

Maschinelles Lernen zur Schadenserkenkung bei Leichtbaubrücken mit beweglichen Massen

Georgios Dadoulis¹, Thamer Al-Zuriqat² und Kosmas Dragos²

¹Laboratory of Experimental Strength of Materials and Structures, Aristotle University of Thessaloniki, Aristotle University Campus, 54124 Thessaloniki, Greece

²Institut für Digitales und Autonomes Bauen, Technische Universität Hamburg, Blohmstraße 15, 21079 Hamburg, Deutschland

E-Mail: dadoulis@civil.auth.gr

Abstract: Leichtbaubrücken sind Fahrzeugverkehr ausgesetzt, bei dem die Gewichte der Fahrzeuge – insbesondere bei Schwerverkehr – häufig mit dem Eigengewicht der Brücke vergleichbar sind. Diese beweglichen Massen führen zu einem komplexen dynamischen Verhalten, das über herkömmliche Strategien zur Schadenserkenkung nur unzureichend analysiert werden kann. Dieser Beitrag beschreibt ein Konzept zur Schadenserkenkung bei Leichtbaubrücken unter Verwendung von maschinellem Lernen (ML). Es wird ein ML-Modell zur Klassifizierung von Bauwerksschäden entwickelt, unter Verwendung von Simulationen trainiert und mittels realen Messdaten in einem Laborversuch validiert. Die Validierung zeigt, dass das ML-Modell geeignet ist, Bauwerksschäden an Leichtbaubrücken zu identifizieren, auch unter variierenden dynamischen Bedingungen, die durch die beweglichen Massen verursacht werden.

Keywords: Künstliche Intelligenz, Maschinelles Lernen, Schadenserkenkung, Fahrzeuglasten, Leichtbaubrücken, bewegliche Massen.



Erschienen in Tagungsband 35. Forum Bauinformatik 2024, Hamburg, Deutschland, DOI: 10.15480/882.13527
© 2024 Das Copyright für diesen Beitrag liegt bei den Autoren. Verwendung erlaubt unter Creative Commons Lizenz Namensnennung 4.0 International.

1 Einleitung

Die Materialreduzierung beim Brückenbau erfordert besondere Sorgfalt, um unerwartete Schwingungen zu vermeiden, wie sie beispielsweise bei der Eröffnung der Millennium Bridge (London, Großbritannien) im Jahr 2000 auftraten [1]. Brücken müssen den Schwingungen, die durch Betriebslasten wie dem Fahrzeugverkehr verursacht werden, standhalten [2]. Bei Leichtbaubrücken können Gewichte der Fahrzeuge mit dem Eigengewicht vergleichbar sein und werden als „bewegliche Massen“ betrachtet, was zu einem komplexen, nicht-stationären Problem führt, das vom klassischen Problem der „bewegliche Massen“ in der Strukturdynamik abweicht [3]. Herkömmliche Strategien zur Schadenserkenkung gehen kaum auf das Problem der beweglichen Massen ein, da experimentelle Techniken sowie Strategien zum Bauwerksmonitoring (engl. Structural Health Monitoring, SHM) in der Regel auf Datenanalysemethoden basieren, die auf stationäre Probleme abgestimmt sind [4]. Auch Studien zu beweglichen Lasten auf Balken erfassen das Problem der Schadenserkenkung bei beweglichen Massen nicht.

Die Schadenserkennung an Trägern mit beweglichen Massen wurde bisher nur sporadisch behandelt. In [5] wurden von einem intakten Brückenmodell vor und nach einer Schadensinduktion die Verschiebungen über die Zeit verglichen, in [6] wurden die Reaktionsgrenzen von Euler-Bernoulli-Trägern mit strukturellen Schäden aus theoretischer Sicht abgeschätzt, und in [7] wurden die Wavelet-Transformationskoeffizienten des experimentell abgeleiteten modalen Parameters zur indirekten Schadenserkennung verwendet. Bei den vorgenannten Ansätzen beruht die Schadenserkennung auf Verschiebungen oder auf der Genauigkeit der experimentell abgeleiteten modalen Parametern, was bei instationären Problemen kaum gewährleistet ist. Aus diesem Grund könnte die Schadenserkennung bei Trägern mit beweglichen Massen von einem Ansatz profitieren, der Beschleunigungsdaten verwendet, wie sie beim SHM gesammelt und von modernen Berechnungswerkzeugen verarbeitet werden [8].

Diese Arbeit stellt einen Ansatz zur Schadenserkennung vor, der auf früheren Arbeiten zur Fehlerdiagnose mithilfe maschinellen Lernens (ML) aufbaut [9]. Insbesondere wird ein ML-Modell zur Klassifizierung von Mustern in Beschleunigungsdaten für strukturelle Schadensszenarien entwickelt. Das in dieser Arbeit verwendete ML-Modell ist ein faltendes neuronales Netzwerk (engl. Convolutional Neuronal Network, CNN), das mit simulierten, „markierten“ Beschleunigungsdaten trainiert wird, die vordefinierten strukturellen Zustandsszenarien entsprechen. Nach Abschluss des Trainings wird die Klassifizierung mit realen Beschleunigungsdaten durchgeführt. Der hier vorgeschlagene Ansatz wird in einem Laborversuch validiert, die Fähigkeit des CNNs zur Erkennung von Bauwerksschäden, die durch den teilweisen Verlust der Festigkeit an einer Stütze eines Stahlträgers dargestellt werden, nachweisen. Die nächsten Abschnitte dieses Beitrags erörtern den Ansatz, gefolgt von den Validierungstests. Der Beitrag endet mit einer Zusammenfassung und Schlussfolgerungen sowie einem Ausblick auf zukünftige Forschungsarbeiten.

2 Schadenserkennung bei Leichtbaubrücken mit beweglichen Massen

In diesem Abschnitt werden zunächst die Simulationen zur Erstellung von markierten Beschleunigungsdaten erläutert. Anschließend werden das CNN-Modell und der Trainingsprozess vorgestellt, einschließlich der Vorbereitung der Beschleunigungsdaten.

2.1 Simulation des Problems der bewegten Masse

Die Bewegungsgleichung einer Leichtbaubrücke mit einer beweglichen Masse, modelliert als kontinuierlicher Euler-Bernoulli-Balken (Abbildung 1), wurde in einer früheren Arbeit der Autoren nachgewiesen [10]:

$$\rho A \ddot{w} + c \dot{w} + EI w'''' = m [g - \ddot{w} - 2v \dot{w}' - v^2 r''] \delta(x - vt), \quad (1)$$

wobei ρ die Materialdichte, A die Querschnittsfläche des Balkens, E der Elastizitätsmodul, I das Trägheitsmoment des Balkens in vertikaler Richtung, c der Dämpfungskoeffizient, w die vertikale Verschiebung der neutralen Achse des Balkens, δ das Dirac-Delta, x die räumliche Koordinate in Längsrichtung des Balkens und t die Zeit ist. Die Geschwindigkeit der bewegten Masse wird mit v und die Oberflächenrauheit mit r bezeichnet. Schließlich gilt $\dot{w} = \partial w / \partial t$, $\ddot{w} = \partial^2 w / \partial t^2$, $w'''' = \partial^4 w / \partial x^4$, $r'' = \partial^2 r / \partial x^2$.

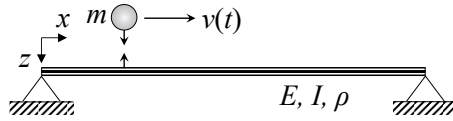


Abbildung 1: Bernoulli-Euler-Balken mit beweglicher Masse

Gleichung (1) wird durch die Modalanalyse $w(x,t) = \varphi_n(x)q_n(t)$ ($n = 1 \dots p$) gelöst, wobei $\varphi_n(x)$ die n -te sinusförmige „Eigenfunktion“ (Wellenfunktion) und $q_n(t)$ die n -te „generalisierte“ Koordinatenfunktion ist. Durch Einsetzen von p Modal-Analyse-Komponenten in Gleichung (1) und Ausnutzung der Orthogonalität der Eigenfunktionen und Vormultiplikation von Gleichung (1) mit $\varphi(x)^T$ ($\varphi(x) = [\varphi_1(x), \varphi_2(x), \dots, \varphi_p(x)]$) wird ein $p \times p$ -System von Differentialgleichungen erhalten (ausgedrückt in Matrixform):

$$\begin{aligned} \mathbf{M}(t)\ddot{\mathbf{q}}(t) + \mathbf{C}(t)\dot{\mathbf{q}}(t) + \mathbf{K}(t)\mathbf{q}(t) &= \mathbf{F}(t) \\ \mathbf{M}(t) &= \mathbf{I} + m\boldsymbol{\varphi}\boldsymbol{\varphi}^T, \quad \mathbf{C}(t) = \text{diag}(2\xi_n\omega_n) + 2mv\boldsymbol{\varphi}\boldsymbol{\varphi}^T, \quad \mathbf{K}(t) = \text{diag}(\omega_n^2) + mv^2\boldsymbol{\varphi}\boldsymbol{\varphi}^T, \\ \mathbf{F}(t) &= mg\boldsymbol{\varphi} - mv^2r''(x(t))\boldsymbol{\varphi}, \quad \xi_n = c/(2\rho AL\omega_n), \quad \omega_n = n^2\pi^2/L^2 \sqrt{EI/(\rho A)}. \end{aligned} \tag{2}$$

Gleichung (2), die eine Reihe gewöhnlicher Differentialgleichungen zweiter Ordnung repräsentiert, wird mithilfe der Zustandsraumformulierung in eine Reihe gewöhnlicher Differentialgleichungen erster Ordnung umgewandelt,

$$\dot{\mathbf{y}}(t) = [-\mathbf{A}^{-1}(t)\mathbf{B}(t)]\mathbf{y}(t) + \mathbf{A}^{-1}(t)\mathbf{h}(t) \tag{3}$$

$$\mathbf{A}(t) = \begin{bmatrix} \mathbf{0} & \mathbf{M}(t) \\ \mathbf{M}(t) & \mathbf{C}(t) \end{bmatrix}, \quad \mathbf{B}(t) = \begin{bmatrix} -\mathbf{M}(t) & \mathbf{0} \\ \mathbf{0} & \mathbf{K}(t) \end{bmatrix}, \quad \dot{\mathbf{y}}(t) = \begin{bmatrix} \dot{\mathbf{q}}(t) \\ \mathbf{q}(t) \end{bmatrix}, \quad \mathbf{h}(t) = \begin{bmatrix} \mathbf{0} \\ \mathbf{F}(t) \end{bmatrix}, \tag{4}$$

die für jeden Zeitpunkt k durch eine Eigenwertanalyse der nicht symmetrischen Matrix $(-\mathbf{A}_k)^{-1}\mathbf{B}_k$ gelöst wird, was zu komplexen Eigenwerten $\boldsymbol{\Lambda}_k$ und einer Eigenvektormatrix $\boldsymbol{\Psi}$ führt:

$$\boldsymbol{\Lambda}_k = \text{diag}(\boldsymbol{\lambda}_{n,k}), \quad \boldsymbol{\lambda}_{n,k} = \begin{bmatrix} e^{\lambda_{n,k}} & 0 \\ 0 & e^{\bar{\lambda}_{n,k}} \end{bmatrix}, \quad \lambda_{n,k} = \xi_{n,k}\omega_{n,k} \pm i\omega_{n,k}\sqrt{1-\xi_{n,k}^2}, \quad \xi_{n,k} \neq \xi_n. \tag{5}$$

$$\mathbf{y}_k = \boldsymbol{\Theta}_k (\boldsymbol{\Theta}_{k-1})^{-1} \mathbf{y}_{k-1} + \boldsymbol{\Theta}_k \int_{k-1}^k (\boldsymbol{\Theta}_s)^{-1} \mathbf{A}_k^{-1} \mathbf{h}_k ds, \quad \boldsymbol{\Theta}_k = \boldsymbol{\Psi}_k \boldsymbol{\Lambda}_k. \tag{6}$$

Die $q(t)$ -Funktionen werden zusammen mit den Eigenfunktionen verwendet, um die vertikalen Verschiebungsfunktionen $w(x,t)$ zu rekonstruieren. Die vertikalen Verschiebungsfunktionen werden wiederum verwendet, um die Beschleunigungsdaten („Simulationsdaten“) zu erzeugen, die für das Training des CNN-Modells verarbeitet werden.

2.2 Training eines neuronalen Netzwerks zur Erkennung von Schäden

Aufgrund der Nichtstationarität des Problems der beweglichen Massen ist eine Vorbereitung der Daten erforderlich, sodass der gekoppelte Zeit-Frequenz-Inhalt der Simulationsdaten wiedergegeben wird [11]. Deshalb erfolgt die Vorbereitung der Daten über die Gabor-Transformation [12],

$$G_w(\kappa) = S \left[\sum_{j=1}^N \dot{w}(x, t_j) z_j e^{-i2\pi\kappa j/N} \right]^2, \quad \kappa = N \cdot f / f_s, \tag{10}$$

wobei G_w der Gabor-Koeffizient ist und κ der Index des diskreten Frequenzbereichs der Frequenz f , für den eine diskrete Menge von Simulationsdaten der Länge N betrachtet wird, die mit der Frequenz f_s ($f_s = 1/\Delta t$) abgetastet werden. Die Fensterfunktion wird als z_j (z. B. Gauß-Funktion oder Hann-Fenster) bezeichnet, und S ist ein Skalierungsfaktor. Gabor-Koeffizienten werden in der Regel als Farbkarte dargestellt, im Folgenden als Gabor-Transformationsbild bezeichnet.

Nach der Erstellung der Gabor-Transformationsbilder wird das CNN-Modell zur Klassifizierung der Bilder in Schadensszenarien entwickelt. Ein typisches CNN-Layout, das in Abbildung 2 dargestellt ist, besteht aus einer Eingabeschicht (engl. „Input Layer“, IL), einer Abfolge von faltenden Schichten (engl. „Convolution Layers“, CL) und Pooling-Schichten (engl. „Pooling layers“, PL), (mindestens) einer vollständig verbundenen Schicht (engl. „Fully Connected Layer“, FC), einer Ausfallschicht (engl. „Dropout Layer“, DrL) und einer Klassifizierungs- oder Ausgabeschicht (engl. „Output Layer“). Die CL führen „gleitende“ Faltungsoperationen über Bildbereiche mit quadratischen Kernels aus. Das Ergebnis jeder CL ist eine Merkmalskarte, die einer Aktivierungsfunktion zugeführt wird, um die CNN mit einer Nichtlinearität zu versehen, die für die Lösung nicht-trivialer Probleme erforderlich ist. Die Aktivierungen werden an die PL weitergeleitet, die die Aktivierungen über gleitende Fensterberechnungen entweder den Durchschnitts- oder den Maximalwerte verdichtet. Auf die letzte PL folgen mindestens eine FC und eine DrL, die die verdichteten Aktivierungen kombinieren und zufällige Ausschlüsse von Aktivierungen vornehmen, um eine Überanpassung zu verhindern. Die DrL ist mit der Ausgabeschicht verbunden, die die Klassen enthält.

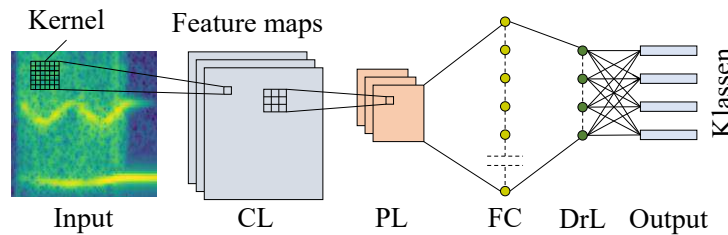


Abbildung 2: Layout einer typischen CNN-Seite

Das Training (d.h. die Feinabstimmung der Kernels) erfolgt mithilfe eines „markierten“ Datensatzes von Gabor-Transformationsbildern, einschließlich Bildern von der intakten Struktur und von Simulationen von Schadensszenarien. Der markierte Datensatz wird zu 70 % in Trainingsdaten, 20 % in Validierungsdaten und 10 % in Testdaten aufgeteilt. Der Trainingssatz wird in Iterationen durch das CNN geleitet, um den Fehlergradienten an der Ausgabeschicht vorherzusagen, und zurückgeleitet, um die Kerne gemäß einem Optimierungsalgorithmus und einer Lernrate zu aktualisieren. Der Validierungssatz wird nach einer Reihe von Iterationen regelmäßig durch das CNN geleitet, um eine Überanpassung des CNN an den Trainingssatz zu vermeiden. Der Testsatz wird am Ende des Trainings durch das CNN geleitet, um die Leistung des CNN zu überprüfen. Die Validierungstests, einschließlich Simulationen eines Balkens mit einer beweglichen Masse, Datenvorverarbeitung und Training eines CNN-Modells zur Schadensklassifizierung, werden im nächsten Abschnitt vorgestellt.

3 Laborversuch zur Validierung

Die Validierung des vorgestellten Ansatzes umfasst einen Laborversuch, die in diesem Abschnitt vorgestellt werden. Zunächst wird der Versuchsaufbau erörtert, gefolgt von den Simulationen zur Erstellung des markierten Datensatzes, dem Training des CNN-Modells, dem Laborversuch und den Ergebnissen.

3.1 Versuchsaufbau und Simulationen

Der Laborversuch wird an einem einfach gestützten HEB100-Stahlträger durchgeführt, der ein verkleinertes Modell einer Leichtbaubrücke darstellt, die von einer beweglichen Masse überquert wird (Abbildung 3). Der Querschnitt des Trägers hat die Abmessungen $100 \times 100 \times 10 \times 6$ (mm) (Breite \times Höhe \times Flanschdicke \times Stegdicke). Die Gesamtlänge der neutralen Achse des Trägers beträgt 5830 mm. Der Träger ist Teil eines Versuchsaufbaus, der verstellbare Streben umfasst, die sicherstellen, dass der Träger waagrecht ausgerichtet ist. Der Träger wird nur an seinen Enden gestützt, d. h. die mittleren Streben stützen den Träger nicht. Die Versuchsanordnung umfasst Rollen, die es ermöglichen, dass sich Massen entlang der Längsachse des Trägers bewegen können, sowie einen Motor, der die Geschwindigkeit der Massen steuert.

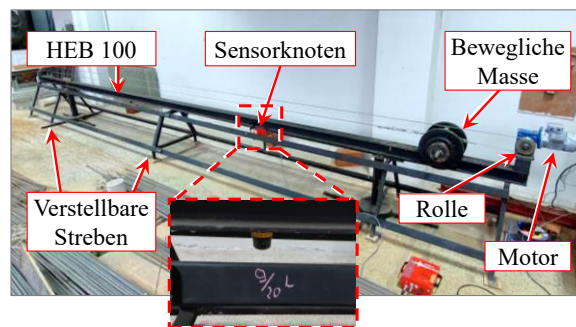


Abbildung 3: Versuchsaufbau

Es wird ein analytisches Modell des Balkens entwickelt und zur Simulation der Bewegung einer sich bewegenden Masse über den Balken verwendet, wobei E und ρ jeweils 198.5 GPa und $7650 \cdot \text{kg/m}^3$ betragen, wie aus Vorversuchen geschätzt wurde. Die Masse und die Geschwindigkeit werden im Zufallsverfahren gewählt, um mehrere Simulationsszenarien zu entwickeln, wobei m Werte von 10 kg bis 40 kg und v Werte zwischen 0.2 m/s und 0.5 m/s annehmen. Es werden zwei strukturelle Zustände betrachtet, ein Zustand für den intakten Träger und ein Zustand, bei dem von einem Verlust der Festigkeit an einer Stütze ausgegangen wird, d. h. durch Ersetzen der vertikalen Stütze durch eine Translationsfeder, was einer Erosion des Untergrunds des Fundaments ähnelt, z. B. als Folge von Abnutzung. Für jeden Zustand werden 500 Simulationen durchgeführt, wobei die Beschleunigungsdaten bei $9L/20$ vom Träger mit $f_s = 128$ Hz berechnet werden. Der Grund, aus dem die Mitte des Balkens vermieden wird, ist, dass die zweite Eigenfunktion, die zur Reaktion beiträgt, durch eine antisymmetrische Sinusform mit einem Nulldurchgangspunkt in der Mitte charakterisiert ist. Für jede Klasse werden 500 Gabor-Transformationsbilder erstellt, die den markierten Satz bilden, der für das Training des CNN-Modells verwendet wird. In Abbildung 4 sind beispielhafte Abbildungen von Gabor-Transformationsbildern für beide Zustände dargestellt, die die Variation der ersten beiden Eigenfrequenzen für beide strukturellen Zustände veranschaulichen. Wie in den Abbildungen zu

sehen ist, verringert der Verlust der Fixierung an der Stütze die Eigenfrequenzen, der Variationstrend ist jedoch in beiden Zuständen ähnlich.

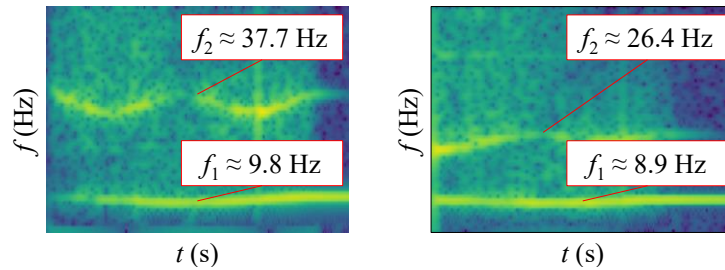


Abbildung 4: Gabor-Transformation für den intakten Balken (links) und für den beschädigten Balken (rechts)

3.2 Training des CNN-Modells

Zur prototypischen Implementierung des Ansatzes wird das MATLAB-Tool für maschinelles Lernen eingesetzt und ein einfaches Layout für das CNN-Modell entwickelt, das in Abbildung 5 dargestellt ist. Es ist an die Eingabe von Gabor-Transformationsbildern und die Anzahl der berücksichtigten Klassen angepasst.

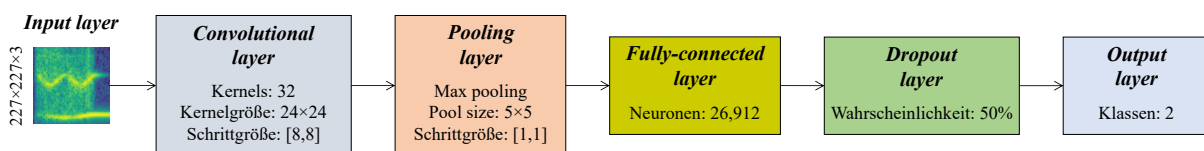


Abbildung 5: Das in dieser Studie definierte CNN-Layout

Während des Trainings wird der „Adam“-Optimierer – ein stochastischer Gradientenabstieg-Optimierungsalgorithmus – für die Feinabstimmung der Kernel in jeder Iteration eingesetzt [13]. Die anfängliche Lernrate wird auf 0,001 festgelegt, die Batch-Größe (Anzahl der Bilder für jede Iteration) beträgt 30 Bilder und die Validierungsfrequenz wird auf alle 30 Iterationen festgelegt. Die Lernrate wird alle 117 Iterationen um den Faktor 0,5 reduziert. Nach 122 Iterationen wird eine hohe Trainingsgenauigkeit von 98 % erreicht. Vor der Anwendung des CNN-Modells mit Beschleunigungsdaten aus dem Versuchsaufbau wird die Klassifizierungsfähigkeit des CNN-Modells anhand des Testdatensatzes überprüft. Der Klassifizierungstest ergab, dass jedes Bild des Testdatensatzes korrekt klassifiziert wurde, was einer Genauigkeit von 100 % entspricht.

3.3 Laborversuch

Es werden Beschleunigungsdaten mithilfe eines drahtlosen Lord Microstrain G-Link-200-Sensorknotens erfasst, der Beschleunigungen von bis zu ± 8 g in drei Achsen und einer Abtastrate von bis zu 4096 Hz messen kann [14]. Der Laborversuch besteht aus zwei Experimenten, die den in den Simulationen verwendeten Zuständen entsprechen und 30 Kombinationen von Massen- und Geschwindigkeitswerten für jeden Zustand verwenden. Die Massen nehmen Werte von 13 kg, 18 kg, 23 kg, 27 kg und 38 kg an, und die Geschwindigkeitswerte reichen von 0,25 m/s bis 0,5 m/s mit einer Schrittweite von 0,05 m/s. Der Verlust der Festigkeit wird durch das Anbringen einer Feder zwischen dem Trägerende und seiner Stütze realisiert. Die Beschleunigungsdaten werden für jedes Experiment in der Nähe der Mitte des Balkens an einem Punkt erfasst, der sich $9L/20$ vom

Balkenende entfernt befindet, wie in Abbildung 4 dargestellt. Wie in den Simulationen wird eine Abtastfrequenz von 128 Hz verwendet. Die Beschleunigungsdaten aus jedem Experiment werden zur Erstellung eines Gabor-Transformationsbildes verwendet, sodass ein Satz von 60 Gabor-Transformationsbildern entsteht, die in das CNN-Modell eingespeist werden. Die Gabor-Transformationsbilder werden dem CNN-Modell zugeführt und in die beiden Klassen eingeteilt, die den betrachteten Zuständen entsprechen. Die Ergebnisse werden in der folgenden Matrix zusammengefasst:

Tabelle 1: Matrix für die Experimente

	Intakt	Beschädigt
Intakt	30	0
Beschädigt	1	29

Die Ergebnisse zeigen deutlich, dass das CNN-Modell in der Lage ist, reale Beschleunigungsdaten korrekt zu klassifizieren. Von den Gabor-Transformationsbildern des intakten Balkens wird nur ein Experiment als „beschädigt“ falsch klassifiziert, während die entsprechenden Bilder des beschädigten Balkens alle korrekt klassifiziert werden. Darüber hinaus dienen die Ergebnisse als Beweis für die Aussagekraft des vorgeschlagenen Konzepts und für die Übertragbarkeit des CNN-Trainings von Simulationen auf die reale Welt.

4 Zusammenfassung und Schlussfolgerungen

In dieser Arbeit wurde ein Ansatz zur Erkennung von Schäden an Leichtbaubrücken vorgestellt, die durch Euler-Bernoulli-Balken repräsentiert werden. Dieser Ansatz basiert auf simulierten Beschleunigungsdaten, mit denen ein CNN trainiert wird, um die Daten in vordefinierte Schadensszenarien einzuordnen. Anschließend wird das CNN mit realen Beschleunigungsdaten angewendet. Der vorgeschlagene Ansatz wurde in einem Laborversuch an einem Stahlträger mit einer beweglichen Masse validiert. Die Ergebnisse haben gezeigt, dass das CNN (1.) mit hoher Genauigkeit anhand von simulierten Beschleunigungsantwortdaten trainiert werden kann und (2.) auf reale Anwendungen übertragen werden kann, sofern das analytische Modell hierfür kalibriert wird. In zukünftigen Arbeiten könnten mehrere Schadensszenarien, die während der Lebensdauer von Leichtbaubrücken auftreten können, untersucht werden und der vorgeschlagene Ansatz zu einem Konzept für SHM weiterentwickelt werden.

Danksagungen

Die Autoren danken der Deutschen Forschungsgemeinschaft (DFG) für die Unterstützung im Rahmen der Sacheihilfe mit dem Geschäftszeichen SM 281/20-1, für die Unterstützung der Hellenic Foundation for Research and Innovation (HFRI) unter Förderkennzeichen 6255 sowie dem Bundesministerium für Digitales und Verkehr (BMDV) für die Unterstützung im Rahmen der Projekte „IDA-KI“ (Förderkennzeichen 19FS2013B) und „KaSyTwin“ (Förderkennzeichen 19FS2059C) im Rahmen des mFUND-Programms. Die in dieser Arbeit vorgestellten Ergebnisse wurden mithilfe der Hochleistungsrecheninfrastruktur der Aristotle University Thessaloniki im Rahmen der Deutsch-Griechischen Kooperation mit der Technischen Universität Hamburg erzielt. Alle in diesem Beitrag

geäußerten Meinungen, Erkenntnisse, Schlussfolgerungen oder Empfehlungen sind die der Autoren und spiegeln nicht unbedingt die Ansichten der DFG, der HFRI oder des BMDV wider.

Referenzen

- [1] P. Dallard, T. Fitzpatrick, A. Flint, A. Low, R. Ridsdill Smith, M. Willford, and M. Roche, "London Millennium Bridge: Pedestrian-induced lateral vibration", *ASCE Journal of Bridge Engineering*, 6(6), pp. 412-417, 2001.
- [2] K. Smarsly and Y. Petryna, "A Decentralized Approach towards Autonomous Fault Detection in Wireless Structural Health Monitoring Systems," in *Proceedings of the 7th European Workshop on Structural Health Monitoring (EWSHM) 2014*, Nantes, France, July 8, 2014.
- [3] A. Firus, *A Contribution to Moving Force Identification in Bridge Dynamics*. Wiesbaden, Germany: Springer Vieweg, 2023.
- [4] K. Dragos, F. Magalhães, G. D. Manolis, and K. Smarsly, "Frequency-domain synchronization of structural health monitoring data", *Journal of Sound and Vibration*, 571(118017), 2024.
- [5] C. Bilello and L. A. Bergman, "Vibration of damaged beams under a moving mass: theory and experimental validation", *Journal of Sound and Vibration*, 274(3-5), pp. 567-582, 2004.
- [6] A. Cicirello, "On the response bounds of damaged Euler–Bernoulli beams with switching cracks under moving masses", *International Journal of Solids and Structures*, vol. 172-173, pp. 70-83, 2019.
- [7] Y. Zhan, F. T. K. Au, and J. Zhang, "Bridge identification and damage detection using contact point response difference of moving vehicle", *Structural Control Health Monitoring*, 28(12), e2837, 2021.
- [8] K. Smarsly, D. Hartmann, and K. H. Law, "Structural Health Monitoring of Wind Turbines Observed by Autonomous Software Components – 2nd Level Monitoring," in *Proceedings of the 14th International Conference on Computing in Civil and Building Engineering*, Moscow, Russia, June 27, 2012.
- [9] T. Al-Zuriqat, C. Chillón Geck, K. Dragos, and K. Smarsly, "Adaptive fault diagnosis for simultaneous sensor faults in structural health monitoring systems", *Infrastructures*, 8(3), p. 39, 2023.
- [10] G. Dadoulis and G. D. Manolis, "Dynamic response of a damaged bridge model traversed by a heavy point mass", *Journal of Sound and Vibration*, 551(117613), 2023.
- [11] H. Fritz, J. Peralta, D. Legatiuk, M. Steiner, K. Dragos, and K. Smarsly, "Fault diagnosis in structural health monitoring systems using signal processing and machine learning techniques", in *Structural Health Monitoring Based on Data Science Techniques*, A. Cury, D. Ribeiro, F. Ubertini, and M. D. Todd, Eds. Cham, Switzerland: Springer, pp. 143-164, 2022.
- [12] S. Qian and D. Chen, "Discrete Gabor transform", *IEEE Transactions on Signal Processing*, 41(7), pp. 2429-2438, 1993.
- [13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization", presented at the *3rd International Conference on Learning Representations*, San Diego, CA, USA, May 7, 2015.
- [14] Microstrain Sensing, *Microstrain Sensing Product Datasheet: G-Link-200*. Williston, VT, USA: Parker Hannifin Corp., 2020.