres Problem ist, dass der Cobot momentan nicht in der Lage ist, zu erkennen, ob ein Werkstück an der Station ankommt. Hier kommen im Wesentlichen drei Möglichkeiten in Frage:

1. Die Kamera, mit der der QR-Code eingescannt wird, überprüft das Förderband.
2. Die Station wird an das gemeinsame Kommunikationssystem angeschlossen und erfährt so, wenn ein Bauteil ankommt.
3. Der Roboter überprüft das Förderband nicht, sondern arbeitet zeitgesteuert. Da der Produktionsprozess sich an allen Stationen periodisch wiederholt sollte dies möglich sein. Die Synchronisierung mit dem restlichen System stellt dann allerdings ein Problem dar.

Die einfachste Lösung wäre hierbei die zweite. Um die Umsetzbarkeit zu prüfen, müssen wir uns in der nächsten Theoriephase mit den Gruppen, die sich mit dem Förderband und dem PTZ-Kommunikationssystem beschäftigen zusammensetzen. Optimalerweise wird hierfür ein Digitaleingang des Cobots genutzt. Ein High-Pegel liegt vor, wenn das Förderband sich bewegt. Eine positive Flanke bedeutet also den Abtransport eines Werkstücks, eine negative die Ankunft.

Die erste Lösung ist in der Umsetzung relativ kompliziert. Es muss ein Algorithmus entwickelt werden, der anhand von Videodaten erkennt, ob sich das Band bewegt. Da das Band jedoch vollständig schwarz ist, ist dies sehr kompliziert. Auch die Erkennung eines Werkstücks ist nicht gerade einfach. Grundsätzlich wäre der Mikado-Objekterkennungsalgorithmus dazu in der Lage, bräuchte aber eine auf das Förderband gerichtete 3D-Kamera. Auch würde die Rechenzeit hierdurch drastisch steigen. Die Entwicklung eines eigenen Algorithmus auf der anderen Seite wäre sehr zeitintensiv, und würde den Rahmen der Studienarbeit sprengen.

Die Zeitsteuerung wäre auch eine praktikable Lösung, die einfach zu implementieren ist. Jedoch stellt sich hier die Problematik der Synchronisierung der Systeme. Diese lässt sich auf zwei Arten realisieren. Entweder automatisch, wofür aber Kommunikation mit dem Förderband nötig ist. Wenn diese Kommunikation möglich ist, kann auch einfach der zweite Ansatz gewählt werden, welcher ähnlich komplex, dafür aber zuverlässiger und Ressourceneffizienter ist. Die Systeme können auch manuell synchronisierbar sein. Hier spielt jedoch die menschliche Ungenauigkeit eine große Rolle, und ein Mensch müsste diese repetitive und eigentlich gut automatisierbare Tätigkeit regelmäßig durchführen.

Zur Lösung dieses Problems ist es zunächst nötig, sich mit den Gruppen des Förderbands und des Kommunikationssystems in Verbindung zu setzen, und entsprechend den so erlangten Informationen eine Lösung auszuwählen. Danach geht es an die Umsetzung eben dieser.

6.3.2. Keine Griffe möglich

Es kann auch dazu kommen, dass kein Griff möglich ist. Wenn das passiert, kann kein Bolzen ins Werkstück eingesetzt werden. Es ist möglich, eine Funktion zu implementieren, welche in diesem Fall den Greifer durch die Schüssel fahren lässt. So können die Bolzen durchgemischt und eventuell doch mögliche Griffe gefunden werden. Die Implementierung hiervon ist relativ einfach. Es wird eine Sequenz „noGrip“ erstellt. Diese Sequenz kann in der „Teil greifen“ Task in den Sequenzen „firstBolt“ und „secondBolt“ eingetragen werden, und so aufgerufen werden, falls kein Griff möglich ist. Sie bewegt den Endeffektor durch die Kiste, nimmt eine neue Punktwolke auf und startet eine neue Griffsuche. Danach springt sie zurück zur Sequenz „firstBolt“ oder „secondBolt“. Es ist auch möglich gezielt einzelne Objekte zu verschieben, um die Bolzen durchzumischen. Beide Lösungen könnte jedoch zu Problemen beim Einhalten der Zykluszeit führen, da die Griffsuche ein relativ langwieriger Prozess ist.

Ein Spezialfall ist hier die leere Kiste. Ist kein Objekt in der Kiste, so kann auch keins gegriffen werden. Es ist möglich eine Sequenz zu implementieren, in die das Programm springt, falls kein Objekt gefunden wurde. Diese Sequenz löst dann einen Alarm aus, und wechselt bei Quittierung des Alarms in die „startSequence“.

Eine Lösung dieses Problems ist momentan nicht vorgesehen. Sie ist nicht in den Anforderungen enthalten. Der Zeitplan der Studienarbeit Arbeit ist sehr eng, und dieses Problem ist eines, das erst angegangen wird, wenn die restlichen Aufgaben erfüllt sind.

6.3.3. Position des Werkstücks

Um die Bolzen korrekt einzusetzen, muss das System die Position des Werkstücks auf dem Förderband kennen. Die Lösung dieses Problems ist jedoch denkbar einfach. Das Werkstück hält immer an derselben Stelle vor der Station. Der Umbau der Station ermöglicht eine gleichbleibende Platzierung dieser. Dem Roboter muss also nur die korrekte Bewegung beigebracht werden, und führt diese dann quasi „blind“ aus.

7. Machine Learning

Es soll ein Machine Learning Algorithmus erstellt werden, der in der Lage ist, verschiedene Bolzen zu erkennen.

7.1. Anforderungen

1. Der Algorithmus soll mit Machine Learning Methoden realisiert werden.
2. Er soll in der Lage sein, verschiedene Bolzen zu erkennen. Das bedeutet:
   2.1. Der Input des Algorithmus ist ein zweidimensionales Kamerabild.
   2.2. Der Output des Algorithmus ist eine Liste der erkannten Bolzen.
      2.2.1. Diese Liste soll in eine .txt Datei geschrieben werden.
      2.2.2. Inhalt dieser Liste sind die Namen der erkannten Bolzen in der Form M<Durchmesser in mm>x<Länge in mm>, also beispielsweise M5x40. Auch die Anzahl der jeweiligen erkannten Bolzen wird benötigt.
      2.2.3. Die Liste sollte übersichtlich formatiert sein.
   2.3. Die Größe der Bolzen, die erkannt werden müssen, ist auf einen sinnvollen Bereich begrenzt. Es werden keine Bolzen schwerer als 1 kg benötigt, und keine die länger als 15 cm sind.
3. Es gelten generelle Qualitätsanforderungen:
   3.1. Der Algorithmus soll so einfach wie möglich sein.
   3.2. Der Algorithmus soll so schnell wie möglich arbeiten.

7.2. Mögliche Vorgehensweise

Unsere Aufgabe ist eine einfache Kategorisierung. Hier eignen sich die supervised oder semi-supervised Lernmethoden. Die Auswahl eines konkreten Algorithmus gestaltet sich etwas komplizierter, üblich für Bilderkennung ist jedoch ein Deep Learning Ansatz.

Bei unserer Aufgabe handelt es sich um ein Klassifikationsproblem. Eingangswerte sind die einzelnen Pixel eines Bildes. Auf diesem Bild können ein oder mehrere Objekte vorhanden sein, welche klassifiziert werden müssen. Die zufällige Platzierung erschwert diesen Prozess.

Wären die Bolzen fest platziert, so würde sich ein „nearest neighbor“ Algorithmus anbieten. Dieser vergleicht seinen Input mit seinen Trainingsdaten, und gibt zurück welche eingelernte Kategorie die ähnlichsten Inputs hat. Da die Bolzen aber zufällig platziert sind, würde dies nur zur Kategorisierung nach Anordnung führen. Auch die Bayes Methodiken, welche mit Wahrscheinlichkeitsbäumen arbeiten, würden daran scheitern. Dasselbe gilt für Klassifikationsbäume, bei denen die Inputs durch einen Entscheidungsbaum laufen, und für Diskriminanzanalysen, welche statistische Methoden sind. [10]

Eventuell ist es sinnvoll, die Aufgabe in zwei Stufen zu unterteilen. Die erste Stufe dient hierbei der Erkennung einzelner Objekte und könnte eventuell ohne Machine Learning umgesetzt werden. Die zweite Stufe klassifiziert einzelne Objekte. Auch sie könnte eventuell ohne Machine Learning umgesetzt werden oder unter Nutzung der erwähnten Methoden.

Als einfachste Lösung bietet sich jedoch Deep Learning an. Die Verarbeitung so vieler Inputs ist mit den anderen Methodiken sehr schwierig, das Deep Learning ist nicht grundlos der Standard für Bilderkennung.

Auch das Nutzen von AutoML ist eine nicht zu verachtende Möglichkeit. Im Hinblick auf den engen Zeitplan und unsere mangelnde Expertise wird sich dies als der beste Weg erweisen.

Eine finale Entscheidung muss zu einem späteren Zeitpunkt getroffen werden.

Ein weiterer wichtiger Punkt ist die Erstellung der Trainingsdaten. Sie sollte optimalerweise von einer Software vorgenommen werden. Es muss ein Programm geschrieben werden, welches ein Bild der Bin-Picking Station sowie Bilder verschiedener Bolzen erhält. Es platziert die Bolzenbilder in der Kiste und übergibt das Ergebnis dem Machine Learning Algorithmus, gemeinsam mit der Liste, die der Algorithmus bei diesem Bild erstellen soll.

8. Fazit und Ausblick

Der erste Teil der Bearbeitung der Studienarbeit war stark von theoretischer Natur. Es fand umfangreiche Recherche zu den Fachgebieten und Anwendungsfällen des Bin-Picking-Prozesses und den vielen Unterkategorien des Machine Learnings statt. Zudem wurde sich mit der Bin-Picking-Steuerung über Mikado Control sowie der damit verbundenen Software vertraut gemacht.

Die Prozessintegration in das Produktionsnetzwerk wurde umfangreich geplant und auch mögliche Probleme berücksichtigt und angesprochen, ebenfalls wurden Anforderungen an den Machine Learning Algorithmus formuliert.

Der Umbau der Bin-Picking-Station wurde durchgeführt und verschiedene Strebenprofile montiert, mit deren Hilfe das Festspannen der Bin-Picking-Station ermöglicht wird. Hierbei wurden alle in Kapitel 5.1 formulierten Anforderungen an den Umbau eingehalten.

Darunter fällt auch der Umbau der Befestigung der für den Bin-Picking-Prozess essentiellen 3D-Kamera, die durch den Umbau um wenige Millimeter sowie wenige Grad Neigung in jede der drei Drehachsen verrückt sein könnte. Daher sollte die Kamera noch kalibriert werden. Dies erfolgt mithilfe der in nebenstehender Abbildung 27 dargestellten Kalibrierplatte der Optonic GmbH. Diese Kalibrierplatte wird in der Werkzeugaufnahme des Roboterarms befestigt und über die manuelle Ansteuerung in den Sichtbereich der 3D-Kamera positioniert. Über die Mikado-Steuerung lässt sich die Kalibrierplatte nun in verschiedene vordefinierte Positionen fahren, die dann eine Kalibrierung der Kamera bewirken.

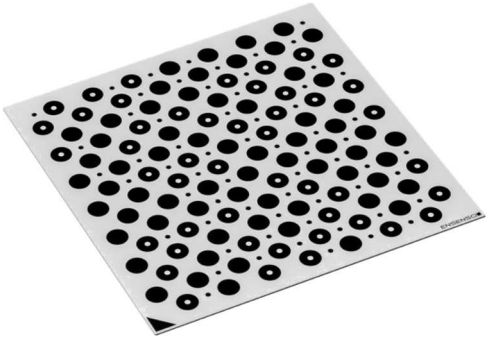


Abbildung 27: 3D-Kamera Kalibrierplatte [13]

Zur Nutzung des Bin-Picking-Systems muss von der Roboterzelle in ein digitales Abbild erschaffen werden, um die Routenplanung und Kollisionsüberwachung realisieren zu können. Dieses wurde bereits erstellt, allerdings muss noch die umpositionierte Kamerahalterung aktualisiert werden. In Abbildung 28 ist der digitale Zwilling der Bin-Picking-Station vor dem Umbau dargestellt. Hier muss zusätzlich der neue Vakuumgreifer eingerichtet werden.

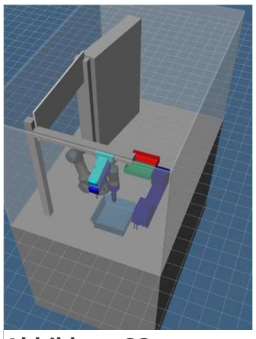


Abbildung 28: Digitaler Zwilling

Die Entscheidung auf die Auswahl eines KI-Algorithmus muss gefällt werden, dies wurde bereits in Kapitel 7.2 diskutiert, allerdings noch keine finale Auswahl getroffen. Auf Basis des gewählten Algorithmus muss das Trainieren des Algorithmus erfolgen, um unbekannte Schrauben und Bolzen erkennen zu können. In nebenstehender Abbildung ist der erste Schritt des Bin-Picking-Prozesses, die Objekterkennung dargestellt. Die Einzelteile verschiedener Dimensionen werden erkannt und in die Kategorien a bis f eingeteilt. Dies muss auch für die unbekannten Einzelteile erreicht werden.

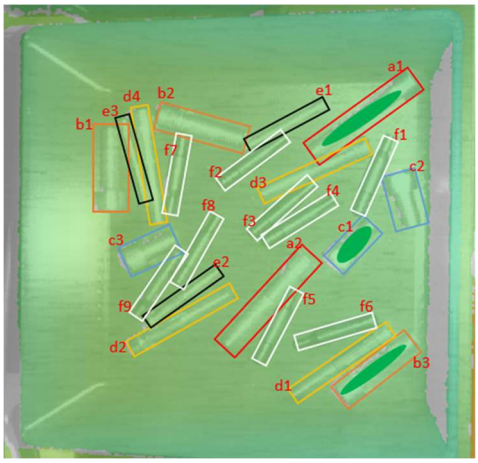


Abbildung 29: Visualisierung der Objekterkennung

Für die Montage der Kamerahalterung an der Bin-Picking-Station haben die Studenten aus einem höheren Jahrgang im Rahmen ihrer Studienarbeit einen gleichen Ausschnitt in der Arbeitsplatte gesägt, wie der, der in dieser Studienarbeit in Kapitel 5.3 erfolgt ist. Da die Kamerahalterung nun versetzt wurde, fehlt ein kleiner Teil der Arbeitsplatte. Diese kann aus rein kosmetischen Gründen aufgefüllt werden. Der gesägte Ausschnitt ist nicht wiederverwertbar, daher könnte etwas mit dem 3D-Drucker realisiert werden.



Abbildung 30: Ausschnitt in der Arbeitsplatte

Zusammenfassend lässt sich sagen, dass in dieser Studienarbeit der gesamte theoretische Teil des Projektes durchgeführt wurde und mit Start der nächsten Theoriephase an der Hochschule am 11.04.23 der praktische Teil umgesetzt werden kann.

9. Literaturverzeichnis

[1] DHBW Ravensburg: Produktionsnetzwerk Fallenbrunnen. Online verfügbar unter https://www.ravensburg.dhbw.de/forschung-transfer/kompetenzzentren/zentrum-fuer-digitalisierung-in-produktion-und-produktentwicklung#produktionsnetzwerk-fallenbrunnen, zuletzt geprüft am 09.01.2023.

[2] B+M Blumenbecker GmbH: Bin Picking. Automatische Objekterkennung und -positionierung. Online verfügbar unter https://www.blumenbecker.com/de/industrie-automation/industrierobotik/bin-picking, zuletzt geprüft am 12.01.2023.

[3] TeDo Verlag GmbH: Bin-Picking-System mit KI-Vision. Online verfügbar unter https://robotik-produktion.de/allgemein/bin-picking-system-mit-ki-vision-2/, zuletzt geprüft am 12.01.2023.

[4] Bosch-Rexroth AG: Mechanik-Grundelemente. Online verfügbar unter https://esd.equipment/de/amfile/file/download/file/132/product/25909/, zuletzt geprüft am 09.01.2023.

[5] Fraunhofer IKS: Mensch-Roboter-Kollaboration: Wie wir in Zukunft mit dem Roboter zusammenarbeiten werden. Online verfügbar unter https://www.iks.fraunhofer.de/de/themen/industrie-40/mensch-roboter-kollaboration.html, zuletzt geprüft am 06.01.2023.

[6] Universal Robots: Universal Robots e-Series Benutzerhandbuch. UR3e. Online verfügbar unter https://s3-eu-west-1.amazonaws.com/ur-support-site/43878/UR3e_User_Manual_de_Global.pdf, zuletzt geprüft am 07.01.2023.

[7] olympustechnologies (Hg.): TeachPendant. Online verfügbar unter https://olympustechnologies.co.uk/wp-content/uploads/2018/08/TeachPendant-e1535039100750-1024x676.jpg, zuletzt geprüft am 07.01.2023.

[8] Optonic GmbH (2022): Handbuch mikado control. Version 2.1.

[9] Huawei Technologies Co., Ltd (2023): Artificial Intelligence Technology. 1st ed. 2023. Singapore: Springer Nature Singapore; Imprint Springer.

[10] Awad, Mariette; Khanna, Rahul (2015): Efficient learning machines. Theories, concepts, and applications for engineers and system designers. New York, NY: Apress Springer Science+Business Media (Books for professionals by professionals).

[11] Bosch-Rexroth AG: MTpro Online Designer. Online verfügbar unter https://www.boschrexroth.com/en/pl/products/product-groups/assembly-technology/topics/engineering-software-mtpro/mtpro-online-designer/, zuletzt geprüft am 13.01.2023.

[12] Coval Vakuumtechnik GmbH: Venturi-Vakuumpumpen. Online verfügbar unter https://www.coval-germany.com/Vakuumtechnik/Leitfaden-f%C3%BCr-das-Greifen-mit-Vakuum/Verfahren-zur-Vakuumerzeugung/, zuletzt geprüft am 12.01.2023.

[13] IDS Imaging Development Systems GmbH: Ensenso Kalibrierplatte. Online verfügbar unter https://de.ids-imaging.com/store/ensenso-calibration-pattern-390-mm-grid-48-mm.html, zuletzt geprüft am 13.01.2023.