

# ERWEITERUNG EINES KI-GESTÜTZTEN ASSISTENZSYSTEMS ZUR OPTIMIERUNG VON LASERSCHNEIDPARAMETERN FÜR EDELSTAHLBLECHE

## Projektarbeit T2000

des Studienganges Elektrotechnik  
Fachrichtung Automation  
an der Dualen Hochschule Baden-Württemberg  
Standort Stuttgart

**Tudor Lupsa**

08.09.2025

<b>Bearbeitungszeitraum</b>	02.06.25 - 08.09.25
<b>Matrikelnummer, Kurs</b>	1491114, TEL23GR3
<b>Dualer Partner</b>	TRUMPF SE+Co.KG, Ditzingen
<b>Betreuer des Dualen Partners</b>	Manuel Geiger, M.Sc

## **Sperrvermerk**

Die vorliegende Projektarbeit beinhaltet interne und vertrauliche Informationen der Firma TRUMPF SE + Co. KG. Die Weitergabe des Inhalts, der Arbeit im Gesamten oder in Teilen, sowie Anfertigen von Kopien oder Abschriften, auch in digitaler Form, sind grundsätzlich untersagt. Ausnahmen bedürfen der schriftlichen Genehmigung durch Herrn Manuel Geiger, Betreuer dieser Projektarbeit bei TRUMPF SE + Co. KG Ditzingen.

Dieser Sperrvermerk gilt zum 12. Januar 2026.

## **Eidesstattliche Erklärung**

Hiermit versichere ich, die vorliegende Projektarbeit selbstständig und nur unter Verwendung der von mir angegebenen Quellen und Hilfsmittel verfasst zu haben. Sowohl inhaltlich als auch wörtlich entnommene Inhalte wurden als solche kenntlich gemacht. Die Arbeit hat in dieser oder vergleichbarer Form noch keinem anderem Prüfungsgremium vorgelegen.

Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_

## Kurzreferat

Diese Arbeit erweitert ein KI-gestütztes Assistenzsystem für das Laserschneiden von Baustahl auf Edelstahl. Ziel ist eine zuverlässige Schätzung der Grathöhe und die Vorbereitung einer belastbaren Rauheitsbewertung. Dazu wurden Schnittabrißgrenzen systematisch bestimmt, neue Bild- und 3D-Daten in der Messzelle aufgenommen und die Messmethodik für Edelstahl angepasst. Ein kompaktes, VGG-ähnliches **CNNs!** (**CNNs!**) wurde mit Edelstahldaten neu trainiert. Die Ergebnisse zeigen eine deutlich verbesserte Gratschätzung mit geringerem Dickenbias, besonders bei 10 mm und 15 mm. Für die Rauheit ist die Datengrundlage noch zu klein, die Analyse bleibt jedoch konsistent. Die Arbeit schafft die Basis für eine erweiterte Datenerfassung und ein künftiges gemeinsames Modell über Werkstoffe hinweg.

## Abstract

This thesis extends an AI-assisted cutting assistant from mild steel to stainless steel. The goal is reliable burr prediction and the preparation of a robust roughness assessment. Cut-off limits were determined systematically, new image and 3D data were acquired in the measurement cell, and the measurement procedure was adapted to stainless steel. A compact VGG-like **CNNs!** was retrained on stainless data. Results show clearly improved burr prediction with reduced thickness-dependent bias, especially for 10 mm and 15 mm sheets. The dataset for roughness modeling is still limited, although the current analysis remains consistent. The work lays the groundwork for extended data collection and for a unified model across materials.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>VI</b>
<b>Abbildungsverzeichnis</b>	<b>VII</b>
<b>Tabellenverzeichnis</b>	<b>VIII</b>
<b>Quellcodeverzeichnis</b>	<b>IX</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Zielsetzung . . . . .	1
1.2 Vorgehensweise . . . . .	1
<b>2 Grundlagen</b>	<b>3</b>
2.1 Laserlichtschneiden . . . . .	3
2.2 Grat und Rauheit eines Blechteils . . . . .	4
2.3 Convolutional Neural Networks (CNNs) . . . . .	5
2.3.1 Faltungsblöcke . . . . .	6
2.3.2 Aktivierungsfunktion . . . . .	7
2.3.3 Pooling und Downampling (mit Global Average Pooling) . . . . .	7
<b>3 Stand der Technik</b>	<b>8</b>
3.1 KI-gestütztes Laserschneiden – Cutting Assistant . . . . .	8
3.2 3D-Kantenmesssystem: Keyence LJ-X8060 . . . . .	9
<b>4 Datengenerierung von Edelstahldatensätzen</b>	<b>11</b>
4.1 Bestimmung der Schnittabrißgrenzen und Erstellung des Datensatzes . . . . .	11
4.2 Messmethodik und Datenerfassung in der Messzelle . . . . .	12
4.2.1 Anpassung der Handscanner Einstellungen für Edelstahl . . . . .	16
<b>5 Optimierung der Vektorberechnung aus der 3D-Punktwolken</b>	<b>19</b>
5.1 Gratverlaufsanalyse bei Edelstahl . . . . .	19
5.1.1 Optimierung der Ausreißererkennung . . . . .	20
5.1.2 Optimierung der Interpolation . . . . .	21
5.2 Rauheitsanalyse bei Edelstahl . . . . .	22
5.3 Validierung der Messoptimierungen . . . . .	23
5.3.1 Validierung der verbesserten Ausreißererkennung im Burr-Verlauf . . . . .	23
5.3.2 Validierung der bestehenden Rauheitsanalyse an den neuen Edelsstahldaten	24
<b>6 Qualitätsschätzungsmodell für Edelstahl</b>	<b>25</b>
6.1 Bewertung des bestehenden Modells . . . . .	26
6.2 Training für Edelstahldatensatz . . . . .	27
<b>7 Reflexion und Ausblick</b>	<b>28</b>
7.1 Reflexion . . . . .	28
7.2 Ausblick . . . . .	28

<b>Anhang</b>	<b>30</b>
<b>Literaturverzeichnis</b>	<b>33</b>

## **Abkürzungsverzeichnis**

**RCP** Rapid Control Prototyping

**IMU** Inertial Measurement Unit, deutsch: Inertialmesseinheit

# Abbildungsverzeichnis

1	Schema zur Bestimmung der Rauheit . . . . .	5
2	Schematischer Aufbau eines CNN . . . . .	6
3	Beispiele zur Plausibilitätsprüfung durch „Iron-hunter“: links eine gültige Aufnahme der Schnittkante, rechts eine Fehlaufnahme mit verdeckender Hand, die verworfen wird. . . . .	8
4	Prozesskette des Cutting Assistant . . . . .	9
5	Lasertriangulation zur Profilaufnahme. ( <b>SensorInstruments _LLASLT_ Manual _2023</b> ) . . . . .	9
6	Messaufbau und Beispelpunktwolke der Schnittkante für den Gratverlauf. . . . .	10
7	Handscanner (1), ID-Lesegerät (2), Keyence 3D-Messsystem (3) in der Messzelle . . . . .	13
8	Smartphone (4) und Industriekamera (5) in der Messzelle . . . . .	14
9	Vergleich zwischen vorhergesagten und tatsächlichen Werten für den Grat (Burr). . . . .	15
10	Vergleich zwischen vorhergesagten und tatsächlichen Werten für die Rauheit (Roughness). . . . .	15
11	Dunkle Belichtung für die Rauheitsschätzung (i, erstes Bild links), zwei helle Belichtungen für die Gratschätzung (ii, mittlere Bilder) und das daraus generierte HDR-Bild (rechts). . . . .	16
12	Handcanner Bild für die Gratschätzung (Baustahl) . . . . .	17
13	Handcanner Bild für die Gratschätzung (Edelstahl) . . . . .	17
14	Handscanner Bild für die Rauheitsschätzung (Baustahl) . . . . .	17
15	Handscanner Bild für die Rauheitsschätzung (Edelstahl) . . . . .	17
16	Beispiel-Proben-ID eines Blechstücks aus den Experimentalplänen . . . . .	18
17	Beispielhafter Verlauf der Rauheitskennzahl $R_a$ entlang einer Schnittkante im ersten Abschnitt mit der Breite 0,3 mm. . . . .	22
18	Ausreißererkennung im Burr-Verlauf . . . . .	23
19	Schematische Darstellung der VGG-ähnlichen Architektur zur Regression der Prozessparameter aus einem Schnittkantenbild. Links der Bildeingang, gefolgt von einem Eingangsblock mit $7 \times 7$ -Faltung und vier Blöcken mit jeweils drei $3 \times 3$ -Faltungen. Die dritte Faltung eines Blocks reduziert die räumliche Auflösung. Global Average Pooling verdichtet die 256 Feature-Maps zu einem 256-dimensionalen Vektor, der den Ausgabe, dem Burr-Wert, speist. ( <b>Tatzel-2021</b> ) . . . . .	25
20	Burr: wahre gegen vorhergesagte Werte für das <b>ST!</b> ( <b>ST!</b> )-Modell. Die Farben kennzeichnen 5 mm, 10 mm und 15 mm. Die gestrichelte Diagonale markiert $y=x$ . . . . .	26
21	Burr: wahre gegen vorhergesagte Werte für das <b>SS!</b> ( <b>SS!</b> )-Modell. Die Farben kennzeichnen 5 mm, 10 mm und 15 mm. Die gestrichelte Diagonale markiert $y=x$ . . . . .	27

## **Tabellenverzeichnis**

1	Formelzeichen für Rauheit und Grat . . . . .	4
2	Formelzeichen zu Gl. (2.1) (Faltung) und Gl. (2.2) (Global Average Pooling) . . . . .	7

# Quellcodeverzeichnis

5.1	Pipeline-Parameter Outlier Correction (Edelstahl, Burr-Verlauf) (zu Algorithm 1)	21
5.2	Pipeline-Parameter Outlier Correction (Baustahl, Burr-Verlauf) (zu Algorithm 1)	21
7.1	Werkstoffabhängiges Routing der Handscanner-Setups . . . . .	30
7.2	Interpolation begrenzter Lücken in 1D-Profilen . . . . .	30
7.3	Achsweise Interpolation von NaNs in Matrizen . . . . .	31
7.4	Interpolation von X-, Y- und Z-Ebenen . . . . .	32
7.5	Kapselnde 1D-Pipeline: begrenzte Interpolation mit optionaler Glättung . . . . .	32
7.6	Schließen kurzer Täler (Valleys) via begrenzter Interpolation . . . . .	32

# 1 Einführung

Dieses Kapitel gibt eine Einführung in die Thematik der Projektarbeit. Es werden die Zielsetzung und die geplante Vorgehensweise beschrieben.

## 1.1 Zielsetzung

Das Ziel dieser Projektarbeit ist es eine Versuchsplattform für die Automatisierungstechnik zu entwickeln bei der typische Regelungstechnische Methoden experimentelle untersucht werden und anschließend auf die Zielpfaltform übertragen werden können. Hierfür soll ein Rapid Control Prototyping (RCP)-System verwendet werden, um die Regelungsalgorithmen in Echtzeit auf der Zielpfaltform auszuführen, indem bereits vorhandene Hardware und Software Bausteine genutzt werden. Das RCP ist ein Verfahren mit dem zu regelnde Systeme schnell und flexibel entwickelt und getestet werden können. Hierbei ist es nicht notwendig manuelle Implementierung in Programmiersprachen für die Zielhardware zu erstellen, sondern es kann direkt von einer grafischen Simulationsumgebung wie z.B. *Matlab/Simulink* auf die Zielhardware übertragen werden mithilfe von einer automatischen Codegenerierung. Dies ermöglicht eine schnelle Iteration und Anpassung der Regelungsalgorithmen, was besonders in der Entwicklungsphase von Vorteil ist, da somti schnell und kosteneffizient Prototypen erstellt und getestet werden können. Die Versuchsplattform soll einen durchgängigen Prozess von der Modellerstellung über die Simulation bis zur Echtzeitausführung auf der Zielhardware abbilden und dabei die experimentelle Parametrierung, Optimierung sowie die Beobachtung relevanter Signale und Messgrößen ermöglichen, indem eine Kopplung zwischen Entwicklungsrechner und Zielpfaltform zur Signal und Parameterkommunikation genutzt wird (Hoyos-Gutiérrez et al. 2023).

## 1.2 Vorgehensweise

Die Projekarbet setzt auf eine Einarbeitung in das Rapid Control Prototyping, um die theoretischen Grundlagen, die verwendeten Begriffe und typische Prozessabläufe einzuordnen und daraus geeignete Vorgehensprinzipien abzuleiten. Aufbauend auf diesem Kenntnisstand werden Anforderungen an die zu entwickelnde Versuchsplattform definiert. Dabei werden funktionale Anforderungen beschrieben, die sich aus den geplanten Experimenten ergeben, sowie nicht funktionale Anforderungen festgelegt, die unter anderem die Umsetzbarkeit, Erweiterbarkeit und Randbedingungen der Nutzung betreffen.

Nachdem die Randbedingungen gesetzt wurden, erfolgt die Auswahl geeigneter Hardware und Software Werkzeuge für die Umsetzung der Versuchsplattform. Ziel ist es, eine Kombination

aus Hardware und Software zu identifizieren, die eine effiziente Entwicklung, Simulation und Echtzeitausführung der Regelungsalgorithmen ermöglicht.

Parallel zum physischen Aufbau wird die Versuchsplattform in Matlab und Simulink modelliert, um ein ausführbares Systemmodell für die Simulation bereitzustellen. In diesem Modell werden Sensorik, Aktorik und die wesentlichen dynamischen Eigenschaften des Prozesses abgebildet, sodass Algorithmen vor der Implementierung auf der realen Plattform unter definierten Bedingungen untersucht werden können. Abschließend wird der Ansatz des Rapid Control Prototyping exemplarisch angewendet, indem ausgewählte Funktionen in Simulink entworfen, simuliert und anschließend mittels automatischer Codegenerierung auf die Zielhardware übertragen werden. Die Implementierung wird getestet und validiert, um die Funktionalität und Leistungsfähigkeit der entwickelten Versuchsplattform zu gewährleisten.

Für die Umsetzung der RCP-Methodik wird in der Projektarbeit ein Anwendungssezenario der "Lageschätzung" genutzt. Hierbei soll eine Inertial Measurement Unit, deutsch: Inertialmesseinheit (IMU) simuliert werden und gegebenenfalls aus der ausgewählten Hardware eine reale IMU ausgelesen werden. Anschließend soll die Lage des Systems in Form von Quaterionen/Eulerwinkel geschätzt werden ([HIER VERWEIS KAPITEL 2.4](#)). Die Lageschätzung soll auf Grundlage einer Sensorfusion funktionieren, indem Beschleunigungs- und Gyroskopdaten kombiniert werden. Hierfür sollen verschiedene Algorithmen implementiert und getestet werden, um die Genauigkeit und Robustheit der Lageschätzung zu bewerten. Ziel ist es, eine zuverlässige Methode zur Bestimmung der Systemlage zu entwickeln, die in Echtzeit auf der RCP-Plattform ausgeführt werden kann.

## 2 Grundlagen

Für die vorliegenden Arbeit sind Kenntnisse in den Bereichen Laserlichtschneiden, Blechverarbeitung, Bildverarbeitung und **CNNs!** erforderlich. In diesem Kapitel werden die wichtigsten Grundlagen erläutert.

### 2.1 Laserlichtschneiden

Beim Laserlichtschneiden wird ein fokussierter Laserstrahl auf die Werkstückoberfläche gerichtet. Das Material schmilzt lokal auf und wird mit einem zugeführten Gas aus dem Schnittspalt ausgeblasen. So entsteht eine schmale, definierte Trennfuge. Die grundlegenden Begriffe und Kenngrößen sind in (**ISO11145-2018**) beschrieben.

In der Produktion kommen vor allem Faser- und Scheibenlaser zum Einsatz. CO<sub>2</sub>-Laser werden seltener verwendet. Faser- und Scheibenlaser erlauben hohe Energiedichten im Fokus und damit hohe Schnittgeschwindigkeiten, besonders bei dünnen bis mittleren Blechdicken. Die Wahl des Prozessgases beeinflusst Schnittqualität und Kantenfarbe. Beim Schneiden mit Stickstoff bildet sich kaum Oxid an der Schnittkante, die Kante bleibt hell und metallisch. Das ist vorteilhaft, weil in der Regel weniger Nacharbeit erforderlich ist, wie z.B. dem Nachbürsten der Kante, so dass wieder eine metallische Oberfläche zu sehen ist. Beim Schneiden mit Sauerstoff reagiert das Material mit dem Gas und somit entsteht eine dunkle Oxidschicht, die die Kante verfärbt. Der Prozess kann zwar schneller sein, führt aber demnach zu zusätzlicher Nacharbeit, wenn eine helle und einwandfreie Kante gefordert ist.

Die wichtigsten Einstellgrößen, welche in dieser Arbeit betrachtet werden, sind die Laserleistung, Schnittgeschwindigkeit, Fokuslage und Gasdruck. Eine höhere Leistung erlaubt höhere Geschwindigkeit, bis die Qualitätsgrenzen erreicht sind. Eine unpassende Fokuslage führt schnell zu Grat oder unvollständigem Materialaustrag. Der Gasdruck muss so gewählt werden, dass die Schmelze zuverlässig aus dem Schnittspalt entfernt wird, ohne die Kante aufzurauen. Für die Bewertung der Schnittqualität sind u.a. Grat und Rauheit relevant. ISO 9013 fasst hierfür Klassen und Toleranzen zusammen (**ISO9013-2017**).

Für Edelstahl gelten gegenüber Baustahl angepasste Einstellungen. Gründe sind unter anderem eine höhere Reflexion und eine geringere Wärmeabfuhr. In der Praxis bedeutet das, dass die Leistung und die Fokuslage sorgfältig abgestimmt werden müssen. Die Geschwindigkeit muss demnach passend gewählt werden. Als Prozessgas wird meistens Stickstoff eingesetzt, um eine helle Kante und geringen Grat zu erreichen (**TRUMPF \_ FusionCutting**).

## 2.2 Grat und Rauheit eines Blechteils

Für das Verständis der vorliegenden Arbeit sind Grat und Rauheit wichtige Kenngrößen der Schnittqualität. Dies wird in diesem Abschnitt erläutert.

Grat und Rauheit beschreiben die Qualität der Schnittkante. Der Grat ist ein aufgeworfener Materialrand an der Unterkante und entsteht durch unvollständigen Schmelzaustrag oder Anhaftungen von Schmelze im Schnittspalt. Die Rauheit beschreibt die feinen Höhenabweichungen der Schnittfläche entlang eines Profils  $Z(x)$ .

Die in Tabelle 1 zusammengefassten Formelzeichen werden in diesem Dokument verwendet.

Tab. 1: Formelzeichen für Rauheit und Grat

Zeichen	Bedeutung	Einheit
$Z(x)$	gefiltertes Profil entlang der Auswertelänge $L$	$\mu\text{m}$
$Z_i$	diskreter Profilwert an Position $x_i$	$\mu\text{m}$
$L$	Auswertelänge	mm
$n$	Anzahl der Stützstellen in $L$	–
$X_{s,j}$	$j$ -te Teilstrecke innerhalb von $L$	mm
$R_a$	arithmetischer Mittenrauwert	$\mu\text{m}$
$R_z$	mittlere Rautiefe aus fünf Teilstrecken	$\mu\text{m}$
$P_j^{\max}, P_j^{\min}$	höchster bzw. tiefster Punkt in $X_{s,j}$	$\mu\text{m}$
$h_b$	Grathöhe an der unteren Schnittkante	$\mu\text{m}$

Die Berechnung der Kenngrößen folgt den nachfolgenden Rechenregeln. Der arithmetische Mittendrauwert  $R_a$  ist das Mittel der Beträge der Profilabweichung über die Auswertelänge  $L$  (**MitutoyoQuickGuide**).

$$R_a = \frac{1}{L} \int_0^L |Z(x)| \, dx$$

und in diskreter Form mit  $n$  Stützstellen  $Z_i$ :

$$R_a = \frac{1}{n} \sum_{i=1}^n |Z_i|.$$

Die mittlere Rautiefe  $R_z$  wird über fünf Teilstrecken  $X_{s,1}$  bis  $X_{s,5}$  bestimmt (**KeyenceISO4287**). In jeder Teilstrecke wird die Differenz zwischen höchstem und tiefstem Profilpunkt gebildet.  $R_z$  ist das arithmetische Mittel dieser fünf Differenzen:

$$R_z = \frac{1}{5} \sum_{j=1}^5 (P_j^{\max} - P_j^{\min}).$$

Die Grathöhe  $h_b$  wird als maximale positive Auslenkung des Profils im Randbereich der unteren Schnittkante bestimmt. Grundlage ist dasselbe Profil  $Z(x)$  oder ein aus einer Punktfolge abgeleitetes Profil senkrecht zur Kante. Neben  $h_b$  können Breite und Form des Grates angegeben werden. Die Profilwerte  $Z_i$  stammen aus einer 3D-Punktfolge oder aus einer bildbasierten Profilerfassung. Aus diesen Werten werden  $R_a$  und  $R_z$  berechnet. Die Grathöhe  $h_b$  wird im Kantenbereich aus demselben Profil ermittelt. Einen Überblick zum verwendeten 3D-Messsystem gibt Abschnitt 3.2.

Die Abbildung 1 zeigt das gefilterte Rauheitsprofil  $Z(x)$  über der Messstrecke  $X$ . Die rote Linie kennzeichnet  $R_a$ ,  $Z_i$  sind die diskreten Profilwerte und  $X_{s,j}$  die Teilstrecken für  $R_z$ . Die Grathöhe  $h_b$  wird an der unteren Schnittkante als größte positive Auslenkung im Kantenfenster bestimmt.

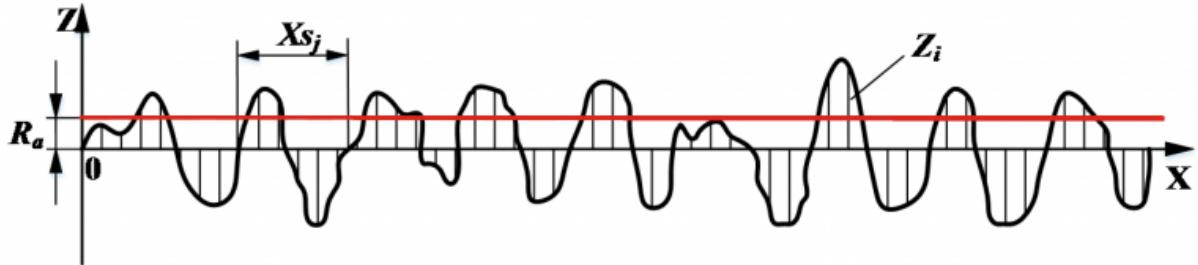


Abb. 1: Schema zur Bestimmung der Rauheit mit Profilwerten  $Z_i$ , Teilstrecken  $X_{s,j}$  und markiertem  $R_a$  (**Timesavers\_Rauheit\_Ra**).

## 2.3 Convolutional Neural Networks (CNNs)

Das vorliegende soll ein Grundverständnis zu **CNNs!** vermitteln, da dies die Grundlegende Architektur der neuronalen Netze ist und darauf der Cutting Assistant aufbaut (siehe Kapitel 3.1).

**CNNs!** sind neuronale Netze für Bilddaten. Sie bestehen aus wiederholten Blöcken aus Faltung (Convolution), nichtlinearer Aktivierung (z. B. **ReLU!** (**ReLU!**)) und Pooling. Die wichtigsten Bausteine sind in den Unterkapiteln 2.3.1, 2.3.2 und 2.3.3 näher beschrieben.

Die Abbildung 2 zeigt den Aufbau eines **CNNs!** am Beispiel eines Roboterbildes. Zu Beginn laufen kleine Filter (typisch  $3 \times 3$ ) über das Pixelgitter und reagieren auf lokale Muster, auch Kernel genannt. In den ersten Schichten entstehen somit Merkmalskarten für einfache Strukturen wie Kanten und Ecken, etwa an der Kontur des Kopfes oder entlang der Armsegmente des Roboters. Eine Aktivierungsfunktion unterdrückt schwache oder negative Antworten und erhöht den Kontrast zwischen relevanten und irrelevanten Bildbereichen. Pooling verdichtet die Merkmalskarten und macht die Darstellung unempfindlicher gegenüber kleinen Verschiebungen. Demnach ist die genaue Position der runden Augen oder Schrauben wird damit weniger wichtig als ihr Auftreten.

Mit zunehmender Tiefe kombinieren weitere Faltungen diese einfachen Muster zu komplexeren Teilen wie Gesicht, Gelenken oder ganzen Körpersegmenten. Am Ende wird die verdichtete Repräsentation *geflattet* und von vollverbundenen Schichten zu einer Ausgabe verdichtet, etwa zu Klassenwahrscheinlichkeiten („Roboter ja/nein“) oder zu kontinuierlichen Werten. Das Netzwerk lernt dabei sämtliche Filtergewichte gemeinsam, sodass frühe und späte Schichten aufeinander abgestimmt sind und eine konsistente Merkmalshierarchie vom Lokalen zum Globalen entsteht (**FischerPochwyt\_NeuronaleNetze\_2017**).

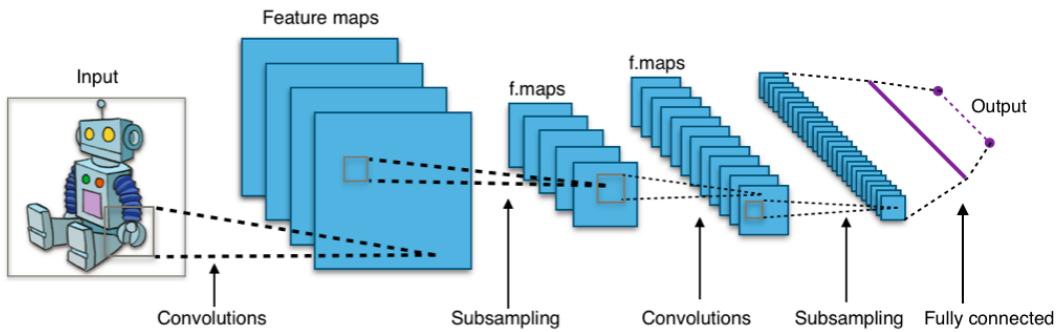


Abb. 2: Schematische Pipeline eines CNN am Beispiel eines Roboterbildes: Faltung extrahiert lokale Muster, Pooling verdichtet die Repräsentation, tiefere Stufen kombinieren Teile zu Objekten, die Entscheidung erfolgt in vollverbundenen Schichten (**FischerPochwyt\_NeuronaleNetze\_2017**).

Nach demselben Prinzip arbeitet der in dieser Arbeit weiterentwickelte „Cutting Assistent“ (siehe Kapitel 3.1). Frühe Filter reagieren auf Kanten und Texturwechsel in Schnittkantenbildern, Pooling sorgt für Robustheit gegen kleine Lageänderungen, und tiefere Schichten fassen wiederkehrende Muster wie Rifen, Rauheitsstrukturen oder Gratbildung zusammen. Die abschließenden Schichten liefern je nach Aufgabe eine Klassifikation oder Regressionswerte wie Rauheit oder Grathöhe.

Die nachfolgenden Unterkapitel erläutern die zuvor genannten zentralen Bausteine eines **CNNs!** im Detail.

### 2.3.1 Faltungsblöcke

Die 2D-Faltung berechnet an jeder Position  $(i, j)$  einen gewichteten Mittelwert der lokalen Nachbarschaft. Die Gleichung (2.1) definiert dies präzise. Der Kernel mit Kantenlänge  $k$  und Gewichten  $w_{u,v}$  wird zentriert über das Eingabebild  $x$  gelegt, die Offsets  $(u, v)$  laufen über das Fenster. Das Ergebnis ist der Ausgabewert  $y_{i,j}$ . Die in Tabelle 2 aufgeführten Symbole legen die Notation fest. Durch das gemeinsame Nutzen der Kernelgewichte über alle  $(i, j)$  sinkt die Parameterzahl und die Abbildung bleibt verschiebungsgleich. In praktischen Netzen ist die Faltung mehrkanalig, dies bedeutet dass mehrere Eingabekanäle gemeinsam gefaltet werden und zu mehreren Ausgabekanälen kombiniert werden.

Tab. 2: Formelzeichen zu Gl. (2.1) (Faltung) und Gl. (2.2) (Global Average Pooling).

Symbol	Bedeutung	Einheit/Bereich
$x_{i,j}$	Eingabewert an Position $(i, j)$	reell
$y_{i,j}$	Ausgabewert der Faltung an Position $(i, j)$	reell
$w_{u,v}$	Gewicht des $k \times k$ -Kerns an Offset $(u, v)$	reell
$k$	Kantenlänge des Faltungskerns	Pixel
$i, j$	Räumliche Indizes (Zeile, Spalte)	—
$u, v$	Kernelindizes	—
$H, W$	Höhe und Breite der Merkmalskarte für GAP	Pixel
$z_c$	GAP-Ausgabe für Kanal $c$	reell
$c$	Kanalindex	—

$$y_{i,j} = \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} w_{u,v} x_{i+u-\lfloor k/2 \rfloor, j+v-\lfloor k/2 \rfloor} \quad (2.1)$$

$$z_c = \frac{1}{HW} \sum_{i=1}^H \sum_{j=1}^W y_{i,j,c} \quad (2.2)$$

### 2.3.2 Aktivierungsfunktion

Die Faltung liefert lineare Antworten  $y_{i,j}$ , die im nächsten Schritt nichtlinear transformiert werden. Eine Aktivierungsfunktion  $\phi(\cdot)$  wird punktweise angewandt und macht die Abbildung modellierfähig, etwa durch ReLU, die negative Werte auf null setzt. Die ReLU-Aktivierung ist eine üblich genutzte Aktivierungsfunktion.

### 2.3.3 Pooling und Downsampling (mit Global Average Pooling)

Pooling fasst lokale Nachbarschaften zusammen und reduziert die räumliche Auflösung. Dadurch sinkt der Rechenaufwand und die Darstellung wird unempfindlicher gegenüber kleinen Verschiebungen. Am Ende des Merkmalsextraktors wird häufig ein *Global Average Pooling* eingesetzt. Die Gleichung (2.2) mittelt für jeden Kanal  $c$  alle Werte  $y_{i,j,c}$  über Höhe  $H$  und Breite  $W$  zu einem einzigen Skalar  $z_c$ . Die so entstehende Vektordarstellung ist kompakt, senkt die Parameterzahl des Ausgabekopfes und eignet sich für Klassifikation und Regression. Andere Pooling-Varianten wie Max- oder lokales Average-Pooling können in früheren Stufen eingesetzt werden, ohne die in Gleichung (2.2) definierte globale Verdichtung am Ende zu ersetzen. Das Max-Pooling behält pro lokalem Fenster den größten Wert und hebt starke Aktivierungen hervor und das Min-Pooling behält den kleinsten Wert und betont schwache Antworten oder dunkle Strukturen. Beide Varianten reduzieren die Zahl der Werte und wirken als nichtlineare Verdichtung.

### 3 Stand der Technik

Dieser Stand der Technik ordnet die Arbeit in den aktuellen Forschungsstand ein. Inhaltlich wird der zuerweiternde KI-gestützte Cutting Assistant beschrieben. Außerdem wird das 3D-Kantenmesssystem erläutert, mit dem die Trainingsdaten erfasst werden.

#### 3.1 KI-gestütztes Laserschneiden – Cutting Assistant

Der Cutting Assistant unterstützt die Qualitätssicherung und die Parametrierung beim Laserschneiden. Abbildung 4 zeigt den Gesamtablauf von der Datenerfassung bis zur rückgekoppelten Optimierung der Prozessparameter. Zunächst erfassst einer am Schneidsystem montierter Handscanner die Schnittkante und übergibt die Bilddaten automatisiert an die Auswertekette. Ein vorgeschaltetes CNN („Iron-hunter“, vgl. Abschnitt 2.3) prüft die Plausibilität der Aufnahme und verwirft Störbilder oder Fehlperspektiven. Beispiele für eine valide Kantenaufnahme und eine Fehlaufnahme sind in Abbildung 3 dargestellt.

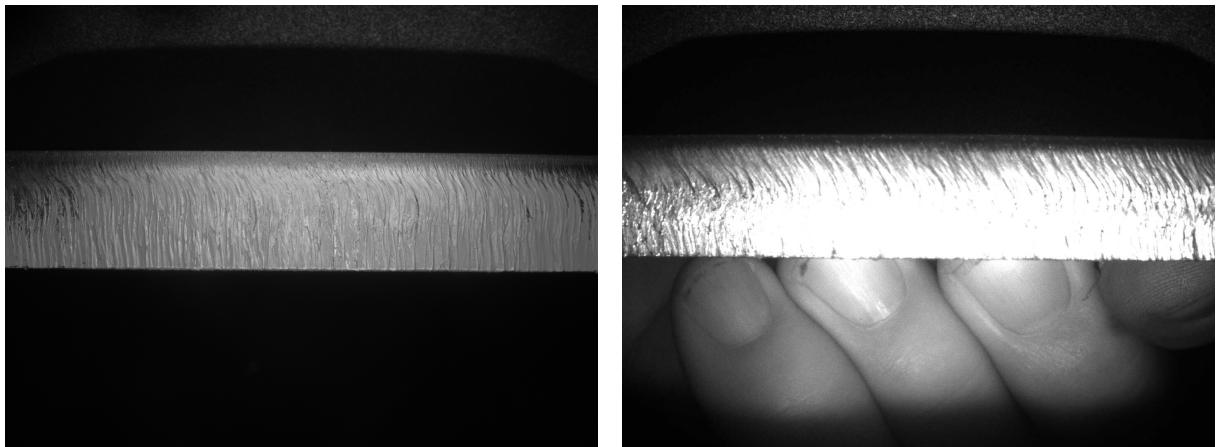


Abb. 3: Beispiele zur Plausibilitätsprüfung durch „Iron-hunter“: links eine gültige Aufnahme der Schnittkante, rechts eine Fehlaufnahme mit verdeckender Hand, die verworfen wird.

Nur valide Daten gelangen zur Detektion, wo ein zweites Netz eine Qualitätsschätzung durchführt und Merkmale wie Rauheit und Grat schätzt. Die Kennzahlen fließen in ein Optimierungsmodell ein, wobei Gasdruck, Fokuslage und Schneidgeschwindigkeit angepasst werden. Wie diese Schneidparameter die Schneidqualität beeinflussen wird im oben Grundlagenkapitel 2.1 näher erläutert.

Der Zyklus aus Aufnehmen, Bewerten und Anpassen wiederholt sich, bis ein Abbruchkriterium erfüllt ist, etwa wenn keine weitere Qualitätssteigerung zu erwarten ist oder die Spezifikation erreicht wurde. Das System ist für Baustahl validiert und wird im Rahmen dieser Arbeit auf Edelstahl erweitert, indem Datensätze ergänzt, Netzparameter feinjustiert und Schwellwerte an das Werkstoffverhalten angepasst werden.

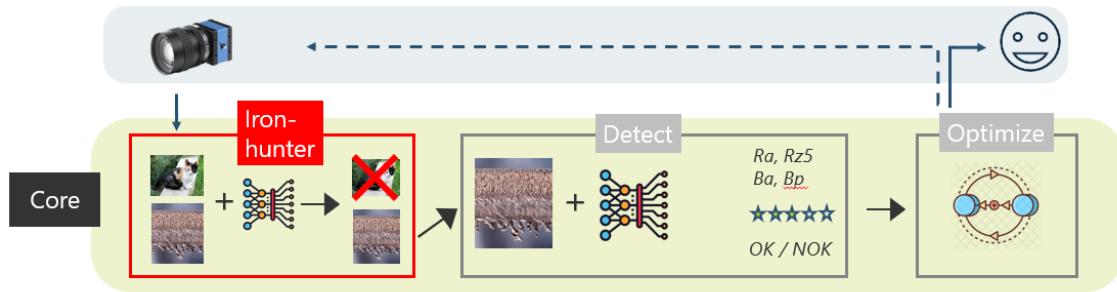


Abb. 4: Prozesskette des Cutting Assistant.

### 3.2 3D-Kantenmesssystem: Keyence LJ-X8060

In diesem Abschnitt wird die Funktionsweise des bereits eingesetzten dreidimensionalen Messsystems beschrieben, mit dem die Schnittkanten der Trainingsdaten für das KI-Modell erfasst wurden. Das System wurde zunächst für das Baustahlmodell verwendet und wird für die vorliegende Arbeit auf die Vermessung von Edelstahl angepasst. Die dafür notwendigen Optimierungen sind in Kapitel 5 beschrieben.

Abbildung 5 zeigt den Aufbau eines Sensors auf Basis der Lasertriangulation. Eine Sendeoptik projiziert eine Laserlinie auf die Oberfläche des Werkstücks. Versetzt zur Sendeeinheit erfasst eine Empfangsoptik das reflektierte Licht und bildet es auf einen zeilenförmigen Detektor ab. Die Position der Lichtlinie auf dem Detektor hängt von der Objektentfernung ab. Aus dieser Geometrie wird für jedes Zeilenprofil die Höhe des zugehörigen Oberflächenpunkts berechnet. Durch die Relativbewegung zwischen Sensor und Objekt entstehen aufeinanderfolgende Profile, die zu einer dreidimensionalen Punktwolke mit  $(x, y, z)$  zusammengeführt werden.

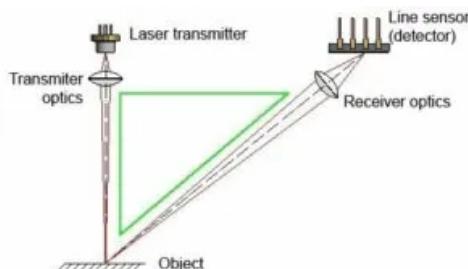


Abb. 5: Lasertriangulation zur Profilaufnahme. ([SensorInstruments\\_LLASLT\\_Manual\\_2023](#))

Abbildung 6 fasst die Erfassung mit dem Keyence LJ-X8060 zusammen. Links ist der schematische Messaufbau mit Koordinatensystem gezeigt. Die Laserlinie tastet die Kante quer zur Scanrichtung ab und die Bewegung erfolgt entlang  $x$ . Rechts ist eine registrierte dreidimensionale Punktwolke im Werkstückkoordinatensystem  $(x, y, z)$  dargestellt. Für die Auswertung werden die Daten durch eine Koordinatenrotation so ausgerichtet, dass die Kante mit der  $x$ -Achse und die Blechebene mit der  $xy$ -Ebene zusammenfällt.

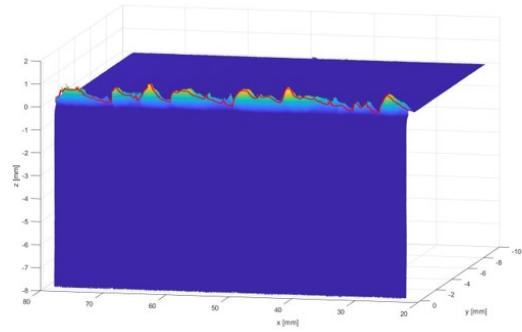
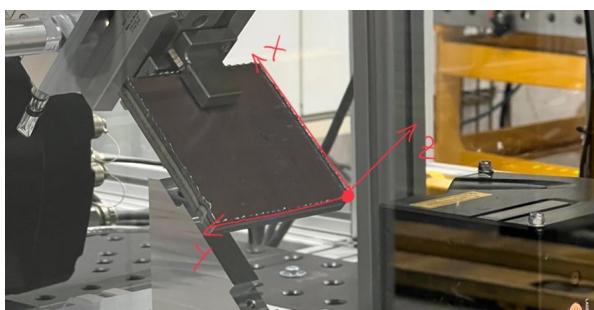


Abb. 6: Messaufbau und Beispieldatensatz der Schnittkante für den Gratverlauf.

Aus der erfassten dreidimensionalen Punktwolke wird ein zweidimensionaler Vektor abgeleitet, der den Gratverlauf im  $yz$ -Schnitt beschreibt. Da die Schnittkante schräg erfasst wird, richtet eine Koordinatenrotation die Profile auf das Werkstückkoordinatensystem aus. Die Kante fällt danach mit der  $x$ -Achse zusammen und die Blechebene mit der  $xy$ -Ebene. Die Rauheit der Schnittfläche wird als zweidimensionaler Vektor in der  $xy$ -Ebene beschrieben. Die Auswertung erfolgt entlang  $x$  und wird über  $y$  gemittelt. Eine ausführliche Beschreibung der Berechnung von Grat und Rauheit ist in Abschnitt 2.2 zu finden.

## 4 Datengenerierung von Edelstahldatensätzen

In dem folgendem Kapitel wird die Versuchsplanung zur Identifikation von Schnittabrißsgrenzen sowie die Messmethodik und Datenerfassung in der Messzelle für Edelstahlbleche beschrieben. Ziel ist es, den prozesssicheren Arbeitsbereich beim Laserschneiden von Edelstahl systematisch abzugrenzen und eine belastbare Datenbasis für die Erweiterung des KI-gestützten Laserschneidassistenten aufzubauen. Anschließend wird die Anpassung des Handscanner-Setups erläutert, um die Bildqualität für die bildgestützte Qualitätsschätzung zu optimieren.

### 4.1 Bestimmung der Schnittabrißsgrenzen und Erstellung des Datensatzes

Ziel dieses Kapitels ist die systematische Abgrenzung des prozesssicheren Arbeitsbereichs beim Laserschneiden von Edelstahlblechen sowie die präzise Identifikation der Parameterbereiche, in denen Schnittabrisse auftreten. Unter einem Schnittabriß wird im Folgenden ein Zustand verstanden, in dem das Werkstück infolge ungeeigneter Parameterkombinationen nicht vollständig getrennt wird, weil der Schnittspalt lokal verschweißt oder die Schmelzaustragung unzureichend ist. Die auf diese Weise gewonnenen Grenzwerte bilden die Grundlage für belastbare Parametrierungsempfehlungen und fließen zugleich in die Ausarbeitung strukturierter Versuchspläne zur Datengenerierung ein.

Die Experimente sind als zweidimensionale Rasterstudien ausgelegt, bei denen jeweils zwei der drei wesentlichen Prozessparameter, in dem Fall der Arbeit ist es die Fokuslage, die Schnittgeschwindigkeit und dem Gasdruck, variiert werden, während der dritte Parameter konstant gehalten wird. Die Laserleistung bleibt in diesen Studien konstant. Für jede untersuchte Parameterpaarung wird ein  $3 \times 3$ -Feld gefertigt, in dem eine Größe entlang der horizontalen und die andere entlang der vertikalen Richtung stufenweise verändert wird. Formal seien die Stufen der beiden variierten Parameter  $x \in \{x_1, x_2, x_3\}$  und  $y \in \{y_1, y_2, y_3\}$ . Der jeweils dritte Parameter  $z$  ist auf einem Referenzniveau  $z_0$  fixiert. Die Studie wird sequenziell für alle drei Kombinationen wiederholt, sodass das Prozessfenster in den relevanten Teirläumen konsistent erfasst wird. Die Wahl der Stufen erfolgt material- und dickenspezifisch, aus praxisüblichen Startwerten und internen Erfahrungswerten wird ein plausibler Arbeitsbereich abgeleitet.

Für jede Parameterkombination wird ein Viereck geschnitten. Ein Schnitt gilt als erfolgreich, wenn der Trennschnitt vollständig ist, weder Durchhang noch Wiederaufschmelzen im Schnittspalt beobachtet wird und der Schmelzaustrag kontinuierlich erfolgt. Die Beurteilung erfolgt unmittelbar an der Maschine sowie nachgelagert in der Messzelle durch optische Inspektion und Dokumentation der Schnittkante. Sobald ein erster Grenzbereich identifiziert ist, wird das umliegende Parametergebiet gezielt erkundet, bis ein stabiler Übergang zwischen den Zuständen

„Schnitt möglich“ und „Schnittabriß“ reproduzierbar nachgewiesen ist. Die so gewonnenen Grenzpunkte werden im jeweiligen Parameterraum verortet und bilden eine aus Messdaten abgeleitete Näherung des Prozessfensters je Blechdicke.

Auf Basis dieser Grenzanalysen erfolgt die Datengenerierung in Form strukturierter Experimentalpläne. Hierfür wurden insgesamt 17 Edelstahlblechtafeln ( $1\text{ m!} (\text{m!}) \times 2\text{ m!}$ ) in den Dicken 5 mm, 8 mm, 10 mm, 15 mm und 20 mm eingesetzt. Auf jeder Tafel wurden 128 quadratische Proben ( $100\text{ mm} \times 100\text{ mm}$ ) geschnitten, sodass ein Gesamtdatenumfang von 2 176 Bauteilen entstand. Die Schneidparameter wurden pro Bauteil innerhalb der vorab definierten, dicken-spezifischen Grenzen variiert und nach einem vorgegebenen Schema zufällig ausgewählt. Dieses Design stellt sicher, dass das Datenset sowohl hochwertige als auch ausdrücklich minderwertige Schneidergebnisse enthält, einschließlich Fehlschnitten und Parameterkombinationen nahe der Schnittabrißgrenze. Solche Negativbeispiele sind für die spätere Modellierung essentiell, um die Trennschärfe zwischen »prozesssicher« und »instabil« zu erhöhen und Fehlklassifikationen zu vermeiden.

Fehlschnitte wurden vollständig protokolliert. Auch wenn betroffene Proben in Einzelfällen nicht aus der Großtafel entnommen werden konnten, gingen diese Versuche mit eindeutiger Kennzeichnung in die Datenbank ein; damit ist bekannt, dass die jeweilige Parameterkombination für die gegebene Blechdicke kein akzeptables Schneidergebnis liefert. Sämtliche entnehmbaren Bauteile werden in der Messzelle vermessen, identifiziert und mit ihren Soll-/Ist-Parametern verknüpft. Die Datenmenge und der Versuchsplan wurden auf Basis der Erfahrungen mit dem bereits für Baustahl trainierten Modell gewählt, um eine ausreichende Abdeckung des Parameterraums und eine robuste Generalisierungsfähigkeit für Edelstahl sicherzustellen.

In Summe ermöglicht die kombinierte Vorgehensweise aus Grenzbestimmung und gezielter Datengenerierung sowohl die belastbare Identifikation der Schnittabrißgrenzen als auch den Aufbau einer ausgewogenen Datenbasis. Diese bildet die Voraussetzung für die Erweiterung und Validierung des KI-Modells, das künftig die Qualität von Edelstahlschnitten prädiktiv bewerten und prozesssichere Parameterbereiche verlässlich empfehlen soll.

## 4.2 Messmethodik und Datenerfassung in der Messzelle

Die Messzelle dient der reproduzierbaren Erfassung aller Messdaten zu den im Rahmen der Experimentalpläne geschnittenen Edelstahlbauteilen. Sie ist als sequenzieller Messprozess ausgelegt, in dem ein mehrachsiger KUKA-Industrieroboter die Proben zwischen den Stationen handhabt. Die Bauteile werden an der Startposition gestapelt bereitgestellt, vom Roboter mittels Vakuumgreifer aufgenommen und der ersten Station zugeführt. Dort erfolgt die automatisierte Probenidentifikation über einen aufgebrachten QR-Code (siehe ID-Lesegerät in Abbildung 7 und

Edelstahlprobe mit QR-Code in Abbildung 16). Die ermittelte Proben-ID wird mit den Metadaten aus den Experimentalplänen (z. B. Blechdicke, Soll-Parameter) verknüpft und dient in der Folge als Schlüssel für die Mess- und Auswertedaten.

Im Anschluss werden an einer Station hochauflöste Aufnahmen der ersten Schnittkante erfasst. Hierzu kommt ein Handscanner (siehe Abbildung 7) zum Einsatz, dessen Aufnahmeparameter, wie z.B. Arbeitsabstand, Belichtung und Auflösung, konstant gehalten werden. Diese Bilddaten bilden die Grundlage für die bildgestützte Qualitätsschätzung des KI-Systems. Ergänzend dazu wird die gleiche Schnittkante mit einem Keyence-3D-Messsystem (siehe Abbildung 7) dreidimensional vermessen, sodass eine 3D-Punktwolke des Kantenverlaufs entsteht. Aus dieser Punktwolke werden definierte Profilverläufe abgeleitet und geometrische Kenngrößen berechnet, die der Erfassung von Gratbildung und der Oberflächenrauheit dienen. Die Berechnung des Grates und der Rauheit ist im obigen Grundlagenkapitel 2.2 näher erläutert. Die genaue funktionsweise des 3D-Messsystems ist im folgenden Kapitel 3.2 ausgiebig erläutert. Die so gewonnenen Ist-Kenngrößen fungieren als Referenz für den späteren Abgleich mit der Bildqualitätsschätzung. In der Abbildung 7 sind die im obigen Abschnitt beschriebenen Komponenten der Messzelle während einer Messung.

In der folgenden Abbildung 7 sind die im obigen Abschnitt beschriebenen Komponenten der Messzelle während einer Messung dargestellt.

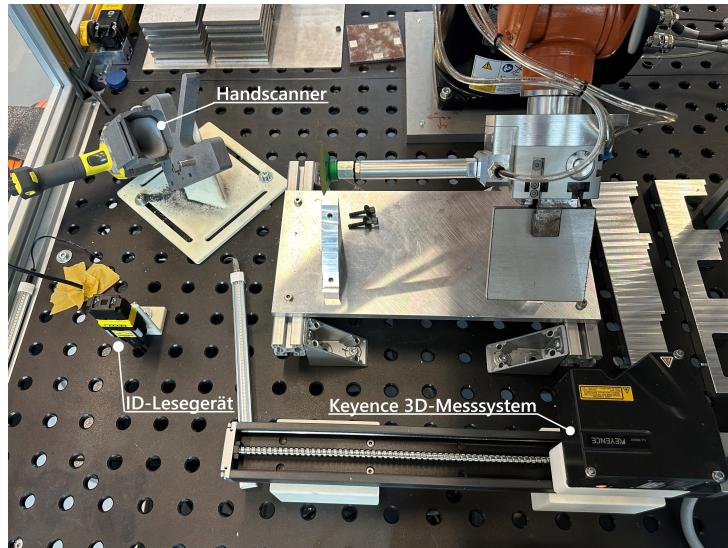


Abb. 7: Handscanner (1), ID-Lesegerät (2), Keyence 3D-Messsystem (3) in der Messzelle

Zur vollständigen Dokumentation werden die Schnittkanten zudem mit einer Industriekamera und einem stationären Smartphone-Setup unter verschiedenen Beleuchtungsbedingungen aufgenommen. Die Kombination aus unterschiedlichen Kameras und Beleuchtungen erhöht die Robustheit der visuellen Beurteilbarkeit und unterstützt die spätere manuelle Nachvollziehbarkeit von Auffälligkeiten. Die erfassten Messdaten des Werkstücks, sowie die daraus abgeleiteten Kenngrößen der Proben-ID zugeordnet und in die zentrale Datenbank überführt.

Die Abbildung 8 zeigt die zuvor beschriebenen Komponenten der Messzelle. Oben im Bild ist das Smartphone (4) mit LED-Ringlicht zu sehen, welches die Schnittkante aus einem 90° Winkel aufnimmt. Unten im Bild ist die Industriekamera (5) mit Auflichtbeleuchtung dargestellt, welche die Schnittkante ebenfalls aus einem 90° Winkel erfasst. Beide Kameras sind fest in der Messzelle montiert und werden automatisch durch den Roboter angesteuert.

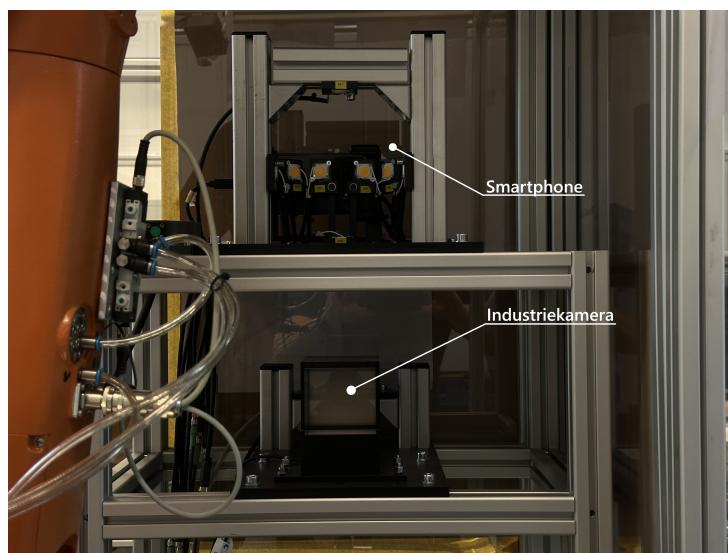


Abb. 8: Smartphone (4) und Industriekamera (5) in der Messzelle

Nach der Datenerfassung werden aus der 3D-Messung die tatsächlichen Kenngrößen der Schnittkante berechnet und den Ergebnissen der bildbasierten Qualitätsschätzung gegenübergestellt. Dieser Abgleich ermöglicht die Beurteilung der Übereinstimmung zwischen qualitativer, bildstützter Bewertung und quantitativer Geometriemessung. Die beschriebenen Messschritte werden für alle vier Schnittkanten jedes Bauteils identisch wiederholt. Abschließend legt der Roboter die vollständig vermessenen Proben an der Endstation geordnet ab.

Anbei ist ein Beispielhafter Vergleich der geschätzten Kenngrößen einer Schnittkante im Vergleich zu den gemessenen Kenngrößen in den Abbildungen 9 und 10 dargestellt. Es sind jeweils zwei Diagramme zu sehen, einer für den Gratwert und einer für die Rauheit. Auf den y-Achsen sind die geschätzten Kenngrößen und auf der x-Achse die gemessenen Kenngrößen aufgetragen. Idealerweise liegen alle Punkte auf der Diagonalen, was eine perfekte Übereinstimmung zwischen Schätzung und Messung bedeuten würde. In diesem Beispiel überschätzt das KI-Modell den Rauheitswert leicht, während der Gratwert überwiegend "gut" geschätzt wird. Mit einer leichten

Abweichung ist stets zu rechnen, da die bildbasierte Schätzung eine Näherung darstellt und nicht alle Details der 3D-Messung erfassen kann.

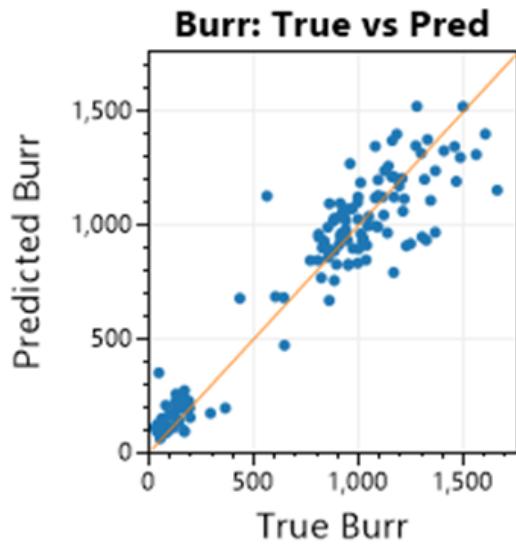


Abb. 9: Vergleich zwischen vorhergesagten und tatsächlichen Werten für den Grat (Burr).

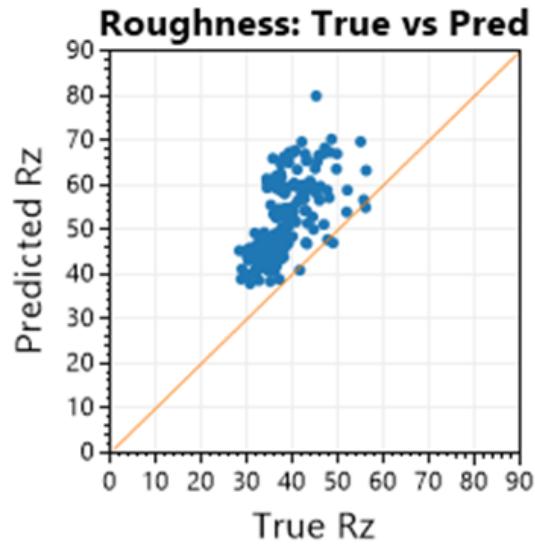


Abb. 10: Vergleich zwischen vorhergesagten und tatsächlichen Werten für die Rauheit (Roughness).

### 4.2.1 Anpassung der Handscanner Einstellungen für Edelstahl

Für die bildgestützte Qualitätsschätzung werden die mit dem Handscanner aufgenommenen Schnittkantenbilder als zentrale Eingangsgröße verwendet. Die bisher im Einsatz befindlichen Aufnahmeparameter waren für Baustahl optimiert. Baustahl weist im Vergleich zu Edelstahl eine geringere Oberflächenreflexion und eine tendenziell matte Erscheinung auf. Werden diese Einstellungen unverändert auf Edelstahl angewandt, führt die höhere Reflexion zu Bildartefakten und zu einer unzureichenden Abbildung der relevanten Mikrostruktur (**ZahnerStainlessReflect**). In der Folge würden Grate (*engl. Burr*) unterrepräsentiert und die Rauheit (*engl. Roughness*) potenziell verfälscht erscheinen. Da die Klassifikation der Bildqualität und die darauf basierende Schätzung von *Burr* und *Roughness* unmittelbar in die Parametrierung des Laserschneidprozesses zurückwirken, ist eine werkstoffabhängige Anpassung des Handscanner-Setups zwingend erforderlich.

Das Aufnahmeprotokoll sieht pro Schnittkante drei Bilder vor: (i) ein bewusst dunkler belichtetes Bild, das primär der Beurteilung der Schnittflächenrauheit dient, sowie (ii) zwei überbelichtete Bilder, die gemeinsam mit dem ersten zu einem HDR-Komposit zusammengeführt werden, um die Kontur und Ausprägung des Grats sicher zu erfassen (siehe beispielhafte Aufnahme in Abbildung 11).

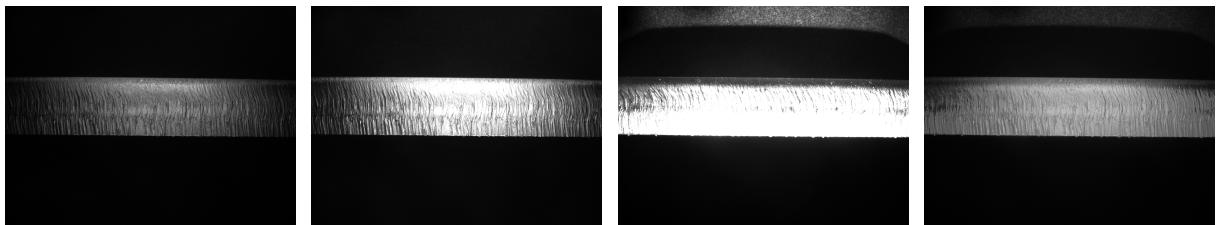


Abb. 11: Dunkle Belichtung für die Rauheitsschätzung (i, erstes Bild links), zwei helle Belichtungen für die Gratschätzung (ii, mittlere Bilder) und das daraus generierte HDR-Bild (rechts).

Die Kalibrierung der Belichtung erfolgt schrittweise. Zunächst wird die Belichtungszeit für das Rauheitsbild so eingestellt, dass die Textur der Schnittfläche ohne Sättigung und mit klarer Detailzeichnung sichtbar ist. Diese Entscheidung erfolgt in dieser Phase bewusst subjektiv, jedoch anhand vorab definierter visueller Kriterien, wie z.B. ausreichender Tonwertumfang und erkennbarer Strukturkontrast. Im Anschluss werden die Belichtungsparameter der beiden HDR-Bilder iterativ variiert, bis der Grat entlang der Schnittkante über den gesamten Bildbereich eindeutig detektierbar ist, ohne dass umliegende Bereiche vollständig verloren gehen. Da die HDR-Komposition durch die Eingangsbilder beeinflusst wird, erfolgt die Abstimmung der HDR-Belichtungen stets nach der Festlegung des Rauheitsbildes.

Zur Sicherstellung der Kompatibilität mit dem bestehenden KI-Modell wird die Anpassung an Referenzaufnahmen aus der bereits validierten Baustahlkonfiguration ausgerichtet. Praktisch bedeutet dies, dass eine Baustahlschnittkante mit den etablierten Baustahleinstellungen aufgenom-

men wird und die Edelstahlaufnahmen so justiert werden, dass die resultierenden Bildcharakteristika in qualitativer Hinsicht vergleichbar sind. Auf diese Weise wird gewährleistet, dass die Edelstahlbilder in das bestehende Modell eingebunden und mit den vorhandenen Trainings- und Bewertungsroutinen verarbeitet werden können.

Die Abbildungen 12, 13, 14 und 15 zeigen exemplarisch die finalen Handscanner Bilder, welche für die Qualitätsschätzung genutzt werden, für Edelstahl im Vergleich zu den bisherigen Bildern für die Qualitätsschätzung von Baustahl. In den Abbildungen ist jeweils die gleiche Schnittkante eines Blechteils aus den geschnittenen Experimentalplänen dargestellt mit einer Dicke von 15 mm, einmal ein Baustahlblech und einmal ein Edelstahlblech.

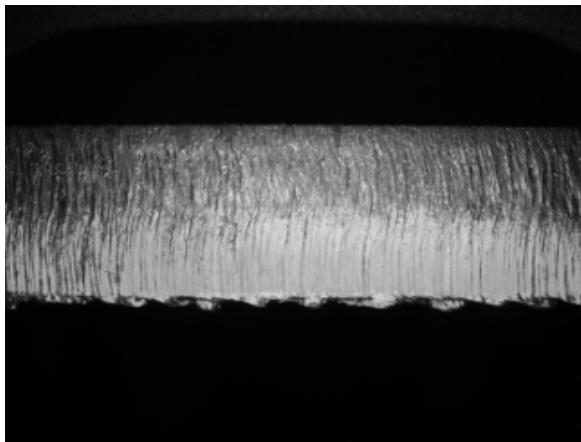


Abb. 12: Handcanner Bild für die Gratschätzung (Baustahl)

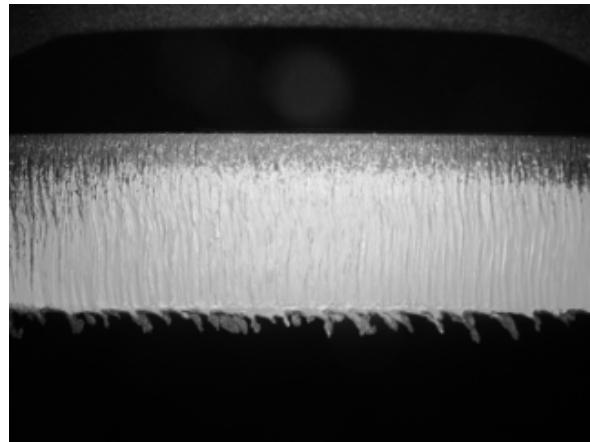


Abb. 13: Handcanner Bild für die Gratschätzung (Edelstahl)

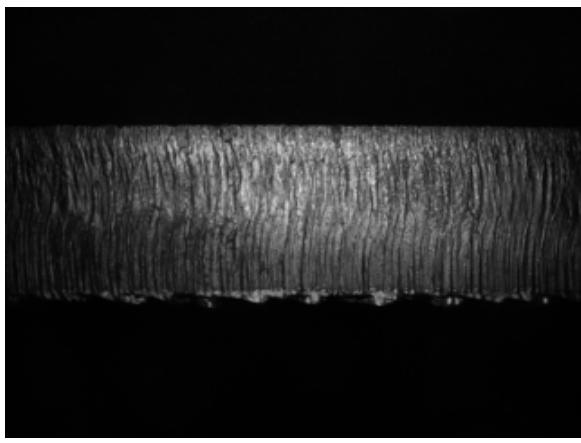


Abb. 14: Handscanner Bild für die Rauheitschätzung (Baustahl)

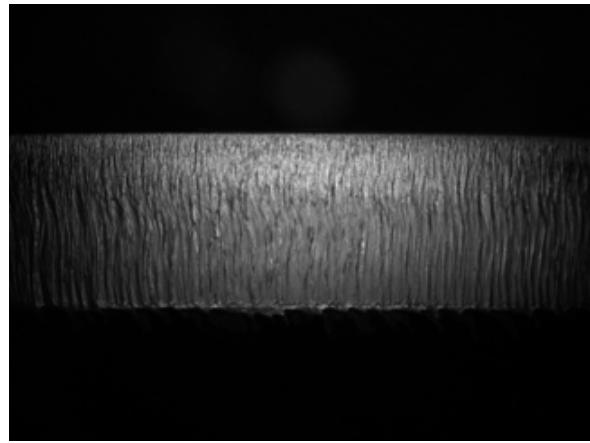


Abb. 15: Handscanner Bild für die Rauheitschätzung (Edelstahl)

Zur konsistenten Anwendung der angepassten Handscanner-Parameter wird das Messzellen-Skript so erweitert, dass das passende Setup automatisiert auf Basis der Bauteilbezeichnung gewählt wird. Die Benennung folgt dem Schema **Maschinename-Experimentalplannname-Materialnummer-Bauteildicke-Bauteilnummer**, z. B. A02280E0005-AiMuWrCjd0-3-050-0176. Die fol-

gende Abbildung 16 zeigt eine Beispiel-Proben-ID eines Blechstücks aus den Experimentalplänen mit den einzelnen Segmenten.

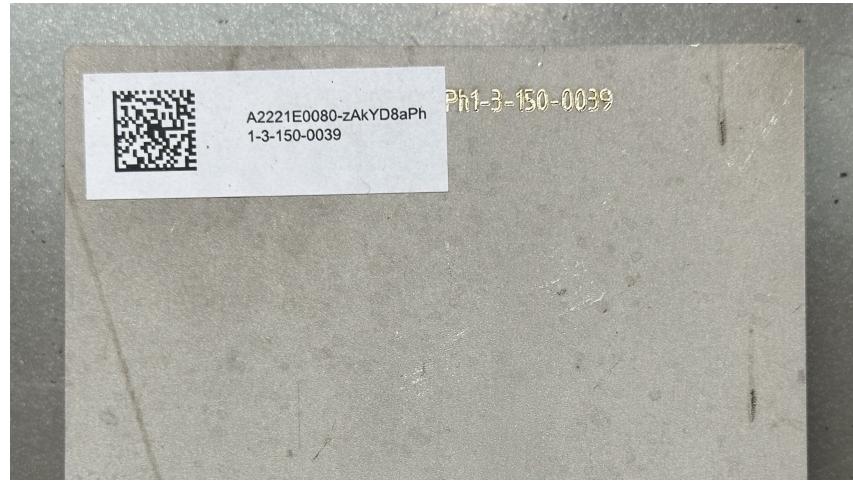


Abb. 16: Beispiel-Proben-ID eines Blechstücks aus den Experimentalplänen

Das Skript parst die Zeichenkette, prüft die Zahl nach dem zweiten Bindestrich und lädt abhängig davon die vordefinierten Handscanner-Einstellungen für den jeweiligen Werkstoff. Auf diese Weise wird sichergestellt, dass die für Edelstahl kalibrierten Belichtungen und Aufnahmeparameter reproduzierbar zur Anwendung kommen und die so erzeugten Bilder ohne systematische Verzerrungen in die Qualitätsmodellierung eingehen. Dies ist im folgendem C-Sharp Quellcode 7.1 dargestellt und im Messzellenskript implementiert.

## 5 Optimierung der Vektorberechnung aus der 3D-Punktwolken

Die mit dem Keyence-System aufgenommene 3D-Punktwolke der Schnittkante bildet die Grundlage für die geometrische Qualitätsauswertung. In der bestehenden Auswertepipeline wird die Punktwolke zunächst segmentiert und in ein lokales Kantenkoordinatensystem überführt. Anschließend erfolgt eine Projektion aus der dreidimensionalen Repräsentation in einen zweidimensionalen Profilverlauf, sodass ein 2D-Vektor entsteht, der den Verlauf des Schneidgrats entlang der Schnittkante beschreibt. Diese Vorgehensweise wurde ursprünglich für Baustahl entwickelt und auf dessen charakteristisch eher wellige, kontinuierliche Gratmorphologie abgestimmt.

Die nachfolgend beschriebenen Anpassungen betreffen ausschließlich den **Grat-/Burr**-Verlauf unterteilt in der Ausreißererkennung von Messdaten und der darauffolgenden Interpolation. Analyse ob die Rauheitsanalyse des Baustahl-Modells auch auf Edelstahl erfolgen kann wird in Kapitel 5.2 beschrieben.

### 5.1 Gratverlaufsanalyse bei Edelstahl

Bei Edelstahl zeigt sich eine abweichende, ausgeprägt zackige Gratstruktur mit höheren lokalen Gradienten und diskontinuierlichen Profilabschnitten. Die bislang implementierte Outlier-Korrektur ist konzipiert zur Eliminierung sporadischer Messfehler bei Baustahl. Demnach stuft diese den Gratverlauf von Edelstahl fälschlich als Ausreißer ein und glättet sie übermäßig. Dadurch werden relevante Merkmale des Edelstahlgrats unterdrückt und der resultierende Vektorverlauf in Richtung eines künstlich „glatten“ Profils verzerrt. Den unterschiedlichen Gratverlauf ist in den obigen Handscanneraufnahmen von Baustahl und Edelstahl ersichtlich (siehe Abbildungen 12 und 13).

Erschwerend kommt hinzu, dass im Messprozess partiell überbeschattete Bereiche auftreten können, die vom Sensor nicht erfasst werden. In der bisherigen Pipeline werden solche Lücken durch Interpolation geschlossen, deren Parameter auf die kontinuierlichen Profile von Baustahl zugeschnitten sind. Für den zackigen Edelstahlgrat führt dies zu einer zu starken Annäherung an glatte Zwischenverläufe und damit zu einem Verlust an formcharakteristischer Information.

Zur materialspezifischen Anpassung werden daher zwei Kernmodule überarbeitet. Die Ausreißererkennung Section 5.1.1 mit nachgelagerte Korrektur und die Interpolation Section 5.1.2 fehlender Stützstellen. In der Ausreißererkennung werden die Schwellwerte und die zugrunde liegenden Sensitivitätsmaße an die höhere lokale Krümmung und den gesteigerten Kantenkontrast des Edelstahlgrats angepasst. Ziel ist eine Differenzierung zwischen echten Messfehlern und materialtypischen Hochfrequenzanteilen. Entsprechend werden Glättungsschritte zurückgenommen, sodass signifikante Gratflanken erhalten bleiben.

Für die Interpolation wird ein konservativer Ansatz gewählt, der Lückenschlüsse bevorzugt entlang lokal konsistenter Nachbarschaften vornimmt und globale, stark glättende Approximationen vermeidet. Damit wird erreicht, dass der rekonstruierte 2D-Vektor fehlende Messpunkte plausibel ergänzt, ohne die charakteristische Zackigkeit des Edelstahlgrats zu nivellieren.

### 5.1.1 Optimierung der Ausreißererkennung

Ziel der Ausreißerbehandlung im **Burr**-Verlauf ist die zuverlässige Trennung zwischen echten Messfehlern und materialtypischen Hochfrequenzanteilen des Edelstahlgrats. Die Verarbeitung ist zweistufig aufgebaut. Zuerst erfolgt ein Vergleich des gemessenen Profils mit einem lokal geglätteten Referenzsignal zur Detektion potenzieller Ausreißerpunkte. Anschließend werden entstandene Lücken kontrolliert rekonstruiert. Der zugehörige Pseudocode ist in Algorithm 1 dargestellt.

In `outlier_correction_burr` dient ein gleitender Mittelwert auf `z_vec` als Referenzsignal. Die punktweise Abweichung  $\Delta = |z - \bar{z}_{MA}|$  wird mit dem Schwellwert `threshold` verglichen. Alle Punkte oberhalb dieses Schwellwertes bilden die Ausreißermaske. Überschreitet der Anteil markierter Punkte den Grenzwert `max_nan_values_perc`, wird der Lauf verworfen und `None` zurückgegeben. Dieses Abbruchkriterium wirkt als Qualitätswächter bei stark gestörten Profilen. Liegt der Anteil darunter, werden die Ausreißer in `z_vec` zu `NaN` gesetzt und mit `smooth_nan_values` rekonstruiert. Die Rekonstruktion schließt kurze Lücken entlang der lokalen Nachbarschaft und erhält charakteristische Gratflanken, sodass der resultierende Vektor die materialspezifische Zackigkeit beibehält.

Die Parametrisierung orientiert sich an der spektralen Struktur des Profils und an der Messrauschenamplitude. Das Fenster `window` bestimmt die lokale Glättung des Referenzsignals. Ein kleineres Fenster reagiert sensibler auf kurzwellige Änderungen, während ein größeres Fenster stärker glättet. Der Schwellwert `threshold` definiert die Minimalhöhe einer Abweichung, ab der ein Punkt als Ausreißer gilt. Für Edelstahl wird ein höherer Schwellwert gewählt, um die ausgeprägten kurzweligen Strukturen nicht fälschlich zu maskieren. Der Grenzwert `max_nan_values_perc` begrenzt die zulässige Ausreißerquote und verhindert Rekonstruktionen auf unzureichender Datenbasis. Die konkreten Werte für Edelstahl und die Referenzkonfiguration für Baustahl sind in Quellcode 5.1 und Quellcode 5.2 angegeben.

Aus methodischer Sicht besitzt die Vorgehensweise eine lineare Laufzeit in der Profillänge und ist damit für große Punktzahlen geeignet. Die Nutzung eines zentrierten gleitenden Mittelwerts sorgt für eine symmetrische Referenzbildung. Randbereiche werden durch die Faltung im Same-Modus in der Länge konsistent gehalten. Längere Messausfälle werden bewusst nicht vollständig interpoliert, da dies die Formtreue des Burr-Verlaufs beeinträchtigen würde. In solchen Fällen greift das Abbruchkriterium und markiert die entsprechende Profilzeile für eine erneute Datenerfassung oder für eine angepasste Parametrisierung.

**Algorithm 1:** Einfache Ausreißer-Korrektur im Burr-Profil (vgl. Parameter in Quellcodes 5.1 and 5.2)

---

**Input:** Positionen  $x$ , Höhen  $z$ , Schwelle  $\tau$ , Fenster  $w$ , max. Anteil  $\alpha$

**Output:**  $(x, z_{\text{clean}})$  oder **None**

```

ref ← MOVING_AVERAGE( $z, w$ , centered)
diff ← ABS( $z - \text{ref}$ )
mask ← ( $\text{diff} > \tau$ )
anteil ← COUNT(mask) / LENGTH( $z$ )
if  $\text{anteil} > \alpha$  then
     $\lfloor \text{return } (\text{None}, \text{None})$ 
set  $z[\text{mask}] \leftarrow \text{NaN}$ 
 $z_{\text{clean}} \leftarrow \text{SMOOTH\_NAN\_VALUES}(x, z)$ 
return  $(x, z_{\text{clean}})$ 
```

---

In der Edelstahl-Pipeline wurden der Schwellwert der Burr-Ausreißererkennung angehoben und das Fenster leicht verkürzt. Ziel ist die Erhaltung materialspezifischer Hochfrequenzanteile bei zugleich robuster Erkennung echter Messfehler. Die verwendeten Parameter sind in Quellcode 5.1 dokumentiert. Zum Vergleich ist die bisherige Baustahl-Konfiguration in Quellcode 5.2 angegeben. Für die Rauheitsanalyse wurden keine Parameter geändert.

Quellcode 5.1: Pipeline-Parameter Outlier Correction (Edelstahl, Burr-Verlauf) (zu Algorithm 1)

```

1 # Parameter burr outlier correction (Edelstahl)
2 burr_outlier_threshold : 0.06 # Threshold for Moving Average Difference Filter [mm]
3 burr_outlier_window     : 9    # Window for Moving Average Difference Filter [samples]
```

Quellcode 5.2: Pipeline-Parameter Outlier Correction (Baustahl, Burr-Verlauf) (zu Algorithm 1)

```

1 # Parameter burr outlier correction (Baustahl)
2 burr_outlier_threshold : 0.03 # Threshold for Moving Average Difference Filter [mm]
3 burr_outlier_window     : 10   # Window for Moving Average Difference Filter [samples]
```

### 5.1.2 Optimierung der Interpolation

Die Interpolation rekonstruiert fehlende Messwerte (`NaN`), die durch Ausreißerkennzeichnung oder unvollständige Erfassung entstehen. Ziel ist die Wiederherstellung eines plausiblen Gratverlaufs, daher werden nur kurze, lokal begrenzte Lücken gefüllt und größere Ausfälle bleiben markiert. Durch die angepasste Ausreißererkennung werden weniger echte Messpunkte fälschlich als Ausreißer markiert, sodass deutlich weniger Lücken entstehen und nicht mehr unnötig interpoliert wird. Am Interpolationsskript wurden keine Änderungen vorgenommen. Alle Konstanten bleiben unverändert, insbesondere die maximale Lücke sowie Glättungs- und Fensterparameter. Der zugehörige Quellcode ist im Anhang dokumentiert. Ein veranschaulichter Gratverlauf ist im folgendem Kapitel 5.3 in Abbildung 18 dargestellt.

## 5.2 Rauheitsanalyse bei Edelstahl

Für die Rauheitsanalyse wird die bestehende Auswertelogik aus der Baustahl-Pipeline übernommen und auf Edelstahl angewendet. Ziel ist die Überprüfung, ob die Parameter und Rechenschritte ohne Anpassungen auch für Edelstahl robuste und nachvollziehbare Werte liefern. Dabei liegt der Fokus auf der Erkennung unplausibler Ergebnisse, die auf ungeeignete Aufnahmebedingungen oder auf notwendigen Anpassungsbedarf im Skript hinweisen könnten.

Die Bildaufnahmeeinstellungen des Handscanners wurden zuvor für Edelstahl optimiert, um Effekte durch erhöhte Reflexion zu minimieren, siehe Section 4.2.1. Auftretende Unplausibilitäten wären damit primär dem Auswertungsskript zuzuordnen. Entsprechend wurde die Rauheitsanalyse anhand einer Stichprobe validiert. Untersucht wurden zehn Edelstahlbleche mit jeweils vier Schnittkanten. Die Prüfung umfasste Konsistenztests über die gesamte Kantenlänge, die Kontrolle auf Ausreißer sowie die Sichtprüfung auf Sättigungseffekte und Messlücken. Keine der vierzig untersuchten Kanten zeigte auffällig abweichende Rauheitswerte. Das Skript zur Rauheitsanalyse wird daher unverändert beibehalten.

Die Auswertung erfolgt abschnittsweise entlang der Schnittkante. Die Kante wird von oben nach unten in Segmente der Länge 0,3 mm unterteilt. Die X-Achse gibt die Position entlang der Schnittkante  $x$  [mm] an, die Y-Achse den für eine Segmentlänge von 0,3 mm berechneten Rauheitswert  $R_a$  [ $\mu\text{m}$ ]. Die Skalierung der x-Achse auf den Bereich 2,1 cm bis 7,6 cm ergibt sich aus der Blechbreite von 10 cm. Ausgewertet wird bewusst nur der mittlere Abschnitt, um beschleunigungsbedingte Einflüsse des Laserschneidprozesses an Ein- und Ausfahrt des Schneidkopfes zu vermeiden. Für jedes Segment wird die Rauheitskennzahl  $R_a$  berechnet und als Profil über der Kantenlänge dargestellt. Die Abbildung 17 zeigt ein typisches Ergebnis. Das Profil weist eine geringe Streuung um den Mittelwert auf und enthält keine systematischen Trends oder Ausreißer. Dies bestätigt die Eignung der bestehenden Parametrisierung für Edelstahl.

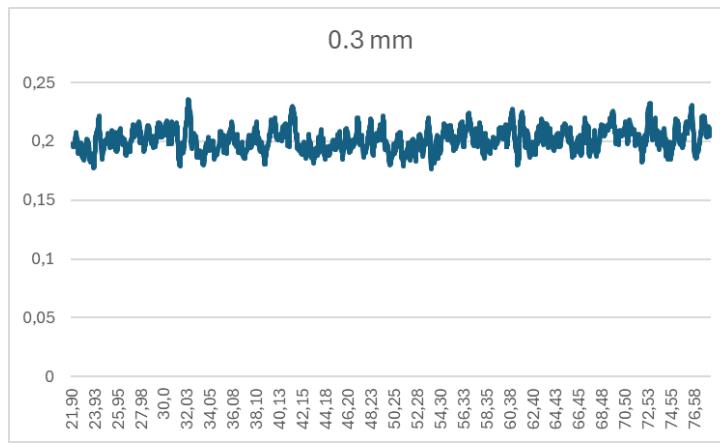


Abb. 17: Beispielhafter Verlauf der Rauheitskennzahl  $R_a$  entlang einer Schnittkante im ersten Abschnitt mit der Breite 0,3 mm.

## 5.3 Validierung der Messoptimierungen

Im folgendem Kapitel werden die verbesserten Messmethoden für die Grat- und Rauheitsanalyse validiert, um bestenfalls die Optimierung für Datenerfassung zu nutzen, so dass das KI-Modell neu trainiert werden kann.

### 5.3.1 Validierung der verbesserten Ausreißererkennung im Burr-Verlauf

In diesem Abschnitt werden die edelstahl-optimierten Einstellungen der Ausreißererkennung validiert. Verglichen wird die frühere Baustahl-Parametrierung mit der angepassten Erkennung für Edelstahl.

Die Abbildung 18 zeigt den Vergleich der verbesserten und der unverbesserten Ausreißererkennung. Hierfür wurden jeweils das gleiche Sektionsbild des gleichen Edelstahlblechs genutzt. Die rote Linie ist der abgeleitete Burr-Verlauf. Auf dem linken Schaubild ist die unverbesserte Ausreißererkennung erkennbar, welche für Baustahl werdet wird und rechts die verbesserte Ausreißererkennung. Demnach ist der verbesserte Verlauf durchgängig, es entstehen weniger fehlerhafte Lücken und die interpolierten Abschnitte sind plausibler. Quantitativ sinkt der Burr-Wert der gezeigten Sektion von  $681,99 \mu\text{m}$  (Baustahl-Parameter) auf  $650,88 \mu\text{m}$  (Edelstahl-Parameter), jedoch erfolgt die einschätzende Bewertung visuell. Es wird geprüft, wie gut die rote Linie dem sichtbaren Gratverlauf folgt. Demnach ist die verbesserte Ausreißererkennung für Edelstahl validiert.

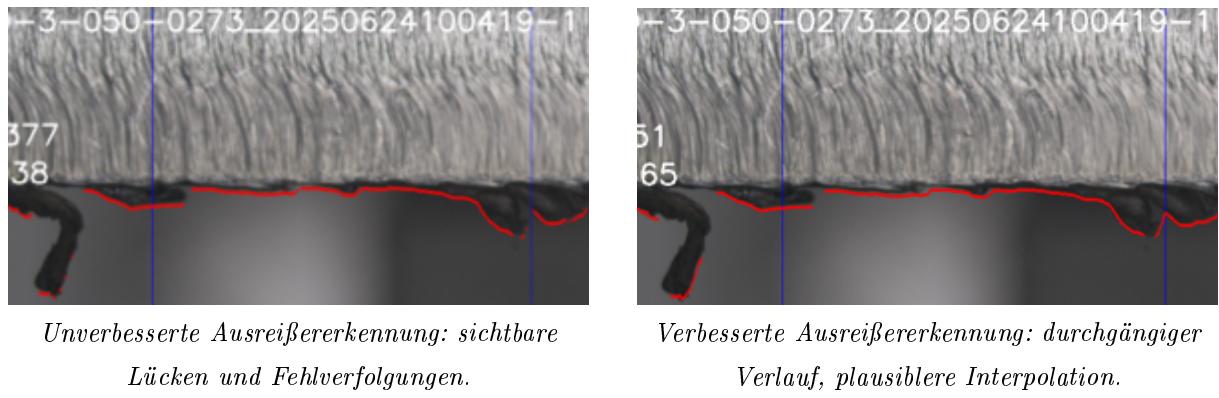


Abb. 18: Links ist die unverbesserte Ausreißererkennung erkennbar, welche für Baustahl werdet wird und rechts die verbesserte Ausreißererkennung. Die rote Kurve ist der abgeleitete Burr-Verlauf.

### **5.3.2 Validierung der bestehenden Rauheitsanalyse an den neuen Edelsdtahldaten**

In diesem Abschnitt wird die Rauheitsanalyse validiert. Die Berechnungsmethode bleibt unverändert. Die Belichtung des Handscanners wurde so angepasst, dass die Struktur der Schnittfläche zuverlässig erfasst wird (siehe Abschnitt 4.2.1). Bei der Analyse in Kapitel 17 traten keine systematischen Ausreißer auf, daher wird die Berechnung vorerst unverändert weitergeführt. Anpassungen werden erst nach weiterführenden Modelltrainings geprüft.

## 6 Qualitätsschätzungsmodell für Edelstahl

Dieses Kapitel beschreibt das bestehende **QA!** (**QA!**) auf Basis eines **CNNs!** und seine Anpassung an Edelstahl. Als Referenz dient VGG16, ein tiefes Netz aus dreizehn Faltungs- und drei vollständig verbundenen Schichten. Kennzeichnend sind gestapelte  $3 \times 3$ -Faltungen in Blöcken, eine schrittweise Verkleinerung der Feature-Maps und eine mit der Tiefe zunehmende Kanalzahl. VGG16 umfasst rund 138 Millionen Parameter und ist damit leistungsfähig, aber für unsere Datenbasis zu groß. Ein Transfer vortrainierter Gewichte aus ImageNet lieferte auf Schnittkantenbildern keine stabilen, domänerelevanten Merkmale, da sich die dort gelernten Objektstrukturen deutlich von den Texturen und Kanten der Laserschnitte unterscheiden.

Aus den Grundideen von **VGG!** (**VGG!**) wurde daher eine kompaktere, VGG-ähnliche Architektur abgeleitet. Sie behält die Blockstruktur und kleine Faltungskerne bei, reduziert jedoch die Komplexität deutlich und verzichtet auf die breite, mehrstufige Klassifikationskopfkette (**Tatzel-2021**).

Nach den Faltungsblöcken liegen mehrere Merkmalskarten vor. Global Average Pooling mittelt jede Merkmalskarte über ihre räumlichen Dimensionen und erzeugt so einen kompakten Merkmalsvektor. Eine schlanke vollständig verbundene Ausgabeschicht bildet diesen Vektor auf die Zielgröße ab, also den geschätzten Gratwert. Das Modell wird von Grund auf auf Schnittkantenbildern trainiert und ist damit auf die materialspezifischen Muster von Edelstahl abgestimmt. Figure 19 zeigt die Blockstruktur und den kompakten Kopf des Netzes.

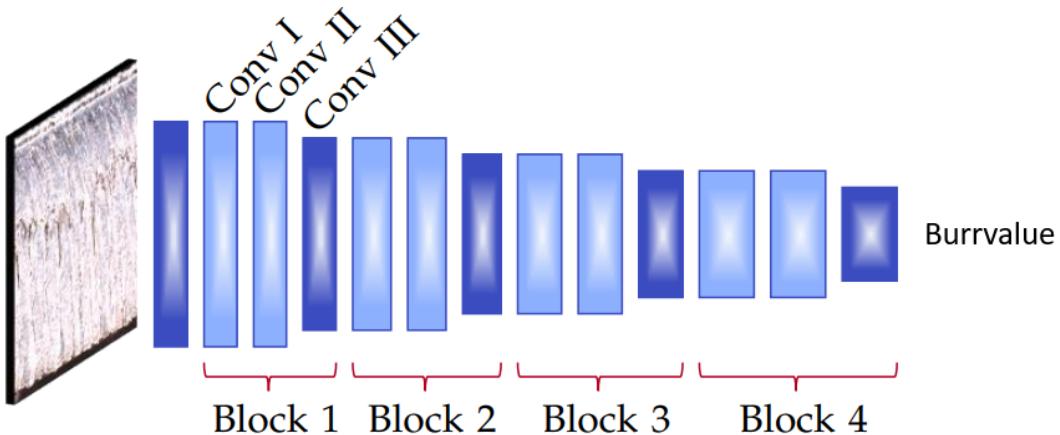


Abb. 19: Schematische Darstellung der VGG-ähnlichen Architektur zur Regression der Prozessparameter aus einem Schnittkantenbild. Links der Bildeingang, gefolgt von einem Eingangsblock mit  $7 \times 7$ -Faltung und vier Blöcken mit jeweils drei  $3 \times 3$ -Faltungen. Die dritte Faltung eines Blocks reduziert die räumliche Auflösung. Global Average Pooling verdichtet die 256 Feature-Maps zu einem 256-dimensionalen Vektor, der den Ausgang, dem Burr-Wert, speist. (**Tatzel-2021**)

## 6.1 Bewertung des bestehenden Modells

Das bestehende **QA!**-Modell für Baustahl wurde auf einen kleinen Edelstahl-Datensatz übertragen und auf die Gratschätzung angewandt. Für die Rauheitsschätzung ist der bestehende Datensatz zu klein, um eine belastbare Bewertung zu ermöglichen. Die Rauheitsschätzung benötigt eine größere Datenbasis, da die Rauheit stark von lokalen Unregelmäßigkeiten abhängt. Die Gratschätzung ist robuster und kann auch mit weniger Daten bewertet werden. Abbildung 20 zeigt wahre gegen vorhergesagte Burr-Werte für die Blechdicken 5 mm, 10 mm und 15 mm. Die gestrichelte Diagonale kennzeichnet die ideale Vorhersage  $y=x$ .

Bei kleinen Burr-Werten (hier: 5 mm) liegen die Punkte noch nahe an der Diagonale. Für 10 mm und 15 mm werden hohe Burr-Werte deutlich unterschätzt und die Streuung nimmt zu. Ursache ist ein Domänenwechsel von Baustahl zu Edelstahl. Edelstahl zeigt bei größeren Dicken (hier: 10 mm und 15 mm) häufiger hohen Burr. Bei einer Blechdicke von 5 mm ist der Burr-Wert im Verhältnis deutlich geringer. Das Baustahlmodell hat solche Ausprägungen im Training nicht gesehen. Die gelernten Merkmale und die Abbildung von Merkmalen auf den Burr sind dadurch auf den Bereich von Baustahl begrenzt. Das führt zu einer Sättigung der Vorhersage und zu einem dickenabhängigen Bias.

Für eine belastbare Vorhersage auf Edelstahl ist ein erneutes Training mit Edelstahldaten erforderlich. Dabei müssen die Faltungsfilter und nicht nur die letzte Schicht angepasst werden. So können die oberflächenspezifischen Strukturen und die höheren Burr-Werte bei 10 mm und 15 mm korrekt erfasst werden.

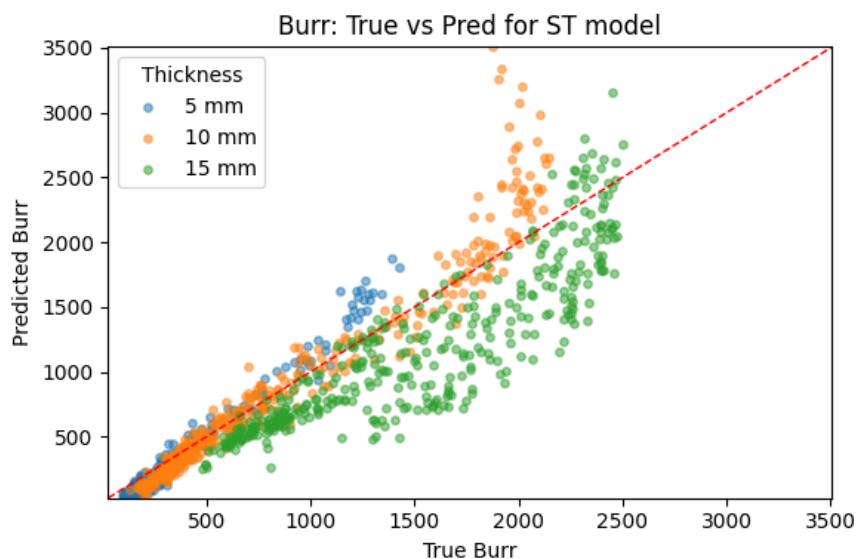


Abb. 20: Burr: wahre gegen vorhergesagte Werte für das **ST!**-Modell. Die Farben kennzeichnen 5 mm, 10 mm und 15 mm. Die gestrichelte Diagonale markiert  $y=x$ .

## 6.2 Training für Edelstahldatensatz

Nach dem Training auf Edelstahldaten für die Gratschätzung zeigt Abbildung 21 eine enge Übereinstimmung der Punkte mit der Diagonalen. Der zuvor sichtbare dickenabhängige Bias ist deutlich geringer. Hohe Burr-Werte bei 10 mm und 15 mm werden nun besser getroffen und die Streuung fällt kleiner aus. Der Grund ist, dass das Modell beim Training nun Beispiele mit hohen Burr-Werten aus Edelstahl gesehen hat. Diese Ausprägungen waren im Baustahl-Datensatz kaum vorhanden. Die gelernten Merkmale bilden die für Edelstahl typischen Strukturen daher zuverlässiger ab. Im höchsten Burr-Bereich bleiben geringe systematische Abweichungen bestehen. Hier kann eine Erweiterung der Daten mit mehr Fällen hoher Burr-Werte und eine Kalibrierung der Vorhersagen weiter helfen. Insgesamt liefert das neu trainierte Modell eine robuste und weitgehend dickenunabhängige Gratschätzung für Edelstahl.

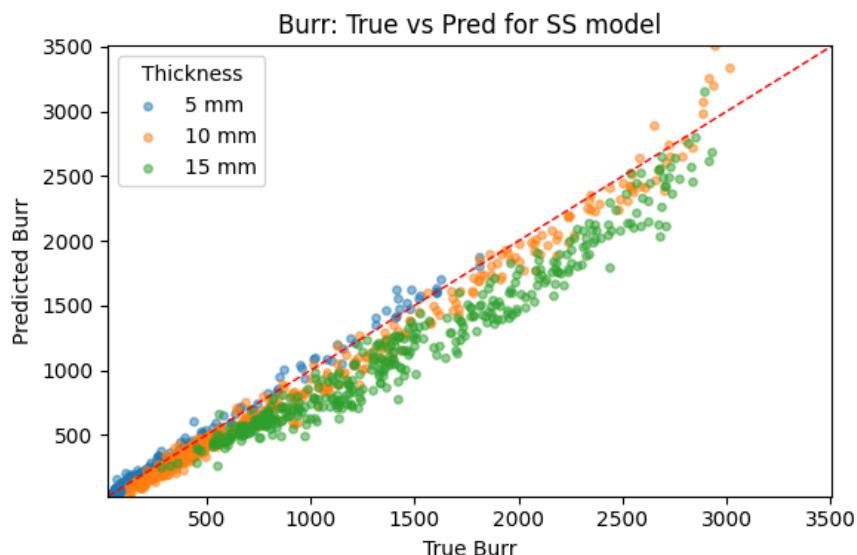


Abb. 21: Burr: wahre gegen vorhergesagte Werte für das **SS!**-Modell. Die Farben kennzeichnen 5 mm, 10 mm und 15 mm. Die gestrichelte Diagonale markiert  $y=x$ .

# 7 Reflexion und Ausblick

Dieses Kapitel bewertet die Arbeit und das gewählte Vorgehen rückblickend. Anschließend werden die nächsten Schritte skizziert.

## 7.1 Reflexion

Das geplante Vorgehen hat sich als zielführend erwiesen. Das Ziel einer verbesserten **QA!** für Edelstahlschnittkanten wurde erreicht. Der wesentliche Engpass lag in der Datenerfassung in der Messzelle. Die Messungen pro Parameterkombination benötigen viel Zeit. Für ein stabiles und robustes Modell werden jedoch viele Beispiele benötigt. Deshalb konnte innerhalb des Bearbeitungszeitraums kein Modell für die Rauheit aufgebaut werden.

Die gewählte Reihenfolge war hilfreich. Zuerst wurde die Schnittabrißsgrenze analysiert. Darauf basierend wurden Daten nach Versuchsplan erzeugt. Die Messmethodik in der Messzelle wurde auf Edelstahl angepasst und anschließend wurde das Modell auf Edelstahldaten neu trainiert. Dieses Vorgehen ist nachvollziehbar und auf weitere Werkstoffe übertragbar, sofern ausreichend repräsentative Daten vorliegen.

## 7.2 Ausblick

Künftige Arbeiten sollten prüfen, ob ein gemeinsames Modell für Baustahl und Edelstahl vorteilhaft ist. Ein Ansatz mit gemeinsamem Merkmalsextraktor und materialspezifischen Ausgabeschichten erscheint geeignet. Als Eingaben kommen reine Bilddaten oder eine Kombination aus Bilddaten und Metadaten in Frage. Relevante Metadaten sind insbesondere Material, Blechdicke, Schneidgas und zentrale Prozessparameter.

Für die Rauheitsschätzung ist eine Erweiterung des Datensatzes mit verlässlichen Referenzmessungen erforderlich. Sinnvoll ist ein gemeinsames Training für Grat und Rauheit, da beide Größen aus ähnlichen Strukturen hervorgehen. Eine nachgelagerte Kalibrierung der Vorhersagen erhöht die Anwendbarkeit im Betrieb.

Die Datenerfassung in der Messzelle sollte weiter beschleunigt werden. Automatisierte Messabläufe und aktives Lernen können die Auswahl informativer Parameterkombinationen steuern. Ergänzend helfen Datenaugmentation und eine gezielte Abdeckung hoher Burr-Werte, die bei dickerem Blech häufiger auftreten.

Für eine belastbare Validierung sollten Modelle an unabhängigen Materialchargen und bisher unbekannten Blechdicken getestet werden. Darüber hinaus ist es sinnvoll, Unsicherheiten der

Vorhersagen auszugeben und potenzielle Verteilungsunterschiede zwischen Trainings- und Anwendungsdaten zu überwachen. Die Präsentation der Ergebnisse sollte sich an anerkannten Qualitätskennzahlen und relevanten Normen orientieren.

Das hier entwickelte Vorgehen lässt sich auf weitere Legierungen und Blechdicken übertragen. Bei Bedarf wird die Messmethodik angepasst und das Modell mit wenigen neuen Beispielen nachtrainiert, um eine robuste Qualitätsschätzung über unterschiedliche Werkstoffe hinweg zu erreichen.

# Anhang

Quellcode 7.1: Werkstoffabhängiges Routing der Handscanner-Setups

```
1 public static string NameParserMaterial(string input)
2 {
3     if (string.IsNullOrWhiteSpace(input))
4         return "ST";                                // Fallback-Wert
5
6     // Position des ersten und zweiten Bindestrichs ermitteln
7     int firstDash = input.IndexOf('-');
8     int secondDash = firstDash >= 0
9         ? input.IndexOf('-', firstDash + 1)
10        : -1;
11
12    // Prüfen, ob ein zweiter Bindestrich und ein Zeichen dahinter existieren
13    if (secondDash < 0 || secondDash + 1 >= input.Length)
14        return "ST";                                // Fallback-Wert
15
16    char digit = input[secondDash + 1];            // Ziffer einlesen
17
18    // Zuordnung: 1 -> SST, 2 -> SST, 3 -> SSS (bei Bedarf anpassen)
19    return digit switch
20    {
21        '1' => "ST",
22        '2' => "ST",
23        '3' => "SS",
24        _ => "ST"                                    // Default
25    };
26 }
```

Quellcode 7.2: Interpolation begrenzter Lücken in 1D-Profilen

```
1 def interpolate_limited_nans(vector, max_gap=5):
2     """
3     Interpoliert nur kleine NaN-Lücken (<= max_gap) mit weicher Spline-Interpolation.
4     """
5     vector = vector.copy()
6     isnan = np.isnan(vector)
7     indices = np.arange(len(vector))
8
9     if not np.any(~isnan):
10         return None
11
12     nan_groups = []
13     in_nan = False
14     start = 0
15     for i, val in enumerate(isnan):
16         if val and not in_nan:
17             in_nan = True
```

```
18         start = i
19         elif not val and in_nan:
20             in_nan = False
21             nan_groups.append((start, i - 1))
22         if in_nan:
23             nan_groups.append((start, len(vector) - 1))
24
25     for start, end in nan_groups:
26         gap_size = end - start + 1
27         if gap_size <= max_gap:
28             left = start - 1
29             right = end + 1
30             if left < 0 or right >= len(vector):
31                 continue
32             if np.isnan(vector[left]) or np.isnan(vector[right]):
33                 continue
34
35         # Verwende CubicSpline statt np.interp
36         x_known = [left, right]
37         y_known = [vector[left], vector[right]]
38         cs = CubicSpline(x_known, y_known, bc_type='natural')
39         interp_indices = indices[start:end+1]
40         vector[start:end+1] = cs(interp_indices)
41
42     return vector
```

Quellcode 7.3: Achsweise Interpolation von NaNs in Matrizen

```
1 def interpolate_nan(matrix_in, axis=0, max_gap=5):
2     """
3     Interpoliert NaNs in einer Matrix entlang der gegebenen Achse mit Begrenzung.
4     """
5     matrix = matrix_in.copy()
6     x, y = matrix.shape
7
8     if axis == 0:
9         for col_index in range(y):
10             if np.isnan(matrix[:, col_index]).any():
11                 vec = interpolate_limited_nans(matrix[:, col_index], max_gap=max_gap)
12                 if vec is not None:
13                     matrix[:, col_index] = vec
14     else:
15         for row_index in range(x):
16             if np.isnan(matrix[row_index, :]).any():
17                 vec = interpolate_limited_nans(matrix[row_index, :], max_gap=max_gap)
```

```
18         if vec is not None:  
19             matrix[row_index, :] = vec  
20     return matrix
```

Quellcode 7.4: Interpolation von X-, Y- und Z-Ebenen

```
1 def interpolate_planes(X, Y, Z, results_directory=None, max_gap=0.5):  
2     """  
3     Interpoliert X, Y, Z Ebenen entlang sinnvoller Achsen mit Lückenbegrenzung.  
4     """  
5     X_interpolated = interpolate_nan(X, axis=1, max_gap=max_gap)  
6     Y_interpolated = interpolate_nan(Y, axis=1, max_gap=max_gap)  
7     Z_interpolated = interpolate_nan(Z, axis=0, max_gap=max_gap)  
8     return X_interpolated, Y_interpolated, Z_interpolated
```

Quellcode 7.5: Kapselnde 1D-Pipeline: begrenzte Interpolation mit optionaler Glättung

```
1 def smooth_nan_values(x, z, max_nan_values_perc=0.4, max_gap=5,  
2                         smoothing=True, window_length=7, polyorder=2):  
3     """  
4     Interpoliert NaN-Werte in z, aber nur bei kleiner Lückenanzahl  
5     und akzeptablem NaN-Anteil. Optional geglättet mit Savitzky-Golay.  
6     """  
7     z = np.copy(z)  
8     nan_indices = np.isnan(z)  
9     num_nans = np.sum(nan_indices)  
10  
11    if num_nans / len(z) > max_nan_values_perc:  
12        return None  
13  
14    z = interpolate_limited_nans(z, max_gap=max_gap)  
15  
16    # Optional glätten  
17    if smoothing and z is not None and np.count_nonzero(~np.isnan(z)) > window_length:  
18        z = savgol_filter(z, window_length=window_length, polyorder=polyorder)  
19  
20    return z
```

Quellcode 7.6: Schließen kurzer Täler (Valleys) via begrenzter Interpolation

```
1 def fill_small_valleys(z, window_size=30, depth_threshold=5.0, max_gap=0.5,  
2                         smoothing=True, smooth_window=7, polyorder=2):  
3     """  
4     Glättet kleine Einbrüche ('valleys') in einem Höhenprofil z.  
5     """  
6     z = np.array(z).copy()
```

```
7     rolling_min = pd.Series(z).rolling(window=window_size, center=True,
  ↪ min_periods=5).min()
8     diff = rolling_min - z
9
10    # Punkte mit tieferem Einbruch als erlaubt
11    valley_mask = (diff > depth_threshold)
12    z[valley_mask] = np.nan
13
14    z_filled = interpolate_limited_nans(z, max_gap=max_gap)
15
16    # Optional glätten
17    if smoothing and z_filled is not None and np.count_nonzero(~np.isnan(z_filled)) >
  ↪ smooth_window:
18        z_filled = savgol_filter(z_filled, window_length=smooth_window, polyorder=2)
19
20    return z_filled
```

## Literaturverzeichnis

**Hoyos-Gutiérrez, Jose; Cardona-Aristizabal, Jaiber; Muñoz-Gutiérrez, Pablo Andrés; Ramirez-Jimenez, Diego (2023): A Systematic Literature Review on Rapid Control Prototyping Applications.** In: *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*. IEEE Xplore Document 10056232. DOI: 10.1109/RITA.2023.3250559. URL: <https://ieeexplore.ieee.org/document/10056232> (Abruf: 12.01.2026).