Classification of High Frequency Impact Signal in Vibrational Analysis of Spur Gearing by using Convolutional Neural Networks

Juvith Ghosh
Department of Sensors and Biomedical Technologies,
School of Electronics Engineering (SENSE),
Vellore Institute of Technology (VIT), Vellore
Tamil Nadu – 632014, India
Email: juvithghosh@gmail.com

Abstract

Spur gears are one of the widely used gears in a gearbox assembly. They often require lubrication and replacement of pinion and gears as prone to damage in high speed shafts with heavy loads and adverse working conditions. These creates spalling and breakage of gear tooth due to material fatigue from excessive loads and also forms pitting corrosion due to reduced lubrication and higher input shaft speeds. Vibrational analysis of these rolling elements is necessary for monitoring the health of gears periodically. These graphs provide a pattern waveform over time to study the characteristic high frequency impact noise signal peaks due to increased vibrations from faulty sections. This paper depicts about the implementation of convolutional neural networks to analyze the vibrational graphs obtained at different rotating speed of shafts for various gear ratios to plot the high frequency impact noise and train the neural networks to identify the peaks and classify among the faulty and healthy spur gears and pinions for a better way to reduce time in estimating the remaining average working life of gears and perform adequate maintenance of components.

Keywords: *Periodic Preventive Maintenance; Fault Analysis; Gear Fault; Convolutional Neural Networks*

1. Introduction

Gears are considered as one of the principle components of the rolling elements. They are generally used in rotatory machines for transmitting power by increasing or decreasing the speed or torque of the element they are attached with. Variety of gears have been manufactured depending upon the applications, but the spur gears are commonly used with every gearbox that have wider applications in Industries, Automotive and in Aerospace Systems [1].

Predictive maintenance of gears is necessary to check for lubrication and proper functioning to increase the remaining useful life of gearboxes and reduce noises due to vibrations. The gears are prone to damage when operating in adverse conditions of heavy axial and radial load or with improper lubrication and thus increasing the temperature that cause wear of tooth and micro-pitting [2]. Broadly, the defects of gears are classified into different types.

1.1 Spalling

It is generally called as fatigue corrosion over the tooth of pinions and gears where they are visible as removed material from the sections around the point of contact and roots of tooth. The small particles removed from the surface makes the gears prone to more friction and noise during operations causing a rattling sound [3]. It can be prevented by handling the gears with manufacturer set conditions as it is dependent on the material of gears for handling the stress limit. This type of corrosion can also be seen in gear units working at high speeds and low lubrication level for longer period of time. A spalled spot at single teeth provides an impact on the vibration analysis chart as sharp peak.

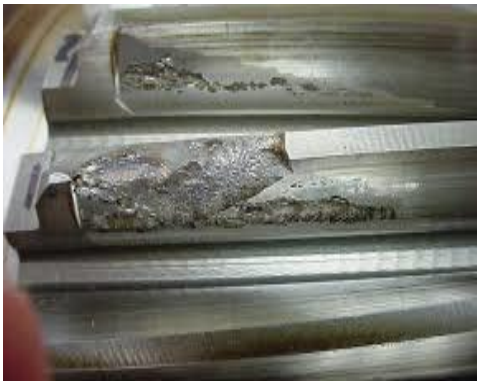


Fig.1. Spalling of helical gear teeth

1.2 Pitting

Pitting corrosion can be found at the gear surfaces as small to large holes [4] that are clearly visible. There are many causes of pitting and the commonest is the contact loss between gears due to improper shaft alignment. It also causes noise in the vibrational analysis and gives increased Hertzian Stress value when a finite element analysis is done.

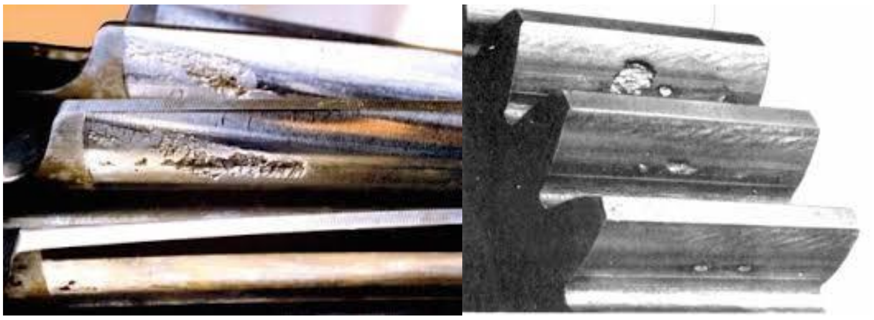


Fig.2. Crates and Pores due to pitting corrosion in spur gears

1.3 Cracks

Cracks can be formed near the roots and breakage of gear teeth may occur when the tress over the point of contact exceeds the bending stress of gear material [2,5]. It causes severe vibrational noise during operation and appear as sharp peak in analysis chart that can be clearly identified. They also occur due to high speed of operation at heavy loads and at higher temperature forming thermal stress over teeth.

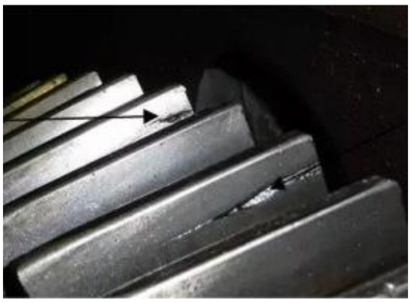


Fig.3. Teeth cracks near gear root due to low bending stress of material

1.4 Abrasive Wear

Abrasive wear can be observed as small grooves and dents over the ger surfaces and they are caused by using the low-quality lubricants containing abrasives. These abrasive materials spoil the smoothed gear surface by bringing more friction to the contact points [6] and as a result causes secondary faults like micro-pitting. It cannot be identified initially due to lesser faulty sections and noise, but a remarkable peak can be found once a secondary fault is generated over the gear surfaces.

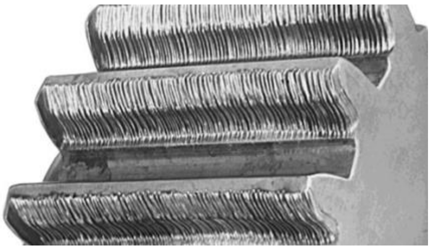


Fig.4. Fine lines over material due to abrasive particles

1.5 Corrosive Wear

This type of corrosion is caused due to the attacks on the material by the lubricant chemicals that have acidic reactions with the material and finally causes a pitting over the surfaces. They cause negligible rise of peaks in vibration analysis [7] chart and cannot be estimated using bare eyes. But severe training the neural network with plots make the system able to classify this defect.

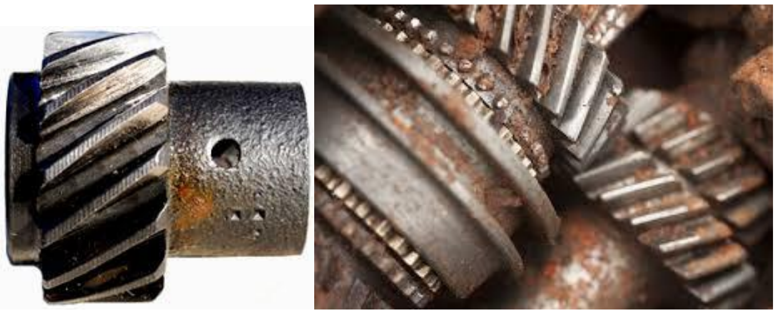


Fig.5. Brown coloration of materials due to corrosion

2. Generation of Vibration Waveform

Vibration analysis is used to detect any defects that persists with the gears of a gearbox. The experimental purpose needs data that can be obtained from vibrational sensors, plotting a frequency graph to visualize the defects. Every plot makes high amplitude spikes and spindles with a corresponding pattern associated with the type of defects. Identification of defects with the proper classification are done by studying these patterns, but during the time of emergency maintenance, the engineers and technicians faces issues due to lack of time studying the vibrational data. The research and development departments are also facing similar issues when predicting the faults of gearboxes [8]. These needs a bulky setup but by the application of signal processing, the concepts of time synchronous averaging with the spectral envelope can be a useful technique to synthetically generate vibrational waveforms for shafts rotating at fixed speed for a certain amount of time. Later on, the plot of the frequency signal can be saved as an image with the gears operating at various speeds and considering the sampling rate of signals to be constant for period of 20 seconds.

A simple arrangement was considered, where the pinion have the input shaft and output was taken from the gear of the gearbox assembly. The sinusoidal waveform for pinion, gear and gear mesh was plotted using the following equations:

$$V_p = 0.4\sin\left(2\pi f_p t\right) \tag{1}$$

$$V_g = 0.2\sin\left(2\pi f_g t\right) \tag{2}$$

$$V_m = \sin(2\pi f_m t) \tag{3}$$

Where,

V_p = Waveform of pinion
V_g = Waveform of gear
V_m = Waveform of gear mesh

The corresponding frequencies of pinion, gear and mesh can further be calculated by using the following equations:

$$f_p = R_p/60 \tag{4}$$

$$f_g = f_p \times N_p/N_g \tag{5}$$

$$f_m = f_p \times N_p \tag{6}$$

Where,

f_p = pinion input shaft frequency
f_g = gear output shaft frequency
f_m = gear mesh frequency
R_p = rotations per minute (RPM) of pinion
N_p = Number of pinion teeth

N_g = Number of gear teeth

The rotation per minute of pinion was kept at 1000 with the spur gear ratio of 2:1 where the gear having 20 teeth corresponding to 10 teeth of pinion and the sampling frequency was set for 2000 kHz to obtain the plot of signal as in figure (6) with acceleration in y-axis calculated by summing the values of equation (1) + (2) + (3) and between time in x-axis for 0.25 seconds.

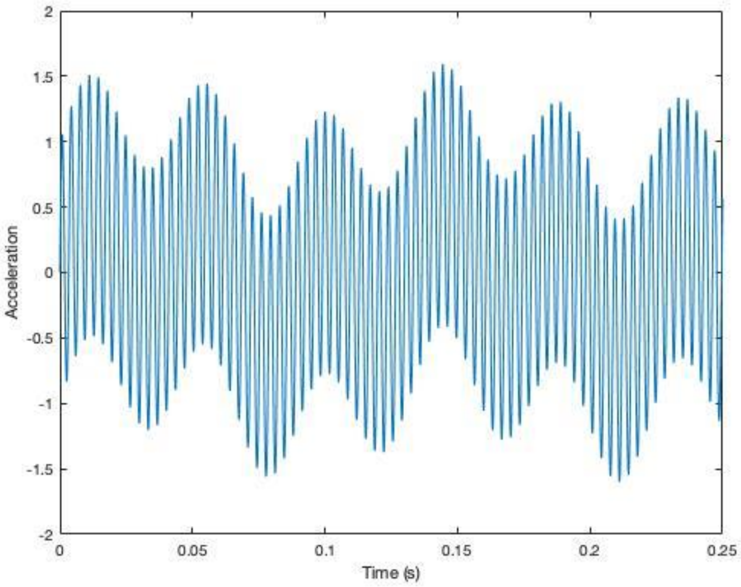


Fig.6. Gear mesh signal analysis for healthy gears of a gearbox

The high-frequency noise signal originating from the faulty location can then be compared with the generated noise signal arriving due to meshing of gear tooth by plotting both the values. Impact frequency was made periodic by convolving the periodic noise with the gear mesh frequency signal by introducing convolution function with the estimated gaussian noise.

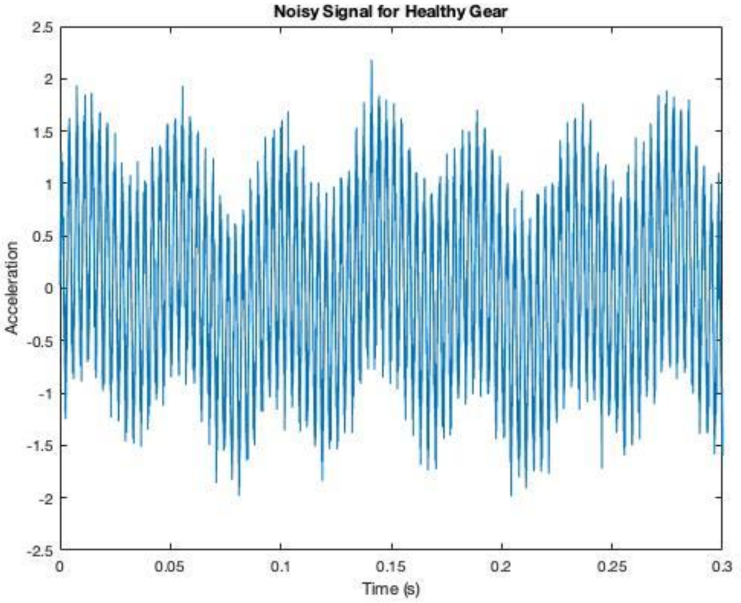


Fig.7. Noisy Signal for Healthy Gear

Figure (8) below shows the frequency plot for the gears with fault. The red marking has been indicated, to bring to notice about the high frequency spikes in the plot for comparison of healthy and faulty gears.

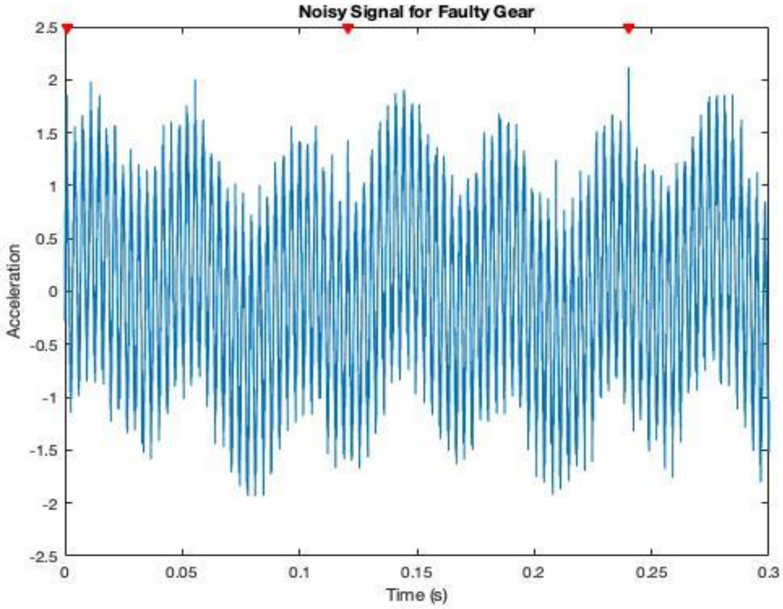


Fig.8. Noisy Signal for Faulty Gear

Table 1. Distribution of datasets on basis of pinion rotation speed

Sl. No.	Speed of pinion (in RPM)	No. of Healthy Gears	No. of Faulty Gears
01	1000	40	40
02	1500	40	40
03	2000	40	40
04	2500	40	40
05	3000	40	40

3. Methodology

Convolutional Neural Networks is the sub-branch of Neural Networks under Artificial Intelligence. This system deals with the process and techniques used by biological neurons in gaining intelligence [9] which includes learning from experience and improving the performance by estimating the mistakes during the training processes and hence the name artificial neural networks. In this experiment, we have introduced convolutional layer with the neural network that assisted the system to consider the image classification problems and solve the issues by behaving as an eye to the brain. Similarly, the simulation values that are in the form of frequency graphs are stored for training and testing with the validation datasets in the ratio of 70:15 and the rest for validating the trained network system. The stored images are then pre-processed before commencing with the training process to perform a better convolution process for increased accuracy of the system.

3.1 Pre-processing of Data

To avoid overfitting of the neural network, it is mandatory to keep the image characters like dimension and pixel resolution similar. The simulation images obtained have a dimension of 400x500x3 that causes the buffer to exceed the limit with a lengthier training time when the process is carried on a single CPU based device. As these are unstructured data, a complimentary function for Augmented Datastore is called in action. The user defines the image morphological operations before a dataset [10] is created. For this experiment, we have set the image resolution for 150x150x3 keeping the contrast of the images as original. The augmented datastore function then collects the specified number of images for training from folders and converts the image data accordingly to store in the buffer space. The remaining number of images from the buffer space are taken for validation and testing purposes.

3.2 Network Design

Convolutional Neural Networks consists of various layers that are connected to each other and states about the errors and losses encountered to every layer during training process. This assists in dividing the complex tasks into simpler as each layer gets experience about learning from preceded and succeeded layers and minimizes losses to increase accuracy. Learning is achieved both through forward and backward propagation techniques where weights and biases are balanced using layer functions. For the experimental purpose by having adequate information about the datasets and system overfitting, seven layers were added to the system.

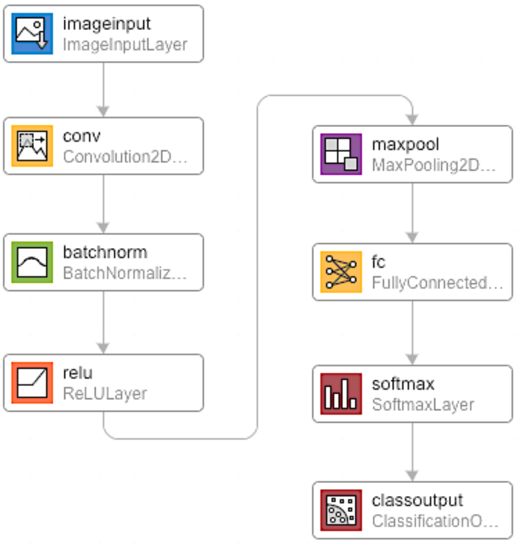


Fig.9. Block diagram showing the designed network layers

3.2.1 Convolution – 3D

Convolution function at three-dimensional level is applied to the images after they are derived from the augmented datastore blocks. This layer performs filtering of images along with padding by selecting image pixels of value and eliminating the least significant data. for the experimental purpose, the filters were hundred by value for zero padding [11]. Pixels of interest were selected as height, width and depth of the image for a better classification of the data. A stride matrix of 3x3 is selected for extracting the pixels of interest. This stride scans over the images from the image input layer and performs convolution. The output values from the layer can be achieved by implementing the equations:

$$\text{Height } h_{out} = \left[\frac{h_{in} - f + 2p}{s}\right] + 1 \tag{7}$$

$$\text{Width } w_{out} = \left[\frac{w_{in} - f + 2p}{s}\right] + 1 \tag{8}$$

Depth $d_{out} = n$, where $n = number\ of\ filters$

Where,
s= Stride length
p=Padding value and
f=filter value

3.2.2 Batch Normalization

A batch normalization function was used to speed up the training process. This function was implemented in the design soon after the convolution layer, before fetching the values to the ReLU layer for further processing. During training the convolutional neural networks, the system experiences internal covariance shift that delays the process of training when a single CPU is used for training. To overcome the issue, the batch normalization processes [12] uses back-propagation method where estimations of errors are calculated and further trains the weights and biases of the layers of network to learn about the error chances by experience and estimation from the succeeded layer.

The data is further divided into numerous batches where normalization of each batch is promoted to calculate the errors efficiently. After each batch is processed, the function trains the layers immediately to reduce chances of further errors [13]. In general normalization technique, a mean of zero and a standard deviation of unity is considered, same is applied in batches and thus the name for batch normalization. Mean of those mini batches can be given by:

$$\mu \leftarrow \frac{1}{m}\sum_{j=1}^{m} y_j \tag{9}$$

Where j =1, 2, 3, … m,
y_j = mini batches

Mini-Batch Variance (σ) can be given as:

$$\sigma \leftarrow \frac{1}{m}\sum_{j=1}^{m}(y_j-\mu)^2$$

(10)

The Normalization after taking out the mean and variance is estimated by the following formula:

$$\widehat{y_j} \leftarrow \frac{y_j-\mu}{\sqrt{\sigma^2+\epsilon}}$$

(11)

3.3.3 Rectified Linear Unit

Rectified linear Unit is commonly termed as ReLU function unit that provides a thresholding property to the data received from the above placed batch normalization layer. By applying this layer, we ensure to increase the performance of the system to some extent by eliminating the vanishing gradient [14] issues due to back propagation learning that was continued in batch normalization method. If the value of a function is less than zero, rectified linear unit provides a null output, this assists in extracting the valuable data to get processed and eliminates the values that are out of interest. The return from the function is provided to the Max Pooling layer sampling it effectively. The function can be denoted by the following equation:

$$r(y) = \begin{cases} y, y \geq 0 \\ 0, y < 0 \end{cases} \tag{12}$$

3.3.4. Max Pooling – 2D

Pooling functions are applied once the data is processed through rectified linear unit. This operation is used to down sample the data or the image size by considering the dimensions like height and width of the input image and then reduces the pixel density. Reduction in density of pixel is carried by selecting an adequate padding information and suitable stride dimensions. For the experimental purpose, the length of stride matrix was [1x1] with zero padding stand. The Max-pooling function [15] also reduces the computational time of the training as it considers maximum value of pixel density by implementing floor values for computation, thereby the reducing the chances for overfitting in the neural network. The floor value can be calculated by applying the following equation:

$$f = \frac{d_{xy}-p}{s}+1$$

(13)

Where,
f = Floor Value of function
s = Stride Dimension
p = Padding Data
d_{xy} = Dimension of Image

3.3.5 Fully Connected Layer

Fully connected layers are generally used for image classification problems where they connect the previous neural layers with each other to train the network better by decreasing the error rate. It makes each layer understand about the errors that are counted from above layers and trains the layers to gain experience by the method of Back – Propagation [16]. The output of this layer is served to the Softmax function to calculate the probability of errors at final stage.

3.3.6 Softmax Output

The output from the fully connected layer is fetched to the Softmax layer to perform a function operation that provides information about the simulated data in manner of probabilistic distribution [17]. A perfect distribution obtained by probability provides a measure of accuracy for the trained network and estimates about the errors and losses occurred while training with the datasets. The output values are stored in the form of vectors where a vector value close to unity is considered strong and weaker values are estimated to be zero count. This is one of the general criteria for multiclass probability where the function can be given as:

$$f_{so}(y) = \frac{e^{a_s(y)}}{\sum_{j=1}^{k} e^{a_j(y)}} \tag{14}$$

Where $0 \leq f_{so} \leq 1$ and $\sum_{j=1}^{k} y_j = 1$

3.3.7 Classification Layer

For a multi-class situation where the classes are mutually exclusive to each other, the Softmax output brings about cross entropy losses [18] that reduces the training time of the network. Classification layer estimates the amount of cross entropy losses occurred during the training process. This can be calculated using the equation:

$$\Delta = \sum_{i=1}^{a} \sum_{j=1}^{b} f_{ij} \ln y_{ij} \tag{15}$$

Where,
a = number of samples
b = number of classes
y_{ij} = output for class j_{th} for sample i_{th}

Table 2. Designed neural network parameters

Sl. No.	Name of Layer	Activations	Learnable
01	Image Input	150x150x3	-
02	Convolution	150x150x100	Weights 3x3x3x100 Bias 1x1x100

03	Batch Normalization	150x150x100	Offset 1x1x100 Scale 1x1x100
04	ReLU	150x150x100	-
05	Max pooling	146x146x100	-
06	Fully Connected	1x1x2	Weights 2x2131600 Bias 2x1
07	Softmax	1x1x2	-
08	Classification	-	-

3.3.8 Training and Validation

The training process is carried forward after deciding the type of solver to be used for better accuracy and lower losses to be encountered. For this experiment, we have used Adaptive Moment Estimation method, commonly known as Adams method [19]. It is one of the easiest methods to apply when the datasets consist of noises and the training needs to be done with faster processing speeds as it consumes lesser computational memory. In this method, the learning rates of mini batches are adaptive depending on the amount of data being processed that further increases the overall performance of the system. The past decaying averages and the squared values can be given as:

$$u_t = \alpha_1 u_{t-1} + (1 - \alpha_1) g_t \tag{16}$$

$$w_t = \alpha_2 w_{t-1} + (1 - \alpha_2) g_t^2 \tag{17}$$

These are the mean and variance of the gradient respectively. Now, after computing, the bias corrected values are given by the equation:

$$\hat{u} = \frac{u_t}{1 - \alpha_1^t} \tag{18}$$

$$\widehat{w} = \frac{w_t}{1 - \alpha_2^t} \tag{19}$$

The experiment was carried for 30 epochs with single iteration per epoch and a learning rate was fixed at rate of 0.001 that provided less CPU computation for efficient training.

Table 3. Distribution of datasets before training the neural network

Sl. No.	Gear type	Training (70%)	Testing (30%)	Validation (30%)
01.	Healthy	140	30	30
02.	Faulty	140	30	30

4. Results and Discussions

Image classification can be shown once after the whole training process gets over which is achieved when all the images have been passed through all the layers once per epoch. This is called as number of iterations per epoch, the value of which was set unity for this experiment. It is mandatory to observe the estimations of computing that are continuously being plotted by the software to monitor the losses and redesign the network if needed, when the accuracy of the system comes below rigorously.

Training might take certain amount of time depending on the capacity of RAM and processing power [20] of the system on which training is being carried on. Elapsed time can be reduced by using parallel computing methodology by attaching a GPU with the system by carefully selecting the hardware on basis of supported software specifications.

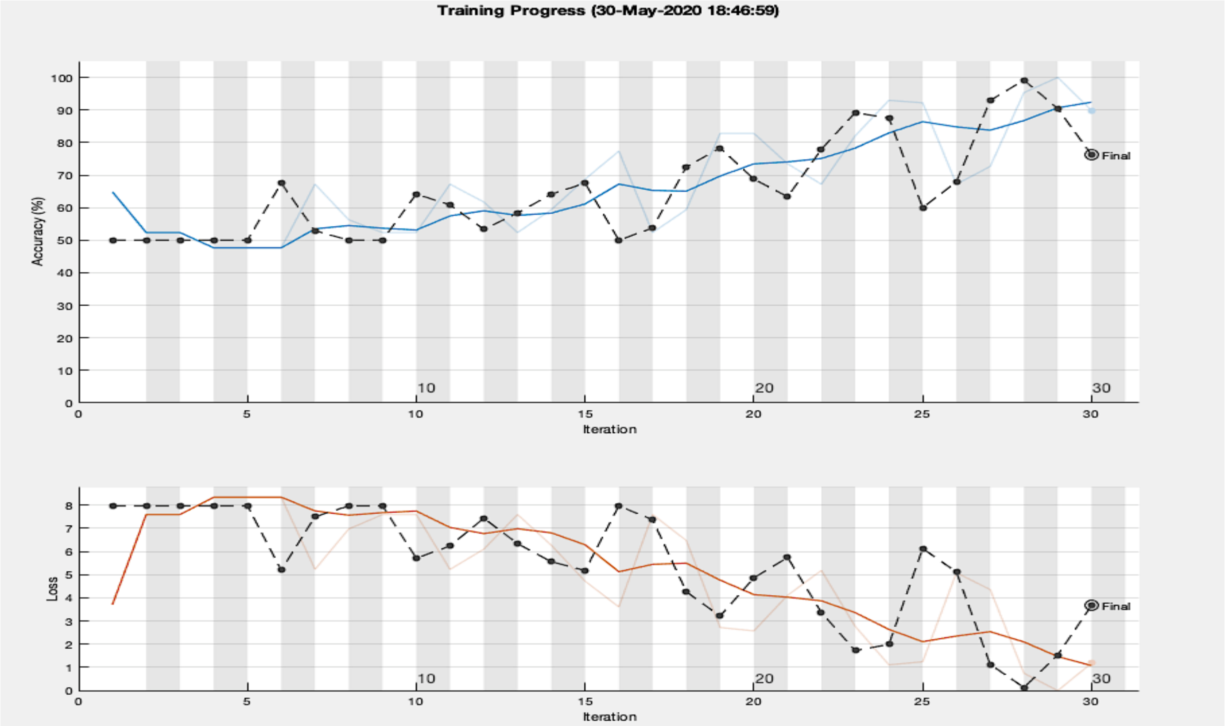
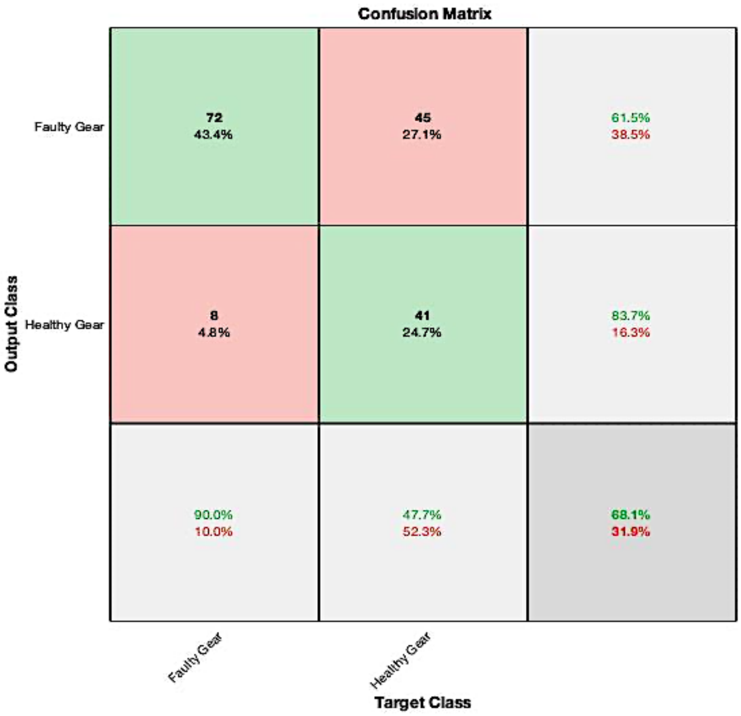


Fig.10. Training report depicting accuracy and losses

The above graph depicts that the training was carried on for total of 30 epochs with an elapsed time period of 89 minutes and 40 seconds with a learning rate of 0.001. Accuracy for validation datasets were found to be 76.25% and training accuracy was around 91% on a single CPU. Estimated loss was found to be near unity by using Adams method and it differs if some other techniques were used, depending upon the network structure and on the number of datasets. It is to be noted that as the training progresses, the losses get reduced and accuracy increases. One can estimate about the overfitting or underfitting of the network if the values differ much with steep elevations and depressions.

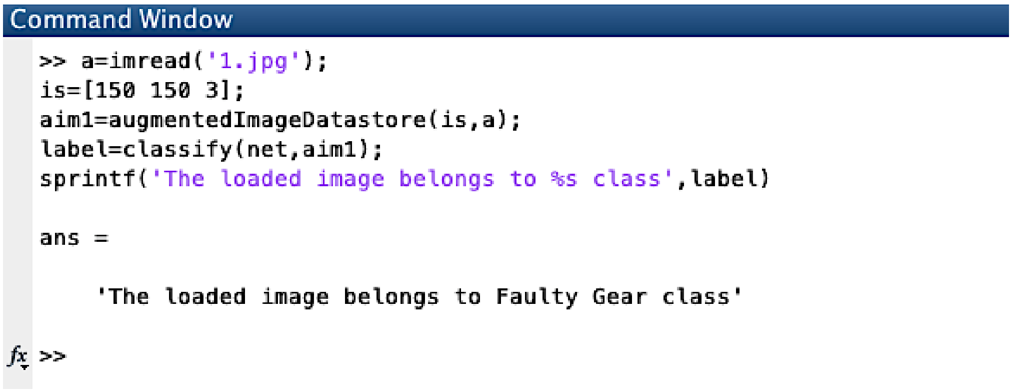


Output Class \ Target Class	Faulty Gear	Healthy Gear	
Faulty Gear	72 43.4%	45 27.1%	61.5% 38.5%
Healthy Gear	8 4.8%	41 24.7%	83.7% 16.3%
	90.0% 10.0%	47.7% 52.3%	68.1% 31.9%

Fig.11. Confusion matrix plot after training process

Confusion matrix was plotted to verify the accuracy of the system where the True values are marked by green labelling and the False values by red coloration. The numeric inside the boxes shows the number of image plots with the corresponding percentages of total datasets.

Target class about x-axis shows the required expected output and at the y-axis, the output class depicts about the output from the trained system and similarly, the x-axis and y-axis values are matched with the colors accordingly.



```
>> a=imread('1.jpg');
is=[150 150 3];
aim1=augmentedImageDatastore(is,a);
label=classify(net,aim1);
sprintf('The loaded image belongs to %s class',label)

ans =

    'The loaded image belongs to Faulty Gear class'

fx >>
```

Fig.12. Output of trained model showing results of gear class

The command window takes the image input saved as ‘1.jpg’ of a faulty gear and using augmented dataset functions, reconfigures the image to classify it with the trained neural network system defined as ‘net’ variable. Thus, it finally tells about the class of the gear after matching it with the loaded datasets to predict the healthiness of the gears.

Conclusion

The results above concluded that with the application of Convolutional Neural Networks, the predictive maintenance of gear units for various applications becomes faster. It becomes more errorless and effortless approach to confirm health of gear units by conventional use of vibrational analysis techniques. This approach makes lesser human interventions in determining the high frequency noise signal spikes and classify between healthy and faulty gears which further accelerates the research process of R&D Departments as well for the maintenance sections.

Acknowledgement

It would be an honorable moment for me to proudly acknowledge Mr. Subhadip Das of SAIL – Durgapur Steel Plant for his guidance and kind motivations to develop systems involving Artificial Intelligence and application in practical field to ease the workload of Research and Development departments in production of mechanical components.

References

[1] M. Watson, J. Sheldon, S. Amin, H. Lee, C. Byington, M. Begin, A Comprehensive High Frequency Vibration Monitoring System for Incipient Fault Detection and isolation of Gears, Bearings and Shafts/Couplings in Turbine Engines and Accessories, in: Volume 5: Turbo Expo 2007, ASMEDC, 2007. https://doi.org/10.1115/gt2007-27660.

[2] D.C.H. Yang, J.Y. Lin, Hertzian Damping, Tooth Friction and Bending Elasticity in Gear Impact Dynamics, Journal of Mechanisms, Transmissions, and Automation in Design. 109 (1987) 189-196. https://doi.org/10.1115/1.3267437

[3] S.Theodossiades, M. Gnanakumarr, H. Rahnejat, M. Menday, Mode identification in impact-induced high-frequency vehicular driveline vibrations using an elasto-multi-body dynamics approach, Proceedings of the Institution of Mechanical Engineers, Part K: Jpurnal of Multi-Body Dynamics. 218 (2004) 81-94. https://doi.org/10.1243/146441904323074549.

[4] F.K. Choy, V. Polyshchuk, J.J. Zakrajsek, R.F. Handschuh, D.P. Townsend, Analysis of the effects of surface pitting and wear on the vibration of a gear transmission system, Tribology International. 29 (1996) 77-83. https://doi.org/10.1016/0301-679x(95)00037-5.

[5] G. DALPIAZ, A. RIVOLA, R. RUBINI, EFFECTIVENESS AND SENSITIVITY OF VIBRATION PROCESSING TECHNIQUES FOR LOCAL FAULT DETECTION IN GEARS, Mechanical Systems and Signal Processing. 14 (2000) 387-412. https://doi.org/10.1006/mssp.1999.1294

[6] M. Izciler, M. Tabur, Abrasive wear behavior of different case depth gas carburization AISI 8620 gear steel, Wear. 260 (2006) 90-98. https://doi.org/10.1016/j.wear.2004.12.034

[7] I.B. Goldman, Corrosive wear as a failure mode in lubricated gear contacts, Wear. 14 (1969) 431-444. https://doi.org/10.1016/0043-1648(69)90006-4

[8] Vara Prasad, P. (2015). Detection of Gear Fault Using Vibration Analysis. International Journal of Research in Engineering and Science (IJRES). 3. 2320-9356.

[9] V. Sze, Y.H. Chen, T.J. Yang, J.S. Emer, Efficient Processing of Deep neural Networks: A Tutorial and Survey, Proceedings of the IEEE. 105 (2017) 2295-2329. https://doi.org/10.1109/jproc.2017.2761740.

[10] [Dataset] Juvith Ghosh, 2020. https://www.kaggle.com/dataset/889ac6ca70a0f06f52268b59e7964ead95a9c2a55a040ab1799194a443b00801

[11] Nikhil B., Nicholas L., 2017. Fundamentals of Deep Learning, first ed. United States of America, O.Reilly.

[12] Batch normalization in Neural Networks: https://towardsdatascience.com/batch-normalization-in-neural-networks-1ac91516821c.

[13] Batch Normalization – Speed up Neural Network Training, Ilango R – Medium: https://medium.com/@ilango100/batch-normalization-speed-up-neural-network-training-245e39a62f85.

[14] Mark Hudson B., Martin T.H., Howard B.D., Deep Learning Toolbox – User's Guide, MATLAB 2018b. https://www.mathworks.com/products/deep-learning.html

[15] What are Max Pooling, Chris – MACHINECURVE: https://www.machinecurve.com/index.php/2020/01/30/what-are-pooling-average-pooling-global-max-pooling-and-global-average-pooling/

[16] Fully Connected Layer: The brute force layer of a Machine Learning model, Surya Pratap Singh, OpenGenus IQ: https://iq.opengenus.org/fully-connected-layer/

[17] Softmax Layer by DeepAI: https://deepai.org/machine-learning-glossary-and-terms/softmax-layer

[18] Phil K. 2017. MATLAB Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence, APress.

[19] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization, International Conference on Learning Representations.

[20] Vincent V., Andrew S., Mark Z.M., 2011. Improving the speed of neural networks on CPUs. Deep Learning and Unsupervised Feature Learning Workshop, NIPS, California.