Feature Extraction from Raw Vibration Signal and Classification of Bearing Faults Using Convolutional Neural Networks

Tanju Gofran[1], Peter Neugebauer[1], and Dieter Schramm[2]

[1] IEEM - Institute for Energy Efficient Mobility
Karlsruhe University of Applied Sciences
gota0001@hs-karlsruhe.de
peter.neugebauer@hs-karlsruhe.de
[2] University of Duisburg-Essen
schramm@mechatronik.uni-duisburg.de

Abstract. Safe, efficient and uninterrupted continuous operation of an electric motor requires real-time condition monitoring of its rotating parts. Other than knowledge based signal analysis, fault feature extraction with statistical information or signal processing methods can be used to classify different fault patterns. But rule based feature extraction methods do not have domain adaptability, so the fault classification working in one system may not work for another system. A deep learning algorithm - Convolution Neural Networks approach is shown in this paper to classify different bearing faults and the trained network shows a good fault prediction capability for other systems.

Keywords: CNN, CMCNN, Conv Layer, Feature Map Analysis

1 Introduction

To avoid unwanted shut-down due to rotating machine fault in a manufacturing line , in a remote power plant and in many other applications a real-time monitoring of the machine condition is a high demand.The aim of Condition Monitoring (CM) of electric machines is to acquire a “health” indication in real-time; in order to identify possible failures in advance, thus avoiding costly and unscheduled down time, upholding accurate servicing schedules [1]. An ideal CM method should be non-destructive and should depend on easily measurable parameters[2]. In contrast to sensor-based techniques, data-driven condition monitoring methods are interesting because they do not require any knowledge about the machine parameters; instead, they only require a database of both healthy and faulty conditions of the machine for eventual feature extraction and classification. Rule based feature extraction and classification of machine conditions are studied in many publications[2–5]. The accuracy of classifying faults mainly depends on how accurate the feature extraction is.The limitation of hand crafted feature extraction is, the classification working in one domain most cases do not work in other domains. Deep Learning is a subclass of Machine Learning (ML) algorithm which can extract features directly from data without prior knowledge or mathematical background of input and then classify as required. Convolutional Neural Networks (CNN) is one Deep Learning technique believed to be the most popular ML algorithm in present time. Because of the advancement of computation technology and efficiency of CNN, it is now a widely used algorithm among researchers to solve many real time problems in various fields of natural science, computer science and engineering. In recent years many publications studied CM using CNN and Deep Neural Networks (DNN)[6–9]. [10]show a domain adaptability of classifying different bearing vibration signal using CNN . [11]proposed hierarchical deep architecture of CNN in which original data is converted into 2D data to classify bearing faults and their sizes. Many of the published work of CM with CNN approaches show very high accuracy, but these are mostly tested on the same dataset. In this work, we propose a simple CNN architecture for classifying bearing faults from 1D vibration signal, which is trained with one test bench dataset and tested on different test bench data. We propose an analysing approach of extracted features by the trained CNN model.

2 Convolutional Neural Networks (CNN)

Convolutional Neural Networks were inspired from the biological process of connectivity pattern of cells of visual cortex of cat and monkey published by Hubel and Wiesel in 1962 and 1968. They showed how without moving the eyes individual cortical neurons respond to stimuli only in a restricted region of the visual field known as receptive field. Modern CNN models also use the similar kind of approach to extract simple to more complex features of input image for classification or detection.

3 Proposed CNN Approach

The main constraint of training CNN models for classifying faults of motor is to have access of a large dataset. In our research project we develop a test-bench (IEEM-CMTestBench) to create different faulty bearing vibration data, but the challenge remains to implement real life faults on bearings. While the project is still in process, to continue our investigation we used a public dataset to train CNN model to classify bearing faults. We named our CNN model as CMCNN-Condition Monitoring Convolutional Neural Network. The trained CMCNN shows high accuracy to classify different bearing faults and finally we test the trained CMCNN model with a different system dataset to verify the model accuracy. The proposed approach is described in the following chapters:

3.1 Training Dataset

The bearing dataset for different faults is prepared by Case Western Reserve University (CWRU)[12] and the dataset is used to train CNN models in number of publications[8, 10, 6]. CWRU dataset provides vibration data for normal bearings (No Fault) and faulty bearings. Faults are artificially implemented on Inner Race (IR Fault), Rolling Element (RE Fault) and Outer Race (OR Fault) of both Drive-End (DE) and Fan-End (FE) bearings using electro-discharge machining (EDM). Fault diameters are ranging from 0.007 inches to 0.028 inches in diameter and vibration measurements are taken at 4 different loads.In order to quantify vibration response effect with load zone, measurements were conducted for both bearings with OR Faults located directly in the load zone(OR@3), at orthogonal to the load zone (OR@3), and at 12 o'clock(OR@12).

3.2 Data Preparation and Data Argumentation

The CWRU dataset includes 6 classes: No Fault, IR Fault, RE Fault, OR@6 Fault, OR@3 Fault and OR@12 Fault. Each class has less than 30 or sometimes even less

than 15 datasets. Generally for successful classification with a Deep Learning Algorithm a minimum of 1000 dataset per class is required. Similarly as[10] we also apply data argumentation to segment each signal data into smaller sizes to increase the number of datasets per classes. In this work, we consider to segment the signal in a way that the input data fed to the CNN model should be equivalent approximately to unit revolution of the bearing. To differentiate random vibrations to faulty or healthy vibration a set of Random Noise is also included in the Training Data.
Our work is comparable with [10] for CNN model design in some extend, where the authors used CWRU datasets to classify 10 classes. In their work each class has a certain size of fault (0.007 to 0.021 inch) and the 10 classes data are decided into 4 combinations of loads for training. In our work we classify five bearing fault classes and one No fault class, by extracting feature from data of all different fault sizes and loads of measurements.

3.3 CMCNN Architecture

The CMCNN contains five layers each containing Convolution (Conv) Layer, Rectified Linear Unit (ReLU) or Activation Layer and Max Pool (Pool) Layer for feature extraction and three Fully Connected Layers (FC) with 50% dropouts for classification. The model is designed and trained with MATLAB deep learning functions. The CMCNN architecture is shown in 1. The main block of a CNN is the Conv Layer, which is done by sliding

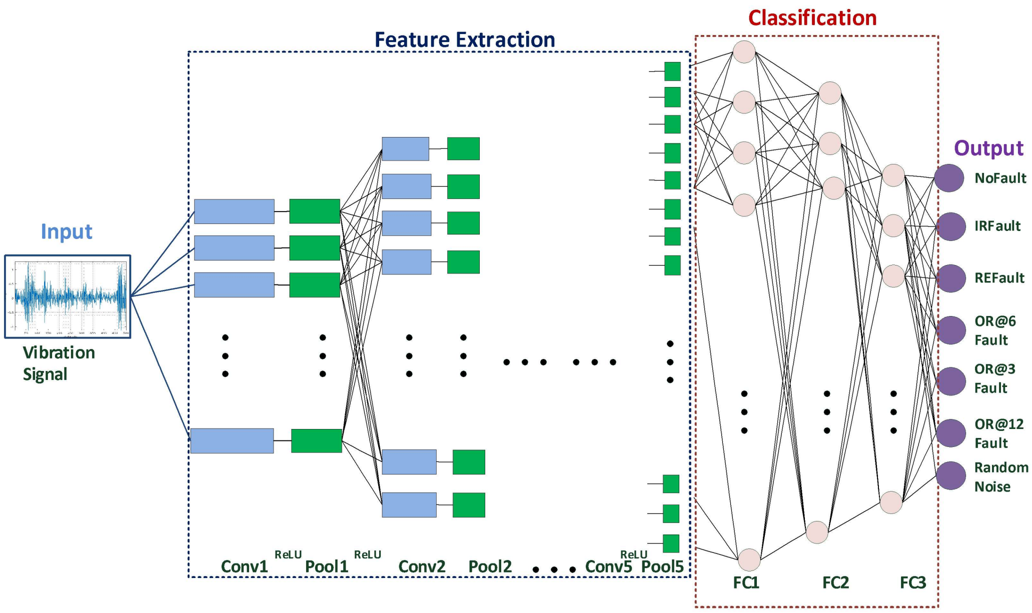


Fig. 1. CMCNN Architecture

a filter or kernel over the input data producing feature map for each covered location (receptive field) of the input and each layer contains multiple numbers of trainable filters. The two main parameters to modify the behaviour of each Conv layer are Stride(s) and Padding (p) and filter size. Stride controls how filters convolves around the input and padding is often use to preserve the information of original input. The size of output feature of Conv Layer can determine by Equation 1.

$$Outout_{size} = \frac{Input_{size} + 2p - filter_{size}}{s} + 1 \quad (1)$$

Generally in image recognition problems the filter size is very small and most of the time zero padding is considered. As the target of CMCNN is to find features from raw vibration data, it does not make sense to extract a very small scaled feature and the Conv Layers should preserve original information of input as much as possible. The idea of implementing wide filters in the first convolution layer is shown in [10]. The Conv Layers of CMCNN starts with wider filters and reduced gradually in later layers. Stride and Padding are parametrized in a way that the original input information is kept as much as possible. In CMCNN the number of filters and neurons at FC layers are chosen similar as VGG16 [13].

3.4 Training CMCNN

The Training dataset is divided 25% for validation and 2.5% data for testing. CMCNN network training parameters are updated with Stochastic Gradient Descent with Momentum (SGDM) algorithm. Stochastic Gradient Descent (SGD) algorithm is used to train network parameter to minimize the Error Function by converging to negative gradient loss, which might oscillate to reach optimum and convergence can be very slow. Momentum can be added to reduce the oscillation. A network parameter update by SGD can be expressed by Equation 2.

$$\Theta_{i+1} = \Theta_i - \alpha \nabla E(\Theta_i) \tag{2}$$

where i is the iteration number $\alpha > 1$ is the learning rate, Θ is parameter vector, $E(\Theta_i)$ is the loss function and $\alpha \nabla E(\Theta_i)$ is the gradient of the Error Function. SGD evaluates the gradient and update the parameter using subset of the training set. Training parameter update with SGDM can be expressed by Equation 3.

$$\Theta_{i+1} = \Theta_i - \alpha \nabla E(\Theta_i) + \gamma(\Theta_i - \Theta_{i+1}) \tag{3}$$

where γ determines the contribution of the previous gradient step to the current iteration. Training performance of CMCNN is shown in 1. True Positive Rate (TPR%) and False Positive Rate (FPR%) is calculated to determine the prediction accuracy per class. 1 also include prediction accuracy of OR and IR Fault class of two new Test data (SpectrQ). SpectrQ dataset is described in later sections.

Table 1. Trained CMCNN 7class performance

	Learned Class and corresponding Labels						
	No Fault	IR Fault	RE Fault	OR@6 Fault	OR@3 Fault	OR@12 Fault	Random Noise
Model: CMCNN_7class Mean Acc.: 94,48% Mean Error: 5,51%	0	1	2	3	4	5	6
Train. Class Dist	4943	4959	4873	3131	3342	2136	2151
Val. Class Dist	1664	1647	1711	1106	1125	687	790
Train. TPR (%)	99,98	90,18	96,80	89,01	91,46	95,37	100
Train. FPR (%)	0,02	9,82	3,20	10,99	8,54	4,63	0
Val. TPR (%)	99,94	89,42	96,54	88,57	90,62	93,53	100
Val. FPR (%)	0,06	10,58	3,46	11,43	9,38	6,47	0
SpectrQuest OR TPR (%)	-	-	-	1,25	0	72,5	0
SpectrQuest OR FPR (%)	14,58	2,5	9,17	-	0	-	0
SpectrQuest OR TPR (%)	-	96,67	-	-	-	-	-
SpectrQuest OR FPR (%)	0	-	0	0,42	0	2,92	0

3.5 Test CMCNN

In this work CMCNN model is trained and tested for a different number of class recognition using CWRU dataset; in 2 four different Models are compared.

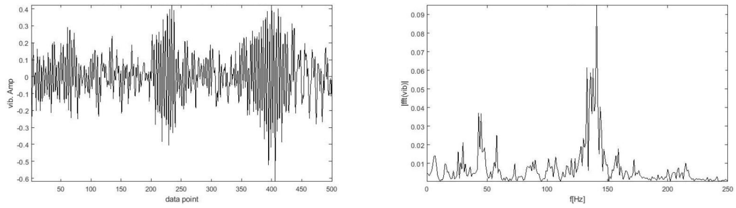
Table 2. Comparison CMCNN for different number of classes and their prediction accuracy for SpetrQ Test data:

Model	CMCNN 7 Class	CMCNN 6 Class	CMCNN 5 Class	CMCNN 4 Class
No Fault	0	0	...	0
IR Fault	1	1	1	1
RE Fault	2	2	2	2
OR@6 Fault	3	3	3	3
OR@3 Fault	4	4	4	3
OR@12 Fault	5	5	5	3
Random Noise	6	...	...	...
Mean TrainAcc (%)	94,48	93,56	99,51	99,78
Mean TrainErr (%)	5,51	6,44	0,48	0,23
Mean ValAcc (%)	94,00	93,58	99,37	99,55
Mean ValErr (%)	6,00	6,42	0,63	0,45
Mean TestAcc (%)	94,2	91,99	99,15	100,00
Mean TestErr (%)	5,80	8,01	0,85	0,00
SpectrQ ORFault PredAcc (%)	73,75	81,17	45,83	0,83
SpectrQ ORFault PredErr (%)	26,25	18,33	54,16	98,75
SpectrQ IRFault PredAcc (%)	96,67	87,50	100	100
SpectrQ IRFault PredErr (%)	3,33	12,50	0	0

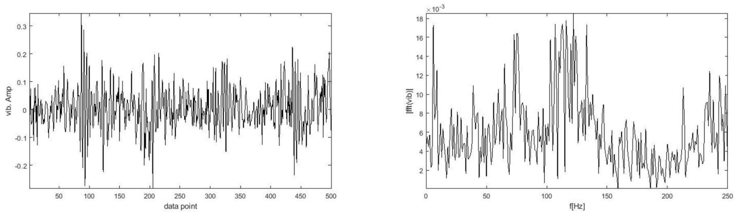
To check the efficiency of CMCNN, we tested this four trained CMCNN to predict OR Fault and IR Fault for a new publicly available (SpectrQ) dataset [14] produced by SepctrQuest test bench [15]. A comparison of Train Data-OR Fault and SpectrQ-OR Fault is shown in Fig. 2. SpectrQ-OR fault location is not known, so we labelled the dataset as target class '3' and checked if the CMCNN can successfully predict class as OR Fault independent where the fault location is. For this reason we trained one CMCNN model for 4 classes where we labelled all OR Fault as '3'. The result shows though the CMCNN accuracy for 4 classes is the highest, the robustness of predicting class for new test data (SpectrQ-OR Fault) is worse.

3.6 Feature Map Analysis

To understand and optimize the feature extraction it is required to visualize deep layer feature map. There are several approaches available to visualize the features, one is to visualize the activations of the network and other is to visualize the convolution weights. In Image recognition problems the output of each convolution layer can be interpretable because the outputs are also another image or sometimes the sharpened edges of the input image. For vibration input signals we should convert the feature output into different domain to understand. In our work we analyse the first Conv layer output by calculating it's Fast Fourier transform (FFT) and compare each filter if significant Feature Frequencies is extracted.In [10] authors also focused feature visualization with FFT and showed feature distribution for each layer and each 10 classes using Stochastic Neighbor Embedding (t-SNE). In our work, we focused on significant feature frequencies of first Conv layer for each filters and for each classes. We found that feature frequencies



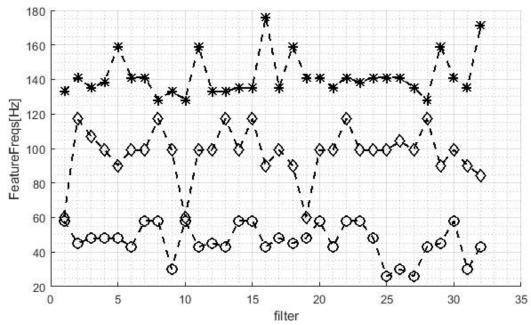
(a) Training



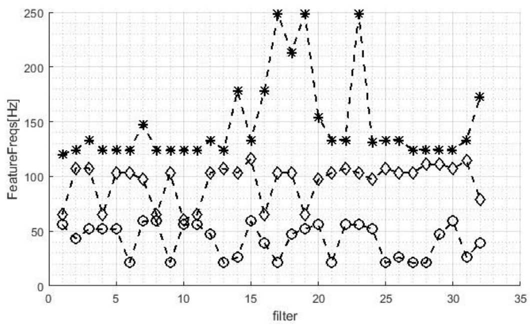
(b) Test

Fig. 2. Comparison of two examples of Training Data for OR Fault (a), and Test data for OR fault (b). Left: time domain data and Right: frequency domain data

are extracted in a range for each class and the extracted feature frequencies for test data (SpectQ) also in similar range of its predicted class. Figure (5) shows the comparison of feature frequencies of all 32 filters of first Conv layer of a Train Input-ORfault and Test Input SpectrQ-OR Fault.



(a) Training Input



(b) Test Input

Fig. 3. Feature Frequencies for all trained filters of first Conv Layer for (a) Train Input: OR Fault and (b) Test Input SpectrQ: OR Fault

4 Conclusion

CMCNN shows good accuracy of predicting OR Fault and IR Fault for SpectrQ Test data which indicates feature extraction of CMCNN is robust for different systems. In future we will compare the filter size and input size effect on feature learning which should find the optimized architecture of the CMCNN. In our research project we aimed to implement real-life bearing faults such as surface fatigue, wear, electrical erosion, plastic deformation due to overload etc. and create dataset using IEEM-CMTestBench. Furthermore, the created dataset will be used to train the optimized CMCNN architecture to achieve robust domain adaptable feature extraction and classification of real-life fault pattern for motor bearing.

References

1. Filippetti, F., Franceschini, G., Tassoni, C., Vas, P.: Recent developments of induction motor drives fault diagnosis using ai techniques. IEEE Transactions on Industrial Electronics **47**(5) (2000) 994–1004
2. Gao, R.X.: Neural networks for machine condition monitoring and fault diagnosis. Nato Science Series Sub Series III Computer and Systems Sciences **185** (2003) 167–188
3. Abbasion, S., Rafsanjani, A., Farshidianfar, A., Irani, N.: Rolling element bearings multi-fault classification based on the wavelet denoising and support vector machine. Mechanical Systems and Signal Processing **21**(7) (2007) 2933–2945
4. Su, H., Chong, K.T.: Induction machine condition monitoring using neural network modeling. IEEE Transactions on Industrial Electronics **54**(1) (2007) 241–249
5. Tóth, D., Szilágyi, A., Takács, G.: Investigation of rolling element bearings using time domain features. In Jármai, K., Bolló, B., eds.: VEHICLE AND AUTOMOTIVE ENGINEERING. Lecture Notes in Mechanical Engineering. Springer, [S.l.] (2017) 3–12
6. Ahmad Shaheryar, Xu-Cheng Yin and Waheed Yousuf Ramay: Robust feature extraction on vibration data under deep-learning framework: An application for fault identification in rotary machines. **167**(4) (2017) 37–45
7. Yin, Z., Hou, J.: Recent advances on svm based fault diagnosis and process monitoring in complicated industrial processes. Neurocomputing **174** (2016) 643–650
8. Guo, X., Chen, L., Shen, C.: Hierarchical adaptive deep convolution neural network and its application to bearing fault diagnosis. Measurement **93** (2016) 490–502
9. Sun, W., Shao, S., Zhao, R., Yan, R., Zhang, X., Chen, X.: A sparse auto-encoder-based deep neural network approach for induction motor faults classification. Measurement **89** (2016) 171–178
10. Zhang, W., Peng, G., Li, C., Chen, Y., Zhang, Z.: A new deep learning model for fault diagnosis with good anti-noise and domain adaptation ability on raw vibration signals. Sensors (Basel, Switzerland) **17**(2) (2017)
11. Guo, X., Shen, C., Chen, L.: Deep fault recognizer: An integrated model to denoise and extract features for fault diagnosis in rotating machinery. Applied Sciences **7**(1) (2017) 41
12. CWRU: Case western reserve university bearing data center website
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations. (2015 http://www.robots.ox.ac.uk/ vgg/publications/2015/Simonyan15)
14. Gareth Forbes: Inner and outer race bearing fault vibration measurements
15. SpectrQuest: Spectrquest test bench