Reducing Experimental Data Requirements in CNN-Based Damage Detection through Transfer Learning

Finja RENTZSCH HOLM[1], Tobias SCHALM[1], Jorge Luis JIMÉNEZ APARICIO[2], Kai-Uwe SCHRÖDER[1]
[1]Institute for Structural Mechanics and Lightweight Design, RWTH Aachen University, Aachen, Germany, finja.rentzsch.holm@sla.rwth-aachen.de
[2]Institute of Man-Machine Interaction, RWTH Aachen University, Aachen, Germany, jimenez@mmi.rwth-aachen.de

Abstract While deep neural networks represent a promising approach for evaluating sensor data to assess damage presence, location and severity, large amounts of data are required for training. However, the generation of experimental data is both labor-intensive and costly. Transfer Learning (TL) is a well-established method in deep learning that enables knowledge transfer from similar domains improving performance and reducing data requirements. While this approach is used in computer vision and natural language processing, its application within the field of strain-based Structural Health Monitoring (SHM) has received little attention in currently published literature. This work evaluates a resource-efficient TL approach for strain-based SHM. An encoder-decoder Convolutional Neural Network (CNN) (UNet) learns relationships between strain distribution and crack geometry of an aluminum tube in a four-point bending test based on Finite Element Method (FEM) data. The UNet is then finetuned to adapt to experimental conditions, accounting for factors such as measurement noise, improving model performance.
This study demonstrates that TL enables efficient adaptation to real-world conditions. Results show, that in settings with large network topologies and limited available training data, finetuning a numerically pretrained UNet shows superior performance over training a UNet only based on experimental data. Therefore, the approach offers a cost-effective and scalable solution for data-driven SHM.

Keywords: Structural Health Monitoring; Convolutional Neural Networks; Transfer Learning; Digital Image Correlation; Finite Element Simulation

Introduction

The application of deep neural networks in SHM poses challenges such as limited availability of real data and limited computing resources. TL addresses these challenges by transferring a priori knowledge from similar domains to a target domain. Different techniques for knowledge transfer in deep neural networks include adapting the input data, the loss-function and the architecture of the neural network. Due to the availability of similar data from numerical simulations, this work focuses on a TL approach in which weights from models trained for the similar source domain are transferred to the model for the target domain. [1, 2]

In this work, strain-based damage detection of aluminum tubes is described as a semantic segmentation task. The structure is discretized through a grid and each pixel of a measured strain field is classified as *damaged* or *undamaged*. This approach enables detection, localisation and dimensioning of a crack, thus categorizing it as Level 3 SHM according

This work is licensed under CCBY 4.0

Media and Publishing Partner NDT.net

to Rytter [3]. A generelised UNet architecture, first introduced by Ronneberger et al. [4] for biomedical image segmentation, is used. The architecture is illustrated in Figure 1 and consists of three main parts: *Encoder*, *Bottleneck* and *Decoder*. The encoder part consists of n_{blocks} blocks and reduces spatial dimension while extracting hierarchical features. The bottleneck can be seen as a compressed representation of the input. The decoder part again consists of n_{blocks} blocks and restores spatial resolution while refining segmentation details. Skip connections support preserving fine-grained spatial information. n_{filter} represents the number of filters used in the first and final layer. The number of filters increases by $2^{n_{block}}$ with decreasing spatial dimensions. Both parameters n_{blocks} and n_{filter} are varied later in this work to evaluate the impact of the UNet topology on network performance. A larger number of blocks enables a higher level of abstraction, whereas an increased number of filters allows the model to capture a greater variety of features within the data. Previous work [5] has shown great applicability for damage identification in Composite Overwrapped Pressure Vessels.

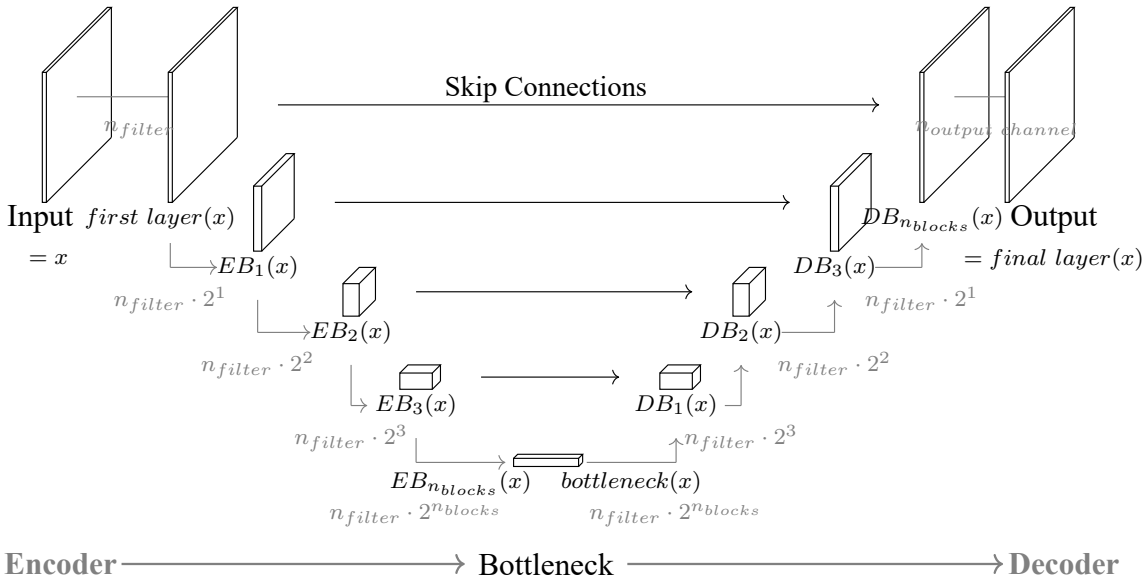


Fig. 1. *Generalized UNet architecture based on [4].*

1. Experimental setup and data creation

The aim of this approach is crack localisation and dimensioning in aluminum beams based on experimental strain data. Within this section, the creation of the experimental strain dataset through Digital Image Correlation (DIC) is described. Furthermore, a second dataset, later used for numerical pretraining of the UNet, is created based on a FEM model and both datasets are compared.

1.1 DIC measurements of an aluminum tube under four-point bending

To create a variety of strain fields for differently damaged parts, 12 aluminum tubes are cut at alternating damage positions x_d and cyclic loading ($5\,kN$ to $15\,kN$ at a frequency of $5\,Hz$) is applied in a four-point bending test at $x = 150\,mm$ and $x = 350\,mm$. The geometry of the aluminum tubes is shown in Figure 2 and the initial crack length varies between $0.5\,mm$ and $1\,mm$. The aluminum tubes are supported at $x = 50\,mm$ and $x = 450\,mm$. DIC measurements of the beams surface are performed every 200 load cycles using a *GOM ARAMIS 4M* system. DIC measurements provide a good balance between measurement effort and the number of generated datapoints. While the aluminium tubes fail after 140 measurement periods on average, crack growth can only be observed during the final measurement periods

before failure. Therefore, 64 dataframes are created containing crack lengths between $1\,mm$ and $46\,mm$. The damage position x_d, the number of dataframes generated from the experiment as well as the assignment of the data to the experimental training (1), validation (2) and testing (3) dataset is summarized in Table 1.

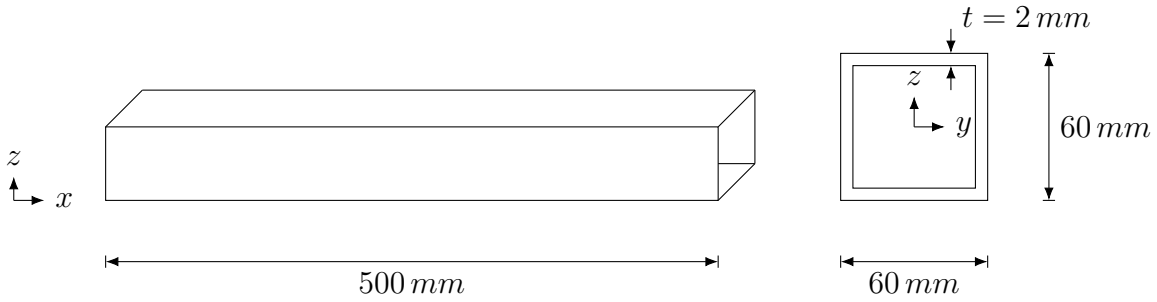


Fig. 2. *Geometry of aluminum tubes used to generate the experimental DIC data.*

index	1	2	3	4	5	6	7	8	9	10	11	12
$x_d\ [mm]$	250	300	250	300	225	225	175	275	200	325	300	340
number of dataframes	5	7	5	5	4	5	5	5	3	11	6	3
dataset	*(1)*	*(2)*	*(1)*	*(2)*	*(1)*	*(1)*	*(1)*	*(1)*	*(1)*	*(3)*	*(1)*	*(3)*

Table 1. Description of the experimental data. The damage position for each of the 12 tubes is given next to the number of dataframes generated during DIC measurements and the assosiated dataset.

A Principal Component Analysis (PCA) is applied to each dataframe to correct axis alignment. The measurement range includes $110\,mm < x < 390\,mm$. To fit the strain measurements into a two-dimensional high-resolution grid, linear interpolation is performed and missing values are replaced by their nearest neighbour and finally 0. Using linear interpolation, the strain data is again transformed into a lower resolution grid that matches the FEM grid.

To create corresponding target data, a binary target mask is created based on the high-resolution strain data. As damage positions are known from the experimental setup (see Table 1), only an area of interest $x_d \pm 10\,mm$ is considered for analytical damage detection. This reduces false crack position and length calculations due to random noise in the strain measurements. Elements of the high-resolution target grid are classified as undamaged (0), if the corresponding strain value $\varepsilon_x \leq 8\,\%$ and classified as damaged if $\varepsilon_x \geq 8\,\%$. $\varepsilon = 8\,\%$ corresponds to the materials elongation at break [6]. A new crack position x_d is calculated as the x-index of the center of gravity within the high-resolution target matrix. The crack length z_d is calculated based on the maximum z-index of a damaged element. Based on the analytically derived x_d and z_d values, new target masks, matching the FEM grid, are created. At the positions $x_d - 2\,mm$ and x_d, all elements with $z \leq z_d$ are set to 1 while the remaining elements are set to 0. This ensures similarity between experimental and numerical targets. Finally, smoothing is applied to all target masks to improve training stability.

This work is licensed under CCBY 4.0

Media and Publishing Partner NDT.net

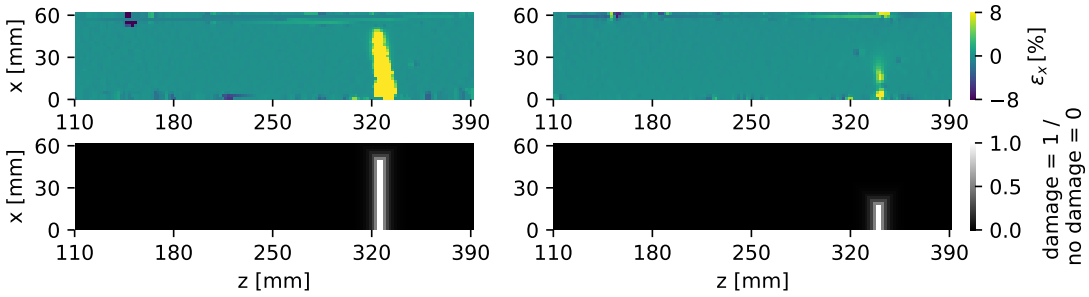


Fig. 3. *Experimental input and target data corresponding to the last frame from tubes 10 and 12.*

1.2 Description of the domain gap between experimental and numerical data

A general FEM model is created based on shell elements with an element size of $2\,mm$ x $2\,mm$ and a shell thickness of $2\,mm$. The bearings are modelled through kinematic coupling of the bottom elements at $x = 50\,mm$ and $x = 450\,mm$ with reference points fixed in the z-direction. Two load introduction surfaces are defined as node surfaces for top elements with $149\,mm \leq x \leq 151\,mm$ and $349\,mm \leq x \leq 351\,mm$. A static load (amplitude of $15\,kN$) is introduced as a central load onto the load introduction surfaces. Cracks are modeled by adding nodes and separating elements along the crack length based on different damage positions x_d and damage sizes z_d (see Figure 4). While the experimental targets are obtained retrospectively, the numerical data generation follows an inverse procedure, in which the damage location and extent are described a priori and the corresponding input fields are then computed using FEM. Numerical FEM models are calculated for every possible combination of damage position and damage size (for $150\,mm \leq x_d \leq 350\,mm$, $z_d \leq 30\,mm$) resulting in 1515 input dataframes. The numerical dataframes corresponding to the last measurement from tubes 10 and 12 are illustrated in Figure 5.

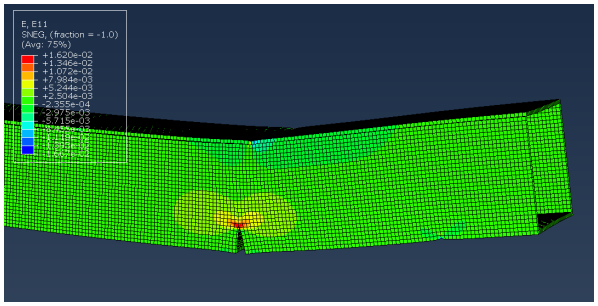


Fig. 4. *Output of the FEM model corresponding to the last dataframe of tube 10 ($x_d = 340\,mm$, $z_d = 16\,mm$).*

While both the numerical FEM data and the experimental DIC data share the same geometrical properties, load conditions and physical mechanisms, a domain gap (also often referred to as *reality gap*) exists in distribution. Qualitative differences between the experimental and numerical input data can be derived from the strain fields in Figures 3 and 5. While in the experimental strain fields, elements along the crack length show strain values above the elongation at break ($\varepsilon = 8\,\%$), in the numerical strain fields, only elements at the tip of the crack exceed the average strain of the tube. Furthermore, maximum strain as well as the number of elements deviating from the average strain value are significantly higher in the experimental strain field compared to the numerical strain field.

This work is licensed under CCBY 4.0

Media and Publishing Partner NDTnet

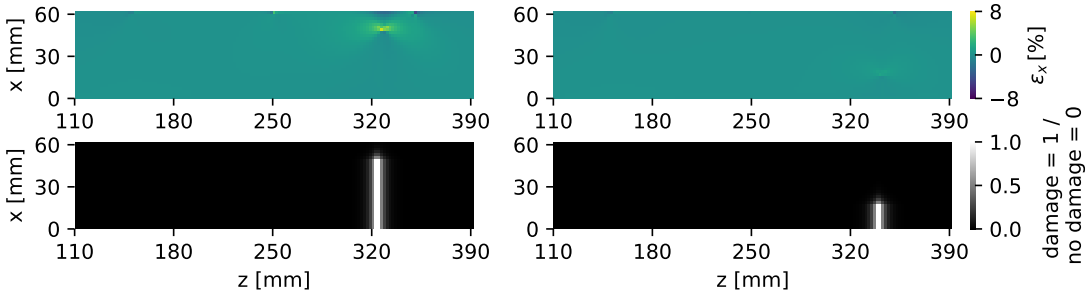


Fig. 5. *Numerical input and target data corresponding to the last frame from tubes 10 and 12.*

The statistical properties of the experimental data as well as the corresponding and random numerical data are listed in Table 2. For the experimental data, the mean strain value $\Delta\varepsilon_x$ amounts to 0.44 % and the standard deviation amounts to 5.48 %. For comparison, in the corresponding numerical data, the mean strain value $\Delta\varepsilon_x$ is 0.17 % and the standard deviation results in 0.21 %. The distribution of the data, as measured by the standard deviation, is significantly wider in the experimental data, and the outliers are more extreme compared to the numerical data. This effect becomes even more significant in comparison to the random numerical dataset that contains the 1515 dataframes used for UNet training with maximum damage sizes $z_d \leq 30\,mm$.

	n	mean [%]	std [%]	min [%]	max [%]	$P_{0.5}$ [%]	$P_{99.5}$ [%]
experimental	64	0.44	5.48	-13.43	198.77	-1.48	10.64
numerical (corr.)	64	0.17	0.21	-4.73	8.66	-0.23	1.02
numerical (rand.)	1515	0.18	0.22	-3.06	3.56	-0.31	1.09

Table 2. Statistical properties of the experimental dataset, the corresponding numerical dataset and the random numerical dataset.

2. Results and Discussion

Trainings are repeated for different network topologies and different TL strategies. First, the results for numerical pretraining are described. Then, the results based on training with only experimental data are presented. Finally, using a Linear Mixed Model (LMM), model performance for the different TL strategies is compared.

2.1 Damage detection based on numerical data

The numerical dataset is randomly split into training (60 %), validation (20 %) and testing data (20 %). The strain data is transformed into matrices of the size 32 x 32 and augmented by a factor of 10 by varying the horizontal position of the damage between 1 and 32. The data is normalised using min-max normalisation and the 0.5 % and 99.5 % percentiles. UNet training is carried out using Intersection over Union (IoU) Loss ($1 - IoU$ and Binary Cross Entropy (BCE) with weights of 0.5 each (see Equation 1[1] and 2). The combined loss function offers balance between a region-based loss function suitable for class imbalance and a pixel-based loss function improving training stability. Initially, the learning rate is set to 10^{-2} and

[1] TP represents the number of damaged pixels correctly predicted as damaged while FP represents the number of undamaged pixels falsely predicted as damaged. The number of damaged pixels falsely predicted as undamaged is abbreviated as FN.

This work is licensed under CCBY 4.0

Media and Publishing Partner NDTnet

reduced after five consecutive epochs without validation loss improvement and training is stopped after 1000 epochs. [7]

$$IoU\ Loss = 1 - \frac{TP}{2 \cdot TP + FP + FN} \tag{1}$$

$$BCE = -(ylog(\hat{y}) + (1-y)log(1-\hat{y})) \tag{2}$$

Two exemplary pairs of input, output and target data from the corresponding numerical testing dataset are shown in Figure 6. Due to target smoothing, damage sizes are slightly overestimated while the damage position is detected correctly. This results in a maximum average IoU of 51.17 %.

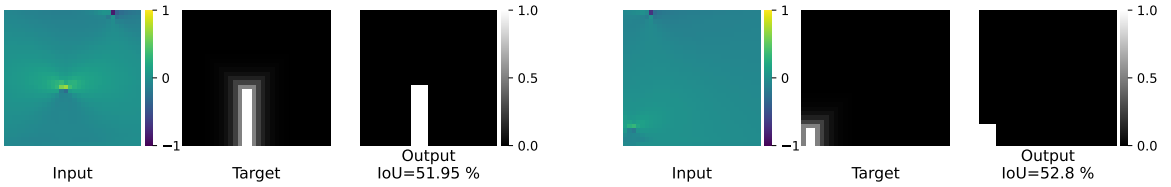


Fig. 6. *Corresponding numerical input, target and output heatmap based on a UNet topology with $n_{blocks} = 1$ and $n_{filter} = 2$ and trained with numerical data.*

2.2 Damage detection based on experimental data

To prevent data leakage, the experimental dataset is split into training, validation and testing data based on a sample level (see Table 1). Training is carried out using the same hyperparameters given for numerical training. Performance measured by the IoU and varies little between different UNet topologies. Exemplary results for the smallest network topology ($n_{blocks} = 1$ and $n_{filter} = 2$) are shown in Figure 7. The best average IoU on the experimental testing dataset amounts to 48.06 % nearly reaching the performance of the numerical model.

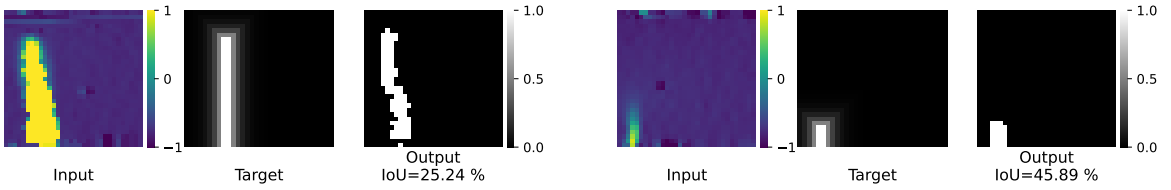
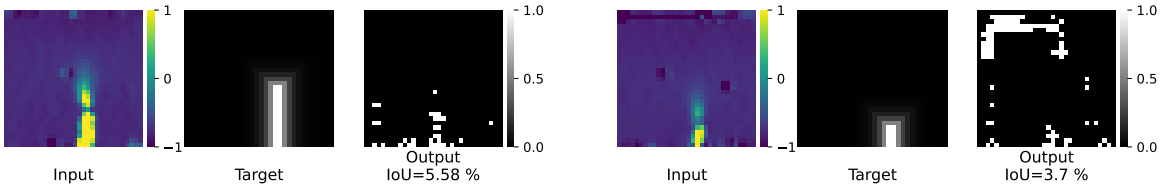


Fig. 7. *Experimental input, target and output heatmap based on a UNet topology with $n_{blocks} = 1$ and $n_{filter} = 2$ and trained with experimental data.*

2.3 Evaluation of Transfer Learning

Three different TL strategies are examined within this study: *full finetuning*, *encoder finetuning* and *decoder finetuning*. During full finetuning, all model parameters are adapted based on the experimental training data, and numerical pretraining is only used as an improved parameter initialisation. On the other hand, for encoder finetuning, only model parameters in the first layer and in the encoder blocks are adjusted based on the experimental training data, while model parameters for the decoder blocks, the final layer and skip connections remain fixed. Similarly, for decoder finetuning, model parameters in the decoder blocks and in the final layer are adjusted. Figure 8 shows poor damage detection for a model which has been trained with numerical data and tested on experimental data. This highlights the domain gap between

the FEM and the DIC strain fields. Below, results for damage detection using a model pretrained with numerical data and fully finetuned with experimental data, are illustrated. While damages are still overestimated, model performance increases through TL. Maximum performance for a fully finetuned model, measured by the average IoU, amounts to 48.40 %. Therefore, TL bridges the domain gap between FEM and DIC data. The following section examines whether TL can yield better results than training solely based on experimental data.



Experimental input, target and output heatmap based on the smallest UNet topology trained with numerical data.

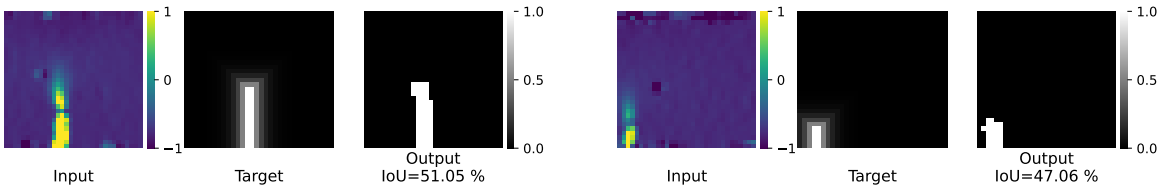
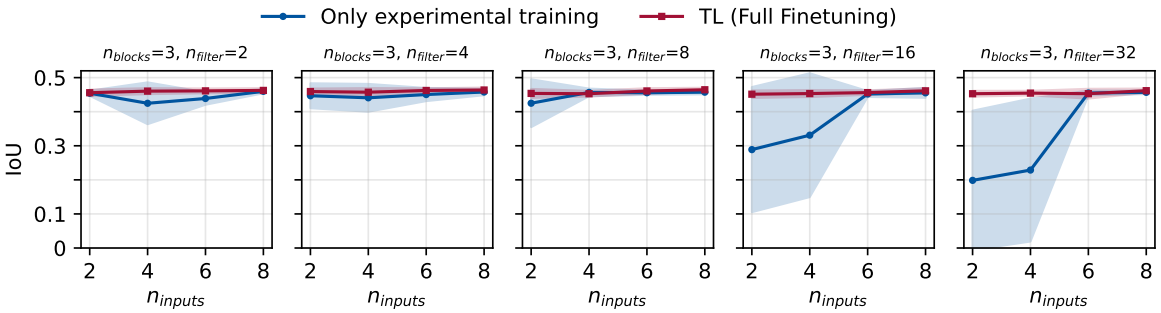


Fig. 8. *Experimental input, target and output heatmap based on the smallest UNet topology trained with numerical data and fully finetuned with experimental data.*

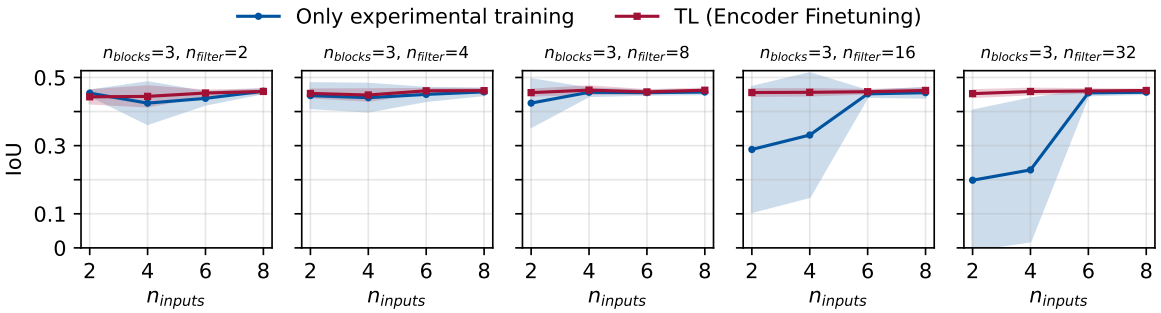
The performance of TL is measured by the average IoU of the model on the experimental testing dataset. In Figure 9, performance of different UNet topologies trained with varying amounts of samples is shown for the baseline strategy (only training the model with experimental data) and TL. Both TL strategies, full finetuning and encoder finetuning, show beneficial results for topologies with 16 and 32 filters. For these topologies and the baseline strategy, 6 training samples are required to achieve maximum performance of around 48 %. Using TL, the same performance can be achieved only using 2 training samples.

As trainings show deviating results for repeated runs and the performance of TL varies across different network topologies, a statistical approach is introduced to statistically evaluate the effect of TL for UNet-based damage detection for aluminum tubes under four-point bending. Training is carried out 20 times for each network topology with $n_{blocks} \in [1, 2, 3, 4, 5]$ and $n_{filter} \in [2, 4, 8, 16, 32]$. The number of experimental training samples n_{inputs} is varied between 2 and 8, resulting in 4000 datapoints. As trainings are repeated across different configurations, data is dependent and the exogeneity assumption is hurt, hence independent variables are systematically related to the error term. Therefore, a LMM is proposed in Equation 3 to quantify the fixed effects of network hyperparameters, the number of input samples and TL. Three interaction terms are added for TL and network topology as well as the number of experimental training samples. While i represents the training configuration, j represents the run. [8]

$$\begin{aligned} IoU_{i,j} = \beta_0 + \beta_{0,i} + (\beta_1 + \beta_{1,i}) \cdot TL + (\beta_2 + \beta_{2,i}) \cdot n_{blocks} + (\beta_3 + \beta_{3,i}) \cdot n_{filter} \\ + (\beta_4 + \beta_{4,i}) \cdot n_{inputs} + (\beta_5 + \beta_{5,i}) \cdot (TL \cdot n_{blocks}) \\ + (\beta_6 + \beta_{6,i}) \cdot (TL \cdot n_{filter}) + (\beta_7 + \beta_{7,i}) \cdot (TL \cdot n_{inputs}) + \epsilon_{i,j} \end{aligned} \tag{3}$$



*Performance of only experimental training compared to **full finetuning**.*



*Performance of only experimental training compared to **encoder finetuning**.*

Fig. 9. *Data requirement for CNN training for damage detection based on experimental DIC data and different UNet topologies with $n_{blocks} = 3$.*

	Full finetuning		**Encoder finetuning**		**Decoder finetuning**	
	coeff.	std. err.	coeff.	std. err.	coeff.	std. err.
$intercept$	0.369^{***}	0.011	0.369^{***}	0.011	0.369^{**}	0.012
TL	0.083^{***}	0.008	0.062^{***}	0.008	-0.016^{*}	0.009
n_{inputs}	0.016^{***}	0.001	0.016^{***}	0.001	0.016^{***}	0.002
$TL \cdot n_{inputs}$	-0.015^{***}	0.001	-0.014^{***}	0.001	-0.013^{***}	0.001
n_{blocks}	-0.002	0.002	-0.002	0.002	-0.002	0.002
$TL \cdot n_{blocks}$	0.001	0.002	0.004^{**}	0.002	0.006^{***}	0.002
n_{filter}	-0.003^{***}	0.000	-0.003^{***}	0.000	-0.003^{***}	0.000
$TL \cdot n_{filter}$	0.003^{***}	0.000	0.003^{***}	0.000	0.005^{***}	0.000
R^2 (marg.)	0.2571		0.2450		0.2369	
R^2 (cond.)	0.3406		0.3263		0.3348	

Table 3. Results for a LMM based on the IoU for experimental training and different finetuning strategies. $^{***}p < 0.01$, $^{**}p < 0.05$, $^{*}p < 0.1$.

Based on the above formulated regression for full finetuning, 25.71 % of the variance in the data can be explained by fixed effects whereas 34.06 % of the variance in the data can be explained by fixed and random effects. The coefficients of determination ($R^2_{marginal}$ and $R^2_{conditional}$) decrease slightly for encoder finetuning and decoder finetuning. In summary, TL, the number of training samples n_{inputs} and the number of filters n_{filter} have a significant impact on model performance for all TL strategies. TL improves model performance for network topologies with more than 16 filters in the first layer while the effect decreases with a decreasing number of filters. Also, the effect increases with a decreasing number of train-

ing samples. Decoder finetuning even has a negative effect on model performance. Model parameters in the first layer and in the encoder blocks are responsible for extracting features directly from the input data. With these parameters remaining frozen for decoder finetuning, the domain gap, that mainly exists between the numerical and the experimental input data, can not be bridged. Finally, full finetuning and encoder finetuning show similar improvement of model performance.

3. Conclusion

This study described the creation of a DIC-based labelled strain dataset for damage detection and dimensioning (Level 3 SHM) in aluminum tubes. A corresponding numerical dataset has been created based on a FEM model and the existence of a domain gap between the two datasets has been shown. By comparing different TL strategies with models only trained with experimental data, a significant positive effect of TL on model performance has been demonstrated. TL allows compensation for the performance drop due to reduced training samples in large network topologies and can reduce data requirements for CNN training based on the example presented within this work. In summary, TL is beneficial, when model capacity is large (e.g. for complex relationships between sensor data and damage mode) and limited training data is available. While full finetuning and encoder finetuning perform similarly well, decoder finetuning does not allow to close the domain gap within this setup.

Training stability has represented a key issue in this work and could be addressed through an improved loss function. Also, this work has only focused on adapting the CNN by finetuning to bridge the domain gap. Alternative methods that address the input data and the loss function might yield further improvement regarding performance and should be considered in future work. Finally, the approach has only been tested on a small dataset of 12 aluminum tubes and further experiments should be conducted for differnt materials or geometries with more complex damages to prove generalisability and applicability.

References

[1] T. Dash, S. Chitlangia, A. Ahuja, and A. Srinivasan. A review of some tequniques for inclusion of domain-knowledge into deep neural networks. Nature Scientific Reports, 2022.

[2] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. Proceedings of the IEEE, 2021.

[3] Anders Rytter. *Vibrational Based Inspection of Civil Engineering Structures*. PhD thesis, Dept. of Building Technology and Structural Engineering, Aalborg University, 1993.

[4] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, pages 234–241. Springer, 2015.

[5] F. Rentzsch Holm, T. Schalm, F. Eggers, and K.-U. Schröder. Lessons learned in developing a convolutional neural network to identify damage modes in type-iv composite overwrapped pressure vessels. In M. Schagerl M. Krommer, M. Nader, editor, *10th EC-COMAS Thematic Conference on Smart Structures and Materials*. 2025.

This work is licensed under CCBY 4.0

Media and Publishing Partner NDTnet

[6] thyssenkrupp Materials Services GmbH. Werkstoffdatenblatt en aw-6060. Technical report, Technology, Innovation Sustainability (TIS), 2017. last accessed: 05.02.2026.

[7] R. Azad, M. Heidari, K. Yilmaz, M. Hüttemann, S. Karimijafarbigloo, Y. Wu, A. Schmeink, and D. Merhof. Loss functions in the era of semantic segmentation: A survey and outlook. 2023.

[8] M. Nguyen. *Regression Techniques for Data Analysis*. Springer Verlag, 2026.

This work is licensed under CCBY 4.0

Media and Publishing Partner