ON THE CONDITION MONITORING OF BOLTED JOINTS THROUGH ACOUSTIC EMISSION AND DEEP TRANSFER LEARNING: GENERALIZATION, ORDINAL LOSS AND SUPER-CONVERGENCE

Emmanuel Ramasso, Rafael de O. Teloli, Romain Marcel Department of Applied Mechanics Institut FEMTO-ST, UFC/CNRS/SUPMICROTECH/UTBM, 24 Rue Alain Savary, 25000 Besançon, France *emmanuel.ramasso@femto-st.fr

June 3, 2024

ABSTRACT

[This paper was accepted in SAGE/SHM journal.]

This paper investigates the use of deep transfer learning based on convolutional neural networks (CNNs) to monitor the condition of bolted joints using acoustic emissions. Bolted structures are critical components in many mechanical systems, and the ability to monitor their condition status is crucial for effective structural health monitoring. We evaluated the performance of our methodology using the ORION-AE benchmark, a structure composed of two thin beams connected by three bolts, where highly noisy acoustic emission measurements were taken to detect changes in the applied tightening torque of the bolts. The data used from this structure is derived from the transformation of acoustic emission data streams into images using continuous wavelet transform, and leveraging pretrained CNNs for feature extraction and denoising. Our experiments compared single-sensor versus multiple-sensor fusion for estimating the tightening level (loosening) of bolts and evaluated the use of raw versus prefiltered data on the performance. We particularly focused on the generalization capabilities of CNN-based transfer learning across different measurement campaigns and we studied ordinal loss functions to penalize incorrect predictions less severely when close to the ground truth, thereby encouraging misclassification errors to be in adjacent classes. Network configurations as well as learning rate schedulers are also investigated, and super-convergence is obtained, i.e., high classification accuracy is achieved in a few number of iterations with different networks. Furthermore, results demonstrate the generalization capabilities of CNN-based transfer learning for monitoring bolted structures by acoustic emission with varying amounts of prior information required during training.

Keywords bolted joints, structural health monitoring, acoustic emission, semi-supervised learning, deep transfer learning

1 Introduction

Bolted structures play a key role in several mechanical systems used in industry to assemble parts together. Ensuring the healthy state of such jointed structures is essential in various industries, such as aerospace, automotive, and construction since failures in bolted joints can lead to significant safety risks, economic losses, affected performances and regulatory compliance.

Detecting bolt loosening or degradation in its early stages enables prompt maintenance or repair, minimizing the risk of expensive downtime. Consequently, this facilitates the optimization of maintenance schedules and contributes to

extending the longevity of equipment and structures. During in-service use, effective implementation of structural integrity monitoring systems is mandatory as the bolts can self-loosen leading to potentially catastrophic failures.

Non-destructive techniques offer valuable tools for monitoring the condition of bolted joints. We can distinguish between active and passive techniques. In the recent review of Huang et al. [1], the authors described active techniques, summarized in three categories: sensor-based [2], percussion-based [3, 4] and vision-based [5]. In the sensor-based category, ultrasonic testing is widely used. The fundamental principle of this active technique consists in an actuator, that generates an ultrasonic signal, and a sensor that receives the response, and in an algorithm that evaluates a change during transmission. The properties of the signal transmitted through the joint interface is modified as the properties of the latter evolve during in-service use. For example, the transmitted energy or spectral features of signals can be correlated with the tightening degree of the bolts. A similar principle was used in various applications such as the monitoring of corroding bolted joints [6], and the detection of composite delamination [7].

One of the causes of self-loosening is the succession of microscale damage related to sticking and slipping between the contact surfaces during vibratory cycles [8]. This form of damage eludes detection through active techniques designed to assess the overall and global behavior of bolted structures. However, such microscale damages dissipate energy by itself, which can be recorded by a passive technique called Acoustic Emission (AE) [9, 10, 11]. AE is defined as the detection of the subnanometric displacements of a material surface caused by the propagation of transient elastic waves generated by a sudden and permanent change in the material integrity. The evanescent nature of these waves requires continuous data collection using piezoelectric (PZTs) sensors, with a typical frequency range between dozens of kHz to 1 MHz, that convert displacements into voltage signals.

The main advantage of this technique lies on its *passive* nature. Indeed, the detection of a defect depends on the energy emitted by the defect's source itself, and not by an external apparatus as in active techniques. Moreover, it is highly sensitive and can detect microscale damages. This accounts for its use in many applications, like material characterization and Structural Health Monitoring (SHM), and has high potential for condition monitoring of bolts. However, very few works have been published on the application targeted by this work. This technique was not considered in the review of Huang et al. [1] because it has only recently been shown that a quantitative relationship can be established between AE signals features and bolt loosening [12]. Independently of the application, the significant challenge behind the use of AE is the *extraction of representative and robust features*, which is of utmost importance for condition monitoring. Therefore, this paper aims at addressing the following questions: what kind of features are the most relevant for tightening level classification? How these features generalize through several measurement campaigns?

A beginning of answer to these questions was proposed by Zhang et al. [12]. The AE technique was used to evaluate the failure of a bolted joint between composite materials. The energy released from AE signals was used to evaluate the residual torque of the joints within a limited range and the authors shown a shift of peak frequencies in the sprectal representation of AE signals. The study provided a proof-of-concept of the use of the AE technique for condition monitoring of bolted joints. However, the authors used handcrafted features based on empirical mode decomposition and Hilbert-Huang transform (HHT) extracted from AE signals. They also considered a sensor with a limited frequency range (20 kHz-160 kHz). Further research is thus needed to improve the feature extraction process as it can be time-consuming and both application/sensor dependent. Further work is also needed to evaluate if sensors with higher frequency range can be more valuable. Finally, it is necessary to study the performance of machine learning algorithms using the proposed features for SHM purposes.

Deep Convolutional Neural Networks (CNNs) are promising candidates for addressing the challenge behind *automatic feature extraction*. This type of *end-to-end* approach is of practical interest because there is no need of data preprocessing since it is already embedded through the numerous layers. It is still a relatively new research area in condition monitoring of bolted joints. The recent study from Fu et al. [13] demonstrated the potential of CNNs for classifying discretized tightening torque levels of bolts from highly noisy AE data and using high-frequency sensors. They considered a recent benchmarking dataset called ORION-AE [14] that allow performance comparison of detection methods. In their study, all tightening conditions were known *in advance* during training and testing, which is not compatible with SHM.

By considering all conditions in a training dataset, a data *normalization* is implicitly applied, whereby the network is able to learn the testing conditions. This is because normalization is embedded in most of pretrained CNNs, like Resnet18 or GoogLeNet, through *batch normalization (BN) layers* [15]. During testing, this type of layer keeps these statistics fixed, therefore the output of a BN layer in testing is normalized according to the mean and standard deviation of the batches estimated during training. If the training dataset comprises all conditions, the testing is thus biased. However, if all conditions are *not* known in advance and different from the conditions represented in training data, then these statistics can be less representative and the generalization of the CNN more difficult. The normalization is an important issue for SHM [16] and is often understated. While there were many studies on the application of deep

learning and transfer learning based on CNN to vibration data [17, 18, 19], the study of generalization capabilities was not tackled so far for AE-based condition monitoring bolted joints.

A partial answer to the aforementioned questions was given in previous work. For the question "What kind of features are the most relevant for tightening level classification?", it was shown in several works that the use of CNN prevents spending time on finding handcrafted features and enables automatic feature extraction through convolutional layers. However, the question "How these features generalize?" remains open - and contributions are needed on this point because generalization is crucial for SHM purposes. To fill this gap, this article contributes to this area by comparing deep learning methods and exploring parameter settings, particularly for the learning rate, to achieve high performance in generalization within a limited training time.

More specifically, our work is particularly innovative with the following contributions:

- **Study of generalization for SHM purpose of bolted joints**: We provide the first study of generalization capabilities of four pretrained networks for condition monitoring of bolts, illustrated on the ORION-AE benchmark dataset. This comprises analyzing how well these networks perform in classifying different tightening levels of bolts based on AE data using distinct campaigns of measurements. By studying generalization, this work assesses the robustness and adaptability of these networks to various conditions encountered in SHM applications for bolted joints.
- **Super-convergence to reduce iterations:** This objective aims to explore the phenomenon of super-convergence in the context of SHM applications for bolted joints. Super-convergence refers to a rapid decrease in the number of iterations required to train a neural network while still achieving high performance. By observing and analyzing this phenomenon, this work provides insights into efficient training strategies that can significantly reduce the computational burden associated with training deep learning models for SHM purposes.
- **Multiple sensors fusion:** This strategy is investigated for condition monitoring of bolted joints, as well as the **impact of pre-denoising** of AE data stream before CNN training and inference. By investigating these aspects, this study enhances the accuracy and reliability of AE-based SHM systems for bolted joints. It achieves this improvement by leveraging complementary information from multiple sensors and optimizing data preprocessing techniques.
- **Several ordinal loss functions are investigated** and compared to the standard cross-entropy loss. This objective focuses on investigating the effectiveness of ordinal loss functions in the context of SHM applications for bolted joints using CNNs. Ordinal loss functions are designed to take into account the order of classes, which is particularly relevant in SHM scenarios where the severity or level of damage may vary. By comparing ordinal loss functions with the standard cross-entropy loss, we aim to identify the most suitable loss function for accurately classifying different tightening levels of bolts based on AE data.

Toward this background, this paper is structured as follows: Section "Benchmark Description: ORION-AE Dataset" presents the ORION-AE benchmark, along with the experimental setup and the measurement protocol used to data acquisition. It also discusses the challenges encountered in performing SHM on bolted structures. Section "Methodology", in turn, describes the developed methodology, including the different modules employed in the workflow. Section "NO-SHM Case: Single Campaign Classification" introduces the results obtained in this work for a so-called NO-SHM Case; in which training, validation and test data come from the same experimental campaign. Therefore, section "SHM Case: Classification on Unseen Campaign" expands the results shown to a so-called SHM case, in which the generalization is of utmost importance, and the campaigns used to test the algorithm are not used at all during the training and validation stages. In addition, the influence of adding *a priori* knowledge about the testing campaign on the classification accuracy of the CNNs is also discussed. Next, "SHM Case: Super-Convergence" discusses, for the first time in the context of SHM on bolted joints, the super-convergence phenomenon. Finally, section "Conclusion" presents concluding remarks and outlines the next steps for further research.

2 Benchmark Description: ORION-AE Dataset

2.1 Experimental setup

The experimental setup is illustrated in Figure 1. The structure used in the work, known as the Orion beam [20, 21], comprises two duraluminium beams, each measuring $200 \times 30 \times 2$ mm. These beams are connected by three M4 bolts, spaced along a length of 30 mm. There are contact patches at each bolt connection to retain the contact between both beams in a small area. These patches consist of a square of 12×12 mm² with an extra thickness of 1 mm. The ORION-AE benchmark dataset, available at Verdin et al.[14] and described in Ramasso et al.[22], encompasses data collected from a structural monitoring campaign using various sensors. The experimental setup includes a TIRA

TV51120 permanent magnetic shaker, which applies a sinusoidal signal with a frequency of 120 Hz to excite the structure. A Polytec PSV500Xtra laser vibrometer is utilized to measure the velocity of the structure near its free end. Additionally, three AE sensors, namely mu80, F50A, and mu200HF manufactured by EuroPhysical Acoustics, are employed. These sensors are connected to a preamplifier set to 60 dB (model 2/4/6 preamplifier made by EuroPhysical Acoustics) and possess frequency ranges of 200 - 900 kHz, 200 - 800 kHz and 500 - 4500 kHz, respectively. The sensors are positioned on the lower beam, approximately 50 mm above the clamped-end, using silicone grease. All data are sampled at a rate of 5 MHz using a Picoscope 4824, which offers a bandwidth of 20 MHz, low noise, 12-bit resolution, and a 256 MS buffer memory. Note that the data collection is performed using a dedicated setup designed to replicate the loosening phenomenon observed in diverse industries, including aeronautics, automotive, and civil engineering, where bolted joints are commonly utilized for assembly.



Figure 1: ORION-AE benchmark and setup configuration [22].

The dataset consists of five measurement campaigns (#B, #C, #D, #E and #F), each involving vibration tests conducted considering different tightening levels. It is important to stress that each campaign started after the structure had been completely disassembled and reassembled. By conducting multiple experimental campaigns, available data allow to assess the robustness of the detection and classification methodology in the presence of variations in the dynamics of the bolted joint. This variability provides valuable insights into the performance and generalization capacity of the proposed approach under different operating conditions and environmental factors.

The upper bolt of the structure was subjected to a sequential decrease in tightening torque, while the other bolts remained fully tightened at a constant torque of 80 cNm. The tightening levels are applied in the following order: 60 cNm, 50 cNm, 40 cNm, 30 cNm, 20 cNm, 10 cNm, and finally 5 cNm. During the vibration tests, the torque modification process differed slightly for the tightening torque of 50 cNm. In this case, the shaker was momentarily stopped before the torque modification and then resumed afterward. For the remaining tightening torque conditions, the modification was made sequentially. Each tightening level corresponds to a specific class, ranging from 1 for 60 cNm to 7 for 5 cNm. Notwithstanding, in each campaign, the displacement at the free-end of the beam was controlled through a feedback strategy from the laser vibrometer signal. The measure of the displacement was also recorded at 5 MHz. This displacement data is used to determine the vibration cycles applied during the experiments.

In recent studies, the detection of torque loss in the bolts of the Orion beam has been investigated [23, 24]. However, these studies primarily rely on modal parameters as features, which are extracted from vibration responses with low-frequency content. Although based on physical parameters, this approach has certain limitations, as the need for data preprocessing and subsequent modal analysis. Therefore, the quality of feature extraction is highly dependent on data and the specific processing techniques employed. Additionally, resonance frequencies, being global characteristics of dynamic systems, exhibit limited sensitivity to variations when compared to measurement uncertainties, particularly at higher levels of applied torque - the effects of microslip arising from relative motion between surfaces are not significant in this context. This work aims to overcome these limitations by utilizing AE data, which possess a rich high-frequency

content and can be directly employed in deep neural networks without the need for extensive preprocessing or feature extraction.

2.2 Challenges - Heterogeneous system

Gardner et al. [25] proposed a general framework for population-based SHM, discussing aspects that allow systems to be classified as homogeneous (e.g., similar geometry) or heterogeneous (e.g., significant difference in material properties). However, the challenge with generic frameworks lies in their inability to represent less typical cases.

Joints are a source of epistemic uncertainties, which mainly appear from shape imperfections (non-flatness of surfaces), dynamic clearance, and pressure distribution over the contact area, they are directly related to mesoscale and microscale parameters in the joint design. Although after each assembly the structural condition is the same, i.e., the overall picture of the test-bench is maintained - which could be seen as a homogeneous system, pressure variations in the contact area caused by the tightening torque, or even by the order in which the bolts are tightened, can lead to significant changes in the vibratory behavior of the system [26, 21]. One can also mention the occurrence of wear between the surfaces due to microslip, which is influenced by the surface roughness and, when accumulated, also results in global changes to the joint dynamics [27, 28]. Note that all these microscale effects can directly influence AE measurements as these sensors, sensitive not only to noise but also to changes in material properties, generate an enormous amount of data.

Figure 2 presents a schematic representation of the Orion beam illustrating, from the design perspective of the structure, zones that are more and less sensitive to contact pressure. Areas where the contact pressure is sensitive to shape imperfections, especially at edges far from the bolt hole, may exhibit wear as vibration tests are run and also show lower repeatability between tests. In general, normal contact areas are formed by micro-contacts randomly distributed over the surface, and thus, they are hardly arranged in the same configuration after complete disassembly [29]. This fact is illustrated by the schematic representation (see top right of Figure 2) considering different campaigns, in such a way that, when there is a reduction of the tightening torque, repeatability of tests between experimental campaigns is difficult to guarantee, since the structure is disassembled and reassembled each time.

The scale of analysis is thus important. From this perspective, micro effects have a non-negligible impact on AE patterns. Clusters of features have minimal intersection between campaigns on the ORION-AE benchmark [30]. As mentioned by Bull et al. [31], systems in a population can be deemed heterogeneous if features differ while the systems are close. Therefore, it is argued that a new system is analyzed at each experimental campaign, hence the difficulty of developing SHM techniques for bolted joints, due to their heterogeneous character (although they are homogeneous systems from the geometrical perspective).



Figure 2: Diagram depicting contact conditions of the Orion beam at interface, and a microscale contact area illustration which highlights the contact state at each campaign.

3 Methodology

The proposed methodology consists of three main modules, illustrated in Figure 3, namely *Signal processing*, *Data preparation* and *Tightening level identification*.



Figure 3: Workflow containing three different modules: (1) Signal Processing, (2) Data Preparation and (3) Tightening Level Identification.

3.1 Signal processing module

Given an AE data stream, the objective is to estimate the level of tightening continuously. Classically, an AE data stream is decomposed into blocks before processing. Blocks can be obtained by different ways: hit detection procedure [32], sliding window [13], or by exploiting additional sensors as in the present application. The vibration cycles obtained from the laser vibrometer signal are used to segment the AE data stream into blocks (top-left of Figure 3). More precisely, a zero-crossing detector is applied to detect the changes, from positive to negative, in the evolution of the vibrometer signal. This leads to N raw AE signals, with about 1200 number signals per tightening level (the number of cycles corresponds to the excitation frequency of 120 Hz during 10 seconds). Due to the cyclic nature of the experiment, the signals obtained are multiplied by a Hanning window before the next step. We can indeed observe that AE events in a given period generates side effects that can be retrieved in the next period as illustrated in Figure 5. Thus, the window acts as a weighting operation that diminishes the influence of data at the signal's edges. The benefit is to focus the analysis on the central portions of cycles.

Samples of AE data stream are shown in Figure 6, which illustrates the amount of noise and the complexity associated. Acoustic emission data are particularly noisy for bolts looseness detection because the damages occur at the microscale (see Figure 2), whereas the structure is under vibration with a relatively large amplitude. An illustration of the amount of noise is given in Figure 6, where bold and dotted horizontal lines represent the approximate range of noise and signal, respectively. By considering peak-to-peak values for the noise and signals, the signal-to-noise ratio (SNR) is approximately

$$\text{SNR} \approx 10 \log_{10} \frac{36}{21} \approx 2.3 \text{dB}.$$

The SNR is highly unfavorable, where only peaks can be distinguished while signal onsets cannot be accurately estimated from the raw data. For comparison, typical AE applications utilize a detection threshold of 40 dB or higher, which significantly exceeds the threshold in our case.



Figure 4: Schematic representation of the studies carried out during the damage detection stage, considering different network architectures, batch sizes, loss functions, fine-tuned or frozen layers, learning rate scheduler and optimizer. In this figure, SGD stands for Stochastic Gradient Descent.







(b) With tapering.





Figure 6: AE raw data stream to illustrate the unfavourable signal-to-noise ratio for two levels of tightening: a) for 05cNm; b) for 60 cNm. Continue and dashed horizontal lines represent the approximate range of noise and signal, respectively.

3.2 Data preparation module

Data preparation (bottom-left of Figure 3) follows with a transformation of the AE signals into images to feed the CNN. This module considers the Continuous Wavelet Transform (CWT) using a filter bank, based on the analytic Morse (3, 60) wavelet (localized in frequency) with scales discretized by 12 filters. It is possible to impose frequency limits in the filter bank, for example by using the bandwidth of the sensors, but this work uses directly the raw signals (which makes the procedure simpler). The result of this step is a set of N RGB images with size $224 \times 224 \times 3$. For each campaign of measurements, about ~ 8400 images are generated (1200 per level), leading to a total of about 42000 images per sensor and for all campaigns.

Figure 7 depicts the average of CWT images in each tightening level, for each sensor. There are about 1200 pairs of images per case. In each sub-figure, the left-hand side image of the pair represents the unsigned integer encoding of the CWT images as used in neural networks, whereas the right-hand side image is encoded using float encoding. These images (where dark blue is for low values, green for median values and yellow for high values) show different aspects of the dataset:

- The amount of noise is important and the data are not trivial. Indeed the whole frequency range is concerned, characterizing a significant noise level, and a large variability can be observed with no clear trend across levels.
- The average responses are very different across sensors but the mu80 seems the most relevant:
 - The mu80 sensor (column 1) shows three different frequency bands (below 100kHz, between 100kHz and 200kHz, and above 200kHz). Of particular interest is the area above 200kHz, which differs clearly from the two other sensors. These patterns are well reproducible across tightening levels.
 - The F50a sensor depicts vertical patterns, indicating that large frequency bands are affected. However, these patterns are not as reproducible as those observed with the mu80 sensor. Furthermore, evaluating the evolution of these patterns across different levels proves to be complex.
 - The mu200HF sensor exhibits horizontal patterns similar to those of the mu80 sensor; however, the highest frequency components (above 100 kHz) are less pronounced, as indicated by the absence of yellow coloration above this threshold.
- Encoding in 8 bits significantly alters the content but it is a necessary evil for the good with reduced memory requirements, lower power consumption, enhanced computational efficiencies, and better quantization and hardware compatibility, particularly with GPUs. Floating-point representations has a very different interpretation as shown in CWT representation.

3.3 Tightening level identification module

The identification of the tightening torque level relies on a training phase of a CNN, using as inputs the CWT images, arranged according to each tightening level. The seven levels correspond to the classes that the CNN has to predict from images. Pretrained neural networks and transfer learning was shown to be efficient in many SHM applications [33, 34, 35, 36, 37, 38]. Figure 4 presents a schematic representation of the main neural network learning configurations discussed in this work, such as batch size, fine-tuned or frozen layers, optimizer, CNN architecture, learning rate scheduler and loss functions. Specifically, these last three items are discussed in detail below:



Figure 7: Average CWT images over about 1200 cycles for each tightening levels. Dark blue is for low values, green for median values and yellow for high values. Lines: tightening levels, columns: Sensors (mu80, F50A, mu200HF). For each couple tightening-sensor, there are two images: One encoded in unsigned integer (first image) as used in networks, one in float (second image). The unit of y-axis is MHz and in log-scale, and the x-axis is normalized time (224 pixels).

3.3.1 Network architecture

An indication of the validation accuracy and prediction time of the 20 most widely used deep neural networks on ImageNet dataset, made of more than 14M natural images representing 1000 classes, is available on Mathworks¹. Although this comparison does not prejudge about their performance on a SHM task, in particular using scalograms that are not natural images as those considered in ImageNet, it does provides a good indication on the difference between the various networks in terms of complexity (number of parameters), accuracy and inference time. Among these networks, the literature on SHM shows that Resnet18 or VGG16 are often used. In many transfer learning applications, all layers in these networks except the top ones were frozen and additional task-specific layers (generally fully connected and dropout ones) with trainable parameters were incorporated at the head of the network. For example, the method proposed by Fu et al. [13] using ORION-AE dataset relies on this strategy. Generally, for a given application, the trade-off behind using frozen layers is to decrease the computation time. However, it also implicitly assumes that the features extracted from ImageNet dataset share common characteristics with scalograms - although the last type of images are not natural ones. An experimental study [39] showed that, in the context of SHM using very small datasets, it can be better to fine-tune all layers instead of freezing them, which is confirmed in our experiments. As presented by Wang et al. [40], while freezing layers can reduce training cost, prematurely freezing under-trained layers in a static manner will impair the final accuracy. Generalization of frozen vs non frozen networks on unseen SHM AE-data are the core of our experiments using the following CNN architectures:

- **GOOGLENET** Proposed in 2015 [41], GoogLeNet architecture is made of "inception modules" which allow sparsely connected blocks, where some layers are arranged in parallel instead of series, to limit overfitting and improve generalization. Those layers select which filter size is pertinent to learn the relevant information. The top-1 accuracy (proportion of images for which the model's single most confident prediction is correct) was 69.778% on ImageNet1K dataset. In the following tests, the two auxiliary classifiers (represented as two additional network's heads equipped with a loss function) are not used since results were almost the same on CIFAR-10 dataset (10 classes) with or without them. The pytorch implementation of GoogLeNet was also made of 1x1 and 1x3 convolution filters in the inception modules. The model considered is composed of 5.6M training parameters.
- **RESNET** Proposed in 2016 [42], the ResNet architecture is made of skip connections which merges the output layers with input ones by addition. The skip connections act as shortcuts between blocks of layers. Bottleneck layers are used to significantly reduce the number of features or channels passing through it, making the network more compact and computationally efficient. In the residual units within ResNet architectures, information can flow more effectively in both the forward and backward pathways. This enhancement mitigates the vanishing gradient issue, facilitating the efficient propagation of gradients across multiple layers, thereby reducing the training process. The main motivation of the ResNet architecture was to increase the number of layers, therefore going deeper for tackling complex problems. For our tasks, we limit the tests to ResNet18, one of the smallest ResNet network used on CIFAR and ImageNet datasets. ResNet18 has yet 11M parameters and led to a top-1 accuracy of 69.758% on ImageNet1K.
- **MOBILENETV2** Proposed in 2018 [43], the MobileNetV2 architecture was motivated by finding an optimal balance between accuracy and performance during tuning of deep neural networks for mobile applications. For that, they used lightweight Depthwise Separable Convolutions blocks, which replace a full convolutional operator with a simplified version that splits convolution into two separate layers. Compared to ResNet, the MobileNet architecture is based on an inverted residual structure, where the input and output of the residual blocks are thin bottleneck layers, unlike traditional residual models that use expanded representations in the input, requiring less computational resources. MobileNetV2 is made of 3.5M parameters and led to a top-1 accuracy of 72.154% on ImageNet1K.
- **EFFICIENTNETB5** Proposed in 2019 [44], the main motivation of EfficientNet architecture was to create a set of networks that carefully balance network depth, width, and resolution for enhanced performance. The most common way of increasing the capacity of a network was to scale up ConvNets by their depth or width, more or less randomly. Another one was to scale up models by image resolution. Based on these observations, they proposed a new scaling method that uniformly scales all dimensions of depth/width/resolution using a "compound" coefficient. They demonstrated the effectiveness of this method on scaling up MobileNets and ResNets. Additionally, their research on the neural architecture design resulted in the creation of a novel baseline network, which was further scaled up to produce a family of models known as EfficientNets, ranging from B0 to B7. These models achieved improved accuracy and efficiency than previous ConvNets, with smaller number of parameters. EfficientNets rely on inverted bottleneck convolution blocks developed in MobileNets as well as squeeze-and-excitation blocks [45] for increased efficiency in computing relevant feature maps. In

¹https://fr.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html

this work, we considered the EfficientNetB5 architecture, which leds to a remarkable increase in performance on ImageNet1K with 83.444% using 30M parameters.

3.3.2 Learning rate scheduler

When all parameters are set to be trainable, the training stage becomes complex for large networks and the choice of the initial learning rate as well as the scheduler becomes crucial. The task of the scheduler is to *implement a policy that dynamically adjusts the learning rate over epochs or iterations*. Both learning rate (LR) and choice of policy stand out as important parameters for transfer learning due to their impact on training dynamics.

The 1cycle scheduler[46, 47] outperformed many standard scheduling approaches and is used in our methodology. This learning rate scheduler, a member of the cyclical learning rates (CLR) family, operates in cycles consisting of two "steps": the first step involves a gradual increase in the learning rate from a minimum value to a maximum value, followed by a second step in which the learning rate is reduced. The stepsize is chosen by the end-user and can represent the number of iterations, or epochs. Several functions are used to gradually increase and decrease the LR, as the Pytorch implementation with cosine evolution. The 1cycle LR is thus made of a single cycle, and there is a variant that includes a third step, gradually decreasing the LR. To determine the minimum and maximum LR values, we trained the models through some iterations with different LR values. The maximum LR is identified as the value just before overfitting occurs during the validation process. When implementing the 1cycle LR policy, the minimum LR is typically set at a factor of 10-25 less than the established maximum bound. As underlined by Smith [47], if the speed of the learning rate evolution is too large, then the training becomes unstable, requiring to vary the number of epochs.

This scheduler has emerged as a promising approach due to a phenomenon known as "super-convergence", where training time takes orders of magnitude less than usual. According to Smith [47], a key element contributing to this phenomenon is the utilization of the 1cycle policy with a relatively high maximum learning rate. In our tests, a warm-up step is performed, where the learning rate is set to 1/25 of the maximum allowed value. Then, it is gradually increased until its maximum. This first step takes 30% of the number of iterations. In the second step, the learning is decreased until the maximum number of iterations is reached. In this scheduler, the learning rate is adapted at each mini-batch.

3.3.3 Loss functions

ORION-AE dataset is made of K = 7 classes, therefore the cross-entropy loss (CRE) is a natural choice:

$$CRE(T, P) = -\sum_{i=1}^{N} \sum_{k=1}^{K} T_i(k) \log(P_i(k)),$$
(1)

where $T_i(k)$ is the target (0 or 1 for categorical target) for *i*-th input and *k*-th class, and $P_i(k)$ the prediction made by the network for this input for each of the K classes.

In the present work, additional loss functions are incorporated to account for the specific characteristics of the application. Indeed, the seven classes in ORION-AE dataset can be considered as ordered. Therefore, considering a loss that promotes ordinality in classification can be relevant for SHM purposes. In this type of loss, predictions that are close to the true class, but incorrect, are penalized less severely than predictions that are far from the truth, thereby *encouraging misclassification errors to be in adjacent classes*. The different loss functions considered in this study are illustrated on a simple example in Figure 4.

One natural loss, referred to as CDW1 (Class-Distance Weighted #1) assigns a weight proportional to the difference between predicted and true labels:

$$CDW1(T,P) = \sum_{i,k} \left(\frac{w_{ik}}{K-1} + 1\right) CRE(T_i(k), P_i(k)),$$
(2)

with

$$w_{ik} = \left| \arg\max_{k} T_i(k) - \arg\max_{k} P_i(k) \right|.$$
(3)

This ordinal categorical cross-entropy loss is implemented in Keras² as an adaptation of the standard CRE for ordinal classification problems. An alternative loss tested in experiments is:

$$CDW2(T, P) = \sum_{i,k} \exp(w_{ik}) CRE(T_i(k), P_i(k)),$$
(4)

²https://www.tensorflow.org/guide/keras?hl=fr

which amplifies the difference between the prediction and the true class.

In Hou et al.[48], a loss was proposed for ordinal classification problems using cumulative density functions (cdf) of targets and of predictions:

$$CDF(T, P) = \sum_{i,k} \left(cdf(T_i(k)) - cdf(P_i(k)) \right)^2.$$
(5)

Finally, we propose to consider two losses called POM1a and POM1b ("Plus Or Minus 1"), which considers the probability on adjacent classes:

$$POM1a(T, P) = -\sum_{i,k} T_i(k) \cdot \log \sum_{\substack{l \in \{-1,0,1\}\\s,t,0 < k-l < = K}} P_i(k-l).$$
(6)

In this loss, if the network predicts a high probability to an adjacent class of the true one, then the loss remains low, encouraging mis-classification errors to be in adjacent classes. Similarly, the sum can be taken on the logarithm:

$$POM1b(T, P) = -\sum_{i,k} T_i(k) \cdot \sum_{\substack{l \in \{-1,0,1\}\\s.t.0 < k-l < =K}} \log P_i(k-l).$$
(7)

4 NO-SHM Case: Single Campaign Classification

This section presents the results of CNN architectures for the classification of the tightening levels in a given campaign. In the following tests, we studied the effect of denoising the data before creating images, as well as the fusion of sensors.

These tests are qualified as "NO-SHM", since training and testing phases share data from the same campaign, even if different data are used for each phase. The fact that training and test data originate from the same campaign *makes the task simpler than isolating data from a different campaign for testing*. When mixing images from all campaigns and applying a CNN (as explained in Introduction), the problem boils down to a classification task with a pre-normalization of the data, which is different from a "SHM" task that should aim at generalizing to unseen test conditions [16, 31]. The difficulty of a CNN to generalize from a set of training campaigns to a distinct testing one will be studied in the next section. This distinction is important to position our approach regarding the work of Fu et al. [13], which was a NO-SHM case. In this section, as in Fu et al., accuracy on testing data is estimated based on the number of correct predictions divided by the total number of data, and denoted as ACC - the standard accuracy.

4.1 Effect of denoising

Wavelets denoising (WD) is widely used in applications related to AE [49, 18, 32]. The basic idea of WD is to use a filter bank based on two quadrature mirror filters (low and high pass) derived from a mother wavelet (mw). It works in two stages. The first stage, so-called *decomposition*, is performed from level 0 (raw signal) until level l by applying both filters, with decimation by a factor 2 between each level. A thresholding method (th) is applied at each level to set to zero the wavelets coefficients corresponding to noise [50]. The second stage, so-called *reconstruction*, allows to generate the denoised signal, from level l back to its original size, from the coefficients obtained at each level. Wavelets are used to denoise raw AE data stream before creating images to study whether it improves or not the accuracy of the classification. The three sensors available in ORION-AE dataset are considered (mu80, F50A and mu200HF) individually. Blocks of 1s (5M sampling points) were used. Similar parameters to Kharrat et al. [51] are used for denoising: mw = db45, th set to the universal threshold method with level dependent rescaling except the level of decomposition l.

The level of decomposition l was set to 0, 1, 2..., 9 for campaign #B (identified as the most difficult one [30]) and to 0, 1, 4, 9 for the other campaigns. For each level, the pretrained network GoogLeNet is applied using images generated during the "Signal processing module" and "Data preparation module". All parameters are set as trainable, i.e. without freezing any layer. For training, a Pytorch implementation is put forward considering ADAMW optimizer[52], momentum equal to 0.9, weight decay of 0.0005, constant learning rate equal to 0.001 (no scheduler policy), minibatch size of 16, and 15 epochs. The dataset is divided into training (80%), validation (10%) and testing (10%), as in Fu et al. [13]. This configuration is summarized in Table 1.

Results from campaigns #B to #F are presented in Tables 2 to 6, respectively. These tables show that there is *no improvement in performance when using a pre-filtering* of data, for all campaigns. Conversely, a decrease of performance can be observed. The results indicate that denoising removes relevant information from signals even at low level of decomposition. The tables additionally demonstrate high performances without pre-filtering, with an average

Approach	Net.	Frozen	Loss	Optim.	LR	Mom.	Sched.	MB	Epochs
		layers							
Fu et al. [13]	Resnet18	Yes	CRE	SGDM	N.C.	N.C.	No	N.C.	150
This work	GoogleNet	No	CRE	ADAMW	0.001	0.9	No	64	15

Table 1: Configuration for studying the noise level and sensor fusion. "N.C." stands for "not communicated".

accuracy ACC $\sim 99\%$. This performance is attributed to the pretrained CNN's capacity of managing noisy input through regularization, dropout and batch normalization. The input signals are highly noisy, but the decomposition of the signals into frequency bands in the scalograms facilitates the training process.

	mu80	F50A	mu200HF	All sensors
0	99.61 %	100 %	100 %	99.81 %
1	99.57 %	100 %	100 %	99.94 %
2	99.38 %	99.94 %	99.94 %	99.86 %
3	99.44 %	99.94 %	100 %	99.86 %
4	97.96 %	100 %	100 %	99.50 %
5	99.07 %	100 %	99.94 %	99.65 %
6	98.70 %	99.88 %	100 %	99.55 %
7	98.88 %	99.88 %	99.88 %	99.23 %
8	95.11 %	99.94 %	99.81 %	97.50 %
9	66.05 %	99.94 %	99.76 %	88.71 %

Table 2: #B: Performance according to the level of decomposition in denoising and to the sensors.

	mu80	F50A	mu200HF	All sensors
0	99.21 %	99.88 %	100 %	99.51 %
1	98.07 %	99.86 %	100 %	99.31 %
4	97.72 %	100 %	100 %	99.33 %
9	67.76 %	99.94 %	99.65 %	88.87 %

Table 3: #C: Performance according to the level of decomposition in denoising and to the sensors.

Table 9 presents the confusion matrix obtained by the GoogLeNet on campaign #B, considering the sensor mu80, and without denoising. Note that the matrix is diagonal, showing the high performance of the classification task in the "NO-SHM" case, considering the sensor that provided slightly lower performances than the other sensors (on average, less than 1% difference).

4.2 Performance using sensor fusion

Tables 2 to 6 show the classification accuracy using sensor fusion. In this work, the fusion process consists of considering all images from all sensors during training, validation and testing.

Note that the fusion slightly decreases the accuracy when considering the single best sensor for each line. This result is in accordance with the study made by Ai et al. [53], in the context of AE-based monitoring of reinforced concrete block, who observe that single sensor is more efficient. Other fusion strategies could be considered, e.g. combining the outputs of different networks [54], but the marginal improvement (compared to 99% with the mu80 sensor) does not justify the increase in computation time.

These results on the NO-SHM case *extends the study* of Fu et al. [13] to all campaigns, to every sensor³ as well as to sensor fusion.

5 SHM Case: Classification on Unseen Campaign

In the previous section, data from different levels of tightening torque were mixed, regardless of the campaign. This section extends the analysis to the so-called SHM case, in which one campaign of measurements is isolated for testing, while the CNN model is trained on the remaining four campaigns. Therefore, each campaign is interpreted as being

³Fu et al. [13] do not specify which sensor was used.

	mu80	F50A	mu200HF	All sensors
0	98.5 %	99.25 %	99.83 %	99.46 %
1	97.86 %	98.72 %	99.11 %	98.39 %
4	97.68 %	99 %	99 %	97.96 %
9	74.18 %	98.72 %	97.25 %	90.69 %

Table 4: #D: Performance according to the level of decomposition in denoising and to the sensors.

-				
	mu80	F50A	mu200HF	All sensors
0	99.51 %	100 %	100 %	99.56 %
1	98.84 %	100 %	100 %	99.45 %
4	98.47 %	100 %	100 %	99.55 %
9	72.39 %	100 %	99.57 %	91.24 %

Table 5: #E: Performance according to the level of decomposition in denoising and to the sensors.

generated by a distinct structure (heterogeneous characteristics discussed in subsection "Challenges - Heterogeneous system"). This section begins by establishing a baseline, which will be used to compare the performance of different configurations when training CNNs for the SHM case, as well as to assess the influence of including *a priori* information from the structure (campaign) not initially used for training.

Moreover, accuracy on testing data is calculated in two ways: (1) the standard accuracy ACC introduced previously; and (2) the accuracy on ± 1 class, denoted as ACC $_{\pm 1}$ (see Appendix B). Note that in the ORION-AE dataset, there may be overlapping classes [30] due to uncertainties on the actual value of the tightening levels, as torque measurements were conducted using an analog torque wrench [22]. Therefore, if a network predicts a tightening level of \hat{c} while the ground truth is c, it is considered correct if $|\hat{c} - c| \leq 1$. The dataset from mu80 sensor is considered in the following tests.

5.1 Baseline

Three network architectures are considered: GoogLeNet, Resnet18 and EfficientNetB5, trained with the same parameters presented in Table 1. To establish the baseline, as the campaign #B is the one that presents considerable challenges for the classification [30], it is used for testing. Therefore, the following configurations are explored: SHM vs. NO-SHM (training, validation and test datasets formed from only campaign #B) situations, and freezing vs. non freezing layers, leading to 4 (configurations) \times 3 (architectures) = 12 cases. Results are summarized in Table 8 with the standard accuracy (ACC) and the accuracy at plus or minus one class (ACC_{±1}). In this table, the **NO**-SHM case with Resnet18 corresponds to the same configuration used by Fu et al. [13]. In that paper, the authors froze the layers, but we were not able to reproduce the same results (99%), except when all the parameters were considered as trainable (mentioned as "NO-Freeze" in the table). In this case, the networks converged after 15 epochs, i.e. in 10 times less epochs than Fu et al.[13].

Table 8 also indicates that layer freezing is not relevant compared to training all parameters, neither for the SHM case nor for NO-SHM, in these datasets. For example, if the simpler task of classifying images from the same campaign is considered (NO-SHM case), freezing layers leads to an accuracy $ACC_{\pm 1} \sim 68.8\%$, whereas 100% is obtained when all parameters become trainable. Therefore, if the task is more complex, as in the SHM case, the accuracy drops dramatically by freezing layers - accuracy $ACC_{\pm 1} \sim 42 - 45\%$ using Resnet18 or EfficientNetB5 architectures against $ACC_{\pm 1} \sim 65\%$ when both CNNs have trainable parameters.

	mu80	F50A	mu200HF	All sensors
0	99.56 %	100 %	100 %	99.95 %
1	99.51 %	100 %	100 %	99.90 %
4	99.33 %	100 %	100 %	99.74 %
9	82.65 %	99.57 %	99.33 %	93.46 %

Table 6: #F: Performance according to the level of decomposition in denoising and to the sensors.

Torque Levels	1	2	3	4	5	6	7
1	247	0	1	0	0	2	0
2	1	216	0	0	0	0	0
3	0	0	236	0	0	0	0
4	0	1	0	199	0	1	0
5	0	0	0	0	218	0	0
6	0	0	1	0	0	233	0
7	0	0	0	0	0	0	256

Table 7: Confusion matrix obtained with the configuration presented in Table 1 using GoogLeNet on campaign #B. Accuracy of 98.5% was reached after 9 epochs, and 99.5% after 12 epochs.

	Configurations	GoogleNet		Resnet18		EfficientNetB5	
	Configurations	ACC	$ACC_{\pm 1}$	ACC	$ACC_{\pm 1}$	ACC	$ACC_{\pm 1}$
Baseline 1	NO-SHM + NO-Freeze	99.8	99.8	99.9	100	99.9	99.9
Baseline 2 [13]	NO-SHM + Freeze	49.0	62.2	57.1	68.8	55.0	66.5
Baseline 3	SHM + NO-Freeze	26.4	52.2	31.4	65.4	35.4	65.9
Baseline 4	SHM + Freeze	17.4	43.7	18.6	42.9	17.0	44.9

Table 8: Accuracy for four baseline approaches; baselines 1 & 2 are variants of the Fu et al. [13] work.

5.2 Influence of the amount of prior using the baseline

The influence of the amount of prior data from the tested structure is studied by gradually adding knowledge on tightening levels, from none (fully unsupervised) to the case where all classes, except class 5 cNm, are used in training (almost NO-SHM). The tightening levels are added from 60 cNm to 10 cNm *in this order, to mimic the situation where regular inspections on the structure allow to label the previously collected data*. The pretrained Resnet18 is used *without freezing any layer*, i.e., all the parameters of the model (11M) are considered trainable, as this is a necessary condition for successful generalization in this dataset. Training procedure is performed using the SGDM (Stochastic Gradient Descent with Momentum) optimizer, and a piecewise learning rate schedule starting with an initial learning rate of 0.01 and a drop factor of 0.1. The model leading to the best validation loss was retained after 15 epochs. A 5-fold cross validation is performed in each test, leading to a total $5 \times 7 \times 5 = 175$ runs.

Figure 8 depicts the results. On the left-hand side (fully unsupervised), no images from the unseen campaign are used during training. From left to right, more prior knowledge is added progressively with respect to the state of the structure. Note the natural trend of the model to better classify the data, as expected. Moreover, in most campaigns, an accuracy $ACC_{\pm 1}$ between 98 and 100% is obtained in the most supervised case (almost NO-SHM case with only one level, 5 cNm, remaining). A difference around 15% can be observed between SHM and NO-SHM situations for each campaign.

Campaign #B is the most difficult one to guarantee satisfactory results, with an average classification accuracy ACC_{±1} ~ 65%. In the NO-SHM case, the results of this campaign depicts the highest variability (see confidence intervals in Figure 8), demonstrating the sensitivity of the model to the training data during cross-validation. The average of ACC_{±1} values on all campaigns, except for #B, is about 78% ± 2 with maximum *a priori*.

6 SHM Case: Super-Convergence

In this section, the tests aim at:

- Identifying configurations of the networks able to quickly train with the best performance in generalization, without freezing any layers;
- Demonstrating the phenomenon of super-convergence with the 1cycle scheduler;
- Demonstrating the relevance of the ordinal losses;
- Comparing these configurations with the baseline.

Due to the number of tests required, only campaign #B is considered in the tests. The other campaigns will be studied subsequently, testing the best configurations found for #B. Four types of CNN architectures are considered, as presented in subsection "Network architecture". These architectures comprise one "small", two "medium" and

On the Condition Monitoring of Bolted Joints through Acoustic Emission and Deep Transfer Learning: Generalization, Ordinal Loss and Super-Convergence



Figure 8: ResNet18, sensor mu80: Performance on generalization on all campaigns with gradual amount of prior.

one "large" network, respectively with MobileNetV2 (3M parameters), GoogLeNet (6M), Resnet18 (11M) and EfficientNetB5 (33M).

6.1 Identifying a configuration without freezing any layer

The training uses ADAMW optimizer, considering a weight decay of 5×10^{-4} coupled with the 1cycle scheduler [46, 47], and a maximum learning rate equal to 0.01. The batch size is set to 8. The five loss functions described in Section "Loss functions" are tested. To identify configurations that ensure rapid convergence, the number of epochs is considered from 2 to 5. The tests were repeated five times leading to a total more than $4 \times 5 \times 5 = 100$ tests.

Figures 9 and 10 present the results through box-plots (median, minimum and maximum values, as well as outliers) of accuracy values $ACC_{\pm 1}$ for the four models. Note the important result provided by POM1b, which outperforms all other losses, for all models. Averaging the 140 tests over all networks used to compute the box-plot of this specific loss (right-hand boxes in Figure 10b, a value of $ACC_{\pm 1} \sim 77.6\%$ is obtained - this is 7% higher than the second best, which is CDF. Throughout tests, POM1b also depicts the smallest variability, demonstrating the robustness compared to the other losses. More importantly, *all ordinal losses provided better performances than the* CRE, *the loss commonly used in most publications on AE classification through CNNs*. The CRE loss also depicts the largest variability, inferring a lack of robustness compared to the other losses. This result highlights the importance of considering ordinal losses in SHM applications.

Figure 10 summarizes the performance of networks, independently of the loss (10a), and the performance of the losses, independently of the network (Fig 10b). Based on these figures, note that the architecture EfficientNetB5 has the best potential, with a median value of 72.7% when considering the tests with all losses (146 cases). The second best network is Resnet18, with 71.26% - this network also has the smallest variability. All models were able to exceed 80 - 82% for some initialization (according to random draws of the batches). MobileNetV2 has the largest variability. The 30 best models are given in Table 10 (appendix), confirming the performance of EfficientNetB5 and Resnet18 with the loss POM1b.

It is important to stress that the results presented in this section are more promising than those presented by the baseline configurations (accuracy around 65%), allowing us to assess that better generalization can be achieved (without prior information), without the need for many training epochs - which is desirable in a SHM context. Three epochs seem enough to get good generalization.



Figure 9: Performance of networks according to the loss. • corresponds to outliers, and a red line is the median.



Figure 10: a) Robustness of the networks according to the loss; b) Robustness of the loss according to the networks.

On the Condition Monitoring of Bolted Joints through Acoustic Emission and Deep Transfer Learning: Generalization, Ordinal Loss and Super-Convergence



Figure 11: Campaign #B: Demonstration of super-convergence. Evolution of the learning rate and of the accuracy in testing for the two best models, using only 3 epochs.

6.2 Robustness and super-convergence

Figure 11 depicts the evolution of the accuracy on testing data (in generalization without prior on campaign #B) for the four best models, and using only 3 epochs, as deduced in previous tests. The evolution of the learning rate is also presented, computed by the 1cycle policy. Epochs are represented by dashed lines. The variability on accuracy is due to the mini-batches (size 8) which are drawn randomly at the beginning of each trial, and this variability decreases along iterations as the network is improving. Note that all networks are quite close in terms of accuracy, with a slight advantage to EfficientNetB5 in the first iterations, followed by Resnet18, MobileNetV2 and GoogLeNet. Based on these figures, a *training dynamics* can be stated, with a schematic illustration proposed in Figure 12, by the following

steps: 1. The initial accuracy around 40% obtained from the pretrained networks with ImageNet;

- 2. The warm-up stage during the very first iterations (0-250), where the accuracy increases a little in most cases;
- 3. A phase of network *specialization*, where the accuracy increases quickly, with +40% in some cases. This phase is observed for all models;
- 4. Gradual increase of the accuracy, which corresponds to a *fine tuning* of the network;
- 5. A *plateau* that starts around 4500-5500 iterations (end of the second step of the schedule), showing the convergence.

These figures show that the best configurations found in this study improve the accuracy from 65% (baseline, Figure 8) to 79% (EfficientNetB5 in Figure 11a). Figure 11a depicts a characteristic behavior in which the accuracy of EfficientNetB5 increases slowly in the *fine tuning*, and remains lower than Resnet18 before an improvement in the late iterations, leading to a higher accuracy. This behavior is observed on the five datasets as shown in next subsection.

Note that Smith et al. [46, 47] recommended to reduce all forms of regularization to preserve a balance between underfitting and overfitting. In particular, the authors advised to use large batch sizes. However, in our study, the minibatch size equal to 8 worked well on all datasets and can be fitted on small GPU.

In terms of computational resources, EfficientNetB5 has $3 \times$ more parameters than Resnet18 requiring $2.5 \times$ much space on a hard disk to store the model. The training stage is about $2 - 3 \times$ longer with EfficientNetB5. On a laptop equipped with a RTX4000 GPU, training EfficientNetB5 took 39 s on average for about 8000 images (one campaign of 70 s), whereas Resnet18 required 19 s. Note that these conclusions can vary with the computer used for the test. Tobiasz et al. [55] compared Resnet(s) and EfficientNet(s) in terms of inference time, and showed that they are much quicker than older networks like VGG16 or InceptionNet(s), but largely slower than MobileNet(s). According to our tests, MobileNetV2 is indeed fast during training but is clearly outperformed by the three others networks for the generalization task (Figures 10a). Therefore, a good compromise could be Resnet18 with the loss POM1b or EfficientNetB5 if the computer has enough resources because its performances were generally better.

6.3 Test on all campaigns

Previous tests were made on campaign #B for illustrative purposes, once it represents the most difficult campaign. Based on the previous tests, two efficient configurations stand out. In this section, these same configurations are applied for evaluating their performance on SHM on the other campaigns, namely #C, #D, #E and #F. Results are showed in Figure 13 and performances (using four metrics, see Appendix B) are summarized in Table 9.



Figure 12: Schematic representation of the training dynamics established during the super-convergence study. Training steps illustrated by this figure are the following: (I) Initialization; (II) Warm-up; (III) Specialization; (IV) Fine-tuning; (V) Convergence.

Training			Testin	Performance in testing				
Campaigns	# train.	# valid.	Campaigns	# test.	$ACC_{\pm 1}$	$R_{\pm 1}$	$P_{\pm 1}$	$F1_{\pm 1}$
#C, #D, #E, #F	25244	6311	#B	8064	78.8	71.4	64.1	66.1
#B, #D, #E, #F	22348	5587	#C	7006	86.4	73.7	76.7	75.1
#B, #C, #E, #F	25148	6286	#D	8185	86.3	77.9	72.8	75.3
#B, #C, #D, #F	25152	6287	#E	8180	78.8	71.1	65.8	68.3
#B, #C, #D, #E	25148	6287	#F	8184	86.1	77.1	70.8	73.8

Table 9: Metrics of performance averaged over 5 runs in % for all campaigns and in generalization, with details on the number of images in training, validation and testing datasets.

In these figures, the phenomenon of super-convergence is clearly visible, achieving rapid convergence within just three epochs using the 1cycle policy. Note the characteristic behavior of EfficientNetB5 during the fine-tuning stage. Initially, its generalization accuracy is lower than that of ResNet18, but it subsequently improves and surpasses ResNet18 in the later iterations. The final performance metrics, as shown in Table 9, were achieved using EfficientNetB5 with POM1b loss across all datasets.

On the Condition Monitoring of Bolted Joints through Acoustic Emission and Deep Transfer Learning: Generalization, Ordinal Loss and Super-Convergence



Figure 13: Performance of the best configuration on the other datasets (#C, #D, #E and #F) by using the same configuration of the two best networks as done for #B.

7 Conclusion

Population-based SHM has emerged as an interesting framework to design and validate SHM algorithms. ORION-AE dataset appears to be well-suited to illustrate this framework because the repeated measurements through different campaigns of a bolted joint have modified the contact conditions at the lap-joint, and its microscale topology has changed between experiments. These microscale changes have an important impact on the behavior of the joint and therefore the different measurement campaigns can be considered as a set of heterogeneous data. Moreover, this dataset can be used in two ways: either campaign-wise or for SHM purposes. In the first case, the normalization of the data, i.e. same campaign being used for training, validation and testing, makes the data homogeneous and the classification problem simple, resulting in an accuracy of over 99%, using relatively simple models. However, in the latter, the task becomes more difficult because of the modified operational conditions, which make the data heterogeneous and challenge the generalization capabilities of the models.

The main conclusions of this study are summarized as follows:

- Previous approaches based on deep neural networks with frozen layers developed for homogeneous population failed to generalize, making them inefficient for SHM purposes within bolted joints using the AE technique. Scalograms are not natural images and therefore it is required to update all parameters of the network.
- Denoising AE data before computing the CWT is not necessary. Denoising is embedded in deep neural networks through convolutional layers that process scalograms.
- Using the mu80 sensor was enough to get high performance, without the need of sensor fusion.
- Different loss functions were compared, in particular ordinal losses and the standard cross-entropy (CRE). We identified one loss, called POM1b, that led to the best results overall, independently on the network. Conversely, the CRE depicted the worst results in terms of accuracy and standard deviation.
- To the best of our knowledge, the phenomenon of super-convergence is presented for the first time on SHM data. For that, we explored a scheduling strategy of the learning rate called 1cycle scheduler that allows to get high accuracy in a few number of iterations.

- We conducted many tests, with different architectures of CNNs and our study showed that the two best configurations in terms of accuracy and convergence speed were: EfficientNetB5 and Resnet18, both with POM1b-loss, with accuracy of 78.8%, 86.4%, 86.3%, 78.8%, 86.1% for the five campaigns of measurements #B, #C, #D, #E and #F, respectively.
- Based on the results of this study, a promising direction for enhancing generalization in SHM of bolted joints involves exploring the performance of sensors with higher frequency ranges for condition monitoring. This approach could potentially improve the accuracy and reliability of bolt detection and assessment.

The true labels are not reliable enough because of the uncertainty about the position of the torque screwdriver used to set the tightening levels. Considering a mistake of one class as correct is a possible approximation which can represent a problem in certain applications. As future work, we will focus on how to encode the uncertainty on labels and how to integrate it into deep neural networks and study whether and how it helps at improving the accuracy.

Acknowledgment

This work was partly carried out in the framework of the EIPHI Graduate school (contract ANR-17-EURE-0002) and the project RESEM- COALESCENCE funded by the Institut de Recherche Technologique Matériaux Métallurgie Procédés (IRT M2P) and Agence Nationale de la Recherche (ANR).

A Appendix A: Identifying a configuration with fast results

Table 10 presents the 30 best network configurations found in terms of accuracy as a function of loss, number of epochs, and number of tests carried out for the SHM case. The testing dataset was #B, whereas the training and validation ones are $\{\#C, \#D, \#E, \#F\}$.

Loss	Epoch	Acc	σ
POM1b	3	79.268	2.1144
POM1b	4	79.264	2.1685
POM1b	2	79.045	1.5405
POM1b	5	78.683	1.7855
POM1b	5	77.741	1.6415
POM1b	3	77.545	2.019
POM1b	5	77.235	1.6483
POM1b	3	77.169	2.3464
POM1b	3	76.886	3.2553
POM1b	2	76.704	2.2842
POM1b	4	76.463	2.872
POM1b	5	76.427	1.8958
POM1b	4	76.188	2.2991
POM1b	2	76.052	3.4309
POM1b	4	76.034	3.6481
POM1b	2	75.599	3.5895
POM1a	5	75.164	2.7232
POM1a	3	74.814	2.9309
POM1a	4	74.554	3.1466
POM1a	4	73.78	1.563
CDF	2	73.229	3.2286
CRE	4	73.15	3.2248
POM1a	2	73.125	2.5063
POM1a	2	72.629	4.1272
POM1a	3	72.414	2.9934
CDW1b	4	72.264	1.7702
CDF	3	71.779	5.5462
POM1a	3	71.592	1.6987
CDF	4	71.25	1.4033
POM1a	5	70.96	1.4625
	Loss POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1b POM1a POM1a POM1a POM1a POM1a POM1a POM1a POM1a POM1a POM1a POM1a POM1a	Loss Epoch POM1b 3 POM1b 4 POM1b 2 POM1b 5 POM1b 5 POM1b 3 POM1b 3 POM1b 3 POM1b 3 POM1b 3 POM1b 4 POM1b 4 POM1b 4 POM1b 4 POM1b 2 POM1b 4 POM1b 2 POM1b 4 POM1b 2 POM1b 4 POM1b 2 POM1b 4 POM1a 3 POM1a 4 POM1a 2 POM1a 2 POM1a 3 POM1a 3 POM1a 3 POM1a 3 POM1a 3 POM1a 3 POM1a </td <td>LossEpochAccPOM1b379.268POM1b479.264POM1b279.045POM1b578.683POM1b577.741POM1b377.545POM1b577.235POM1b376.886POM1b376.886POM1b276.704POM1b476.463POM1b576.427POM1b476.052POM1b476.034POM1b275.599POM1a575.164POM1a374.814POM1a473.78CDF273.229CRE473.15POM1a272.629POM1a372.414CDF371.779POM1a371.592CDF471.25POM1a570.96</td>	LossEpochAccPOM1b379.268POM1b479.264POM1b279.045POM1b578.683POM1b577.741POM1b377.545POM1b577.235POM1b376.886POM1b376.886POM1b276.704POM1b476.463POM1b576.427POM1b476.052POM1b476.034POM1b275.599POM1a575.164POM1a374.814POM1a473.78CDF273.229CRE473.15POM1a272.629POM1a372.414CDF371.779POM1a371.592CDF471.25POM1a570.96

Table 10: 30 best models on average for generalization on campaign #B, sorted by accuracy at plus or minus one class, and the standard deviation over five tests.

B Appendix B: Metrics

Figure 14 depicts a confusion matrix, say m, for K = 7 classes based on which the following metrics can be obtained. The matrix is tridiagonal which allows to show the specificity of the metrics at plus or minus one class.

The standard accuracy is

$$ACC = \sum_{i=1}^{K} m(i, i),$$

, while the accuracy at ± 1 is

$$ACC_{\pm 1} = \sum_{i=1}^{K} \sum_{j \in \{i-1, i, i+1\}} m(i, j),$$

with obvious cases at i - 1 > 0, i + 1 < K.

The recall and precision at plus or minus one, $P_{\pm 1}$ and $R_{\pm 1}$ respectively, are obtained as follows. Let #correct k the number of points correctly classified in class k, #predicted k the number of points predicted as class k, and N_k the

number of points in class k, then:

$$R_{\pm 1}(k) = \frac{\#correct \ k}{N_k},$$
$$P_{\pm 1}(k) = \frac{\#correct \ k}{\#predicted \ k},$$

with N_k the number of elements in class k in the ground truth (given by the sum of columns for the k-th row) and

$$\#correct \ k = \sum_{(i,j) \in \mathcal{N}(i) \setminus \{(i+1,i+1),(i-1,i-1)\}} m(i,j),$$

where $\mathcal{N}(i)$ is the 3 \times 3 neighborhood of *i* and

#predicted
$$k = \sum_{i=1}^{K} \sum_{j \in \{-1,0,1\}} m(i,k-j).$$

After taking the average of recall and precision values over classes we can deduce the F1-score:

$$F_1 = \frac{2\overline{R}_{\pm 1} \cdot \overline{P}_{\pm 1}}{\overline{R}_{\pm 1} + \overline{R}_{\pm 1}}.$$

For the example depicted on the figure, we have ACC = 0.36, $ACC_{\pm 1} = 1.0$, $P_{\pm 1}(k) = [0.80.875, 0.778, 0.778, 0.875, 0.8]$, $\overline{P}_{\pm 1} = 0.81$ and $R_{pm_1}(k) = [0.8, 0.875, 0.778, 0.778, 0.875, 0.8]$, $\overline{R}_{\pm 1} = 0.81$, yielding $F_1 = 0.81$.



Figure 14: Schematic representation of a tridiagonal confusion matrix.

References

- [1] Huang J, Liu J, Gong H et al. A comprehensive review of loosening detection methods for threaded fasteners. *Mechanical Systems and Signal Processing* 2022; 168: 108652. DOI:https://doi.org/10.1016/j.ymssp.2021.108652.
- [2] Li XX, Li D, Ren WX et al. Loosening identification of multi-bolt connections based on wavelet transform and ResNet-50 convolutional neural network. *Sensors* 2022; 22(18). DOI:https://doi.org/10.3390/s22186825.
- [3] Du C, Liu J, Gong H et al. Percussion-based loosening detection method for multi-bolt structure using convolutional neural network Densenet-CBAM. *Structural Health Monitoring* 2024; 0(0): 14759217231182305. DOI:10.1177/ 14759217231182305.
- [4] Yang Z and Huo L. Bolt preload monitoring based on percussion sound signal and convolutional neural network (cnn). *Nondestructive Testing and Evaluation* 2022; 37(4): 464–481. DOI:10.1080/10589759.2022.2030735.
- [5] Sun Y, Li M, Dong R et al. Vision-based detection of bolt loosening using yolov5. Sensors 2022; 22(14): 5184.
- [6] Shah JK, Braga HBF, Mukherjee A et al. Ultrasonic monitoring of corroding bolted joints. *Engineering Failure Analysis* 2019; 102: 7–19. DOI:https://doi.org/10.1016/j.engfailanal.2019.04.016.
- [7] Peng T, Saxena A, Goebel K et al. Integrated experimental and numerical investigation for fatigue damage diagnosis in composite plates. *Structural Health Monitoring* 2014; 13(5): 537–547. DOI:10.1177/1475921714532992. URL https://doi.org/10.1177/1475921714532992.
- [8] Rafik V, Combes B, Daidié A et al. Experimental and numerical study of the self-loosening of a bolted assembly. In *Advances on Mechanics, Design Engineering and Manufacturing II*. Springer, 2019. pp. 85–94.
- [9] Ferrer C, Salas F, Pascual M et al. Discrete acoustic emission waves during stick–slip friction between steel samples. *Tribology International* 2010; 43(1): 1–6. DOI:https://doi.org/10.1016/j.triboint.2009.02.009.
- [10] Geng Z, Puhan D and Reddyhoff T. Using acoustic emission to characterize friction and wear in dry sliding steel contacts. *Tribology International* 2019; 134: 394–407. DOI:https://doi.org/10.1016/j.triboint.2019.02.014.
- [11] Feng P, Borghesani P, Smith WA et al. A Review on the Relationships Between Acoustic Emission, Friction and Wear in Mechanical Systems. *Applied Mechanics Reviews* 2019; 72(2): 020801. DOI:10.1115/1.4044799.
- [12] Zhang Z, Xiao Y, Su Z et al. Continuous monitoring of tightening condition of single-lap bolted composite joints using intrinsic mode functions of acoustic emission signals: a proof-of-concept study. *Structural Health Monitoring* 2019; 18(4): 1219–1234. DOI:10.1177/1475921718790768.
- [13] Fu W, Zhou R and Guo Z. Automatic bolt tightness detection using acoustic emission and deep learning. In *Structures*, volume 55. Elsevier, pp. 1774–1782. DOI:https://doi.org/10.1016/j.istruc.2023.06.100.
- [14] Verdin B, Chevallier G and Ramasso E. ORION-AE: Multisensor acoustic emission datasets reflecting supervised untightening of bolts in a jointed vibrating structure, Harvard Dataverse, 2021. DOI:10.7910/DVN/FBRDU0.
- [15] Ioffe S and Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. pmlr, pp. 448–456.
- [16] Farrar C and Worden K. *Structural Health Monitoring: A Machine Learning Perspective*. John Wiley & Sons, Ltd, 2013.
- [17] Chen SX, Zhou L, Ni YQ et al. An acoustic-homologous transfer learning approach for acoustic emission–based rail condition evaluation. *Structural Health Monitoring* 2021; 20(4): 2161–2181. DOI:https://doi.org/10.1177/ 1475921720976941.
- [18] Xin H, Cheng L, Diender R et al. Fracture acoustic emission signals identification of stay cables in bridge engineering application using deep transfer learning and wavelet analysis. *Advances in Bridge Engineering* 2020; 1(1): 1–16. DOI:https://doi.org/10.1186/s43251-020-00006-7.
- [19] Pandiyan V, Drissi-Daoudi R, Shevchik S et al. Deep transfer learning of additive manufacturing mechanisms across materials in metal-based laser powder bed fusion process. *Journal of Materials Processing Technology* 2022; 303: 117531. DOI:https://doi.org/10.1016/j.jmatprotec.2022.117531.
- [20] de Oliveira Teloli R, Butaud P, Chevallier G et al. Dataset of experimental measurements for the Orion beam structure. *Data in Brief* 2021; 39: 107627. DOI:https://doi.org/10.1016/j.dib.2021.107627.
- [21] de Olivera Teloli R, Butaud P, Chevallier G et al. Good practices for designing and experimental testing of dynamically excited jointed structures: The orion beam. *Mechanical Systems and Signal Processing* 2022; 163: 108172. DOI:https://doi.org/10.1016/j.ymssp.2021.108172.
- [22] Ramasso E, Verdin B and Chevallier G. Monitoring a bolted vibrating structure using multiple acoustic emission sensors: A benchmark. *MDPI DATA* 2022; 7: 31–45. DOI:https://doi.org/10.3390/data7030031.

- [23] Miguel LP, Teloli RO, da Silva S et al. Probabilistic machine learning for detection of tightening torque in bolted joints. *Structural Health Monitoring* 2022; 21(5): 2136–2151. DOI:10.1177/14759217211054150.
- [24] da Silva S, Omori Yano M, Teloli RdO et al. Domain Adaptation Of Population-Based Of Bolted Joint Structures For Loss Detection Of Tightening Torque. ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg 2023; : 1–12DOI:10.1115/1.4063794.
- [25] Gardner P, Bull L, Gosliga J et al. Foundations of population-based shm, part III: Heterogeneous populationsmapping and transfer. *Mechanical Systems and Signal Processing* 2021; 149: 107142. DOI:https://doi.org/10. 1016/j.ymssp.2020.107142.
- [26] Brake MRW, Schwingshackl C and Reuß P. Observations of variability and repeatability in jointed structures. *Mechanical Systems and Signal Processing* 2019; 129: 282 – 307. DOI:https://doi.org/10.1016/j.ymssp.2019.04. 020.
- [27] Zhang M, Lu L, Wang W et al. The roles of thread wear on self-loosening behavior of bolted joints under transverse cyclic loading. *Wear* 2018; : 30–39DOI:https://doi.org/10.1016/j.wear.2017.10.006.
- [28] Li D, Botto D, Xu C et al. Fretting wear of bolted joint interfaces. *Wear* 2020; 458-459: 203411. DOI: https://doi.org/10.1016/j.wear.2020.203411.
- [29] Stachowiak G and Batchelor AW. Engineering tribology. London: Butterworth-Heinemann, 2013.
- [30] Ramasso E, Denoeux T and Chevallier G. Clustering acoustic emission data streams with sequentially appearing clusters using mixture models. *Mechanical Systems and Signal Processing* 2022; 181: 109504. DOI:https: //doi.org/10.1016/j.ymssp.2022.109504.
- [31] Bull L, Gardner P, Gosliga J et al. Foundations of population-based shm, part I: Homogeneous populations and forms. *Mechanical Systems and Signal Processing* 2021; 148: 107141. DOI:https://doi.org/10.1016/j.ymssp.2020. 107141.
- [32] Pomponi E, Vinogradov A and Danyuk A. Wavelet based approach to signal activity detection and phase picking: Application to acoustic emission. *Signal Processing* 2015; 115: 110 – 119. DOI:https://doi.org/10.1016/j.sigpro. 2015.03.016.
- [33] Bao Y, Tang Z, Li H et al. Computer vision and deep learning-based data anomaly detection method for structural health monitoring. *Structural Health Monitoring* 2019; 18(2): 401–421. DOI:https://doi.org/10.1177/ 1475921718757405.
- [34] Azimi M and Pekcan G. Structural health monitoring using extremely compressed data through deep learning. *Computer-Aided Civil and Infrastructure Engineering* 2020; 35(6): 597–614. DOI:https://doi.org/10.1111/mice. 12517.
- [35] Liu W, Tang Z, Lv F et al. An efficient approach for guided wave structural monitoring of switch rails via deep convolutional neural network-based transfer learning. *Measurement Science and Technology* 2022; 34(2): 024004. DOI:10.1088/1361-6501/ac9ad3.
- [36] Postorino H, Monteiro E, Rébillat M et al. Cross-structures deep transfer learning through kantorovich potentials for lamb waves based structural health monitoring. *Journal of Structural Dynamics* 2023; 2: 24–50. DOI: 10.25518/2684-6500.135.
- [37] Yu Y, Hoshyar AN, Samali B et al. Corrosion and coating defect assessment of coal handling and preparation plants (chpp) using an ensemble of deep convolutional neural networks and decision-level data fusion. *Neural Computing and Applications* 2023; 35(25): 18697–18718. DOI:10.1007/s00521-023-08699-3.
- [38] Yu Y, Samali B, Rashidi M et al. Vision-based concrete crack detection using a hybrid framework considering noise effect. *Journal of Building Engineering* 2022; 61: 105246. DOI:10.1016/j.jobe.2022.105246.
- [39] Özgenel ÇF and Sorguç AG. Performance comparison of pretrained convolutional neural networks on crack detection in buildings. In *Isarc. proceedings of the international symposium on automation and robotics in construction*, volume 35. IAARC Publications, pp. 1–8.
- [40] Wang Y, Sun D, Chen K et al. Egeria: Efficient dnn training with knowledge-guided layer freezing. In *Proceedings of the Eighteenth European Conference on Computer Systems*. EuroSys '23, New York, NY, USA: Association for Computing Machinery. ISBN 9781450394871, p. 851–866. DOI:10.1145/3552326.3587451.
- [41] Szegedy C, Liu W, Jia Y et al. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 1–9.
- [42] He K, Zhang X, Ren S et al. Deep residual learning for image recognition. In *Proceedings of the IEEE conference* on computer vision and pattern recognition. pp. 770–778.

- [43] Sandler M, Howard A, Zhu M et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4510–4520.
- [44] Tan M and Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, pp. 6105–6114.
- [45] Hu J, Shen L and Sun G. Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141.
- [46] Smith LN. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:180309820 2018; .
- [47] Smith LN and Topin N. Super-convergence: Very fast training of neural networks using large learning rates. In Artificial intelligence and machine learning for multi-domain operations applications, volume 11006. SPIE, pp. 369–386.
- [48] Hou L, Yu CP and Samaras D. Squared earth mover's distance-based loss for training deep neural networks. *arXiv* preprint arXiv:161105916 2016; .
- [49] Bianchi D, Mayrhofer E, Gröschl M et al. Wavelet packet transform for detection of single events in acoustic emission signals. *Mechanical Systems and Signal Processing* 2015; 64: 441–451. DOI:https://doi.org/10.1016/j. ymssp.2015.04.014.
- [50] Donoho D. De-noising by soft-thresholding. IEEE transactions on information theory 1995; 41(3): 613-627.
- [51] Kharrat M, Ramasso E, Placet V et al. A signal processing approach for enhanced acoustic emission data analysis in high activity systems: Application to organic matrix composites. *Mechanical Systems and Signal Processing* 2016; 70: 1038–1055. DOI:https://doi.org/10.1016/j.ymssp.2015.08.028.
- [52] Loshchilov I and Hutter F. Decoupled weight decay regularization. In *International Conference on Learning Representations*. pp. 1–18.
- [53] Ai L, Soltangharaei V and Ziehl P. Evaluation of ASR in concrete using acoustic emission and deep learning. *Nuclear Engineering and Design* 2021; 380: 111328. DOI:https://doi.org/10.1016/j.nucengdes.2021.111328.
- [54] Ai L, Bayat M and Ziehl P. Localizing damage on stainless steel structures using acoustic emission signals and weighted ensemble regression-based convolutional neural network. *Measurement* 2023; 211: 112659. DOI:https://doi.org/10.1016/j.measurement.2023.112659.
- [55] Tobiasz R, Wilczyński G, Graszka P et al. Edge devices inference performance comparison. *Journal of Computing Science and Engineering* 2023; 17(2): 51–59. DOI:10.5626/JCSE.2023.17.2.51.