

Erkennung von Rissen mittels maschinellen Lernens

Entwicklung eines verteilten Systems für die Erkennung von Rissen in Bahnschwellen mithilfe von Künstlicher Intelligenz

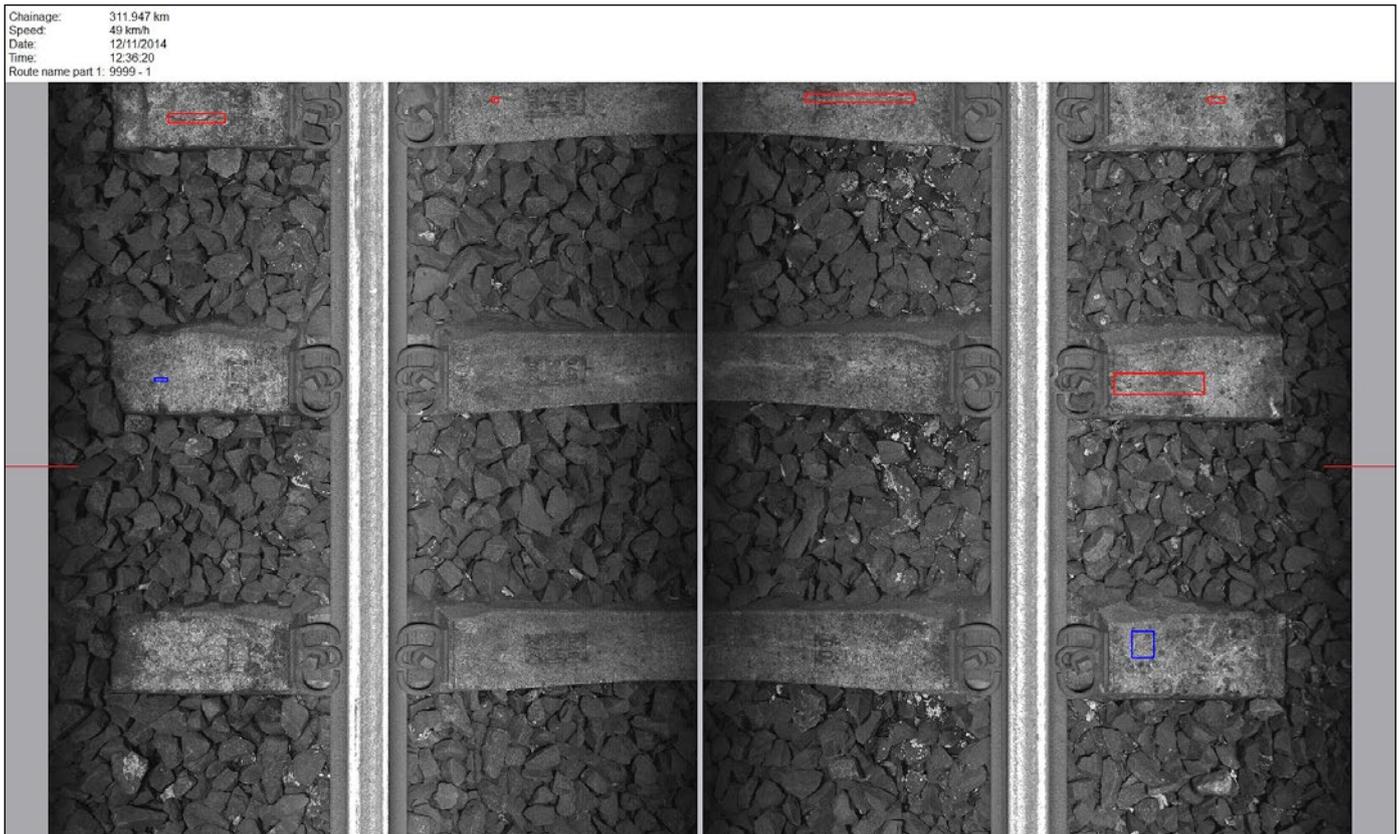


Abb. 1: Ansicht des Gleisbetts im Tool CrackCheck [2]

ANDREAS LEICH | NILS KORNFELD |
JOSHUA NIEMEIJER | MAX KAISER |
MARCEL JÄCKLE

Instandhaltung ist eines der zentralen Themen für den Betreiber eines Schienennetzes. Um einen planmäßigen Schienenverkehr sicherzustellen, muss die zugrunde liegende Infrastruktur betriebsbereit gehalten werden. Dies geschieht, indem der momentane technische bzw. betriebliche Zustand der Infrastruktur bestimmt wird und Schäden, wie z. B. Risse in Schwellen, rechtzeitig erkannt werden. Vor diesem Hintergrund wird in diesem Beitrag dargestellt, wie ein modernes Verfahren des maschinellen Lernens für die Erkennung von Rissen in Bahnschwellen in das bestehende Erfassungssystem eines Messzuges der DB Netz AG (DB Netz) integriert worden ist.

Einleitung

Im Streckennetz der DB Netz sind mehr als 62 Mio. Spannbetonschwellen verlegt, was einem Investitionsvolumen von mehr als 8 Mrd. EUR entspricht [1] (Abb. 1). Da der mechanisch einwandfreie Zustand der Schwellen wesentlich für die Gleisgeometrie und somit für die Sicherheit ist, ist seine regelmäßige und möglichst automatisierte Überprüfung erstrebenswert. Hinzu kommt, dass Schwellen bei frühzeitiger Erkennung der Risse repariert werden können. Ferner ist die Dokumentation von fertigungsbedingten Schäden innerhalb des Gewährleistungszeitraumes ein wirtschaftlicher Vorteil für den Netzbetreiber.

Bahnschwellen unterliegen einer ständigen mechanischen Belastung durch Befahrung sowie einer Beanspruchung durch die Witterung, was über die Zeit zu einem Ausfall der Schwellen führen kann. Daneben gibt es betontechnische Schädigungsmecha-

nismen, wie z. B. das Gipfstreifen, das auch neue Schwellen schnell ausfallen lässt. Ausprägungen dieses Verschleißes sind Risse in den Schwellen (Abb. 4). Mit zunehmender Anzahl und/oder Ausprägung der Risse degeneriert die Integrität der Schwelle und beeinträchtigt deren Funktionalität.

Daher muss der Zustand jeder einzelnen Schwelle regelmäßig erfasst und bewertet werden. Dies kann unter großem Aufwand durch eine von einem Menschen durchgeführte visuelle Inspektion erfolgen.

Die DB Netz betreibt einen Messzug, welcher das Gleisbett bei Geschwindigkeiten von bis zu 140 km/h abfotografiert und die Bilder in einer Datenbank bereitstellt [2]. Mittels digitaler Bildverarbeitung ist es damit möglich, Risse in den Bahnschwellen zu detektieren. Wie solche Verfahren üblicherweise funktionieren, wird z. B. in [3, 4] beschrieben. Allerdings ist klassische Bildverarbeitung in der optisch heterogenen Domäne eines Gleisbettes anfällig für Fehlde-

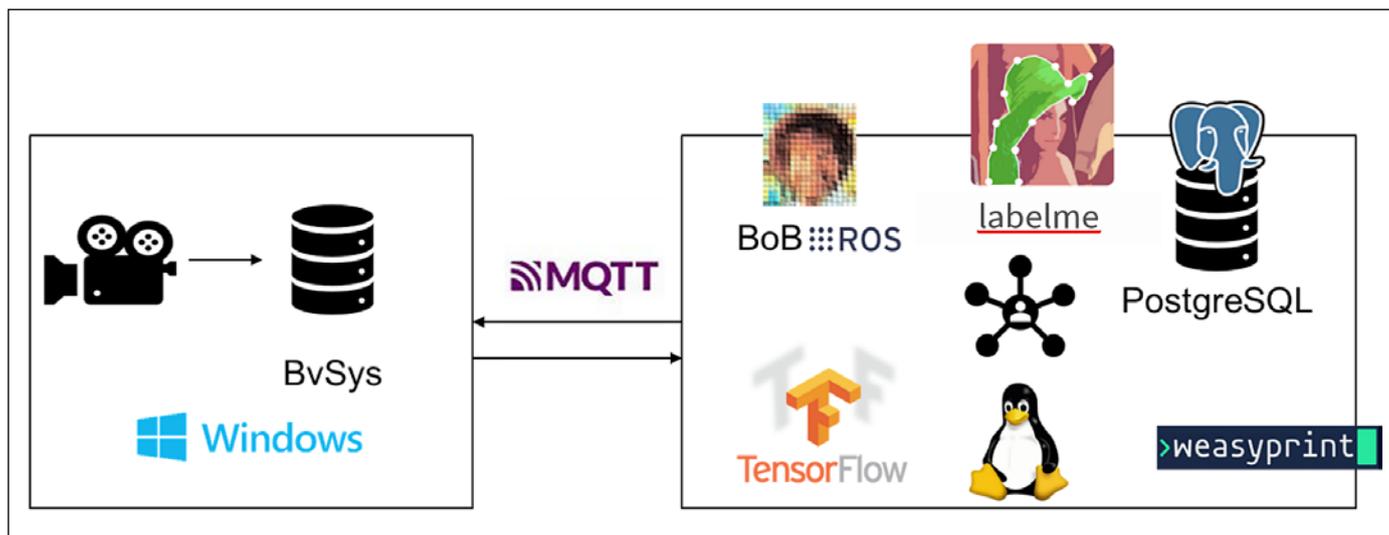


Abb. 2: Zusammenspiel der Komponenten

tektionen. Insbesondere können in erheblichem Umfang falsch positive Fehldetektionen entstehen, d.h. es wird ein Fehler detektiert, der real nicht vorliegt. Diese müssen durch einen Bearbeiter begutachtet werden, um tatsächliche Risse von Zweigen, Verschmutzung und ähnlichen Fehlern zu unterscheiden. Dies ist meist unwirtschaftlich.

Moderne Algorithmen des maschinellen Lernens, sogenanntes Deep-Learning mit Convolutional Neural Networks (CNN), versprechen eine hohe Erkennungsleistung und können erfolgreich bei der automatischen Identifikation von Rissen eingesetzt werden [5]. Besonders vorteilhaft ist die Möglichkeit, ein System zu entwickeln, welches infolge manueller Eingriffe lernt und dadurch die Erkennungsrate steigert. Die Entwicklung eines solchen Systems wird in diesem Beitrag vorgestellt.

Systemkonzept

Aus Gründen der Erweiterbarkeit, der Austauschbarkeit, der Transparenz bei der Datenverarbeitung und der Kosteneffizienz wurde

auf bereits vorhandene und in Betrieb befindliche Komponenten sowie auf frei verfügbare Software zurückgegriffen (Tab. 1).

Für das Zusammenspiel der Komponenten wurde neben der unmittelbaren Einbindung der jeweiligen Bibliotheken als Softwarebibliothek auf lose Kopplung der Softwarekomponenten gesetzt. Das bedeutet, dass unabhängig kompilierbare Softwarekomponenten, sogenannte Knoten, interagieren und gemäß dem Publisher-Subscriber Entwurfsmuster [6] erst zur Laufzeit zum Gesamtsystem zusammengeschaltet werden.

Funktionsumfang, Systemkomponenten und Integration

Um möglichst viel Funktionalität aus dem Bestand, also dem CrackCheck-System nutzen zu können, wurde die Software zunächst in zwei Sub-Systeme zerlegt, welche mittels MQTT miteinander kommunizieren (Abb. 2). Hierbei fungiert CrackCheck als Bildaufnahmesystem und Bilddatenbank.

Eine Adaptierung unter Windows interagiert mit dem ImageServer-API von CrackCheck.

Das erlaubt Zugriff auf die Befahrungsdatenbank, welche Bilddaten mit Informationen zur Streckenverortung verzeichnet. Somit werden über die MQTT-Verbindung Dienste wie der Abruf einer Liste aller Befahrungen, ihrer jeweiligen Attribute, relativer Positionen sowie der Bilder selbst realisiert.

Diese Kommunikation erlaubt es, das unter Windows laufende System CrackCheck mit den unter Linux laufenden Softwarekomponenten für Offline-Prozessierung, Annotation, Training von Netzen, Ergebnisberichtsdatenbank und Ergebnisberichterstellung zu koppeln. In den folgenden Abschnitten wird darauf detailliert eingegangen.

Bedienfluss, Grafische Benutzeroberfläche (GUI) und Reporting

Als Algorithmus für die eigentliche Erkennung (Inferenz) der Risse wird ein Mask R-CNN mit einer Resnet50 backbone Architektur [7] verwendet. Für dieses liegt eine TensorFlow-Implementierung vor. In [5] trat beim Einsatz eines solchen Verfahrens eine falsch-positiv Rate von ca. 50 % auf, wenn unbekannte Arten von Ris-

Komponente	Entwickler	Genutzte Funktionalität
CrackCheck	bvSys GmbH	Bildaufnahme, Datenbank Befahrungsbilder mit API
ROS	Willow Garage, Stanford Artificial Intelligence Laboratory	Build System, Integrationsframework, Prozessübergreifende Kommunikation
BoB-ROS	DLR, Institut für Verkehrssystemtechnik	Erweiterung von ROS speziell für die Sensorverarbeitung
labelme	Kentaro Wada	Werkzeug mit grafischer Benutzeroberfläche für die Annotation
PostgreSQL	PostgreSQL Global Development Group	Datenbankmanagementsystem für die Ergebnisdatenbank
TensorFlow	Google Brain Team	Training des CNN, Inferenz CNN
MQTT, Paho Implementierung	OASIS Message Queuing Telemetry Transport Technical Committee, Eclipse Paho messaging project	Asynchrones, bi-direktionales Protokoll zur Übertragung von Anfragen und Befahrungsbildern
Jinja weasy	Armin Ronacher	Template Generator für HTML & CSS
Print	Kozea	HMTL zu PDF-Konverter zur Erzeugung des Ergebnisreports
OpenCV	Intel, Willow Garage	Bildverarbeitungsbibliothek für metrologische Auswertung

Tab. 1: Systemkomponenten

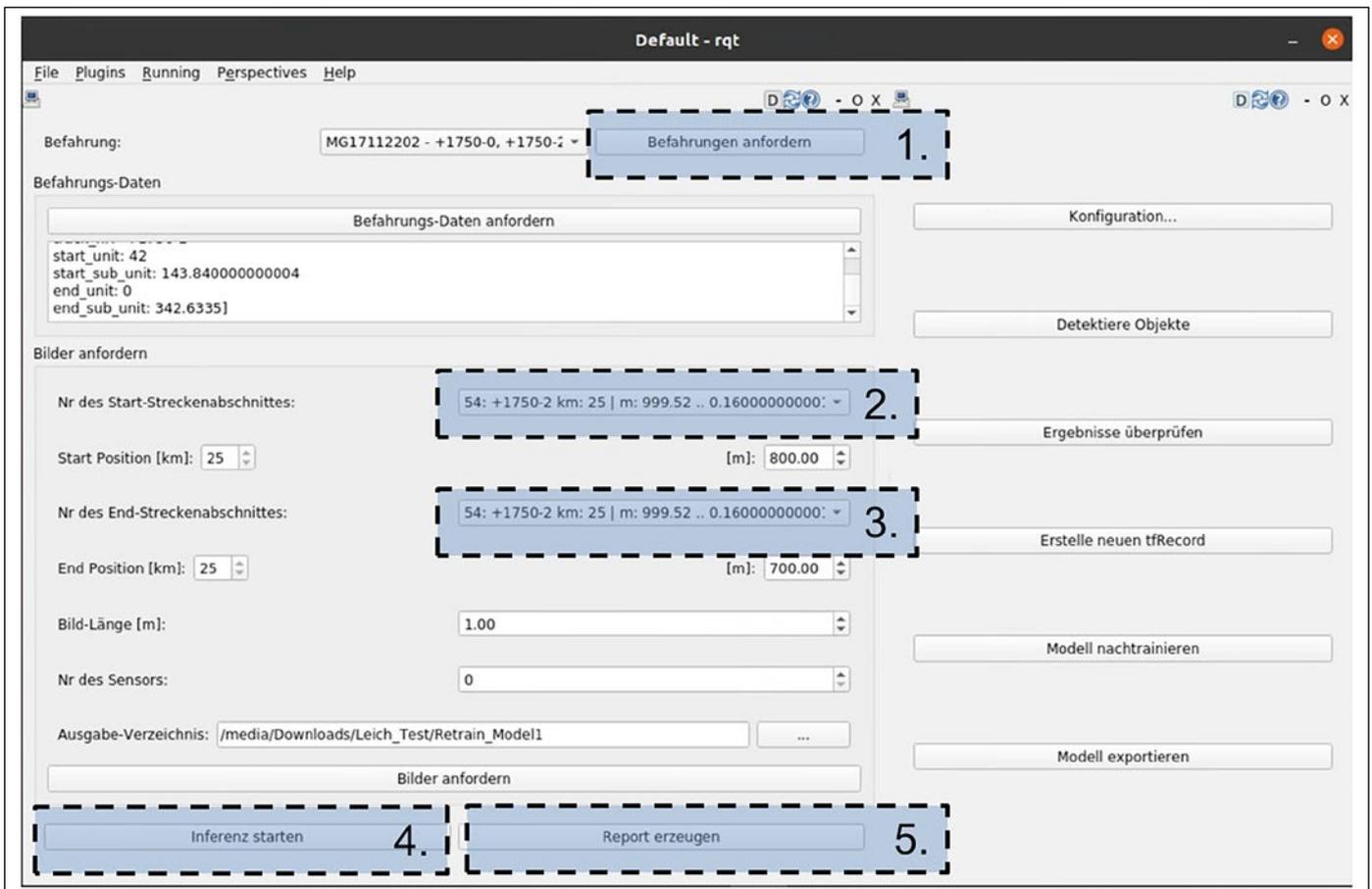


Abb. 3: GUI des Systems

sen auftreten (z.B. in Holzschwellen). Ein weiteres Training mit repräsentativem Bildmaterial ließ die falsch-positiv Rate auf ca. 25 % sinken. Dies macht deutlich, dass stetiges Nachtrainieren eine große Bedeutung für den erfolgreichen Betrieb der Erfassungslösung hat. Das hier vorgestellte System realisiert daher den Bedienfluss für eine solche kontinuierliche Verbesserung der Trainingsdaten im produktiven Betrieb, der wie folgt abläuft:

1. Der Bearbeiter erzeugt Ergebnisberichte und stellt diese den Anlagenverantwortlichen (ALV) zur Verfügung.
2. Die ALV begutachten die Ergebnisse der Inferenz und teilen dem Bearbeiter etwaige Fehler mit.
3. Der Bearbeiter sichtet die Erkennungsfehler der Inferenz, korrigiert diese und legt sie als neue Trainingsdaten ab.
4. Je nach Bedarf wird das System mit den aktualisierten Trainingsdaten erneut trainiert. Dies ist mit der im Modell verkörperten Erfahrung zunehmend seltener nötig.

Für das nachträgliche Training müssen die fehlerhaften Daten zunächst korrigiert werden. Hierzu besteht die Möglichkeit, sich die betreffende Stelle als Bilddatei herunterzuladen (Abb. 3 „Bilder anfordern“) und die Inferenz lokal zu wiederholen (Abb. 3 „Detektiere Objekte“). Mittels „Ergebnisse prüfen“ wird das Annotierwerkzeug labelme gestartet, um das

fehlerhafte Erkennungsergebnis nachträglich händisch zu korrigieren (Abb. 4), mittels „Erstelle neuen tfRecord“ dann eine Trainingsdatenbank im TensorFlow-Format erstellt und mittels „Modell nachtrainieren“ das nachträgliche Training angestoßen.

Algorithmen und Datenverarbeitung

In den folgenden Abschnitten wird detaillierter auf Konzepte der Algorithmen eingegangen, welche sich im Rahmen der Realisierung als notwendig erwiesen haben. Die Beschreibung folgt der Verarbeitung. Die Verarbeitungskette beginnt mit der Inferenz, welche parallel auf mehreren Grafikkarten der Workstation läuft. Das Ergebnis der Inferenz ist eine mit Trefferbildern und -polygonen befüllte Ergebnisdatenbank. Auf dieser werden metrologische Analysen durchgeführt, um die Rissgeometrien für einen Ergebnisbericht zu bestimmen. Nach Sichtung dessen werden ggf. aus Erkennungsfehlern neue Trainingsdaten generiert.

Parallelisierung

Die Verarbeitung der Bilder ist rechenintensiv. Um bei der Verarbeitung längerer Befahrungen akzeptable Verarbeitungsgeschwindigkeiten erreichen zu können, müssen alle Möglichkeiten zur Parallelisierung genutzt werden. Diese wurde durch zeitliche Staffelung, sogenanntes Pipelining, und Verteilung der Infe-

renzprozesse auf mehrere Grafikprozessoren (GPU) erreicht (Abb. 5).

Pipelining bedeutet in diesem Fall, dass das nächste Bild bereits übertragen und entpackt wird (Image Acquisition), während für das aktuelle Bild eine Inferenz stattfindet (Inference) und das Ergebnis der Verarbeitung des vergangenen Bildes in die Datenbank geschrieben wird (DB Writer).

Verteilung der Inferenzprozesse bedeutet, dass aufeinanderfolgende Bilder gleichzeitig auf verschiedenen GPU verarbeitet werden.

Metrologische Auswertung

Mit der Segmentierung durch die Inferenz werden die Umrisspolygone der gefundenen Vorkommnisse zusammen mit dem Pixelmaßstab, der Streckenreferenzierung und den Rohbildern der Treffer in der Ergebnisdatenbank abgelegt. Ab diesem Schritt können mittels deterministischer, also „klassischer“ Bildverarbeitung die geometrischen Parameter der Risse ausgewertet werden (Abb. 6). In einem ersten Schritt werden die Polygone in eine Pixelmaske konvertiert, und die Fläche in Pixel wird berechnet. Aus dieser kann für jeden Punkt per Abstandstransformation der euklidische Abstand zum Perimeter, also der Rissumfang, berechnet werden. Mittels des gefundenen Maximalwertes kann somit die maximale Rissweite und deren Ort inner-



Abb. 4: Riss in Betonschwelle, Schienenbefestigungsmittel, Segmentierung

halb des Risses ermittelt werden. Auch die Länge der Rissperimeter wird aus der Pixelmaske ermittelt. Für die Bestimmung einer praktikablen Risslänge wird der Algorithmus „Longest Shortest Path“ angewendet. Hierfür wird ein Riss zunächst skelettiert. Die Mitte des Risses wird dabei als Linie, sogenannte Skelettlinie, approximiert. Die Gesamtlänge der Skelettliniensegmente wäre bereits eine mögliche Metrik zur Feststellung der Risslänge. Dabei würden aber verästelte Risse eine große Risslänge suggerieren. Da aber die Länge der Grobgestalt eines Risses eher

als maßgeblich für die Schwächung des tragenden Querschnittes angesehen wird, wird der vorgenannte Algorithmus auf das Netz von Risssegmenten angewendet. Hierbei wird der längste Pfad über die Knotenpunkte zwischen den einzelnen Risssegmenten ausgewählt. Nebenverästelungen werden so automatisch ignoriert. Sollte der Grad der Verästelung trotzdem von Interesse sein – etwa zur Charakterisierung von Netzzissen –, ließen sich nachträglich die bereits berechneten Perimeterlängen in Beziehung zu den Rissflächen setzen. Im letzten Schritt werden

alle berechneten Größen mit dem Pixelmaßstab kalibriert und in der Ergebnisdatenbank abgelegt.

Erstellung von Ergebnisreports

Die gefundenen und metrologisch ausgewerteten Risse sollen dem ALV in leicht interpretierbarer Form zur Verfügung gestellt werden. Zum derzeitigen Stand geschieht dies als Ergebnisreport in Form von PDF-Dateien. Jeder Report hat eine erste Seite mit Basisinformationen zur Befahrung wie z.B. Befahrungsname, -zeitpunkt und -länge. Ferner werden die Start- und

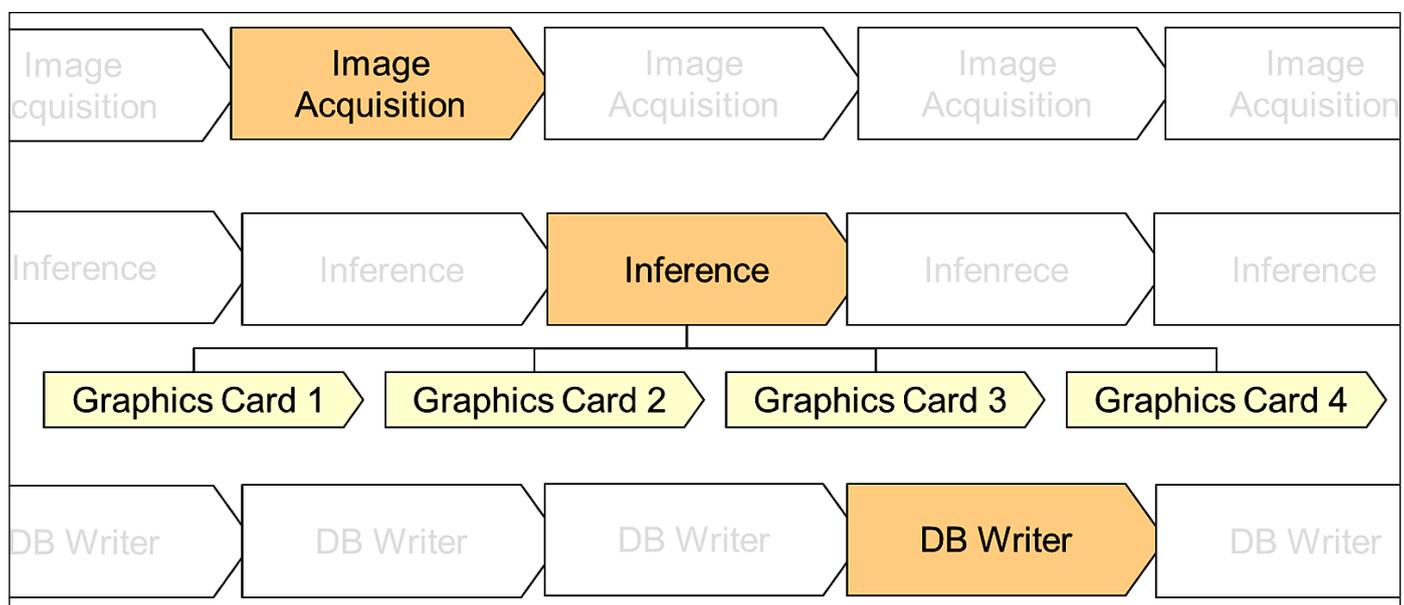


Abb. 5: Pipelining und Parallelisierung im System

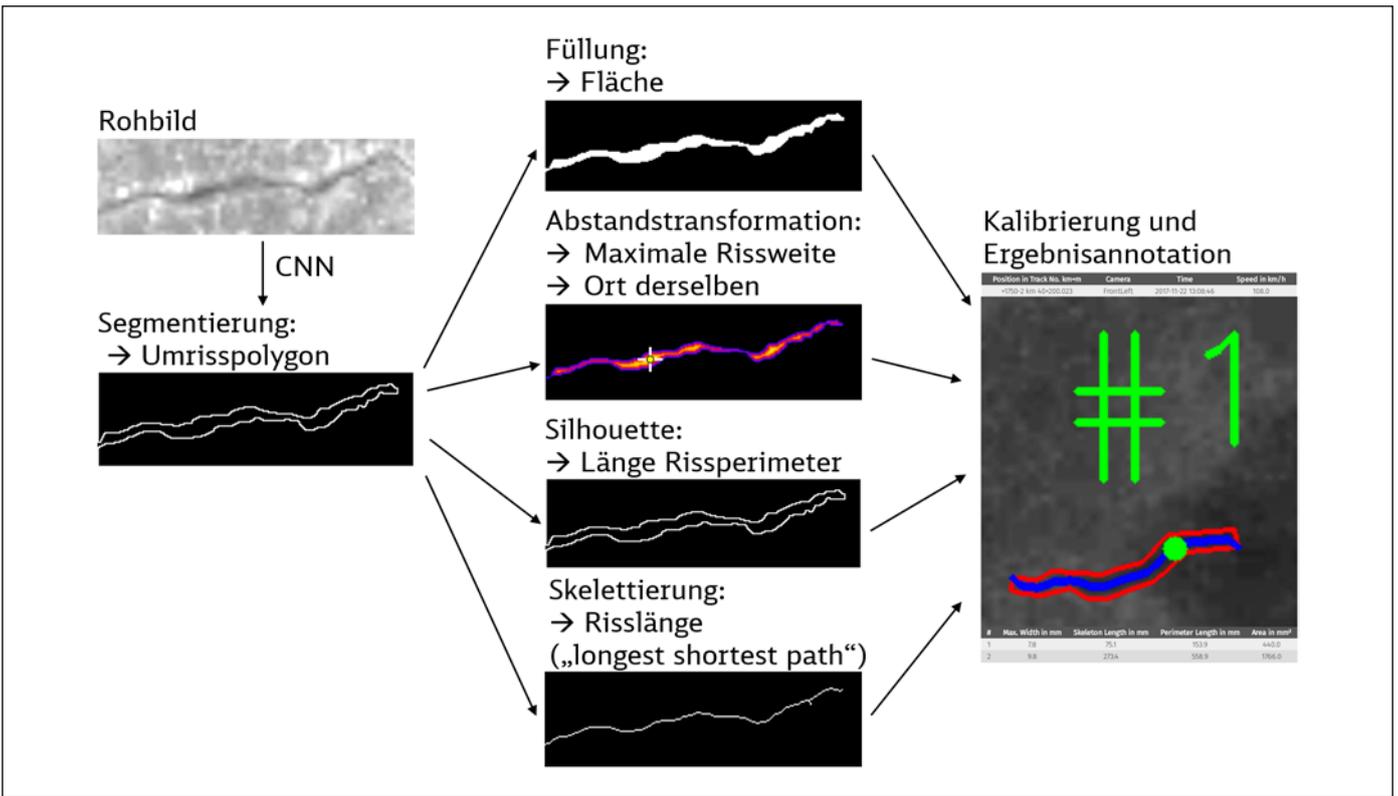


Abb. 6: Metrologie

Endpositionen der Befahrung und die Anzahl der gefundenen Risse angegeben. Für einen schnellen Überblick sorgt darüber hinaus ein maßstäblich dargestelltes Streckenband, auf dem Fundstellen als Punkte visualisiert werden. Hierauf folgt für jedes Bild mit Rissen eine stan-

dardisierte Darstellung von Bildposition und Zeit. Auf dem Bild selbst werden die Rissperimeter, -mittellinien und Orte der maximalen Rissweite mit Nummerierung als „Overlay“ dargestellt. Die zugeordneten Messwerte werden darunter tabellarisch aufgeführt (Abb. 7).

Zusammenfassung und Ausblick

In diesem Beitrag wurde ein System vorgestellt, welches die Erkennung von Rissen mittels modernem Deep-Learning in der Produktivumgebung eines Messzuges ermöglicht. Dieses fügt sich in die bestehen-

DLR SLEEPER INSPECTION 2.0

Report on Cracks in Concrete Sleepers

Inspection Run

Run Name: MG17112202
 CrackCheck ID: {bb0bbb90-a6f2-4b46-8f5d-6bba9e3d93e4}
 Date and Time: 2017-11-22 13:07:46
 Operator:
 Length of Run: 41.867 km

Track Overview

Start Position in Track No. km+m	End Position in Track No. km+m	Crack Count
+1750-2 km 41+600.0	+1750-2 km 4+300.0	4

Analyzed portions of tracks.

Graphical representation of analyzed portions of tracks with tagged cracks (orange spots).

Maßstab Orangener Punkt: Riss Graphische Streckenübersicht

Gesamtzahl an Rissen

List of Images with Cracks

Position in Track No. km+m	Camera	Time	Speed in km/h
+1750-2 km 40+200.023	FrontLeft	2017-11-22 13:08:46	108.0

Streckenposition Bild Nummerierte Risse

Messwerte für jeden Riss

#	Max. Width in mm	Skeleton Length in mm	Perimeter Length in mm	Area in mm ²
1	7.8	75.1	153.9	440.0
2	9.8	273.4	558.9	1766.0

Abb. 7: Aufbau Ergebnisreport

Homepageveröffentlichung unbefristet genehmigt für Deutsches Zentrum für Luft und Raumfahrt e.V., DB Systemtechnik GmbH, DB Netz AG / Rechte für einzelne Downloads und Ausdrücke für Besucher der Seiten genehmigt / © DWV Media Group GmbH

de Verarbeitungskette der DB Netz ein. Es ermöglicht die Auswertung von Befahrungen, die Erzeugung von Berichten über das Ergebnis sowie die Ableitung zusätzlicher Trainingsdaten aus Erkennungsfehlern und ein nachträgliches Training des Mustererkennungsalgorithmus. Derzeit findet der Hochlauf des Systems in einem Testbetrieb statt. Dabei wird die noch geringe Qualität der Trainingsdaten verbessert, und es werden Softwarefehler behoben, die sich bei intensiver Nutzung offenbaren. Es besteht die Erwartung, dass die Risse nach weiteren Trainingsläufen präziser von den Risspolygonen umfasst werden. Ferner erlaubt es die bereits realisierte Ablage in eine relationale Datenbank, die

Entwicklung einzelner Risse über mehrere Befahrungen hinweg zu verfolgen sowie die spätere Prognose im Sinne einer „Predictive Maintenance“.

Die beschriebene Vorgehensweise beim Training des Systems konzentriert sich auf die Korrektur von falsch-positiven Detektionen. „Übersehene“ Risse bleiben jedoch für den Bearbeiter unerkannt. Eine gezielte Suche nach Kandidaten für falsch-negative Detektionen sollte somit Gegenstand weiterführender Untersuchungen sein. ■

QUELLEN

[1] Hentschel, V.: „Fahrwegstrategie der DB Netz AG“, gehalten auf der 3. Symposium Lebenszyklus System Betonschwelle 2017, 14. November 2017. Zugriffen: 22. November 2022. [Online]. Verfügbar unter: [https://](https://docplayer.org/72192880-Fahrwegstrategie-der-db-netz-ag-anlagenverfuehrgarkeit-im-lebenszyklus.html)

docplayer.org/72192880-Fahrwegstrategie-der-db-netz-ag-anlagenverfuehrgarkeit-im-lebenszyklus.html

[2] bvSys Bildverarbeitungssysteme GmbH, „CrackCheck - bvSys“. <https://www.bvsys.de/crack-check.html> (zugegriffen: 9. Dezember 2022)

[3] Yamaguchi, T.; Hashimoto, S.: „Fast crack detection method for large-size concrete surface images using percolation-based image processing“, Mach. Vis. Appl., Bd. 21, Nr. 5, S. 797–809, Aug. 2010, doi: 10.1007/s00138-009-0189-8

[4] Oliveira, H.; Correia, P. L.: „CrackIT — An image processing toolbox for crack detection and characterization“, in 2014 IEEE International Conference on Image Processing (ICIP), Okt. 2014, S. 798–802, doi: 10.1109/ICIP.2014.7025160

[5] Kornfeld, N.; Luber, A.; Leich, A.; Kaiser, M.; Schubert, L. A.; Groos, J. C.: „Zustandsüberwachung der Bahninfrastruktur mit KI“, Eisenbahn Ingenieur Kompendium, 6. Dezember 2019

[6] Grimm, R.: „Klassifikation von Mustern in der Softwareentwicklung | heise online“. <https://www.heise.de/blog/Klassifikation-von-Mustern-in-der-Softwareentwicklung-7203402.html> (zugegriffen: 9. Dezember 2022)

[7] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R.: „Mask R-CNN“, in 2017 IEEE International Conference on Computer Vision (ICCV), Okt. 2017, S. 2980–2988, doi: 10.1109/ICCV.2017.322



Andreas Leich

Gruppenleiter Geo-Datenverarbeitung und Informationsfusion
Deutsches Zentrum für Luft- und Raumfahrt e.V., Berlin
andreas.leich@dlr.de



Nils Kornfeld

Wissenschaftlicher Mitarbeiter
Deutsches Zentrum für Luft- und Raumfahrt e.V., Braunschweig
nils.kornfeld@dlr.de



Joshua Niemeijer

Wissenschaftlicher Mitarbeiter
Deutsches Zentrum für Luft- und Raumfahrt e.V., Braunschweig
joshua.niemeijer@dlr.de



Max Kaiser

Referent für Perzeptionstechnik
DB Systemtechnik GmbH, Minden
max.kaiser@deutschebahn.com



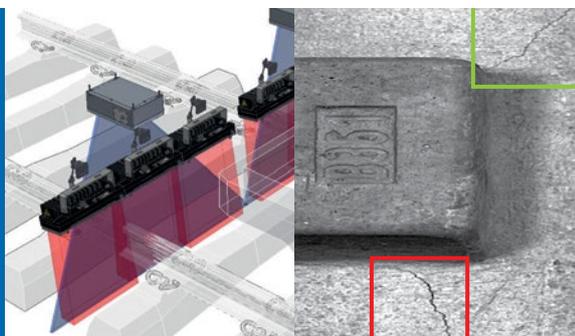
Marcel Jäckle

Systemingenieur Lichtraum/
Schwellendiagnose
DB Netz AG, Minden
marcel.jaekle@deutschebahn.com

Crack Check

KI-basierte Risserkennung

Spezialist für feste Fahrbahn
und Beton-Schwellen



Das System zeichnet sich aus durch:

- Hohe Auflösung bei hoher Geschwindigkeit
- Hohe Erkennungsleistung
- Geringe Fehldetektionsrate
- Leichte Bedienung; intuitiv bedienbares User Interface
- Modulares Konzept
- Geringer Wartungsaufwand
- Nachgewiesene Bahnfestigkeit
- Leichte Integrierbarkeit, da geringer Platzbedarf sowohl unter dem als auch im Fahrzeug
- Offenes System; dadurch leichte Adaption an landesspezifische Datenbanksysteme bzw. zentrale Datenerfassung auf den Fahrzeugen

bvSys Bildverarbeitungssysteme

☎ +49 421 460 461-0
✉ info@bvsys.de

Alle Informationen zu unseren Systemen finden Sie auf www.bvsys.de

bvSys – führend in der Entwicklung hochpräziser Bahninspektionssysteme