Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Gear Grinding Monitoring based on Deep Convolutional Neural Networks

Chenyu Liu * **Alexandre Mauricio** * **Zhuyun Chen** **
Katrien Declercq *** **Yannick Meerten** ***
Yann Vonderscher *** **Konstantinos Gryllias** *

* *Department of Mechanical Engineering, KU Leuven Dynamics of Mechanical and Mechatronic Systems, Flanders Make, Leuven, Belgium (e-mail: konstantinos.gryllias@kuleuven.be)*
** *School of Mechanical and Automotive Engineering, South China University of Technology, Guangzhou, China (e-mail: mezychen@gmail.com)*
*** *VCST Industrial Products, Sint-Truiden, Belgium (e-mail: Katrien.Declercq@vcst.be, Yannick.Meerten@vcst.be)*

Abstract: Grinding plays a vital role in modern gear manufacturing industry while the need for high quality products is continuously increasing. A methodology for gear grinding monitoring, exploiting the power of Deep Learning architectures and 2D representations, is presented in this paper. Vibration signals, measured during the grinding process under healthy and faulty conditions, are classified with high accuracy. Three types of faults i.e., a high profile form error, a high lead error, and a high profile slope variation, have been emulated. The Short-Time Fourier Transform (STFT) of each vibration signal is calculated, and the 2D time-frequency representations are input to a Deep Convolutional Neural Network (DCNN) for classification. Different filter sizes are tested, and the classification accuracy of 95.0% has been achieved, demonstrating the efficiency of the methodology for gear grinding monitoring.

Keywords: Process monitoring, deep learning, convolutional neural network, class activation map, gear grinding.

1. INTRODUCTION

The concept of Industry 4.0 sparks a revolution in the manufacturing industry. Traditional industrial automation is upgrading to a data stream-based, fully connected, and more flexible manufacturing system, also known as the smart factory (Albers et al., 2016). Supported by highly digitalized facilities, advanced sensor technologies as well as powerful signal processing tools, engineers now focus more on optimizing the manufacturing processes to achieve higher quality and accuracy products. Grinding is one of the most widely applied machining processes. In the gear manufacturing industry, grinding presents, compared to other processes, higher abrasives delivering stability, higher material removal efficiency, and longer tool life. The performance of the gear grinding process depends mainly on the condition of the tool, the correct positioning of the workpiece and the machine stability. In practice, a lousy execution of grinding will impact both the gear surface integrity and the geometric accuracy and will lead to various manufacturing faults, which may result in low quality gear products and finally in increased machine noise and reduced machine performance. Gupta et al. (2017) illustrated that, during the gear grinding process, several manufacturing faults can occur and can be classified under three major categories: high profile form error $ff\alpha$, high lead error $ff\beta$ and high profile slope variation $fh\beta$. When the defected gears are not identified on time, they are normally delivered to the customer, leading to complaints, dissatisfaction, and economic loss. Therefore there is a strong need for automated systems that can monitor the grinding process and detect in almost real-time any abnormalities in the quality of the produced gears. Strangely enough, until today, only a limited number of papers on online monitoring of grinding processes are available. Several frequency domain features have been proposed by Gryllias et al. (2017) which shed some light on the study of gear grinding characteristics. However, due to the high complexity of the grinding process, extracting such frequency domain features mostly depends on the diagnostic expertise. Besides, such features in some cases may have low generalization ability and therefore might be valid only in specific cases.

Recently, deep learning gained much attention from researchers in different fields. In the area of process monitoring, deep learning provokes a leap forward from traditional engineering crafted features towards intelligent feature extraction. Thanks to the advantage of self-learning, an efficient and automatic deep learning architecture can adaptively capture the representative information from any input through multiple non-linear transformations and approximate complex non-linear functions with very small error (Jia et al., 2016). So far, various attempts have proved the effectiveness of deep architectures in machine fault detection and classification, including 1D Convo-

lutional Neural Networks (CNN) (Zhang et al., 2018b), deep autoencoder (Chen and Li, 2017) and Long-Short Term Memory (LSTM) networks (Sang et al., 2018). These models have been successfully applied in vibration-based signal classification problems. The usage of 2D images to feed the deep neural network has been recently exploited by Fanioudakis et al. (2018) and Huang et al. (2017) using respectively Short-Time Fourier Transform (STFT) and Wavelet Transform representations as inputs for fault classification.

The work of this paper focuses on the monitoring of gear quality and more specifically on the gear grinding fault identification and classification, exploiting the power of Deep Convolutional Neural Networks (DCNN). Gear grinding experiments are conducted with 6 types of artificial defects embedded, and the vibration signals are collected during the experiments, which are further transformed into 2D representations using STFT. The 2D images are adopted as the inputs of a specially designed DCNN architecture with selected hyper-parameters. A comparative analysis is conducted to investigate further the influence of filter size on the output classification performance. The testing results show that the proposed methodology achieves high accuracy in grinding faults classification. Furthermore, the Gradient-weighted Class Activation Mapping (Grad-CAM) is used to get more insights into the network operations.

The rest of the paper is organized as follows: an introduction of CNN is presented in Section 2, while the methodology is briefly presented in Section 3. The detailed experimental study is discussed in Section 4, and the results are provided in Section 5. In Section 6, a first effort to understand the interior of CNN via visualization is demonstrated. The paper finally closes with some conclusions.

2. THEORETICAL BACKGROUND

CNN is nowadays one of the dominate deep learning approaches, proposed initially for image processing. So far, it has been widely and successfully applied in image classification (Wenhui and Fan, 2017), moving object detection (Zhang et al., 2018a), (Grassi and Grieco, 2003) and semantic segmentation (Noh et al., 2015).

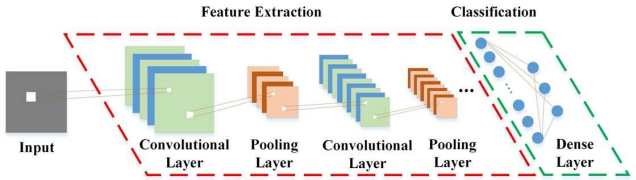


Fig. 1. Classic CNN structure

A classic CNN structure for classification is formed by two major modules i.e., the feature extraction and the classification module, as shown in Fig. 1. In the feature extraction module, convolutional layers and pooling layers are combined. The convolutional layer is structured by a certain number of filter kernels, which represent the weights in the neural network. These filters are convolved with the input to extract local features and are followed by the activation functions, which lead to the output feature maps. The mathematical description of the convolutional operation is described by Noh et al. (2015) as following:

$$\mathbf{X}_k^{(m)} = f(\sum_{c=1}^{c} \mathbf{W}_k^{(c,m)} * \mathbf{X}_{k-1}^{(c)} + \mathbf{B}_k^{(m)}) \tag{1}$$

where $*$ is the convolution operator; k represents the number of layers, c is the number of channels, and m is the number of the filters. $\mathbf{X}_{k-1}$ denotes the input, and $\mathbf{X}_k$ is the output feature map. $\mathbf{W}_k$ is the weight i.e., the filter with the corresponding bias $\mathbf{B}_k$. Finally, the activation function f is applied to the summation of all convolution operations. A pooling layer is usually set after the convolutional layer in the CNN architecture. It acts as the downsampling process to reduce the spatial size of the feature map. Max-pooling is the most commonly used pooling method which can be described as:

$$p^{l(i,j)} = max_{(i-1)\mathbf{W}+1 \le t \le jw}\{\alpha_{l(i,t)}\} \tag{2}$$

where $\{\alpha_{l(i,t)}\}$ represents the i-th neuron in the t-th frame of the l layer. $\mathbf{W}$ is the width of the pooling region, and t is the step for each pooling operation. The output of the max-pooling layer is a maximum local value $p^{l(i,j)}$, which can be considered as the extracted feature. The purpose of pooling is to reduce the number of parameters of the network, meanwhile obtaining location-invariant features. The classification module is formed by fully connected layers i.e. dense layers. For most classification problems, the dense layer usually uses a softmax activation function:

$$y_k = \frac{e^{a_k}}{\Sigma_{i=1}^{n} e^{a_i}} \tag{3}$$

where y_k is the output of the k-th neuron, and a_k is the input signal from the former layer. Softmax function is often used in the output layer of the neural network to give a final probability of the output class. The final decision of the classifier is made based on this probability. In the case of multiple convolutional and pooling layers, CNN is constructed as a DCNN. With the increasing of the layer depth, more abstract features can be captured with deeper layer filters. Typically, the number of filters for each convolutional layer will increase simultaneously with the depth of layers. Layers with other functions, such as drop-out layers and batch-normalization layers, can also be added into the structure for particular purposes. The training of the DCNN mainly includes three parts: the loss function, the optimization algorithm, and the backpropagation process. Among others, the cross-entropy error and the mean square error are the most widely used loss functions. After applying the loss function, the gradient should be calculated to find the fastest descent using an optimization algorithm.

3. METHODOLOGY

Instead of directly using features extracted from a 1D vibration signal, the combination of a 2D spectrogram with a DCNN architecture is used in this paper to detect and classify gear grinding faults. The flow chart of the proposed STFT-CNN methodology is shown in Fig. 2. The measured vibration signal is firstly transformed using the Short-Time Fourier Transform (STFT) to obtain the spectrogram. Then the spectrogram is converted to a 2D matrix and

is used as an input to the DCNN. A sufficient number of labeled data, including data from all classes, are used for training. The trained DCNN model can be used for the identification and classification of process faults. In this framework, the spectrogram can practically be replaced by any time-frequency (i.e., Wigner-Ville spectra) or time-scale representations (i.e., wavelets).

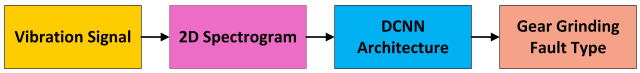


Fig. 2. Flow chart of the proposed methodology

4. EXPERIMENTAL STUDY

To test and evaluate the effectiveness of the abovementioned methodology, a series of measurements, emulating healthy and faulty operating conditions, have been performed on an industrial gear grinding system (Fig. 3). The workpiece is mounted on the spindle, and the grinding worm moves and machines the workpiece. Three types of signals are captured simultaneously: the worm speed signal, the worm current signal, and the vibration signals. Two accelerometers are mounted on the grinding worm holder and on the workpiece spindle base, respectively, to capture the vibration during the grinding process. The emulated process faults are achieved by adjusting the operational parameters of the grinding program. The three types of faults i.e. the $ff\alpha$, the $ff\beta$, and the $fh\beta$ are emulated by controlling the feed rate, the infeed distance, and the workpiece eccentric distance respectively. The detailed implementation and the physical explanation of these operational parameters can be found in (Gryllias et al., 2017). Each process fault is realized in two groups of parameters to simulate two levels of fault severity. In total, 120 measurements have been captured with 7 labels, as listed in Table. 1. For each fault type, five sequential measurements have been captured before redressing the grinding worm. Due to production needs, a rather limited dataset has been made available for deep learning research. Therefore to prevent overfitting on the small number of training samples, a cross-validation strategy has been adopted, as described in the following section.

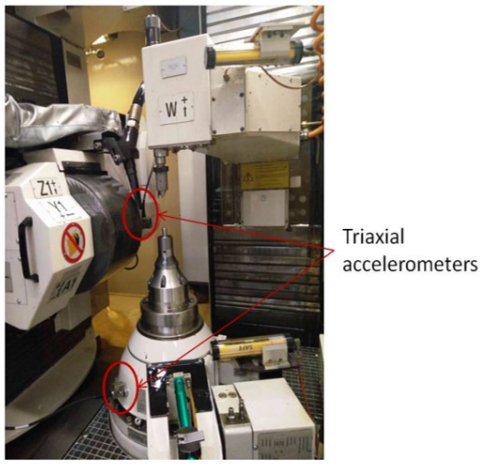
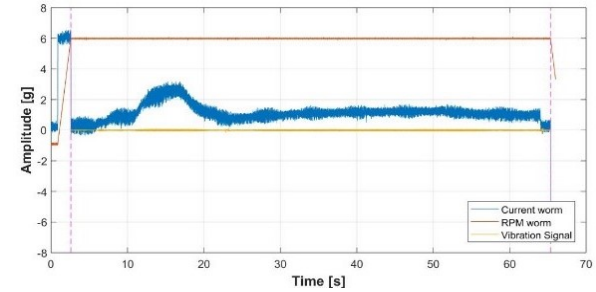


Fig. 3. Gear grinding machine

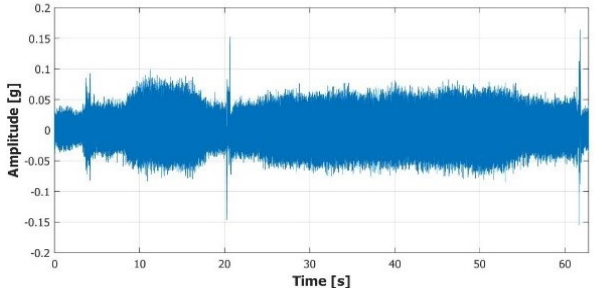
Table 1. Gear grinding dataset

Label	Operation Parameters	Number
Healthy	Feed rate 0.2 mm/rev; Infeed 0.1 mm	60
Feed 1	Feed rate +0.15 mm/rev	10
Infeed 1	Infeed +0.08 mm	10
Excenter 1	Excenter 20 μm	10
Feed 2	Feed rate +0.30 mm/rev	10
Infeed 2	Infeed +0.15 mm	10
Excenter 2	Excenter 40 μm	10

A vibration signal, captured under normal gear grinding operation (healthy condition - baseline), is illustrated in Fig. 4. The signals have been captured from the accelerometer mounted on the grinding worm holder with a sampling frequency of 25600 Hz. The signal measured at the z-axis presents higher energy and represents the vibration characteristics better than the other two axes. For this reason, only the z-axis signals are used. The accelerometers record the whole grinding operation, including the run-up of the grinding machine and its final run down after the end of the grinding process. The vibration signal, the worm motor current signal, and the worm motor speed signal (in RPM) are presented in Fig. 4a. A speed increasing and decreasing section at the beginning and the end can be identified. The grinding process takes place only during the stationary speed condition. Therefore the worm speed signal is used to remove the starting and ending sections from the vibration signals to reduce the irrelevant noise, as shown in Fig. 4b. The spectrogram of the baseline signal, estimated with an overlap of 1 second, is presented in Fig. 5. Several excited frequency bands can be observed: a) below 1000 Hz, b) between 3000 and 4000 Hz, c) around 10000 Hz, and d) around 12000 Hz.



(a)



(b)

Fig. 4. Gear grinding signal in normal (healthy) operation (a) original signals from multiple sensors (b) segmented vibration signal

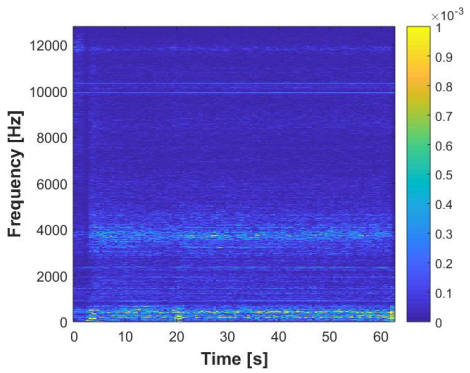


Fig. 5. 2D STFT representation of a healthy baseline signal

5. APPLICATION

Signal processing results show that gear grinding is a rather complex process due to time-varying cutting forces, high level of noise, and various interfering parameters. It is rather difficult to identify different types of faults by signal processing tools during the grinding process. Therefore DCNN is adopted in this work to capture the hidden patterns embedded in the signals and extract information that can be used for the discrimination of the classes/fault types. Inspired by the classic LeNet-5, a 5 block DCNN is built with Keras and Tensorflow. The main structure of the DCNN is shown in Table. 2. The spectrogram is initially downsampled to the size of 64×224, which is selected as a trade-off between computational efficiency and feature representing ability. The first 4 blocks are constructed with the same frame: a convolution layer with a Rectified Linear Unit (ReLU) activation, a max pooling layer, and a batch normalization layer. The last block consists of a flatten layer and two dense layers. In order to conduct the classification, the 2D representations have to be transformed into 1D before being input to the classifier. The last dense layer uses softmax as the final activation function.

Table 2. DCNN architecture

Input 1×64×224
Layer 1: Convolution (stride-1)-6-ReLu
Layer 2: 2×2 Max Pooling-BN
Layer 3: Convolution (stride-1)-6-ReLu
Layer 4: 2×2 Max Pooling-BN
Layer 5: Convolution (stride-1)-12-ReLu
Layer 6: 2×2 Max Pooling-BN
Layer 7: Convolution (stride-1)-12-ReLu
Layer 8: 2×2 Max Pooling-BN
Layer 9: Flatten
Layer 10: Fully Connected (dim-500)-ReLu
Layer 11: Fully Connected (dim-7)-Softmax

The training of the DCNN is realized using the adaptive moment estimation optimization algorithm (Adam) and the categorical cross-entropy as a loss function. The selection of hyper-parameters is one of the toughest challenges in deep learning research. Szegedy et al. (2016) proposed that 2×2 max pooling can effectively reduce the scale of hidden layers, which makes it the common choice for deep architectures. Zhang et al. (2018b) has reported that the increasing of kernel numbers can help to increase the depth of the feature space and results in sufficient learning of global abstract features. Based on the abovementioned remarks, the proposed DCNN architecture uses 2×2 as the size of max pooling and 6-6-12-12 as the kernel depth with 3×3 kernels. Since the total number of the available measurements is relatively limited, a 5-fold cross-validation approach is adopted to guarantee the generalization performance of the proposed DCNN. For each fold, 96 samples are used for training, and 24 remain for testing. Training and testing sets are stratified random split for each label group. The classification results are presented in Table 3.

Table 3. 5-fold cross validation results

Fold	1	2	3	4	5	Average
Accuracy	91.6%	91.6%	91.6%	95.8%	100.0%	95.0%

It should be noted a priori that 95.8% and 91.6% correspond respectively to 1 and 2 samples of misclassification. The results show that the proposed DCNN structure is effective, with an average accuracy of 95.0%. Since the selection of kernel size has effects on the feature distribution, 5 DCNN structures, marked from DCNN1 to DCNN5 with various combinations of kernels, are further studied to explore the influence of filter sizes to the global classification accuracy. The results are presented in Table 4.

Table 4. Classification results

Experiment	Filter Sizes	Average Accuracy
DCNN1	3-3-3-3	95.0%
DCNN2	3-3-3-5	91.6%
DCNN3	3-3-5-5	90.8%
DCNN4	3-5-5-5	79.2%
DCNN5	5-5-5-5	80.3%

It can be found that the classification accuracy of the smaller filter size structure is better than the large filter size. Small filters are more suitable to capture the features from the spectrograms. According to Defferrard et al. (2016), the use of a stack of multiple small size filters can reduce the structure complicity significantly compared to large filters. The accuracy and the loss curves of DCNN1 and DCNN5 with different filter sizes are presented in Fig. 6. The model accuracy of the testing process grows with training (Fig. 6a). It can be observed that the two curves almost overlap, highlighting that the model has gained generalization ability after the training. Fig. 6c shows that in experiment DCNN5, the accuracy drops after epoch 25.

The confusion matrix of DCNN1 is presented in Fig. 7. It can be found that the misclassification of each fold is no more than 1 sample. Considering the amount of samples in the dataset, this number is acceptable. It should also be noticed that one sample labelled Feed 1, one labelled Infeed 2, and one labelled Excenter 2 are misclassified as Healthy, while one Healthy and one Infeed 1 are misclassified both as Excenter 2 during this experiment. Mathematically speaking, there is a strong possibility that similar features are extracted for the labels Healthy and Excenter 2 since they are both misclassified

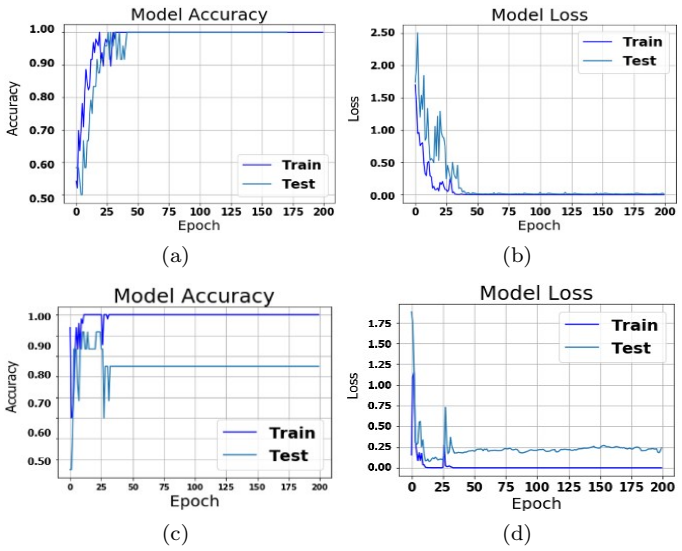


Fig. 6. Training and testing of DCNN models (a) DCNN1 accuracy curve (b) DCNN1 loss curve (c) DCNN5 accuracy curve (d) DCNN5 loss curve

to each other. On the other hand, based on the comments of the machine operators, the eccentricity might have been corrected during the grinding process; therefore, some of the eccentric samples might be finally healthy.

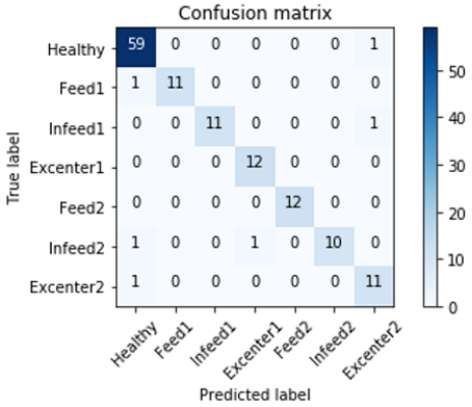


Fig. 7. Confusion matrix of DCNN1

6. VISUALIZATION

In an effort to understand the classification results of the structure with different filter sizes, a number of the trained filters for each layer of DCNN1 and DCNN5 are visualized and presented in Fig. 8. It has been reported that the shallow layer filters of CNN depict the edges of the image and that the deeper ones focus on more complex textures. Szegedy et al. (2015) have found that smaller filter sizes could focus on local features. Thus vast information is extracted, especially in deep layers. Besides, a smaller filter size could reduce the number of hyper-parameters to improve computational efficiency. It can be seen from Fig. 8a that the filters in the first two convolutional layers of DCNN1 have shown some patterns along the horizontal direction. The filters in red dotted boxes show a texture-like spatial arrangement of pixels, which can be interpreted as feature extraction tools. 5×5 filters also show such pattern, as can be seen in Fig. 8b, but in less cases. Other filters behave more noisy compared to the 3×3 filters. The classification results indicate that the proposed DCNN architecture can generalize to unseen data. The neural network is regarded as a black-box model, and its learning process is difficultly interpreted.

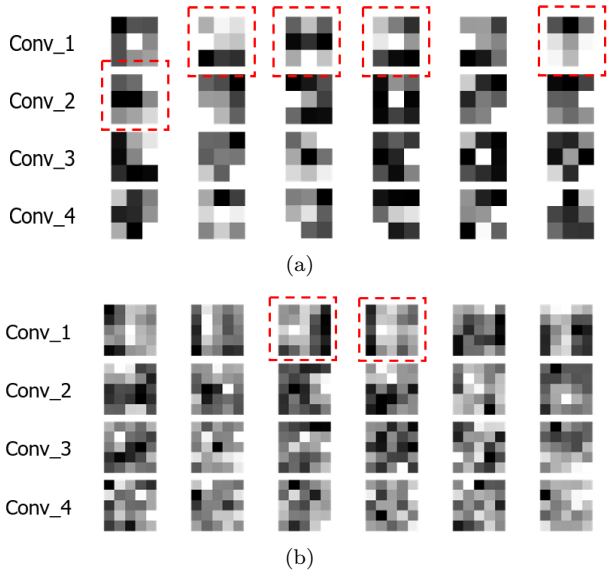
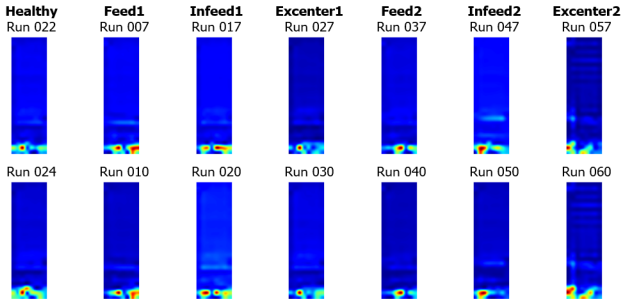
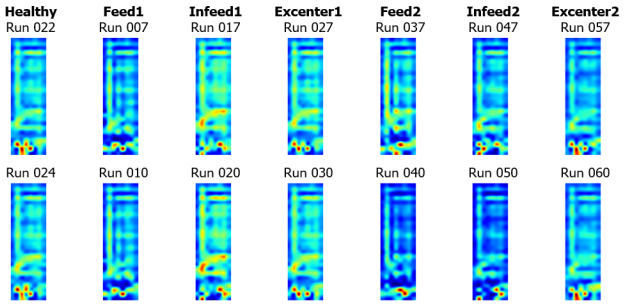


Fig. 8. Visualization of filters in (a) DCNN1 (b) DCNN5

Recently, several attempts have been proposed targeting to peek inside the network to understand and obtain knowledge on what and how the deep networks are learning from data. Gradient-weighted Class Activation Mapping (Grad-CAM) is one way to visualize the attention of the network. It generates a localization map highlighting important regions based on the gradient information from the last convolutional layer (Selvaraju et al., 2017). With the 2D spectrogram input, the Grad-CAM can be used to trace the discriminative time-frequency regions of the different classes according to the ground-truth. The Grad-CAM is derived from the convolutional layer in block 4 of the proposed DCNN architecture. 14 Grad-CAMs from testing samples of the 7 classes (same class each column) of DCNN1 and DCNN5 are presented in Fig. 9. The red color regions represent the attention of the network, and it can be found that the network is focusing on diverse regions for samples from different classes. In Fig. 9a, the attention region of DCNN1 is concentrated for the samples with the same label. The frequency band below 1kHz is the decisive area for all classifications. The focused regions of Feed1, Infeed1, Infeed2, and Excenter 2 are distributed in the longer area. On the other hand, the attention of the network is more centralized for Healthy, Feed2, and Excenter1. The results show that 3×3 filters can capture the features in the characteristic frequency zone. On the contrary, the Grad-CAMs of DCNN5, shown in Fig. 9b, present a widespread of activations with lower intensity of activation for all labels. It is noticed that samples Run 017, Run 037, and Run 047 are misclassifications in this case. Since a larger filter size extends the receptive field, it is obvious that DCNN5 lacks the ability to capture small and complex features, and results in more misclassifications with lower classification accuracy.



(a)



(b)

Fig. 9. Grad-CAMs of samples in (a) DCNN1 (b) DCNN5

7. CONCLUSION

A novel gear grinding monitoring methodology, exploiting the power of DCNN and 2D spectrograms, has been proposed in this paper. The methodology has been tested and validated on an industrial gear grinding system, and vibration signals have been classified in 7 classes with an average accuracy of 95.0%, proving the efficiency of the method. To understand the influence of the kernel size, varying sizes of filters are tested, and the results show that 3×3 filters can achieve a better classification performance compared to 5×5 filters. Moreover, the visualization of the filters and the Grad-CAM method can provide some explanation about the results as well as about the learning process of DCNN. Globally the proposed deep architecture is proved extremely effective for the process monitoring of gear manufacturing.

ACKNOWLEDGEMENTS

This research was supported by Flanders Make, the strategic research centre for the manufacturing industry and VLAIO in the frames of MODA project.

REFERENCES

Albers, A., Gladysz, B., Pinner, T., Butenko, V., and Stürmlinger, T. (2016). Procedure for defining the system of objectives in the initial phase of an industry 4.0 project focusing on intelligent quality control systems. *Procedia Cirp*, 52, 262–267.

Chen, Z. and Li, W. (2017). Multisensor feature fusion for bearing fault diagnosis using sparse autoencoder and deep belief network. *IEEE Transactions on Instrumentation and Measurement*, 66(7), 1693–1702.

Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, 3844–3852.

Fanioudakis, E., Geismar, M., and Potamitis, I. (2018). Mosquito wingbeat analysis and classification using deep learning. In *2018 26th European Signal Processing Conference (EUSIPCO)*, 2410–2414. IEEE.

Grassi, G. and Grieco, L.A. (2003). Object-oriented image analysis using the cnn universal machine: new analogic cnn algorithms for motion compensation, image synthesis, and consistency observation. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 50(4), 488–499.

Gryllias, K., Kilundu, B., Devos, S., Vonderscher, Y., and Vandewal, B. (2017). Condition monitoring of gear grinding processes. In *International Conference Surveillance, Date: 2017/05/22-2017/05/24, Location: INSA Euro-Mediterranean, Fes, Morocco*.

Gupta, K., Jain, N.K., and Laubscher, R. (2017). *Advanced gear manufacturing and finishing: classical and modern processes*. Academic Press.

Huang, H., He, R., Sun, Z., and Tan, T. (2017). Wavelet-srnet: A wavelet-based cnn for multi-scale face super resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, 1689–1697.

Jia, F., Lei, Y., Lin, J., Zhou, X., and Lu, N. (2016). Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mechanical Systems and Signal Processing*, 72, 303–315.

Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, 1520–1528.

Sang, J., Park, S., and Lee, J. (2018). Convolutional recurrent neural networks for urban sound classification using raw waveforms. In *2018 26th European Signal Processing Conference (EUSIPCO)*, 2444–2448. IEEE.

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, 618–626.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Wenhui, Y. and Fan, Y. (2017). Lidar image classification based on convolutional neural networks. In *2017 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, 221–225. IEEE.

Zhang, M., Li, W., and Du, Q. (2018a). Diverse region-based cnn for hyperspectral image classification. *IEEE Transactions on Image Processing*, 27(6), 2623–2634.

Zhang, W., Li, C., Peng, G., Chen, Y., and Zhang, Z. (2018b). A deep convolutional neural network with new training methods for bearing fault diagnosis under noisy environment and different working load. *Mechanical Systems and Signal Processing*, 100, 439–453.