

Detektion von Konstruktionsfehlern durch eine automatisierte Objekterkennung mittels Deep Learning

Patrik Müller, Paul Christoph Gembarski, Roland Lachmayer

Institut für Produktentwicklung und Gerätebau, Gottfried Wilhelm Leibniz Universität Hannover

An der Universität 1, 30823 Garbsen

Email: {Mueller, Gembarski, Lachmayer}@ipeg.uni-hannover.de; Internet: <https://www.ipeg.uni-hannover.de>

Inhalt: In der Produktentwicklung gelten diverse Richtlinien, an denen sich Ingenieure orientieren. Letztendlich sind aber die Erfahrungen des jeweiligen Anwenders dafür verantwortlich, ob eine Konstruktion korrekt gestaltet wird. Dieser Beitrag bietet einen Ansatz zur automatisierten Anwendung dieses Erfahrungswissens durch die Objekterkennung mittels Deep Learning. Dafür wird ein neuronales Netz anhand eines Beispiels so trainiert, dass es gießgerechte Gestaltungsmerkmale in einer Konstruktion erkennt und klassifiziert. Die Objekterkennung wird mithilfe des YOLOv4-Algorithmus realisiert, indem zweidimensionale Schnittansichten des Bauteils vom neuronalen Netz analysiert werden. Des Weiteren werden zwei Methoden zur dreidimensionalen Rekonstruktion der Schnittansichten vorgestellt. Abschließend wird die Methode bewertet und kritisch gewürdigt.

Stichwörter: Computer Vision, Deep Learning, CNN, YOLO, YOLOv4, KI, benutzerdefinierte Objektdetektion

Identification of constructional faults through automated object detection via Deep Learning

Abstract: Engineers are guided by multiple design guidelines in product development. Certainly, it is the experience of the respective user that determines whether a construction is designed correctly. This paper offers an approach to the automatized implementation of this experience knowledge by object detection using Deep Learning. For this purpose, a neural network is trained by means of an example in such a way that it detect and classify casting-compatible design features in a construction. The object recognition is realized with the YOLOv4-algorithm by analyzing two-dimensional sectional views of the component by the neural network. Furthermore, two methods for three-dimensional reconstruction of the sectional views are presented. Afterwards the method is evaluated and critically appraisal.

Keywords: Computer Vision, Deep Learning, CNN, YOLO, YOLOv4, AI, Custom Object Detection

1 Einleitung

„Die Entwicklung künstlicher Intelligenz könnte entweder das Schlimmste oder das Beste sein, was den Menschen passiert ist.“ Eine abschließende Bewertung des Zitats von Steven Hawking aus dem Jahr 2018 ist nicht möglich, da die Anwendung und Weiterentwicklung von KI-Methoden als Entscheidungsunterstützungssystem oder Problemlöser in verschiedensten Anwendungsgebieten noch Gegenstand aktueller Forschung ist. In der rechnerunterstützten Produktentwicklung stehen Ingenieure z. B. häufig vor Problemen, die intuitiv leicht zugänglich sind, sich allerdings schwer in Algorithmen formalisieren lassen und ein hohes Maß an Perzeption erfordern [1]. Ein Beispiel dafür, ist die Identifikation und Auflösung von Knotenpunkten sowie das Vermeiden von Materialanhäufungen innerhalb des gießgerechten Gestaltens [2]. Ein System, welches hierbei unterstützen soll, muss diese Materialanhäufungen wahrnehmen. Auf visueller Ebene kann die automatisierte Bilderkennung als Unterstützung fungieren. Das maschinelle Sehen (*engl.* Computer Vision) wird breit erforscht und bereits in einzelnen Domänen erfolgreich zur Anwendung gebracht, z.B. als Diagnosewerkzeug in der Radiologie [3]. Die reine Perzeption reicht jedoch zum Treffen von Entscheidungen nicht aus. Hierfür ist eine Kopplung mit anderen KI-Mechanismen oder wissensbasierten Systemen erforderlich. Dieser Artikel untersucht die Anwendung von neuronalen Netzen für die Bilderkennung bei ingenieurtechnischen Fragestellungen in Kombination mit fallbasiertem Schließen am Beispiel eines CAD-basierten Assistenzsystems zur Detektion von Materialanhäufungen.

2 Stand der Technik

Die maschinelle Fähigkeit, Bilder zu erfassen, zu erkennen und zu verarbeiten, weist kaum grundlegende Unterscheidungen zur Bilderkennung menschlicher Intelligenz auf. Da die neuronalen Netze dem menschlichen Gehirn nachempfunden sind, finden sich unzählige Analogien im Prozess der Bilderkennung zwischen Mensch und Maschine. [4]

2.1 Bildverarbeitung

Ein zweidimensionales digitales Bild wird durch ein Signal repräsentiert, welches Informationen in Form von Pixeln beinhaltet. In der Regel bildet jeder Pixel einen definierten Farbwert ab. Im RGB-Farbraum, der die Farbwahrnehmung durch die Mischung dreier Grundfarben nachbildet, wird jeder Pixel durch drei Intensitätswerte von 0-255 für die Kanäle

Rot, Grün, Blau beschrieben. In Bild 1 ist hierfür eine Beispielgrafik auf verschiedenen Abstraktionsniveaus mit diskreten Grauwerten dargestellt.

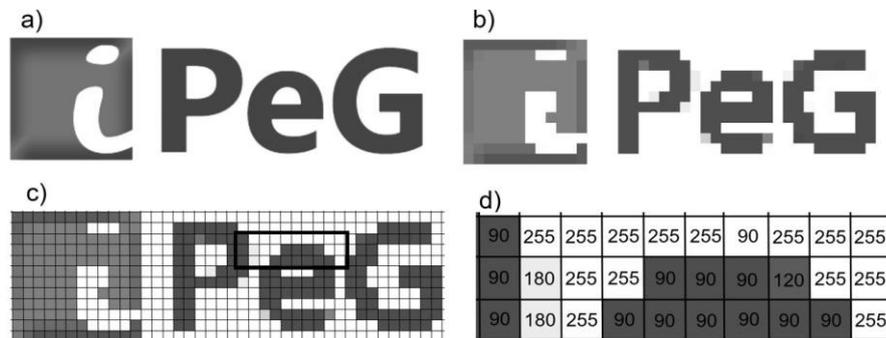


Bild 1: Bilddarstellung auf den verschiedenen Abstraktionsniveaus a) Ursprung, b) Verpixelt, c) Verpixelt mit Raster, d) Ausschnitt von c) mit diskreten Grauwerten.

Auf dieser Basis lassen sich Bilder als Signale rechnerunterstützt bearbeiten, auswerten, verändern und erkennen. Die menschliche Bildverarbeitung weist dabei wesentliche Ähnlichkeiten auf. Das menschliche Auge nimmt Bilder in Form von Lichtwellen wahr, welche je nach Wellenlänge Spannungs- und Stromänderungen auf der Retina erzeugen. Diese Änderungen werden von den Stäbchen und Zapfen detektiert. Bei den drei Arten von Zapfen gibt es jeweils eine für die Lichtkomponente Rot, Grün, Blau (Analogie zum RGB-Farbraum). Die Stäbchen und Zapfen erzeugen daraus Signale, welche im Gehirn z.B. zu Formen, Farben und Bewegungen verarbeitet werden [5]. Die genauen Prozesse innerhalb des Gehirns sind zum derzeitigen Kenntnisstand nicht präzise geklärt und Gegenstand der Hirnforschung bzw. Wahrnehmungspsychologie. Seherfahrungen und Gelerntes werden im Gehirn verknüpft und in Form von Modellen als Wissensbasis gespeichert. Bei einem wiederholten „Sehen“ einer bekannten Szene wird das Retinabild gefiltert und mit der Wissensbasis in Beziehung gesetzt. So ist der Mensch in der Lage innerhalb Bruchteilen von Sekunden eine Flut an Informationen aufzunehmen und zu verarbeiten. [6]

2.2 Convolutional Neural Network (CNN)

Neuronale Netze sind Netze aus künstlichen Neuronen mit einer definierten Architektur. Dabei addiert ein Neuron die gewichteten Eingangssignale und verarbeitet diese Summe gemäß einer Aktivierungsfunktion zu einem Ausgangssignal. Lernfähig ist das Netz unter anderem durch die Veränderung dieser Gewichte, sodass die Differenz zwischen berechnetem und vorhergesagtem Ausgangssignal minimal ist. Ein hochauflösendes Farbbild mit 1.000 x 1.000 Pixel besitzt demnach 3.000.000 Eingangsparameter. Eine

vollständig verbundene neuronale Netzschicht (*engl. Fully Connected Layer*) aus 1.000 Neuronen würde schon hier zu einer Matrix mit 3.000.000.000 Parametern führen. Typischerweise bestehen robuste neuronale Netze aus einer Vielzahl von Schichten (*engl. Layer*). Die hieraus resultierende Anzahl an Parametern ist enorm und nicht angemessen zu verarbeiten. Aus diesem Grund wurden die *Convolutional Neural Networks* (CNN) entwickelt [7]. Innerhalb dieser Struktur werden *Convolutional Layers* verwendet, die Bilder durch lernfähige Filter falten (*engl. to convolve*). Im nächsten Schritt werden die Informationen zusammengefasst (*engl. to pool*). Dadurch werden überflüssige Informationen gelöscht. Durch mehrere hintereinander angeordnete *Convolutional* und *Pooling Layer* werden Merkmale auf verschiedenen Abstraktionsebenen (bspw. Linie, Form, Muster) in *Feature Maps* extrahiert, die Anzahl der zu trainierenden Gewichte reduziert und die räumliche Zuordnung der Merkmale sichergestellt. Die Faltung erzeugt Schichten von *Feature Maps*, die aus dem vorherigen *Layer* zwar identische Eingaben bekommen, aber aufgrund der Gewichtsmatrizen verschiedene Merkmale herausfiltern. [7-9]

Anschließend werden die Signale in einen oder mehrere *Fully Connected Layer* geführt und ein Ausgangswert berechnet (Bild 2).

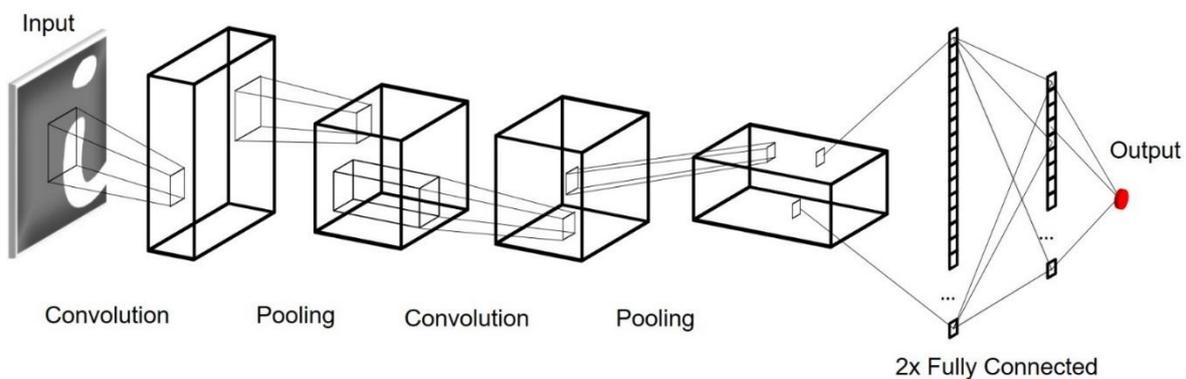


Bild 2: Architektur eines CNN zur Bildklassifikation. Dargestellt sind vier *Layer* (2x *Convolutional+Pooling* sowie 2x *Fully Connected*) mit mehreren Filterkernen, sodass eine hohe Anzahl an *Feature Maps* in den jeweiligen *Layern* gespeichert wird. Abbildung nach [9].

Mit dieser Methode lassen sich beispielsweise Bilder klassifizieren, segmentieren sowie Objekte präzise erkennen [10]. Aufbauend auf den CNNs gibt es diverse Features, Veränderungen und Optimierungen für spezielle Anwendungsfälle, sodass weiterentwickelte Formen des CNN unter diversen Bezeichnungen in der Literatur zu finden sind. In der Produktentwicklung finden CNNs noch keine breite Anwendung. Die Methoden der künstlichen Intelligenz werden hauptsächlich im Qualitätsmanagement sowie der industriellen Bildverarbeitung erfolgreich eingesetzt [11].

3 Verfahrensauswahl und Datengenerierung

Um die Lunkerbildung in Gießkonstruktionen zu vermeiden, werden verschiedene Regeln des gießgerechten Gestaltens vom Konstrukteur angewendet [12]. Wenn Rippen und Wände in Knotenpunkten aufeinandertreffen, entstehen Materialanhäufungen. Unter der Berücksichtigung von verschiedenen Gestaltungsrichtlinien lassen sich diese beispielsweise durch das Einlegen von Kernen auflösen oder durch die Anordnung von Steigern über der Materialanhäufung entschärfen [2]. Um eine oder mehrere günstige bzw. ungünstige Knotengestaltungen in zweidimensionalen Schnittansichten von Gießkonstruktionen zu detektieren, sollte die Konstruktion nicht im gesamten, sondern die gesuchten Merkmale räumlich lokalisiert und klassifiziert werden. Als Methode wird daher die Objektdetektion verwendet. Beim *Deep Learning* gibt es zwei Wissensquellen, die dem Anwender helfen, ein zuverlässiges und robustes neuronales Netz aufzubauen: Gelabelte, zugängliche Daten sowie effektive und erprobte Netzarchitekturen. Als State-Of-Art für die Objektdetektion gilt aktuell die Netzarchitektur YOLOv4 (You-Only-Look-Once Version 4) [13]. Der Algorithmus zeichnet sich dadurch aus, dass das CNN die zu testenden Bilder nur einmal sehen muss, um sie vollständig zu analysieren (daher *You-Only-Look-Once*). Anders als bei Methoden, welche die Regionen eines Bilds sequentiell analysieren (bspw. das bewährte R-CNN [14]), teilt YOLO die Grafik in eine definierte Anzahl an Zellen. Anschließend werden interessante Bereiche innerhalb der Zellen in sogenannten Bounding Boxes, die das gesuchte Objekt umschließen, vorhergesagt [15]. In Bild 3 ist dafür der Algorithmus der Objektdetektion nach YOLO als Regressionsproblem dargestellt.

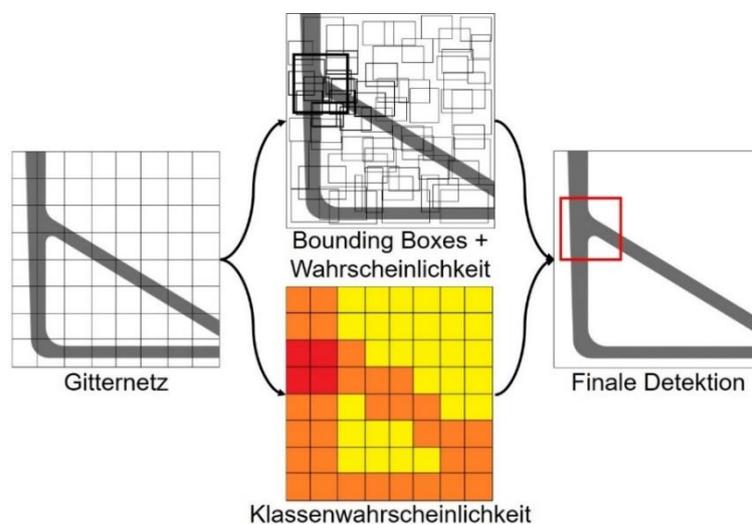


Bild 3: Objektdetektion mit dem YOLO-Algorithmus am Beispiel eines ungünstigen Knotenpunktes.

Abbildung nach [15].

Der regressionsbasierte Algorithmus errechnet die Wahrscheinlichkeiten, dass ein Objekt korrekt innerhalb von einer Zelle mit der Bounding Box umschlossen wurde und führt eine Klassifikation durch. Da durch diesen Algorithmus eine Vielzahl an verschiedenen Bounding Boxes entsteht, werden am Ende nur diejenigen beibehalten, die mit einer definierten Wahrscheinlichkeit ein gesuchtes Objekt detektiert haben. YOLO zeichnet sich nicht nur durch eine ausgezeichnete Netzarchitektur aus, sondern beinhaltet zusätzlich diverse weitere Features. Neben der Möglichkeit, das Netz mit selbsterstellten gelabelten Daten bzw. Grafiken zu trainieren, besitzt es zahlreiche Algorithmen, die das Training optimieren. So werden Features zur Erweiterung der vorhandenen Daten (*engl.* Data Augmentation) verwendet, die Grafiken so verändern, dass das Netz zuverlässiger trainiert wird und robuster auf neue Daten reagiert [13]. Dadurch werden die Trainingsbilder beispielsweise automatisch gespiegelt, verrauscht, verzerrt, gedreht und in YOLOv4 außerdem zu Mosaiken zusammengesetzt. Dadurch steht eine Vielzahl der gegebenen Anzahl an Trainingsbildern zur Verfügung, mit welchen das Netz trainiert wird.

Als Datenbasis wird eine zur Verfügung gestellte Kundenbibliothek verschiedener CAD-Gusskonstruktionen mit jeweils 30 Schnittansichten von günstigen und ungünstigen Knotengestaltungen verwendet sowie durch Data Augmentation Algorithmen vormodifiziert. Das Training wird GPU-basiert in Google Colab durchgeführt.

Das Potential der Methode wird bewusst bei einer kleineren Datenbasis ermittelt, da in der Regel der Teilestamm in einem konstruierenden Betrieb keine unendlich große Datenbasis ergibt. Diese Datenbasis stellt im neuronalen Netz die Grundlage für die Objektdetektion durch fallbasiertes Schließen dar. Im fallbasierten Schließen werden Problemstellungen durch einen Analogieschluss gelöst [16]. Dabei ist ein ständiger Abgleich von neuen und gelernten Fällen innerhalb der Datenbasis notwendig [17]. Auch hier existiert eine Analogie des maschinellen Lernens zur Problemlösung menschlicher Verhaltensweisen. Ein Mensch, der mit einem neuen Problem konfrontiert wird, versucht Lösungsvorgehen zu ähnlichen Problemen aus seiner persönlichen Vergangenheit zu ermitteln. Sollte ein ähnliches Problem schon einmal erfolgreich gelöst worden sein, wird der Mensch versuchen, das aktuelle Problem ähnlich zu lösen. Ist dies nicht der Fall, ist eine Erweiterung der Datenbasis (für Mensch bzw. Maschine) zwingend erforderlich [18]. Neuronale Netze können durch dieses Vorgehen Ähnlichkeiten gewichten und letztendlich abbilden.

4 Benutzerdefinierte Objekterkennung

Im Folgenden werden die Ergebnisse der benutzerdefinierten Objekterkennung mittels YOLOv4 dargestellt und das neuronale Netz anhand verschiedener Beispiele validiert. Abschließend wird die Methode kritisch bewertet und das Potential analysiert.

4.1 Prozesskette und Ergebnisse

Durch die Data Augmentation Features von YOLOv4 wird das Netz, ausgehend von der ursprünglich generierten Datenbasis, mit insgesamt 256.000 individuellen Datensätzen trainiert und optimiert. Innerhalb der Prozesskette werden dafür automatisiert 2D-Schnittansichten von den Gussgehäusen in der Konstruktionsumgebung erstellt und in einem gängigen Bildformat gespeichert. Weitere Nachbearbeitungen der Bilddaten zur Durchführung der benutzerdefinierten Objekterkennung sind nicht notwendig. Das trainierte neuronale Netz kann auf den Speicherort der zu überprüfenden Schnittansicht zugreifen und diese anschließend analysieren. Anhand einer Gegenüberstellung von zwei Konstruktionsschnitten (analog zum eingangs verwendeten Beispiel in Bild 3), die dem Netz unbekannt sind, lässt sich das Potential der Objektdetektionsmethode abschätzen (Bild 4).

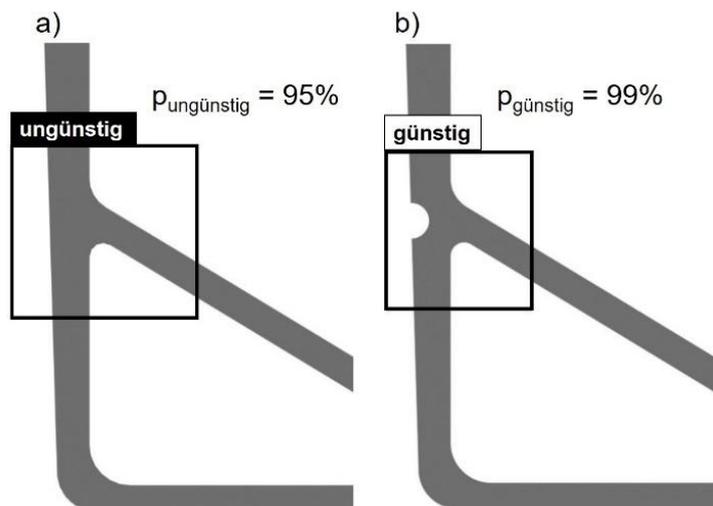


Bild 4: Vorhersage des neuronalen Netzes a) einer ungünstigen Knotenkonstruktion und b) einer günstigen Knotenauflösung mit den jeweiligen Vorhersagewahrscheinlichkeiten.

Es werden für jedes Bild ca. 1.500 Vorhersagen getroffen, von denen alle unter einer Vorhersagewahrscheinlichkeit von 30% verworfen werden. Um eine dreidimensionale Ergebnisdarstellung als ergänzende bzw. zusätzliche Fehlerüberprüfung vorzunehmen, lassen sich weitere Methoden des Image Processing an die zweidimensionale

Objektdetektion anschließen. Bild 5 zeigt dafür zwei Varianten, die eine manuelle Überprüfung erlauben. Dadurch ist eine Aussage darüber möglich, ob eine vom neuronalen Netz erkannte Fehlerstelle des Beispiels in Bild 4a) tatsächlich einen Fehler darstellt. Durch diese rekonstruierenden Methoden der 2D-Schnittansichten in ein 3D-Volumenmodell können die detektierten Fehler dreidimensional verifiziert werden. Beispielsweise lassen sich die Schnittansichten mithilfe von Filtern so bearbeiten, dass das Volumen nach den Intensitäten der Materialanhäufungen durchleuchtet werden kann. In Kombination mit verschiebbaren Ebenen (in alle Koordinatenrichtungen) durch das Bauteil kann so eine zuverlässige Entscheidung durch den Anwender bzw. Konstrukteur getroffen werden.

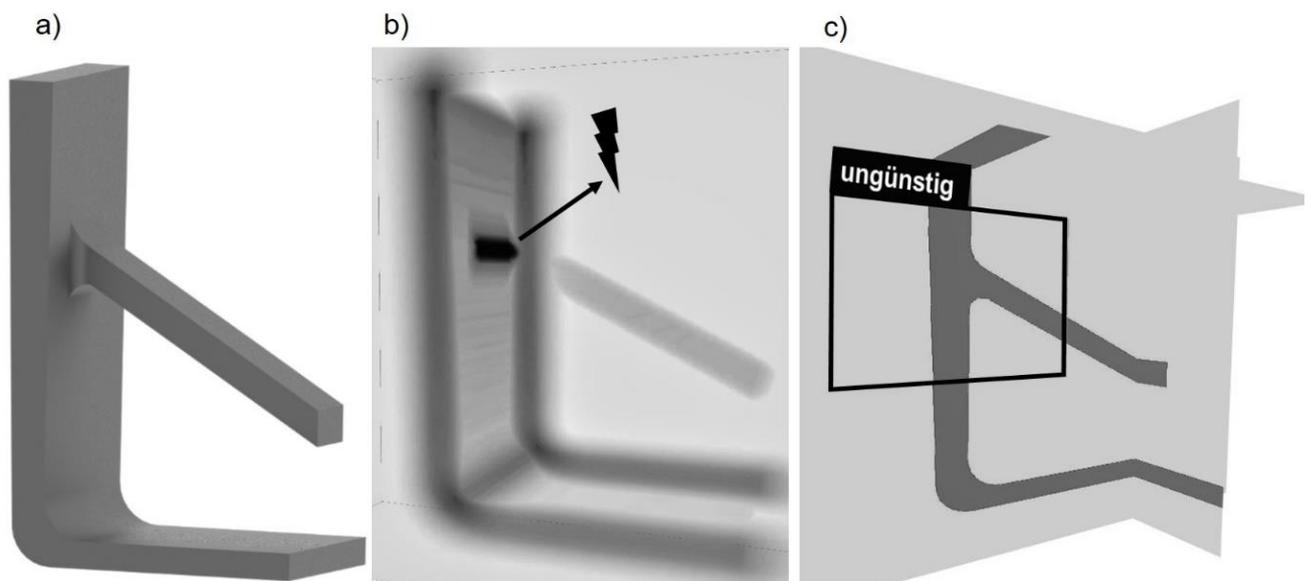


Bild 5: Weiteres Image Processing des a) realen Bauteils mit einer ungünstigen Knotengestaltung zu b) einer gefilterten und intensitätsbasierten Darstellung der Materialanhäufungen sowie c) einer Darstellung mit verschiebbaren Fenstern durch alle Ebenen.

Mithilfe der intensitätsbasierten Darstellung ist eine Weiterverarbeitung der Informationen des Image Processing möglich. Sollte der Anwender die detektierte Materialanhäufung endgültig als Fehlerstelle bewerten, ist eine weitere Filterung nach den Zentren der maximalen Dichteintensitäten möglich. Diese Maximalstellen können in ein Koordinatensystem transformiert und beispielsweise als Marker in die CAD-Konstruktion zurück überführt werden. Hierbei ist auf eine Homogenität des Schnittebenen- und des globalen Konstruktions-Koordinatensystems zu achten. Allerdings zeigen sich bei Robustheitstests des Netzes einige Schwierigkeiten. Im Vergleich zu einem einzigen Knoten entlang einer Wand stellen sich deutlich geringere Vorhersagewahrscheinlichkeiten bei

einer komplizierteren Geometrie mit günstigen und ungünstigen Knotenkonstruktionen in näherer Umgebung zueinander heraus (Bild 6).

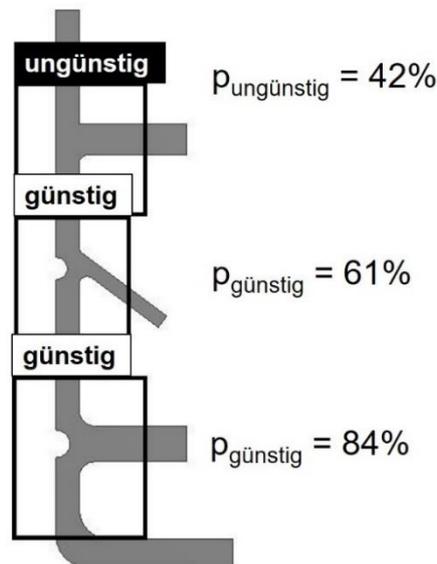


Bild 6: Vorhersage des neuronalen Netzes einer komplizierteren Wandgeometrie mit günstigen und ungünstigen Knotenpunkten sowie den jeweiligen Vorhersagewahrscheinlichkeiten.

Es lässt sich feststellen, dass die Ungenauigkeit der Vorhersagewahrscheinlichkeit mit der Kompliziertheit der Geometrie korreliert. Mit einem gewissen Optimierungsaufwand der Netzparameter lassen sich diese Vorhersagewahrscheinlichkeiten erhöhen.

4.2 Kritische Würdigung des Vorgehens

Grundsätzlich gilt für neuronale Netze der Zusammenhang „je größer die Datenbasis, desto zuverlässiger das Ergebnis“. Dies ist vor allem bei der aufwendigen Aufgabe der Objektdetektion in Bildern zu beachten. Die Objektdetektion mithilfe des YOLOv4-Algorithmus funktioniert für die Datenbasis COCO (Common Objects in Context – 330.000 Bilder, davon über 200.000 gelabelt, über 1.500.000 Objekt-Datensätze in 80 Objektklassen) außergewöhnlich gut. An der Masse der Daten in COCO zeigt sich auch der Grund: Für die 80 Objektklassen, bezogen auf die Objektlabel, sind durchschnittlich knapp 18.000 Datensätze für jede Kategorie vorhanden. Im Rahmen dieser Arbeit wurde das neuronale Netz hingegen lediglich mit 30 Datensätzen pro Objektklasse trainiert. Trotzdem zeigen die Ergebnisse bereits ein wesentliches Potential dieser Methode. Ein maximal robustes neuronales Netz braucht jedoch eine Vielzahl von günstigen und ungünstigen Knotenkonstruktionen, welche die derzeitige Datenlage bei weitem übertreffen muss.

Eine Fragestellung, die sich aus der Diskussion über die Größe der Datenbasis ableiten lässt, ist die Erweiterbarkeit auf weitere Klassen. Neben der gießgerechten Auflösung von Knotenpunkten gibt es weitere Richtlinien, wie beispielsweise schematische Querschnittsabstufungen und das Vermeiden von Hinterschneidungen. Sollten all diese möglichen Fehlerquellen zuverlässig detektiert werden können, ist eine aufwändige bzw. umfassende Datenbasis notwendig, die auch eine lokale Überlagerung verschiedener Fehler sicher erkennen muss. Auch hier ist der Nutzen mit dem Aufwand zur Erzeugung dieser Datenbasis ins Verhältnis zu setzen.

Weitere Herausforderungen finden sich in der Transformation der zweidimensionalen Schnittansicht in ein dreidimensionales Volumen. Dieser Schritt ist bei einer ganzheitlichen und automatisierten Betrachtung notwendig, um alle Gestaltungsrichtlinien vollständig in die Einbindung der Regeln zu integrieren. So sind zum Beispiel Materialanhäufungen nicht als solche zu klassifizieren, wenn darüber (aber außerhalb der analysierten Schnittebene) ein Steiger liegt. Eine mögliche Lösung hierfür wäre die Speicherung der detektierten Objekte in 2D und ein anschließender regelbasierter Abgleich der Ergebnisse in 3D.

5 Zusammenfassung und Ausblick

Die im Rahmen dieses Beitrags vorgestellte Methode der automatisierten Objektdetektion zur Erkennung von Konstruktionsfehlern mittels *Deep Learning* zeigt ein grundsätzlich positives Potential. Durch dessen Anwendung lassen sich Konstruktionsansichten auch zum derzeitigen Kenntnisstand geeignet vorselektieren. Allerdings ist eine manuelle Nachbearbeitung und Überprüfung der Ergebnisse zwingend erforderlich. Des Weiteren sind Verbesserungen der Robustheit des Netzes sowie die regelbasierte Transformation der zweidimensionalen Ergebnisse in ein dreidimensionales Objektdetektionssystem sinnvoll. Eine Möglichkeit zur dreidimensionalen Objektdetektion ist die semantische Segmentierung von 3D-Volumen durch die Verwendung von Punktwolken aus den CAD-Konstruktionen. In der Literatur sind potentialabschätzende Versuche der dreidimensionalen Objektsegmentierung verschiedener Umgebungen aufgeführt. Mithilfe dieser Anwendungen lässt sich beispielsweise zwischen Bäumen, Flugzeugen und Gebäuden im dreidimensionalen Raum unterscheiden [19].

Von diesem Vorgehen wurde allerdings an dieser Stelle abgesehen, da bei Konstruktionsfehler nicht zwischen beispielsweise Menschen und Autos unterschieden wird, sondern zwischen günstigen und ungünstigen Lösungen – also Objektklassen mit teilweise

marginalen Unterschieden. Eine Untersuchung der CAD-Konstruktionen im dreidimensionalen Raum ist demnach erst zielführend, wenn die Methode in 2D für den gegebenen Anwendungsfall hinreichend beherrschbar ist.

Da der Einsatz dieser Methode für zweidimensionale Schnittansichten mit einem anschließenden dreidimensionalen, manuellen Post-Processing vielversprechend ist, ist eine Überprüfung dieser Konstruktionsmerkmale im dreidimensionalen Volumen das Ziel kommender Untersuchungen.

Literatur

- [1] Gembarski, Paul C.: *On the Conception of a Multi-agent Analysis and Optimization Tool for Mechanical Engineering Parts*. In: Agents and Multi-Agent Systems: Technologies and Applications 2020, 21. Mai 2020, S. 93-102.
- [2] Kurz, Ulrich; Hintzen, Hans; Laufenberg, Hans: *Konstruieren, Gestalten, Entwerfen: Ein Lehr- und Arbeitsbuch für das Studium der Konstruktionstechnik*. Wiesbaden. Vieweg+Teubner Verlag, 2009, ISBN: 978-3-8348-0219-4.
- [3] Dheeba, Jacob; Singh, N Albert; Selvi, S Tamil: *Computer-aided detection of breast cancer on mammograms: A swarm intelligence optimized wavelet neural network approach*. In: Journal of Biomedical Informatics, 2014.
- [4] Jesan, John P.; Lauro, Donald M: *Human Brain and Neural Network Behavior: A Comparison*. In: Ubiquity, November 2003.
- [5] Fatt, Irving; Weissman, Barry A.: *Physiology of the Eye*. Butterworth-Heinemann, 1992, ISBN: 978-0-7506-9085-0.
- [6] Kosslyn, Stephen M.: *Mental images and the Brain*. In: Cognitive Neuropsychology, 05. Januar 2007, S. 333-347.
- [7] Krizhevsky, Alex; Sutskever, Ilya; Hinton, Geoffrey E.: *ImageNet Classification with Deep Convolutional Neural Networks*. In: Advances in neural information processing systems, Mai 2012, S. 1097-1105.
- [8] Wang, Tao; Wu, David J.; Coates, Adam; Ng, Andrew Y.: *End-to-End Text Recognition with Convolutional Neural Networks*. In: 21st International Conference on Pattern Recognition (ICPR 2012), 11.-15. November 2012.
- [9] LeCun, Yann; Bottou, Léon; Bengio, Yoshua; Haffner, Patrick: *Gradient-Based Learning Applied to Document Recognition*. In: Proceedings of the IEEE, 1998, S. 2278-2324.
- [10] Ayyadevara, Kishore: *Pro Machine Learning Algorithms: A Hands-On Approach to Implementing Algorithms in Python and R*. Apress, 2018, ISBN: 978-1-4842-3564-5.
- [11] Plappert, Stefan; Gembarski, Paul Christoph, Lachmayer, Roland: *The Use of Knowledge-Based Engineering Systems and Artificial Intelligence in Product Development: A Snapshot*. In: Information

Systems Architecture and Technology: Proceedings of 40th Anniversary International Conference on Information Systems Architecture and Technology (ISAT 2019), 15.-17. September 2019.

- [12] Conrad, Klaus-Jörg: *Grundlagen der Konstruktionslehre: Maschinenbau-Anwendungen und Orientierung auf Menschen*. Carl Hanser Verlag, München, 2018, ISBN: 978-3-446-45321-0.
- [13] Bochkovskiy, Alexey; Wang, Chien-Yao; Liao, Hong-Yuan Mark. *YOLOv4: Optimal Speed and Accuracy of Object Detection*. In: arXiv:2004.10934, 23. April 2020.
- [14] Girshick, Ross; Donahue, Jeff; Darrell, Trevor; Malik, Jitendra: *Rich feature hierarchies for accurate object detection and semantic segmentation*. In: arXiv:1311.2524, 22. Oktober 2014.
- [15] Redmon, Joseph; Divvala, Santosh; Girshick, Ross; Farhadi, Ali: *You Only Look Once: Unified, Real-Time Object Detection*. In: arXiv:1506.02640, 8. Juni 2015.
- [16] Richter, Michael M.; Weber, Rosina: *Case-Based Reasoning: A Textbook*. Berlin Heidelberg, Springer-Verlag, 2013, ISBN: 978-3642401664.
- [17] Aamodt, Agnar; Plaza, Enric: *Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches*. In: AI communications 7, 1994.
- [18] Siqueira, Renan; Bibani, Mehdi; Duran, Deniz; Mozgova, Iryna; Lachmayer, Roland; Behrens, Bernd-Arno: *An adapted case-based reasoning system for design and manufacturing of tailored forming multi-material components*. In: International Journal on Interactive Design and Manufacturing (IJIDeM), 24. April 2019, S. 1175-1184.
- [19] Qi, Charles R.; Su, Hao; Mo, Kaichun; Guibas, Leonidas J.: *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. In: arXiv:1612.00593, 2. Dezember 2016.