PAPER

Deep learning for impact detection in composite plates with sparsely integrated sensors

View the article online for updates and enhancements.

You may also like

Recent citations

Deep learning for impact detection in composite plates with sparsely integrated sensors

Andreas M Damm[1,2,3] , **Claudio Spitzmüller**[1,3], **Andreas T S Raichle**[2,3], **Andre Bühler**[3], **Philipp Weißgraeber**[1,3] **and Peter Middendorf**[2,3]

[1] Robert Bosch GmbH, Corporate Research, Renningen, Germany
[2] Institute of Aircraft Design, University of Stuttgart, Germany
[3] ARENA2036 Research Campus, Stuttgart, Germany

E-mail: Andreas.Damm@de.bosch.com

Abstract

In this paper, both location and energy of impacts on an anisotropic carbon fiber reinforced plate (CFRP) are detected with the help of deep learning. We introduce sparse low-cost sensor array integration in CFRP plates that allows for structural monitoring of lightweight structures. Using a resin transfer moulding process microelectromechanical systems (MEMS) and piezoelectric transducers (PZT) sensors are integrated into CFRP plates. We developed an automated test bench to perform weight drop impact loadings with impact energies ranging between 0.22–0.56 mJ on a 1×1 cm^2-grid with 441 locations. The obtained sensor signals were processed by means of a short-time fourier transformation and used as input for the training of a deep learning model. This model was implemented with a convolutional neural network. To accelerate the training phase we introduce a coarse analytical model that generates artificial sensor signals we use for pretraining of the neural network. Yielding high prediction accuracies of 99.82% and 98.68% for a correct classification of impact location and energy, respectively, the capability of the proposed approach was demonstrated. Despite their limited resolution the low-cost MEMS accelerometers were able to correctly locate an impact and its energy with 99.76% and 97.04%, respectively. The pretraining led to an increased robustness of the training process. Additionally, for the case of PZT sensors, it also reduced the number of required epochs for convergence significantly.

Keywords: structural health monitoring, impact detection, fiber-reinforced composites, microelectromechanical system, piezoelectric transducer, convolutional neural network, artificial intelligence

(Some figures may appear in colour only in the online journal)

1. Introduction

In carbon fiber reinforced plastics (CFRP) impacts can cause critical delaminations. Continuous cyclic loading of a pre-damaged structure leads to an internal crack propagation and further weakening of the component until it experiences critical failure under regular loads [1]. Accordingly, knowledge about the energy of an impact can help quantifying residual fatigue strength and estimating the need for inspecting the damaged component. Since most traditional non-destructive testing methods can only help inspecting a structure locally, knowing the location of an impact proved to be a key factor in successful examination of the component [2]. Quantifying the energy of an impact as well as localize it is the basis for estimating the severity of resulting damages [3]. Because surfaces are most heavily exposed to impacts, the majority of the publications to be presented use composite plate structures as specimen.

Common sensors for measuring accelerations are piezo-electric transducers (PZT). Because they can be produced with small dimensions and allow for high-frequency sampling only restricted by the acquisition equipment, they have been frequently used for impact detection [4–6]. A common technique is comparing the time of arrival (TOA), that triangulates the location of an impact. The arrival of a transient strain wave is registered by synchronized sensors. Together with the material's sound propagation velocity, the time differences yield information about the distance of the impact to each sensor [7]. Although a transfer to plates with anisotropic sound propagation is possible in theory [8, 9], building analytical models for anisotropic structures like FRP remains a challenge. One feasible approach is reconstructing the force history of an impact and using the systems transfer function to calculate sensor signals. By minimizing the error between measured and calculated sensor signal, an impact position can be determined iteratively or in a selective process out of multiple candidate points [10, 11].

Since they allow correlating complex phenomena, correspondingly more and more recent studies resort to artificial intelligence (AI) for impact detection [12, 13]. Artificial neural networks (ANN) have been used in an early work from 1994 for impact localization [14] and since then were improved in different studies. Investigations range from determining the influence of damages in plates on the possibility to detect impacts [15] to using embedded PZT networks for analyzing an impact [16, 17]. However, those approaches all exploit features based on an isotropic propagation of the transient impact wave. By formulating propagation velocities as a function of the fibre angle, these studies were improved [18]. Other approaches are building a simulative model to predict anticipated sensor signals [19, 20] or using phase velocities as features [21]. Further studies using surface applied PZT sensors for impact detection can be found in [22]. Convolutional neural networks (CNN) are able to evaluate spatial informations within data, as well as depicting complex relationships. Due to these properties they have proven superior to classic feed forward neural networks in many applications. CNNs are more complex networks that convolve data with a kernel which effectively reduces the numbers of weights needed, due to weight sharing. Since several years the number of publications related to both structural health monitoring (SHM) and CNN rose constantly. Popular publications are using CNNs for detecting general problems in rotating machinery [23, 24], but also for specific tasks like locating loosened bolts in a steel frame [25], performing a fault diagnosis in gearboxes or analyzing pedestrian load scenarios on bridges [26]. The results seem promising, and one recent publication successfully used a CNN to localize and detect impacts on composite structures with PZT [27]. This emphasizes the potential of a CNN in the context at hand.

Although PZT—alone or in combination with AI—offer a great possibility to detect an impact on a composite structure, a fundamental problem remains. For a mass-scaled application the price of PZTs and their required peripheries such as operator amplifiers and analogue to digital converters is too high. This obstacle can be tackled with an innovation

from consumer electronics. Microelectromechanical systems (MEMS) are using measuring principles based on classical electro-mechanical models, for example the change of capacity. The analogue data is converted to digital values and transmitted as binary data within the sensor package, making communication easy and reliable. An on-board microprocessor offers the possibility to program the sensor and an I/O interface allows for individual applications. Their small size, cheap price, and high robustness are key differences to classical analogue sensors [28]. However, their measurement resolution is limited. While PZTs can measure frequencies up to 10 MHz, a small-sized commercially available MEMS-accelerometer is currently limited to a maximum sampling rate of only a few kHz. Since the time differences of transient signal arrivals in classical TOA methods are very small, MEMS are not able to resolve the required time information. In order to gather information about location and energy of an impact nonetheless, a similar CNN as proposed in [27] is used in our approach.

With this study we show that a CNN can enable embedded MEMS sensors to detect and analyze impacts on a CFRP plate. We validate our approach with common PZT sensors. In section two of this study the production of the sensor integrated specimens and their material parameters are presented. Following, an overview of the measurement strategy and the automated data generation with subsequent signal preprocessing is given. The next section gives a brief introduction into the theory of neural networks with an emphasis on CNNs and is explaining the architecture of the CNN used within the scope of this research in detail. The section continues with an explanation on how the results are retrieved and statistically validated and closes with a description of the analytical model we used to generate artificial data for the pretraining. In the fourth section we present the obtained results for the impact localization and energy determination and discuss them with a subsequent conclusion as the last section of this study.

2. Materials and methods

This section presents the technical specifications of the sensors used and the parameters for the plate specimen's production process. Following the structure of the measurements is defined. The section continues with a description of an automated test bench and the steps of data generation and handling. Finally, the methods for data preprocessing will be presented.

2.1. Sensors and composite plate

The aim of this study is to realize an impact detection with embedded MEMS. Since PZT sensors are traditionally used for impact detection, they serve as a reference to evaluate the potential of MEMS sensors for impact detection. Accordingly, one plate with embedded MEMS and one plate with embedded PZT were fabricated.

The piezo element used has a diameter of 35 mm and a thickness of 0.56 mm and is of the type EPZ-35MS29F. It has a resonant frequency at 2.9 kHz and a resonant impedance of 250 Ω. The sensors were connected by soldering isolated

Isolated PZT

Isolated tin pads

(a) Lower half
dry CF fabric

(b) Top view dry CF
fabric in RTM tool

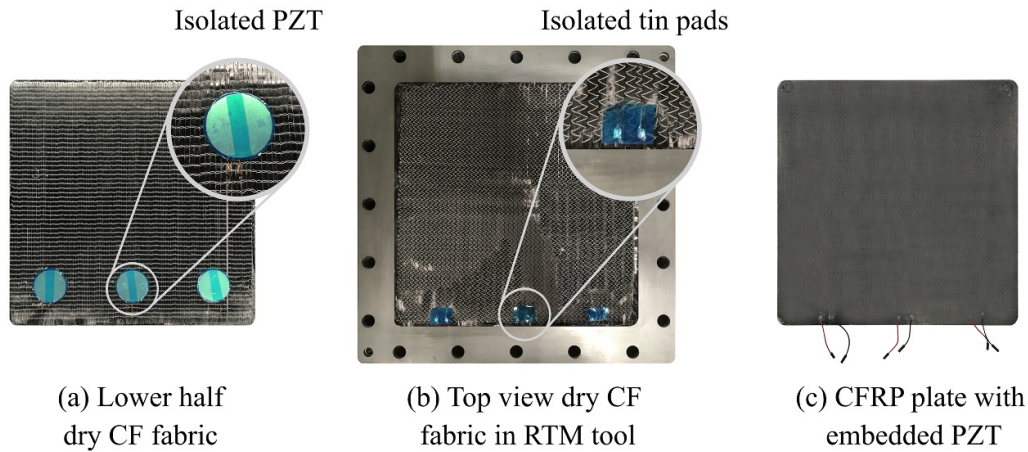(c) CFRP plate with
embedded PZT

Figure 1. Production steps of CFRP plate with embedded PZT sensors.

enamelled copper wires to the electrodes while blue polyester film tape avoided direct contact of the electrodes with the fibers thus preventing a short circuit. Figure 1(a) shows the PZTs as they were placed in between carbon fiber plies during production.

Selecting a suitable MEMS sensor is a balancing act and contradicting requirements have to be weighed up. The BMA 280 (Bosch Sensortec GmbH) offers a good balance between cost, package size and sampling rate for the application. It is an off-the-shelf, three-axis accelerometer in an epoxy compound package with dimensions of $2 \times 2 \times 0.95$ mm^3 with a maximum data output rate of 2 kHz. Because it is a surface-mounted element, soldering wires to the sensor by hand is not possible. Instead, it is mounted on top of a flexible printed circuit (FPC) board, see figure 2(a) [29]. The FPC measures 6×800 mm^2, holds three MEMS connected with one SPI bus, and can be easily connected to a microcontroller thanks to a standard plug.

The specimen is a carbon fiber reinforced plate dimensioned $280 \times 280 \times 2$ mm^3 with a radius of 10 mm at each corner, see figure 3(b). Four non-crimp $0°/90°$ biaxial fabric plies of B-C-416 g m^{-2}-1270 mm (SAERTEX GmbH & Co. KG) with Aksaca A-42 12 K carbon fibers (AKSA Akrilik Kimya Sanayii A.S.) were stacked symmetrically to the layup $[0/90/0/90]_S$ and infiltrated with the epoxy resin system Araldite LY 1564 / Aradur 5003-1 (Huntsman International LLC). The Aksaca roving has a fracture toughness of 4200 MPa and a Young's modulus of 240 GPa and a density of 1.76 g cm^{-3}, leading to an areal weight of 416 g/square/m for the dry SAERTEX fabric. The Huntsman resin system has a fracture toughness of 60 MPa, a Young's modulus of 2750 MPa and a density of 1.1 g cm^{-3}. After the sensors were placed in the neutral plane of the laminate, the mould was closed and resin was injected with 1 bar pressure with an assisting vacuum of 80 mPa using a resin transfer moulding (RTM) process. For two hours the specimen were cured at 75 °C with a holding pressure of 4 bar resulting in a fiber volume content of 47.5%.

The PZT sensors were manually placed inside the dry fabric at the defined positions, see figure 3(a). By pinching the copper wires through the fabric and soldering on solder tin pads,

it was possible to connect the integrated sensors after the manufacturing process. For this, the pads were drilled up after curing and wires were soldered on. To prevent short circuits, the pads were isolated with the same tape as the PZTs themselves which can be seen in figure 1(b). Figure 1(c) shows the finalized specimen with connected PZTs as it was used for impact testing.

The RTM tool was custom-made in order to route the PCB out of the tool into a clamping device, which allows for reproducible integration of MEMS sensors [29, 30]. While this means that an additional strip of PTFE film is necessary to seal the tool, at the same time it offers an easy connection of the sensors both during and after production, see figure 2(b). Figure 2(c) shows the consolidated specimen after demoulding and deburring. The output of the embedded MEMS was validated by evaluating the effect of gravitation on the z-acceleration.

2.2. Measurement definition

Within this investigation the two major tasks of impact localization and energy determination are examined. To ensure a meaningful discussion of the results after the measurements, evaluation parameters are defined beforehand. Hence, for impact localization a tolerance radius as well as a desired prediction accuracy were defined. If the position guessed by the model lies within a radius of the actual impact location, the guess is counted as correct and therefore contributes positively to the overall accuracy. Due to the absence of an external specification, the tolerance radius is chosen to be r $= \sqrt{2}$ cm, which is seen as sufficiently small for the given task. The area spun by the radius represents 1.4% of the total impacted area and 0.8% of the total plate size. Without any prior information about the capability of the proposed method the definition of a target prediction accuracy is challenging. However, an accuracy of greater than 99% can confidently be accepted as an accurate result. Although introducing a circular tolerance is intuitive, it is less practical for the definition of impact locations with G-Code. Therefore, a two-dimensional grid with fine increments of 10 mm is applied. Figure 4 shows the
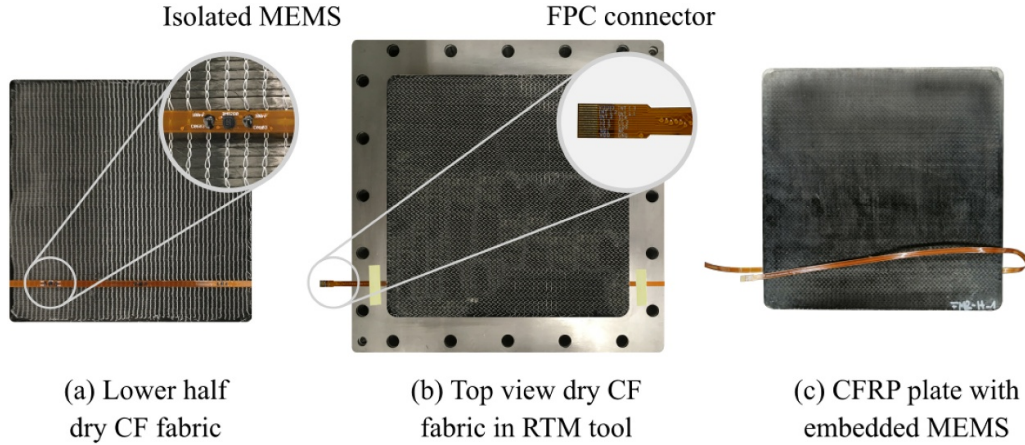
Figure 2. Production steps of the CFRP plate with embedded MEMS sensors.
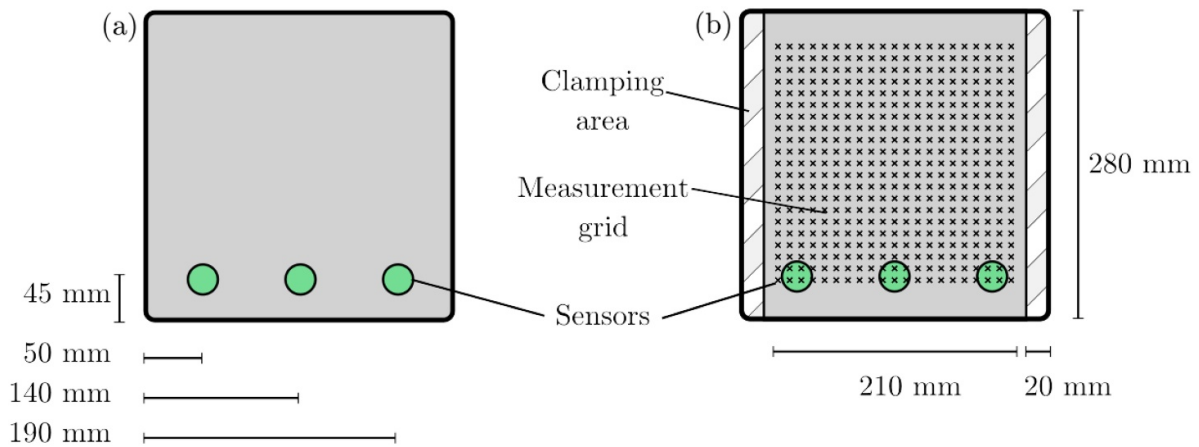

Figure 3. Location of sensors in CFRP plate (a) and plate test set up conditions (b).

training grid, with the green area representing the precision our algorithm is allowed to consider as correct. The grid of crosses represent all 441 impact coordinates. These coordinates were used to label the measurements taken during data acquisition. Throughout training these finer classes are used. This makes the training task more challenging than the actual task, as later nine impact locations lie within the actual tolerance and relate to a correct prediction. The idea is to improve final test results by artificially increasing the difficulty during training. For impact energy detection the measurement grid was adopted, but the labelling is changed to the amount of energy introduced during the impact.

Figure 4. Training grid and accepted tolerance of impact location.

2.3. Automated impact bench

One challenge that deep learning algorithms have in common is the large amount of data needed to train a network to satisfaction. Reproducing manually introduced impacts is very time-consuming and lacks reproducibility. Hence, a test bench based on an assembly kit for additive manufacturing applications was modified for this purpose, see figure 6. With its traverse axes and electric motors, the head can be moved with computerized numerical control (CNC) within a range
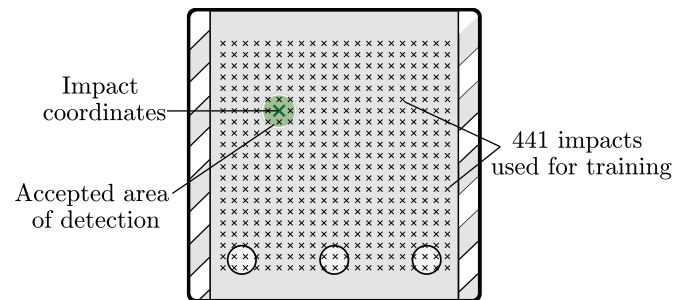
of 350 mm in direction of both *x*- and *y*-axis and 200 mm in direction of the *z*-axis. The motors are driven by an Arduino based motherboard and controlled with the software *Repetier-Host*. In G-code, an arbitrary movement of the head within the maximum ranges can be programmed. As the composite plate is clamped on two opposite sides, the regarded impact area in this study was chosen to be a square of 210×210 mm^2. The head stops at every intersection of the 10 mm grid within this area to perform an impact, see figure 3(b). For this purpose, the
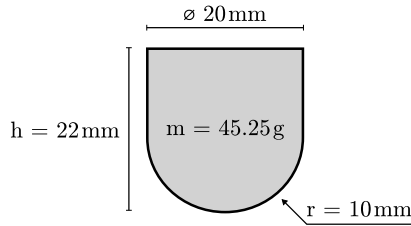
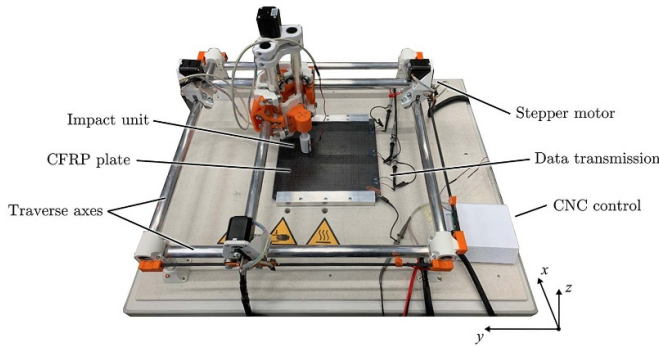**Figure 5.** Dimensions of the drop weight.

Figure 6. Three axis CNC bench set up used to generate reproducible impact data.

CNC head carries an impact unit. It consists of an electromagnet, a drop weight and a fence for an automated lifting after the impact. The drop weight has a spherical tip and a flat bottom which allows for an automated pick up with the magnet, see figure 5. When the impact unit reaches a specified coordinate, a 5 V signal is sent to trigger both the switching off of the magnet and the start of data logging. A PICOSCOPE 6402D four channel oscilloscope (Pico Technology Ltd) digitizes and saves the analogue acceleration data of the piezo sensors. For two seconds, the 8 bit resolved acceleration is recorded within a range of ±0.5 V with a sampling rate of 500 kHz. The communication with the MEMS sensors is realized with an ARDU-INO DUE microcontroller board which stores 1.49 seconds of data sampled with 2 kHz on a SD card. With a resolution of 14 bit, the acceleration is discretised in 0.244 mg bit^{-1} steps for a measurement range of ±2 g.

2.4. Data generation

Table 1 shows the measurement campaigns used for the neural networks in our approach. Each campaign consists of multiple measurement series, each of them containing impacts on all 441 locations. One measurement series is completed, once the time series for all possible impacts were recorded.

For the impact localization each defined position was impacted 48 times for each sensor. Only impacts with a constant drop height of 3 mm and a constant energy of 1.33 mJ have been applied. Each of the 441 impact positions serve as a distinct label for the measured time series. The 48 series contain 441 measurements each, leading to a total data set size of 21 168 time series with 48 measurements for each label.

For impact energy distinction three different energy levels are considered. To vary the energy levels, three different drop

heights h_i were chosen. The impact energy is calculated with the potential energy of the drop weight $E_i = mgh_i$. The three drop heights were 1 mm, 3 mm and 5 mm. Together with a fixed test weight of 45.25 g they correspond to 0.44 mJ, 1.33 mJ and 2.22 mJ of introduced impact energies. In the impact energy measurement campaign each of the 441 positions were impacted 25 times with each drop height, leading to 75 impacts per position in total. In this campaign, the three energy levels are the classes the CNN predicts, so all impacts of the same height share the same class, regardless of their location on the structure. With 75 impacts on each location, this campaign contains 66 150 measurements for each sensor in total.

2.5. Data preprocessing

This subchapter presents the methods of preprocessing used to transform the data to a suitable input for the proposed CNN. In the following the data reduction, artifact elimination and spectrogram preparation are explained.

The PZT sensors measure acceleration for a sampling time $t_s = 2$ s with a sampling rate $f_s = 500$ *kHz*. Applying the Nyquist-Shannon theorem ($f_s > 2f_{max}$) yields a maximum resolvable frequency $f_{max} < 250$ kHz [31]. Although this means that the bandwidth available to extract features is wide, it also results in a densely packed time series with 10^6 data points. In order to reduce the amount of data, only 85 000 data points are processed. This equals a measurement time of 170 ms. By determining the time when the first sensor sees the introduced impact, the main impact behaviour can be isolated. All three sensors subsequently use the same starting point, which allows preserving the information about the wave's time of arrival. Since MEMS sensors have a maximum data output rate of 2 kHz, its corresponding measured signal is of much lower resolution. Due to variations in the micro structures, the real elapsed time between measurements was 503 μs on average, leading to a sampling frequency of 1988 Hz. According to the Nyquist-Shannon theorem the MEMS can detect frequencies up to a maximum of 994 Hz. Since the total amount of data points is only $n = t_s f_s = 1.49$ s $\cdot$ 1988 Hz = 2962, these low resolved measurements need no further downsampling. The maximum acceleration By determining the time ±2g, every acceleration exceeding these limits is cut off. The setting can be changed in the sensor to a higher value, due to the accompanying reduction in resolution we decided in favor of the resolution. Figure 7 gives an overview of the raw time series as they were measured with the piezo and MEMS sensors for a drop height of 1 mm. It shows the measurement ranges of both sensors as well as the ability of the MEMS to measure the gravitation. It's constant has been used to verify the correct positioning and sensitivity of the MEMS inside the structure. The dropping weight induced multiple smaller impacts following the initial one which can also be seen in the sensor signals. For the sake of completeness, artificially created sensor data used for the pretraining of the network is presented in the figure as well. A detailed description of the generation will follow in the next section.

Table 1. Overview of measurements defining the number of measurement series used for training, validation and test set for the respective studies.

	Sensor	Impact energy	Impact grid	No. measurement series			No. labels	Training data per label	Overall data points
				Training	Validation	Test			
Impact location	Piezo MEMS	1.33 mJ	21 × 21	37	8	3	441	37	21 168
Impact energy	Piezo MEMS	0.44 mJ 1.33 mJ 2.22 mJ	21 × 21	19	4	2	3	8379	66 150
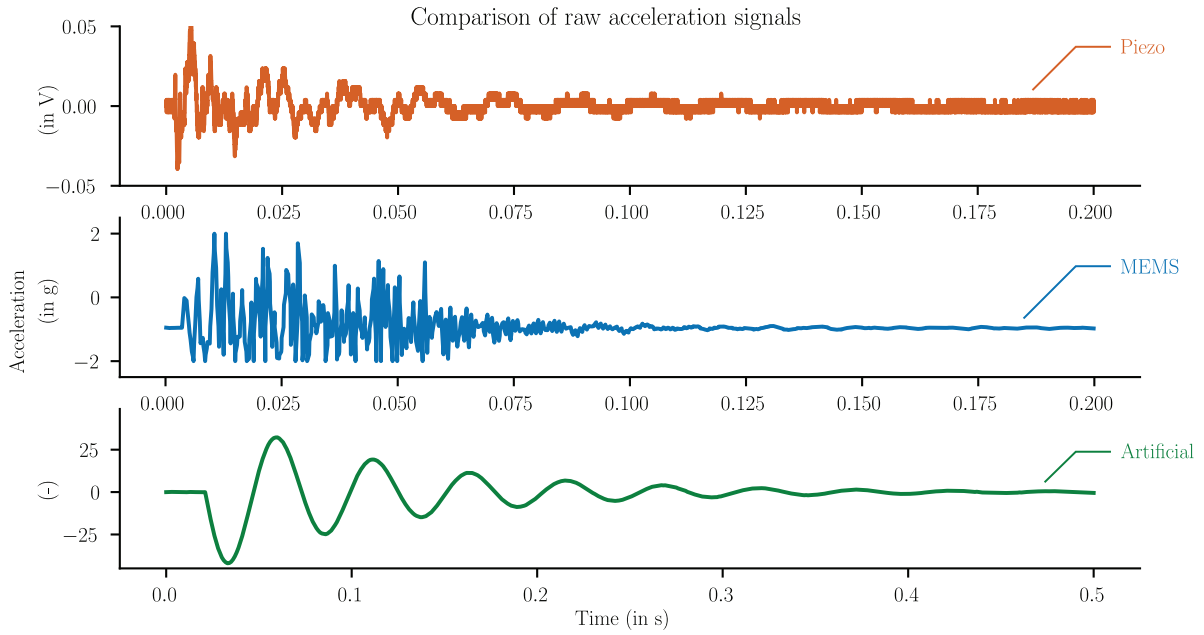
Figure 7. Comparison of raw time histories: Measured by piezo, MEMS and artificially created data.

After data preparation, the occurred frequencies are calculated. A short time fourier transformation (STFT) takes into account the transient nature of the signal and thus was chosen to transform the time series into the frequency spectrum. For a real sequence $x(n)$ the STFT $s_m(\omega)$ can be calculated as :

$$s_m(\omega) = \sum_{m=0}^{N-1} x(n)\, w_K(n - mR)\, \mathrm{e}^{-i\omega n}, \qquad (1)$$

where a Kaiser window w_K of the duration L is shifted N times over the time series with a step width of R [32]. Therefore, the number of steps $N = n/R$ equals the time resolution on the spectrogram's abscissa. For MEMS sensors, with $L_{\mathrm{MEMS}} = 128$ and $R_{\mathrm{MEMS}} = 64$ this results in $N_{\mathrm{MEMS}} = 2962/64 = 46$ steps. For the piezo sensors, with $L_{\mathrm{PST}} = 256$ and $R_{\mathrm{PST}} = 128$, the abscissa is distributed in $N_{\mathrm{PST}} = 85\,000/128 = 664$ steps. The resolution of the ordinate depends on the specified size for the spectrogram, since ω is continuous and the maximum value is the Nyquist-frequency f_{max}. In this work, all spectrograms are resampled with a lanczos filter to a uniform size of 100×100. This provides the neural network with a constant input independent of individual measurements or sensor types and reduces the computing effort. The frequencies' amplitudes are represented as a decimal value which can be plotted using a color map. Figure 8 shows the two-dimensional spectrograms for an introduced impact, calculated from data of the three sensors, respectively. Those three spectrograms are stacked, resulting in a three-dimensional array, which serves as the input for the CNN.

3. Convolutional neural network

After a brief introduction into neural networks, this section will give an overview of the architecture used within this study. A closer look is also taken on a pretraining approach for the CNN.

3.1. Fundamentals

The most basic artificial neural network (ANN) is a standard feedforward net, see figure 9(a). Simple processors s, so called neurons, are arranged in three types of layers: input and output layers l_0 respectively l_k with multiple hidden layers l_h in between. The neurons of the input layer s_0 and output layer s_k are connected through those of hidden layers. Information is fed into the neurons as input $x_{0,n}$, where further information
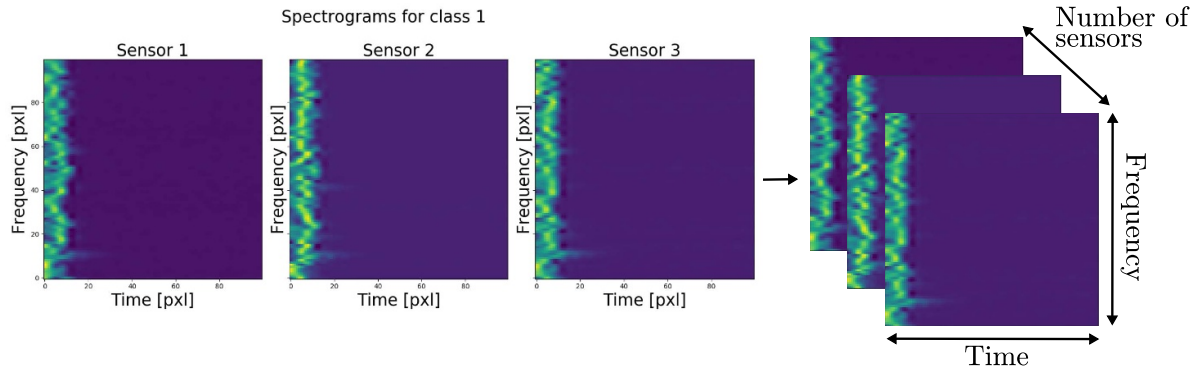
Figure 8. Sample sensor data for one specific class is stacked into a 3D array as input for the CNN.

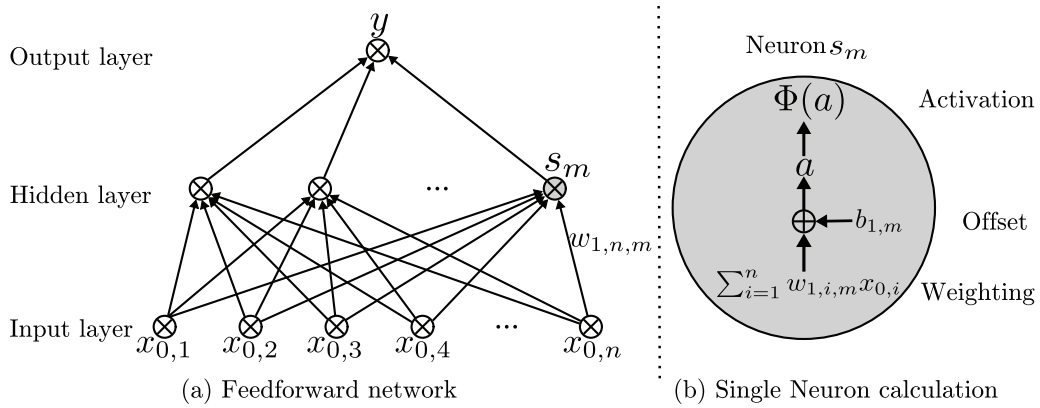(a) Feedforward network   (b) Single Neuron calculation

Figure 9. Single layer feedforward network (a) with calculations done in neuron s_m (b).

propagation is decided individually. This is realized by an activation function $\Phi(a)$ by comparing the input value to a threshold. A step function is an example for such an activation function. It calculates a single neuron's output y for an input $\underline{x}$, weighted with $\underline{w}^\top$ and considering a bias b:

$$y = \Phi(a) = \begin{cases} 1 & \text{if } a = \underline{w}^\top \underline{x} + b \geq 0, \\ 0 & \text{if } a = \underline{w}^\top \underline{x} + b < 0. \end{cases} \tag{2}$$

In a densely connected architecture, each neuron is connected to all neurons of the respective neighboring layers. Thus, all n outputs of the previous layer's neurons combined form the input $\underline{x_{h-1}}$ for the calculation in the layer h for each of the present m neurons. This leads to the following equations calculated inside a single neuron $0 < j \leq m$ in the layer h [33]:

$$a = b_{h,j} + \sum_{i=1}^{n} w_{h,i,j} x_{h-1,i}, \tag{3}$$

$$x_{h,j} = \Phi(a). \tag{4}$$

Figure 9(b) gives an overview of the calculations performed inside a neuron and shows that the output of the last layer's neuron is taken as the result of the network [34]. This proves to be a disadvantage for complex problems since the net's

weights and biases have to be adjusted manually if the result is unsatisfactory. In order to tackle this problem, ANNs often use back propagation for supervised learning. First, a cost function quantifying the error between predicted outputs and the true values is formulated. Following, the cost function is derived with respect to the trainable parameters and minimized over the course of training by using gradient descent [35, 36]. After each iteration p, all weights and biases are updated for the next iteration $p + 1$ following:

$$w^{p+1} = w^p - \alpha \frac{\partial L(\hat{y}, \underline{y})}{\partial w^p}, \tag{5}$$

$$b^{p+1} = b^p - \alpha \frac{\partial L(\hat{y}, \underline{y})}{\partial b^p}. \tag{6}$$

The learning rate α defines the step size taken in the gradient descent. While keeping computing costs as low as possible, the learning rate has to be small enough to find suitable subsequent parameters. Starting with randomly initialized values, this procedure optimizes the net's parameters step by step. To prevent the network from memorizing whole datasets instead of learning to recognize features, a dataset is usually split into training and validation set. The validation data is used to validate the neural network with non-training data after every epoch, *i.e.* every time the whole dataset was completely passed through the net. When a local minimum of the cost function is reached, the training of the neural network is finished.
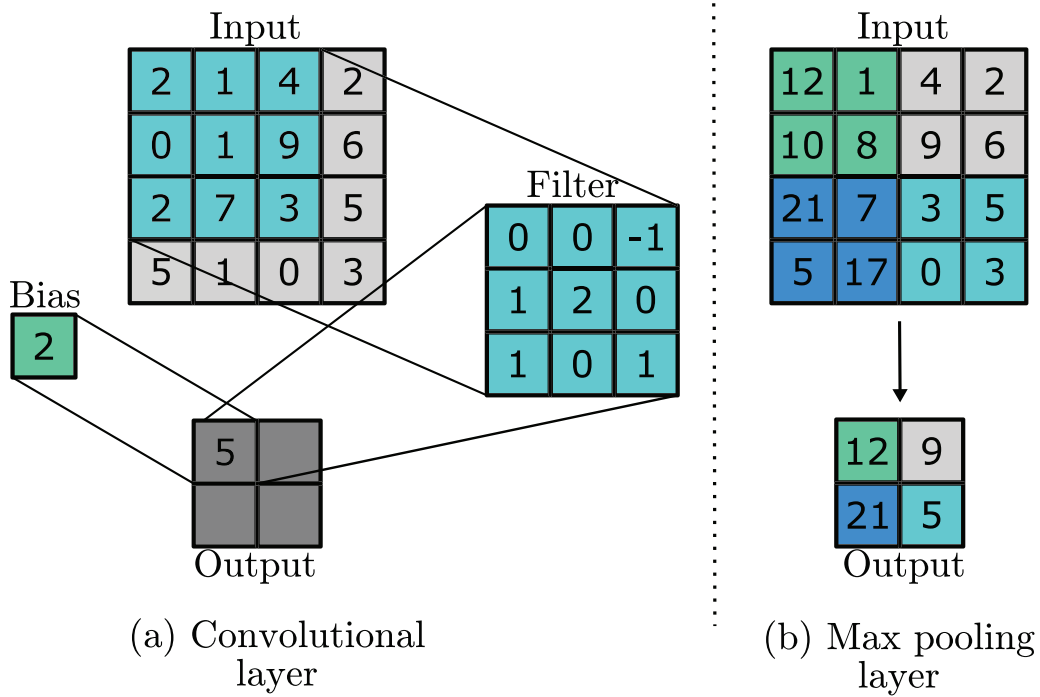
Figure 10. Schematic operation principle of convolutional layer and max pooling layer.
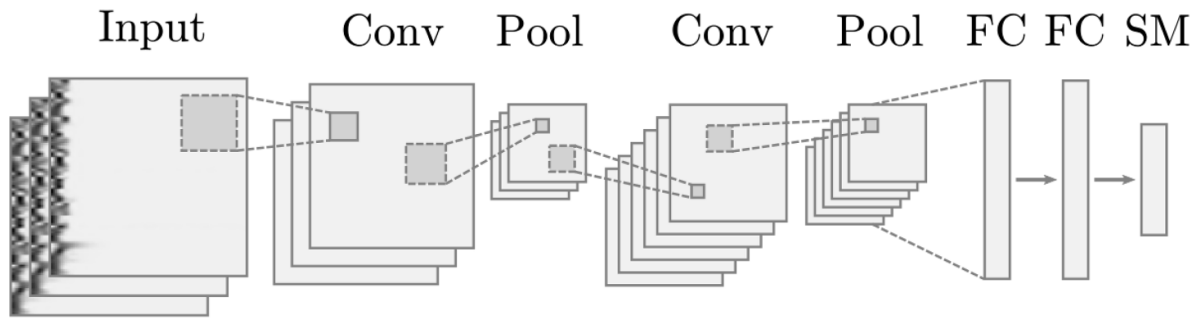

Figure 11. Schematic architecture of the used CNN.

If the task for the ANN gets more complex, more layers with more neurons have to be introduced to be able to depict the complexity of the problem. For sophisticated problems, often with spatial context, computing resources might limit the depth of regular ANNs. For such problems, convolutional neural networks (CNN) have been well established since they were published by Le Cun *et al* [37], even more so after they were run on graphic processors, accelerating the calculations considerably [38]. A CNN is especially well suited for complex tasks, because it efficiently reduces data thus needing fewer neurons to solve problems with spatial dependencies. Applications typically involve text, speech or image processing [39].

A CNN extends the approach of a regular ANN by combining differently structured layers. As shown in figure 10(a), convolutional (Conv) layers reduce the size of the input data by convolving the input and a specified filter kernel as well as adding a bias.

To further reduce data dimensions, pooling (Pool) layers are applied. The input matrices are divided into clusters on which a mathematical operation is applied, resulting in a single output, see figure 10(b). In a max pooling layer for example, only the maximum value inside each cluster is forwarded as an input to the following layer. At the end of such an arbitrarily structured convolutional layer architecture, the output matrix is flattened to a one-dimensional array. The data in this array represents feature weightings extracted from the original input data and serves as input for the following fully connected (FC) layers. These FC layers work as a regular ANN as explained above and their final layer is regarded as the output of the neural net. However, if working with a multi-classification problem, an activation function can be applied to the outputs for better interpretation. The *softmax* function (SM) is a good way to interpret the results statistically, because it normalizes the N input values so they add up to a total of 1. In this way the output of the network $\hat{y}$ represents the probabilities with which
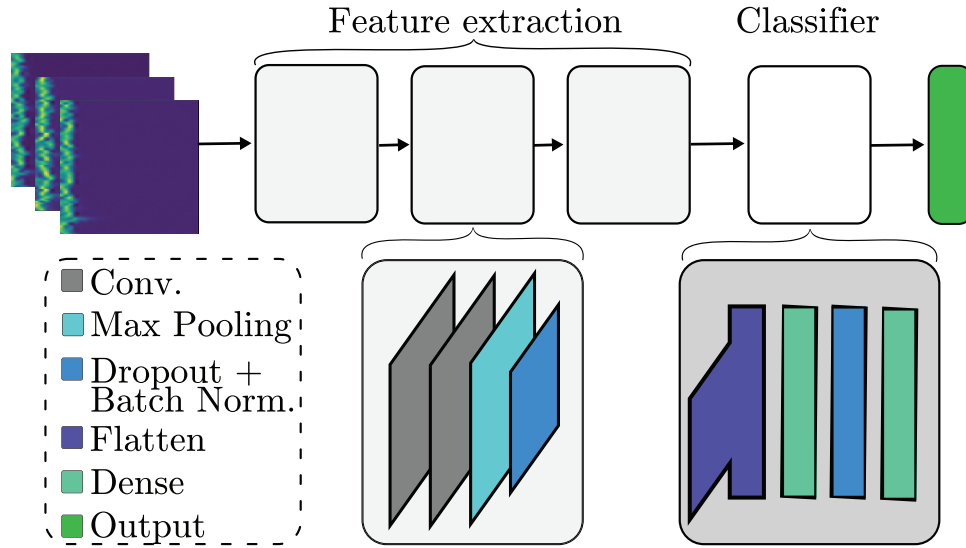
Figure 12. Visualization of architecture of the present CNN.

the CNN considers each of the *N* classes to be the correct one [40, p. 209].

$$\hat{y}_i = \Phi(a)_i = \frac{\exp(a_i)}{\sum_{n=1}^{N} \exp(a_n)} \quad \text{for } i = 1, \ldots, N. \quad (7)$$

Figure 11 shows a basic structure of a CNN containing the mentioned layers, applied on image data. Additional layers can be included in the CNN's architecture to make the result more robust. During training, dropout layers ignore every iteration a specified amount of randomly selected neurons. This helps preventing co-dependencies from developing between neurons, which would lead to an over-fitting of the network. Other sources of code robustness are batch normalization layers, which stabilize the training process. As the training data is fed into the network batch by batch, standardization of these input batches helps accelerate the training. For a more detailed view on the properties of the individual layers, further literature can be recommended [27, 37].

3.2. Network architecture

The CNN we applied is composed by a total of twenty layers, see figure 12. The input layer contains batches of the spectrograms as a four-dimensional array. The first dimension of the array describes the batch size, the other three axes contain the information of the stacked spectrograms. Every iteration image data from 32 measurements is used in order to be able to fit the data into the memory. Afterwards, features are extracted from the input data using a convolution layer. Those layers tend to learn filters identifying features like horizontal or vertical edges. As in all following interconnected layers, a rectified linear unit (ReLu) is used as the neurons' activation function. The ReLu is a common activation function which blocks negative inputs while passing through positive ones:

$$y = \Phi(a) = \begin{cases} a & \text{if } a > 0, \\ 0 & \text{if } a \leq 0. \end{cases} \quad (8)$$

After the first one, a second convolutional layer follows. For CNNs the feature complexity increases throughout the layers. Consequently, the following layer is a max-pooling layer. It is used to reduce the dimensions and thus the amount of data to be processed. Afterwards a dropout layer is following in order to regularize the network by deactivating neurons, randomly selected in every iteration. 20% of the neurons that are in front of the dropout layer are excluded from the training. This prevents a single neuron from specializing on a particular position and therefore enforces better generalization. The dropout layer is followed by a batch normalization to standardize the input for the next convolutional layers.

After the feature extraction, the classification takes place within the CNN. This is done by a classical network of fully interconnected neurons. Since the neurons in the dense layer require two-dimensional arrays, a flatten layer is used to reduce the four-dimensional arrays of the convolution to two dimensions. Within the dense layer, a L2-regularization is used to prevent overfitting and further improve generalization by penalizing large individual weightings and thus distributing the weightings better among the neurons. This is achieved by adding a weighted L2-norm to the cost function minimized during training [41].

A batch normalization is performed before the last neuronal layer. In this process, the output of the first dense layer is normalized. This prevents the gradients from becoming infinitely small or too large which accelerates training further. Finally, a second dense layer with a softmax activation function is used to determine the estimated probability distribution of the classes. Table 2 offers a comprehensive overview of the layers and hyper parameters used for the proposed CNN. During training the parameters are optimized with the Adam algorithm introduced by Kingma and Ba [42]. It is an adaptive

Table 2. Layers of the proposed CNN including their properties and activation functions used. The table presents all the hyperparameters needed to reproduce the networks used within the scope of this study.

Layer	Type	Properties	Activation Function
1	Input	Shape: $100 \times 100 \times 3$, Batch size: 32	—
2	2D convolution	Filters: 40, Kernel size: 3×3, Padding: 'same'	ReLu
3	2D convolution	Filters: 40, Kernel size: 3×3, Padding: 'same'	ReLu
4	Max Pooling	Kernel size: 2×2	—
5	Dropout	Dropout rate: 0.2	—
6	Batch-Normalization	Momentum: 0.9	—
7	2D convolution	Filters: 80, Kernel size: 3×3, Padding: 'same'	ReLu
8	2D convolution	Filters: 80, Kernel size: 3×3, Padding: 'same'	ReLu
9	Max Pooling	Kernel size: 2×2	—
10	Dropout	Dropout rate: 0.2	—
11	Batch-Normalization	Momentum: 0.9	—
12	2D convolution	Filters: 160, Kernel size: 3×3, Padding: 'same'	ReLu
13	2D convolution	Filters: 160, Kernel size: 3×3, Padding: 'same'	ReLu
14	Max Pooling	Kernel size: 2×2	—
15	Dropout	Dropout rate: 0.2	—
16	Batch-Normalization	Momentum: 0.9	—
17	Flatten	—	—
18	Dense	Neurons: 882, L2-Regularization: 0.03	ReLu
19	Batch-Normalization	Momentum: 0.9	—
20	Dropout	Dropout rate: 0.2	—
21	Dense	Neurons: 441(Location) or 3(Impact Energy)	Softmax

learning rate optimization algorithm that bases on the above given gradient descent (see equations (5) and (6)).

The sparse categorical cross entropy

$$L(\underline{\hat{y}}, \underline{y}) = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \qquad (9)$$

is serving as the cost function, with $\hat{y}$ representing the CNNs estimation and y the true label for N classes. The label is encoded in $\underline{y}$ as a sparse vector with a one marking the true labels position. To ensure the network reaches a stable state it is trained for 400 epochs.

3.3. Validation and prediction accuracies

In this paper, two separate CNNs are used to predict the location and the energy of an impact introduced onto the composite plate structure. The two tasks require a specific labelling and are trained independently of each other. For both of the networks the structure presented above is used, the number of output classes being an exception. The classes are adopted from the amount of impact coordinates and impact energies, respectively. The main objective of the CNN is to evaluate the potential of MEMS for structural monitoring purposes. Thus, the hyperparameter optimization of the network was conducted with the goal to improve the prediction accuracy with signals recorded by MEMS and not by the piezo reference sensors. To evaluate the training robustness, a k-fold cross validation was applied to the training data. For this purpose, the data sets were split into six randomly selected subsets, both for localization as well as for energy determination. Each of the subsets serve as a validation set once while the rest is used for training a CNN. In the end, multiple independently trained

CNNs with individual parameters emerge and consequently yield varying prediction accuracies. The average validation accuracy μ and its standard deviation σ is then calculated from the last ten epochs of each of these CNNs to make the validation as conclusive as possible. With a 95% probability, the prediction accuracy p on new data should therefore lie within the borders of $\mu - 2\sigma$ and $\mu + 2\sigma$. To verify this assumption, randomly chosen test sets were excluded from the rest prior to the training process of the CNNs. The trained models are used to predict these three sets for impact localization and six test sets for energy detection. The average of these prediction accuracies is presented as the final prediction accuracy

3.4. Analytical pretraining

Commonly, the weighting matrices of the individual layers are initialized randomly. As this initialization includes no information about the task itself, the idea of pretraining is to find a weight initialization representing pre-existing trained features [43]. This is done by first training the network on a pretraining task and using the obtained weights as initialization when training on real data. If the pretraining task shares similarities with the real task to be trained, it can provide a meaningful offset to start from and therefore help improving the training behaviour [44]. It is not necessary to model the problem in detail but rather to develop a coarse approximation. Therefore, an analytical model was created which simulates impact dynamics in an isotropic plate. The model is based on the propagation velocity of an acoustic wave and the distance between an impact and each one of the sensors. Figure 13 shows the sensor positions and an exemplary impact position. It also highlights the ambivalence of an estimated impact coordinate triangulated with a two-dimensional sensor array.
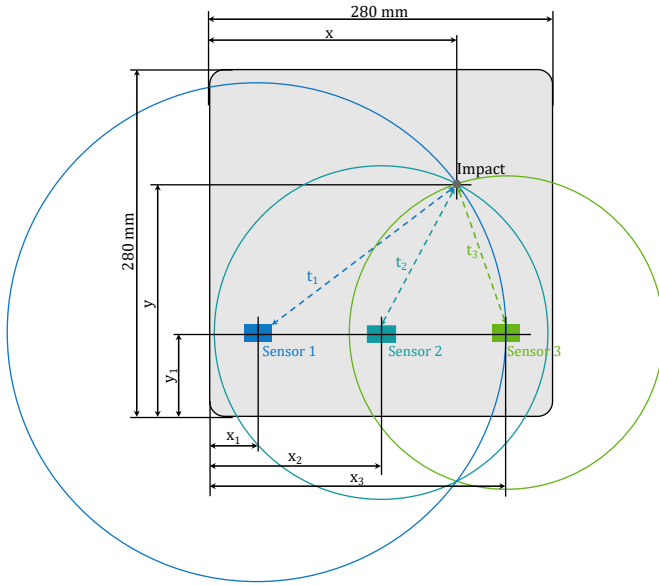
Figure 13. Geometries used by analytical wave propagation model employed in the pretraining.

(a) Loss of prediction for impact localization with piezo sensor for pretrained vs. untrained parameter initialization

(b) Loss of prediction for impact localization with MEMS sensor for pretrained vs. untrained parameter initialization

Figure 14. Analysis of the effect of the pretraining based on a simplified analytical model.

This is a known disadvantage. However, the present sensor network offers better scaling potential in an automated pick-and-place process and was therefore chosen to be suitable. For the model, plate geometry, sensor positions, and impact grid are adopted from the real scenario. A damped sine function serves as excitation source: $f(t) = A \sin(0.001 < wt) e^{-0.001 t}$. To generate multiple different time series, both frequency as well as amplitude of the excitation function are altered for each simulated impact. To mimic different impact energies, the amplitude A is selected randomly in the range of $2 \leq A < 5$. Additionally, a uniformly distributed offset frequency f_o in the range of -2 Hz $\leq f_o < 2$ Hz is applied to an initial frequency of $f = 20$ Hz. After converting the frequency to $\omega = 2\pi f$, gaussian noise with zero mean and a standard deviation of 0.05 is added to the signal. To create the sensor specific measurements, the distance of the impact location to the sensor coordinates r_{Sensor} is taken into account. Both time offset with respect to the propagation velocity and a damping decay factor λ of
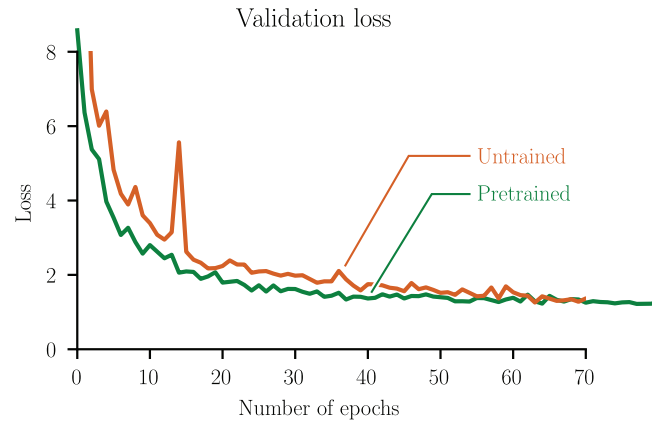
$$\lambda = -10^{-3} 1/s \qquad (10)$$

are added. The decreasing energy density of a spreading wave is considered by making the signal amplitude A_{Sensor} a distance-dependent function
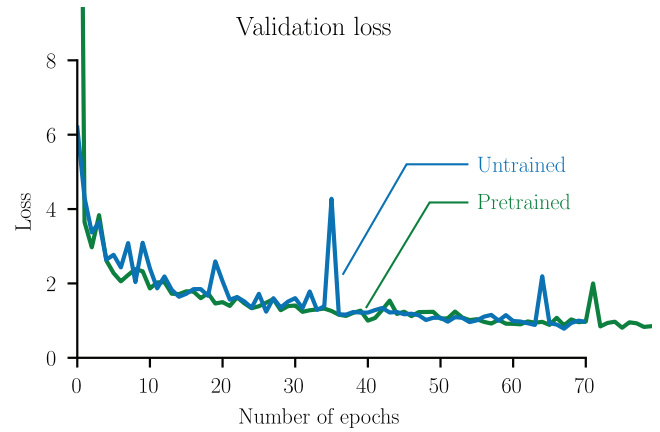
$$A_{Sensor} = A_0 \frac{1}{\sqrt{r_{Sensor}}}, \qquad (11)$$

where the initial signal amplitude A_0 decreases with the travelled distance r.

The pretraining helps the neural network to generate a set of weights that fit more to the present scenario. When the first real sensor data is introduced into the algorithm, the network already tends to extract the features that might be important for classification. This leads to achieving an earlier attainment of the net's final accuracy. Consequently, the amount of data needed for a full training of the CNN is reduced.

3.5. Implementation

The data preprocessing necessary to provide suitable input data is carried out based on the programming language Python. The networks themselves are implemented with Tensorflow, a Python based interface for machine and deep learning algorithms. Tensorflow allows to build custom neural networks while providing an optimized structure for training and optimizing neural networks. Furthermore, Tensorflow grants GPU support, which allows to accelerate the training process significantly [45]. The training of the networks took up to 5 hours and was conducted on a Nvidia Geforce RTX 2080S graphics processing unit.
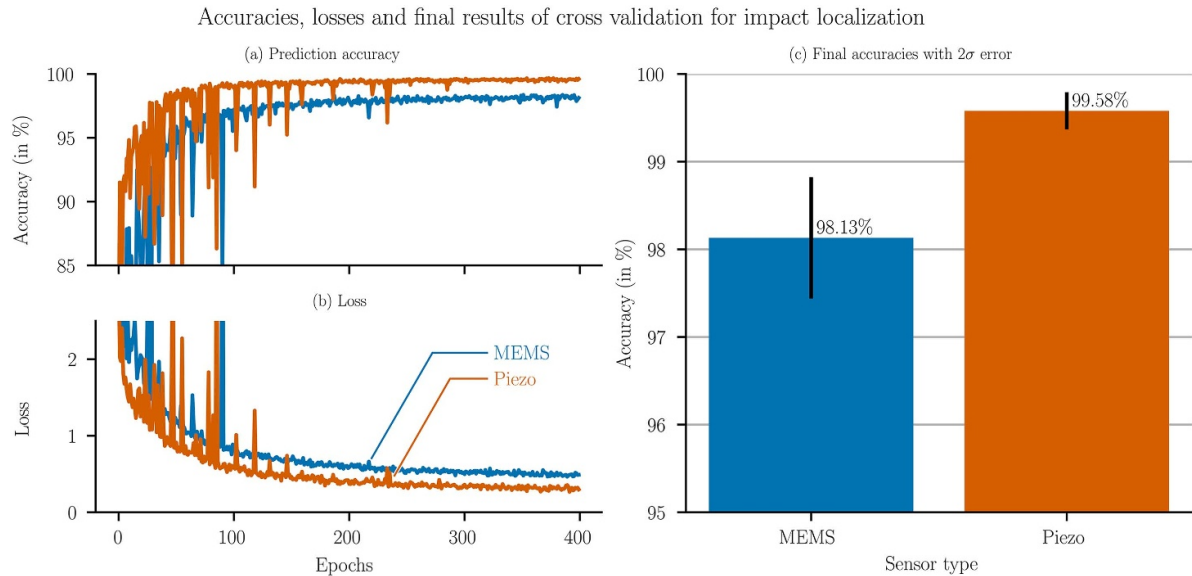
Accuracies, losses and final results of cross validation for impact localization

Figure 15. Comparing (a) validation accuracy, displayed with y-values greater than 85%, (b) loss, and (c) prediction accuracy for impact localization, displayed with y-values greater than 95%.

4. Results and discussion

In this section, first the effectiveness of pretraining a neural network with artificial data is presented, followed by the results for the impact localization and energy prediction.

4.1. Effect of pretraining

To lower the training time and amount of data needed to reach a stationary training point the pretraining approach as described earlier was applied. For testing, a model was trained for 50 epochs using artificial data that was created beforehand. The model was then saved and its weights were used as initialization for training the main model on real data for 150 epochs. Subsequently the results were compared to a randomly initialized model trained for the same number of epochs. This routine was performed for the piezo sensor data as well as the MEMS data.

As expected, the results confirm that there is no significant influence on the final accuracy. When looking at the training progress different training behaviours between the pretrained and non-pretrained model can be described. To evaluate the training behaviour it is useful to take a closer look at the loss value, which is the actual learning parameter that is minimized during training. In figure 14(a) one can observe that with pretraining the loss for piezo data starts with a lower value and reaches the plateau earlier if compared to the course of loss in the untrained model. Therefore the desired effect of faster training progress can be confirmed. Due to the high time resolution of piezo sensors, it is expected that the model makes its decision based on the time of arrival. The generated artificial data also contain this information. Thus the derived feature extractions used for the piezo data as well as the artificial data is similar enough to benefit from the pretraining approach.

However, for MEMS data the proposed pretraining approach could not meet the expectations, as can be derived from figure 14(b). The pretrained model starts off with a bigger loss than the randomly initialized model and afterwards follows a similar course. Hence, no improvement could be achieved by using pretraining. When predicting with signals from MEMS sensors, the neural network most likely needs to rely on other features than the ones learned during pretraining. Due to the limited time resolution of the MEMS sensors, a distinction by time of arrival might not be possible. The generated data seems to differ too much from the real MEMS data, thus the principles used for MEMS data classification are not represented well enough. The decrease in performance is explainable, because a pretraining with too much unrelated data can initiate the feature extractions into a wrong direction. To successfully implement the pretraining approach on MEMS data a more realistic modelling for the artificial data seems to be necessary.

Although the pretraining does not affect the network's final loss and therefore accuracy, it can be noted that the coarse analytical data provided for improving the initial weight matrix led to a smoother course of loss development over the training phase for both sensor types. While the pretraining may not contribute to an accelerated training phase for the MEMS, it still positively influenced the error minimization procedure.

4.2. Impact localization

Figures 15(a) and (b) show the averaged courses of validation accuracy and loss for both MEMS and piezo data over all cross validation splits. The CNNs were trained for a total of 400 epochs with a batch size of 32 without any overfitting occurring. As figure 15(a) shows, the validation accuracy for the piezo sensors reaches a stationary state after 100 epochs whereas it takes approximately 200 epochs for the accuracy
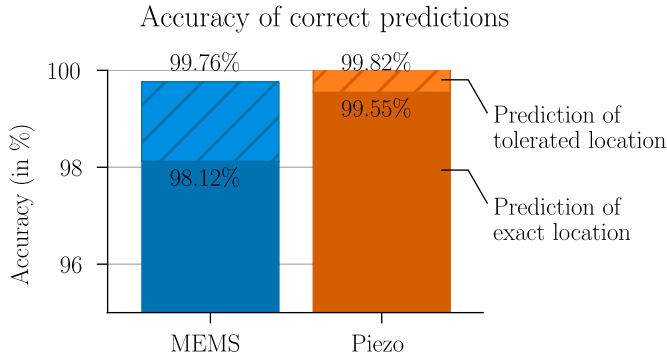
Accuracy of correct predictions

Figure 16. Distribution of valid prediction on exact and tolerated areas for impact localization.

Impact locations with their prediction deviations. Measured with MEMS. Overall average : 0.32 mm

(a) Prediction deviation map for MEMS sensors

Impact locations with their prediction deviations. Measured with piezo. Overall average : 0.16 mm

(b) Prediction deviation map for piezo sensors

Figure 17. Graphical representation of prediction deviation of every impact location, (a) for MEMS sensors and (b) for piezo sensors. The illustrated sensors are shown true to scale.

for the MEMS sensors to converge. This suggests that the information density for the MEMS data, in comparison to the piezo data, is lower and more measurements are required to properly train a CNN. As can be seen by the decreasing loss in figure 15(b), the neural network continuously improves its confidence for a correct prediction. In early epochs, oscillations in both validation accuracy and loss can be observed. This is rooted in the limited amount of data used in a single batch, since it does not represent the whole data set in its entirety. The model weights are updated after passing batches containing a small proportion of possible classes through the CNN. If the parameter optimization for the last iteration is not representative for the whole dataset, the subsequent validation will yield an unfavourable accuracy in the respective cross validation CNN. This resulting in the occasional peaks that are visible in both graphs despite the averaging. Nonetheless the validation accuracies for data recorded both with MEMS and piezo sensors rise continuously until they converge to the values of $\mu_{\mathrm{MEMS}} = 98.13\%$ and $\mu_{\mathrm{PZT}} = 99.58\%$, see figure 15(c). With $2\sigma_{\mathrm{MEMS}} = 0.69\%$ and $2\sigma_{\mathrm{PZT}} = 0.21\%$ the standard deviations are comparably small. Note that the *y*-axis starts with 95%, displaying the small difference in the validation accuracies prominently.

As can be seen in figure 16, the prediction accuracies for both sensors applied on test data position themselves well within the predicted two-σ area of the validation accuracy. The MEMS sensors yield a prediction accuracy of $p_{\mathrm{MEMS}} = 98.12\%$ and the piezo sensors an even higher value of $p_{\mathrm{PZT}} = 99.55\%$. The figure also shows the prediction accuracies for detecting an impact within the predefined toleranced area. The final prediction accuracy for both sensors to detect an impact location within this area of 6.28 cm^2 therefore is $p_{\mathrm{final,MEMS}} = 99.76\%$ and $p_{\mathrm{final,Piezo}} = 99.82\%$. The result striving towards a 100% accuracy emphasizes how well the network was trained and exceeds the desired prediction accuracy of 99%. The majority of wrong classifications were those in which an adjacent field was estimated as the impact location. This can be seen in figure 17, where each impact location is displayed with the deviation to the predicted location. The results are averaged over all six networks used in the cross validation to be able to identify systematic errors in classification. As
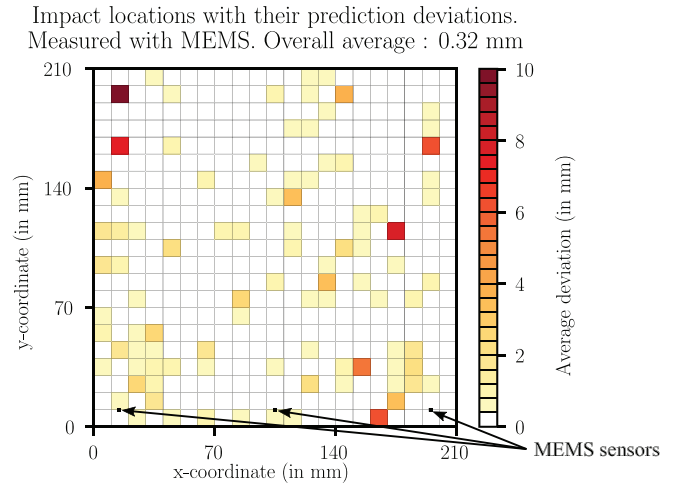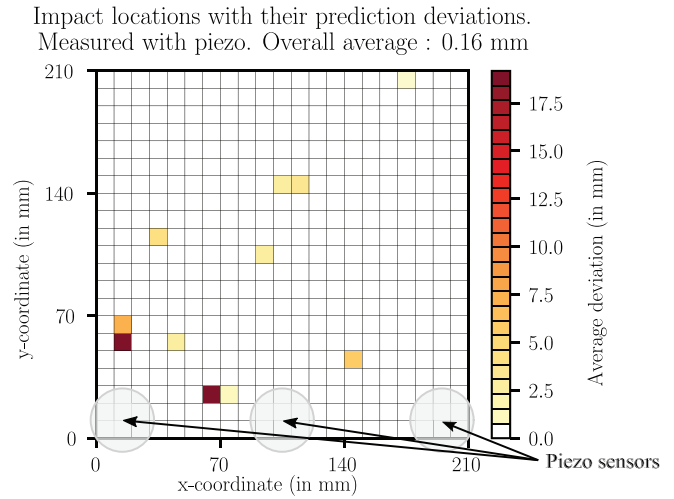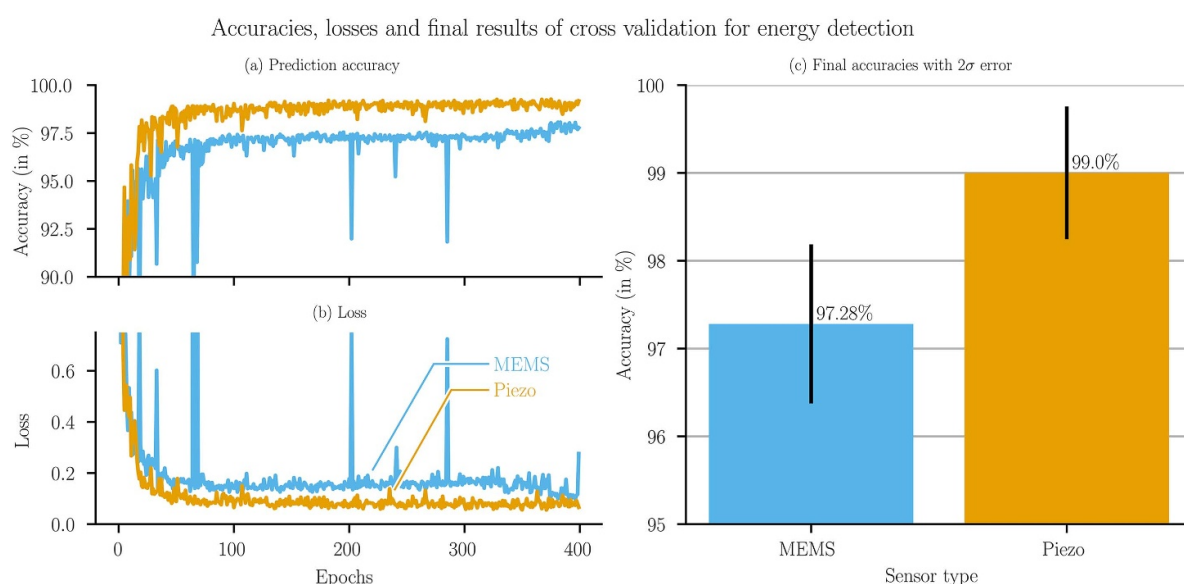
can be seen with 0.32 mm and 0.16 mm there is a small overall average deviation for MEMS and piezo sensors, respectively. Neither the prediction errors for MEMS sensors (figure 17(a)) nor those for the piezo sensors (figure 17(b)) show a significantly higher error density or gravity in the vicinity of boundary constraints or the sensors themselves.

This is a sign that the CNN learned the correct basic principles necessary to estimate impact locations reliably and even solving a regression problem seems conceivable for further studies. The convergence of the results close to a perfect prediction accuracy also confirms the correct assumptions of our training methodology. It suggests that the approach of artificially complicating the training task improves the final results.

Comparing the results with the state of the art emphasizes the great potential of using CNNs on cheap low bandwidth acceleration data recorded by sensors. All current studies

Table 3. Literature review of studies detecting low-energy impacts on carbon fiber reinforced plates by regression or within more than 10 regions.

	Study	No. of sensors	Sensor type	Approach	Average Deviation
Triangulation	Coverley and Staszewski (2003) [18]	12	PZT	Genetic algorithm (GA) triangulates excitation source	11.91%
	Zhao *et al* (2017) [46]	3	PZT	Mindlin Reissner model for dispersive wave velocity, GA for triangulation	1.46%
	Merlo *et al* (2017) [9]	3	PZT	Triangulation w/ exact determination of time of arrival (TOA)	3.41%
Model based	Tracy and Chang (1998) [6, 47]	13	PZT	Iterative comparison of analytically Kirchhoff model	3.90%
	Hu *et al* (2007) [4]	4	PZT	Iterative comparison of Kirchhoff FEA	0.29%
	Zhu *et al* (2018) [48]	4	PZT	Mindlin-Reissner FEA, subsequent iterative location optimization	3.42%
AI supported	Haywood *et al* (2004) [17]	12	PZT	Multi layer perceptron, input: TOA and maximal amplitude	6.09%
	Seno and Aliabadi (2019) [49]	8	PZT	Artificial neural network, input: TOA	0.63%
	Present study	3	PZT, MEMS	Convolutional neural network, input spectrograms	0.08% 0.15%

Accuracies, losses and final results of cross validation for energy detection

(a) Prediction accuracy

(b) Loss

(c) Final accuracies with 2σ error

Figure 18. Comparing (a) validation accuracy, displayed with y-values greater than 90%, (b) loss, and (c) prediction accuracy for energy detection, displayed with y-values greater than 95%.

rely on piezo sensors. They either use analytical or simulated models to compare the recorded data with or apply neural networks on preprocessed data. Analytical triangulation in endless fiber reinforced plastics are challenging and yield similar results with better sensors [9]. In model-based approaches it is difficult to accurately simulate the boundary conditions in accordance to test set ups. Exemplary FEM-based approaches led to prediction accuracies of 91.7% for impacts within an area of 7.07 cm^2 [4] and 87.2% for impacts within an area of 20.29 cm^2 [6]. Comparable approaches using artificial intelligence applied preprocessing techniques that need finely resolved time series as input. In [17] for example, 12 piezo sensors are used to localize the impact within an

area of 3.9 cm^2 on average after manually extracting features from the signal. A paper closely related to the present study successfully located impacts on a curved and stiffened structure. However, 12 piezo sensors were applied to the surface and the structure was only divided into three regions of 1050 cm^2 each [27].

Table 3 offers an overview on studies localizing low-energy impacts continuously or with at least 10 regions on CFRP plates. It displays the amount and type of sensors used as well as the approach the authors took. The average deviation is the averaged prediction error for both x and y in proportion to their respective specimen dimensions in millimeters. One can see that the proposed approach minimizes the state of the art by a
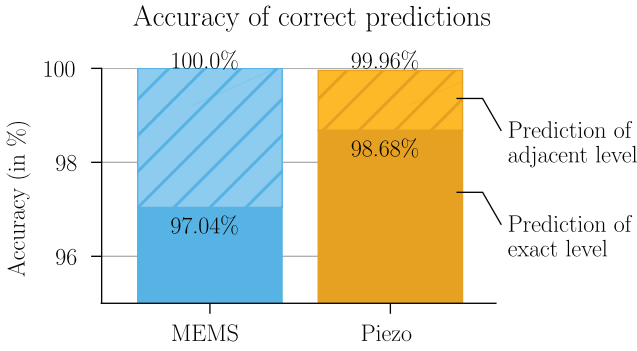
Figure 19. Distribution of valid prediction on exact and tolerated areas for impact energy.

factor of greater than 4 for MEMS and by nearly a magnitude for piezo sensors.

4.3. Impact energy determination

In order to classify impact energies equivalent to drop heights of 1, 3 and 5 mm, the individual CNNs of the k-fold cross validation were trained for 400 epochs. An early stopping criteria triggered the training to be stopped 250 epochs after the loss stopped decreasing. This led to an average training duration of 341 epochs, with the first neural network being stopped at epoch 298. This behaviour stems from the early convergence of both the averaged courses of validation accuracy and loss and can be seen in figures 18(a) and (b). Again, these figures show that the batch size of 32 is small and not representative in comparison to the amount of input values, manifesting itself in the present peaks. However, the consistency of the course is smoother than in figures 15(a) and (b), since the number of classes is reduced significantly from 441 to 3. The improvement of accuracy and loss minimization at the end of the graph on the MEMS data just before the end of training is just an apparent one. It is the effect of successively ending training data series on an averaged course. The continuing data series gain more influence with every stopping one, leading to the agitated behaviour starting with epoch 298. Figure 18(c) shows the averaged validation accuracy of each of the CNN's last 10 valid values for both MEMS and piezo data. Note that the *y*-axis starts with 95%, displaying the small difference in the validation accuracies prominently. With $\mu_{PZT} = 99.00\%$ the piezo sensors were more suitable for the task of distinguishing between tightly spaced energy levels. However, with a validation accuracy of $\mu_{MEMS} = 97.28\%$, the MEMS sensors still yielded a trustworthy result. The area of two times the standard deviations $2\sigma_{MEMS} = 0.90\%$ and $2\sigma_{PZT} = 0.76\%$ to either side of μ allows to assume the prediction accuracy for the test data.

From figure 19 it can be seen, that the test of the CNN on both data sources align well with these assumptions. The correct impact energies were predicted with an accuracy of $p_{MEMS} = 97.04\%$ respectively $p_{PZT} = 98.68\%$. For a better understanding of the nature of a wrongly predicted energy level, figure 19 displays the prediction accuracy when accepting an adjacent energy level as a correct guess. With MEMS
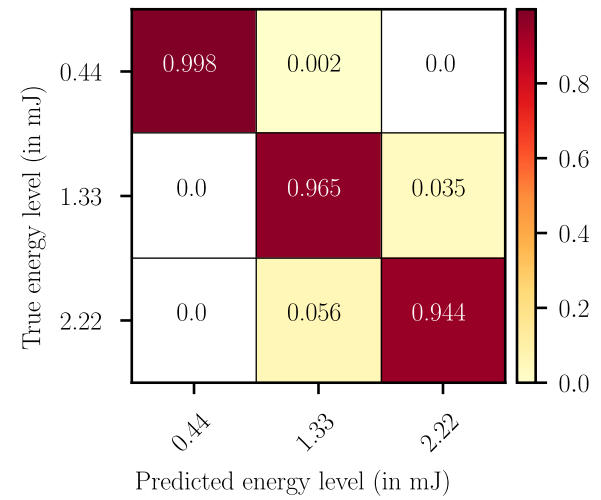
(a) Confusion matrix for MEMS sensors
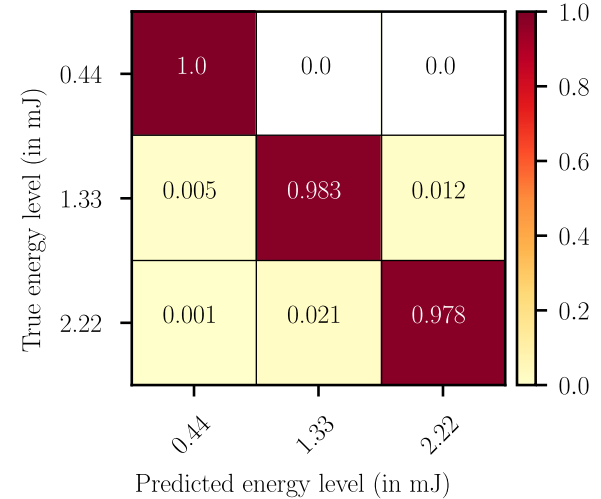
(b) Confusion matrix for piezo sensors

Figure 20. Confusion matrix displaying the amount of correctly classified energy levels and type of classification errors for (a) MEMS and (b) piezo sensors.

data, the CNN never mistakes the highest drop energy (2.22 mJ) with the lowest (0.44 mJ), whereas for piezo data it converges to 99.96%. This behaviour emphasizes that the classification defining principles for this CNN were also built in a robust and general manner. The features obtained by the CNN are modelling energy-level relevant properties well and since the current difference between energy levels is minuscule ($\Delta E = 0.89$ mJ), broadening the tolerance range should be acceptable for most use cases. With a ΔE slightly bigger, the presented approach seems to be able to predict impact

energies compliant with a zero-mistake policy. As can be seen in figure 20 both the MEMS and the piezo sensor were able to distinguish the lowest energy with almost no errors. However, the signals for the other two energy levels could not be distinguished with equal accuracy. Especially with the MEMS sensors, the network had problems distinguishing between 1.33 mJ and 2.22 mJ, see figure 20(a). The limited bandwidth of $\pm 2g$ can be attributed for that effect: accelerations exceeding the predefined limit are cut off which in turn equalizes signals of higher energies. Although not as pronounced, the piezo sensors show a similar trend of misclassification, see figure 20(b). Investigating if this antiproportional trend of decreasing prediction accuracies with increasing impact energies continues should be addressed in further studies. Note that the average accuracy in the confusion matrix differs from the test accuracies shown earlier, as the confusion matrix only takes the final accuracy of the six networks, whereas the test result was averaged over the last ten accuracies of every network to make it statistically more relevant.

Reaching similar accuracies for predicting closely packed energy levels with MEMS sensors has not been performed up to date. In [14] the authors used one set of impacts as a reference and were able to determine impact heights with a deviation up to 10% to the actual values. Two piezoelectric sensors were used to record signals for impacts with energies ranging from 0.84 J to 3.34 J in steps of $\Delta E = 0.84$ J. A prediction accuracy of 98.3% was reached in a study that varied impact heights between 20 and 80 mm in steps of 20 mm. After preprocessing the data was fed into a CNN following a similar approach as presented in this paper [27].

5. Conclusion

Within the scope of this work the suitability of structurally integrated MEMS sensors to localize impacts and determine their energy was examined. A reference based on state-of-the-art piezo sensors was used to evaluate the results. First, a CNC based test setup was defined to ensure reproducible data generation. Three piezo and three MEMS sensors were integrated in separate composite plates which were clamped into the test setup on two opposing sides. After the data generation, a suitable data preprocessing as well as a CNN for the classification tasks were designed and optimized with respect to a maximum prediction accuracy for MEMS data. The results show that although MEMS sensors offer a limited time resolution, class specific information can be extracted with the help of a CNN. As a consequence thereof, a meaningful class prediction can be given for the specific tasks. With a 99.76% accuracy for impact localization and a 97.04% accuracy for impact energy the correct values could be predicted with the help of test datasets. When compared to the piezo sensors, a comparable impact detection performance could be observed. Because the use of low-cost sensors is more cost-efficient than other approaches, the proposed sensor network could be applied for monitoring tasks in mass-produced components. A pretraining of the CNN based on data generated by an analytical model was successfully implemented. However, in its current form only the training on piezo data benefited from it. Based on the current results high potential for following studies is seen in further generalization of this AI approach. With investigating different specimen shapes under varying boundary conditions this approach's ability to transfer model parameters could be evaluated.

ORCID iD

Andreas M Damm ⓘ https://orcid.org/0000-0002-9051-8377

References

[1] Blanco N, Kristofer Gamstedt E, Asp L E and Costa J 2004 Mixed-mode delamination growth in carbon–fibre composite laminates under cyclic loading *Int. J. Solids Struct.* **41** 4219–35

[2] Choi K and Chang F-K 1994 Identification of foreign object impact in structures using distributed sensors *J. Intell. Mater. Syst. Struct.* **5** 864–9

[3] Rytter A 1993 Vibrational based inspection of civil engineering structures PhD thesis Dept. of Building Technology and Structural Engineering, Aalborg University ID 2389379256

[4] Hu N, Fukunaga H, Matsumoto S, Yan B and Peng X H 2007 An efficient approach for identifying impact force using embedded piezoelectric sensors *Int. J. Impact Eng.* **34** 1258–71

[5] Tracy M and Chang F-K 1998 Identifying impacts in composite plates with piezoelectric strain sensors, part i: theory *J. Intell. Mater. Syst. Struct.* **9** 920–8

[6] Tracy M and Chang F-K 1998 Identifying impacts in composite plates with piezoelectric strain sensors, part ii: Experiment *J. Intell. Mater. Syst. Struct.* **9** 929–37

[7] Kundu T 2014 Acoustic source localization *Ultrasonics* **54** 25–38

[8] Nakatani H, Kundu T and Takeda N 2014 Improving accuracy of acoustic source localization in anisotropic plates *Ultrasonics* **54** 1776–88

[9] Merlo E M, Bulletti A, Giannelli P, Calzolai M and Capineri L 2017 A novel differential time-of-arrival estimation technique for impact localization on carbon fiber laminate sheets *Sensors* **17** 2270

[10] Kalhori H, Ye L, Mustapha S, Li J and Li B 2016 Reconstruction and analysis of impact forces on a steel-beam-reinforced concrete deck *Exp. Mech.* **56** 1547–58

[11] Kalhori H, Lin Y and Mustapha S 2017 Inverse estimation of impact force on a composite panel using a single

piezoelectric sensor *J. Intell. Mater. Syst. Struct.*
28 799–810

[12] Khan S and Yairi T 2018 A review on the application of deep learning in system health management *Mech. Syst. Signal Process.* **107** 241–65 cited By 129

[13] Zhao R, Yan R, Chen Z, Mao K, Wang P and Gao R X 2019 Deep learning and its applications to machine health monitoring *Mech. Syst. Signal Process.* **115** 213–37

[14] Hahn H T, Wilkerson B and Stuart J 1994 An artificial neural network for low-energy impact monitoring *J. Thermo. Compos. Mater.* **7** 344–51

[15] Maseras-Gutierrez M A, Staszewski W J, Found M S and Worden K 1998 Detection of impacts in composite materials using piezoceramic sensors and neural networks *Proc. SPIE* **3329** pp 491

[16] Ross R 2006 Structural health monitoring and impact detection using neural networks for damage characterization *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Conf. 14th AIAA/ASME/AHS Adaptive Conf. 7th* p 2117

[17] Haywood J, Coverley P T, Staszewski W J and Worden K 2004 An automatic impact monitor for a composite panel employing smart sensor technology *Smart Mater. Struct.* **14** 265

[18] Coverley P T and Staszewski W J 2003 Impact damage location in composite structures using optimized sensor triangulation procedure *Smart Mater. Struct.* **12** 795

[19] Sharif-Khodaei Z, Ghajari M and Aliabadi M H 2012 Determination of impact location on composite stiffened panels *Smart Mater. Struct.* **21** 105026

[20] Ghajari M, Sharif-Khodaei Z, Aliabadi M H and Apicella A 2013 Identification of impact force for smart composite stiffened panels *Smart Mater. Struct.* **22** 085014

[21] Qiu L, Yuan S, Zhang X and Wang Y 2011 A time reversal focusing based impact imaging method and its evaluation on complex composite structures *Smart Mater. Struct.* **20** 105014

[22] Delgado L E M, Benedé Je R and Casellas Je Ví 2010 A review of impact damage detection in structures using strain data *Int. J. COMADEM* **13** 3–18 ISSN 1363–7681

[23] Janssens O, Slavkovikj V, Vervisch B, Stockman K, Loccufier M, Verstockt S, Van de Walle R and Sofie V H 2016 Convolutional neural network based fault detection for rotating machinery *J. Sound Vib.* **377** 331–45

[24] Guo S, Yang T, Gao W and Zhang C 2018 A novel fault diagnosis method for rotating machinery based on a convolutional neural network *Sensors* **18** 1429

[25] Abdeljaber O, Avci O, Kiranyaz S, Gabbouj M and Inman D J 2017 Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks *J. Sound Vib.* **388** 154–70

[26] Mustapha S, Kassir A, Hassoun K, Dawy Z and Abi-Rached H 2020 Estimation of crowd flow and load on pedestrian bridges using machine learning with sensor fusion *Autom. Constr.* **112** 103092

[27] Tabian I, Fu H and Khodaei Z S 2019 A convolutional neural network for impact detection and characterization of complex composite structures *Sensors* **19** 4933

[28] Mohammed Z, Gill W A and Rasras M 2017 *Modelling and Optimization of Inertial Sensor-Accelerometer* (Cham: Springer Int. Publishing) pp 331–45

[29] Klein L, Kugler A and Schoenfeld D 2018 Method for arranging a number of micromechanical acceleration sensors on or in a plastic component and corresponding plastic component

[30] Klein L 2019 Sensor systems for frp lightweight structures: automotive features based on serial sensor products *Sensors* **19** 3088

[31] Shannon C E 1949 Communication in the presence of noise *Proc. IRE* **37** 10–21

[32] Mertins A 2010 *Signaltheorie* ViewegTeubner

[33] Jain A K, Mao J and Moidin Mohiuddin K 1996 Artificial neural networks: A tutorial *Computer* **29** 31–44

[34] Leshno M, Lin V Y, Pinkus A and Schocken S 1993 Multilayer feedforward networks with a nonpolynomial activation function can approximate any function *Neural Netw.* **6** 861–7

[35] Kelley H J 1960 Gradient theory of optimal flight paths *Ars J.* **30** 947–54

[36] Bryson A E 1961 A gradient method for optimizing multi-stage allocation processes *Proc. Harvard Univ. Symp. on Digital Computers and Their Applications* vol 72

[37] LeCun Y, Boser B E, Denker J S, Henderson D, Howard R E, Hubbard W E and Jackel L D 1990 Handwritten digit recognition with a back-propagation network *Advances in Neural Information Processing Systems* pp 396–404

[38] Krizhevsky A, Sutskever I and Hinton G E 2012 Imagenet classification with deep convolutional neural networks In *Advances in Neural Information Processing Systems* pp 1097–105

[39] Guo M, Xiao M and Fang Y 2020 Research on deep learning algorithm and application based on convolutional neural network *Advances in 3D Image and Graphics Representation, Analysis, Computing and Information Technology* (Singapore: Springer) pp 117–22

[40] Bishop C M 2006 *Pattern Recognition and Machine Learning* (Berlin: springer) (https://doi.org/10.1007/978-981-15-3863-6_13)

[41] Krogh A and Hertz J A 1991 A simple weight decay can improve generalization In *Proc. of the 4th Int. Conf. on Neural Information Processing Systems* NIPS'91 San Francisco, CA, USA Morgan Kaufmann Publishers Inc pp 950–7

[42] Kingma D P and Ba J 2014 Adam: A method for stochastic optimization (arXiv: 1412.6980)

[43] Krig S 2016 *Feature Learning Architecture Taxonomy and Neuroscience Background* (Cham: Springer Int. Publishing) pp 319–74

[44] Yosinski J, Clune J, Bengio Y and Lipson H 2014 How transferable are features in deep neural networks? In Advances in neural information processing systems pp 3320–3328

[45] Abadi M *et al* 2015 TensorFlow: Large-scale machine learning on heterogeneous systems Software available from tensorflow.org

[46] Zhao G, Hu H, Li S, Liu L and Li K 2017 Localization of impact on composite plates based on integrated wavelet transform and hybrid minimization algorithm *Compos. Struct.* **176** 234–43

[47] Tracy M and Chang F-K 1998 Identifying impacts in composite plates with piezoelectric strain sensors, part i: theory *J. Intell. Mater. Syst. Struct.* **9** 920–8

[48] Zhu K, Qing X P and Liu B 2018 A two-step impact localization method for composite structures with a parameterized laminate model *Compos. Struct.* **192** 500–6

[49] Seno A H and Aliabadi M H 2019 Impact localisation in composite plates of different stiffness impactors under simulated environmental and operational conditions *Sensors* **19** 3659