Fault detection in Railway Track using Artificial Neural Network

S.Santhalakshmi
Francis Xavier Engineering college,
Computer science & Engineering,
ravinasantha@gmail.com
9629094684

M.Santhana Priya
Francis Xavier Engineering college,
Computer science & Engineering
santhyamaharajan@gmail.com
8610277692

Dr. I.Jeena Jacob M.E Phd.,
Francis Xavier Engineering college,
Computer Science & Engineering,
9894083125
ijeenajacob@gmail.com

Abstract-Faults in railway tracks cause majority of the train accidents. Images of railway track will be captured by camera which will be provided as an input to the system consisting of neural network. And hence proper measures can be taken by the authorities. If a fault arises in the track section, the system will automatically detect the fault and necessary actions can be taken to avoid any miss happening.

I.INTRODUCTION

We found that using Image processing and putting that as an input to the neural network is more efficient and productive than other methods. Using Long Short Term Memory (LSTM) Neural Network is a better option than Convolution Neural Network (CNN)[1] . While the LSTM network outperforms the convolution network for the track circuit case, the convolution networks are easier to train. Sensors are often located away from energy supplies and hence require either batteries or some form of energy supply to power them. If there are errors in transmission across WSN, then data may be missing. The claim in[2] contradicts itself as it needs to minimize the energy consumption but maximize communication efficiency. For this proper arrangement will be need to be made. The images may be blurred and/or the track may not be visible [3]. It is even possible that two cameras capture the same crack. We need to analyze the proper distance of the cameras from one another and the track so as to avoid the incorrect data and get proper output.

Track condition and presence of cracks in tracks play an important factor in the derailment of the trains. Manual checking of the tracks takes up a huge amount of time and resources. Also it is open to human error. To address this issue of derailment and ensure proper investigation. It tracks, an automated system is needed. The scope of the paper is to develop a robust neural network and train it using Backtracking Algorithm which will compute on the input images of tracks and give an output whether the image is cracked or not.

II.MECHANISM FOR FAULT DETECTION

Back propagation[9-12] is used in artificial neural networks to calculate a gradient and their weights used to train deep neural networks, with more than one hidden layer. This paper proposes a proper method to detect cracks in railway tracks using Image Processing and Neural Networks. The input images of the tracks go through a series of Image Processing functions, discussed later, to the Neural Networks. The Network then computes whether the track is cracked or not.

In the context of leaning, back propagation is used by gradient descent optimization algorithm to adjust the weight of neurons by calculating the gradient of the loss function. This technique is called as backward propagation of errors, and the error is calculated at the output and distributed back through the network layers. The back propagation algorithm has been repeatedly rediscovered and is equivalent to automatic differentiation in reverse accumulation model.

Back propagation requires a known, desired output for each input value—it is therefore considered to be a supervised learning method . Back propagation is a generalization of the delta rule to multi-layered feed forward networks, using the chain rule to iteratively compute gradients for each layer. It is closely related to the Gauss–Newton algorithm, and is part of continuing research in neural back propagation. It can be used with any gradient-based optimizer, such as L-BFGS or truncated Newton.

To implement the proposed algorithm, explicit formulas are required for the gradient of the function

```
{\displaystyle w\mapsto E(f_{N}(w,x),y)}
```

```
{\displaystyle w\mapsto E(f_{N}(w,x),y)}
```

where the function is

```
{\displaystyle E(y,y')=|y-y'|^{2}}
```

```
{\displaystyle E(y,y')=|y-y'|^{2}}.
```

The learning algorithm can be divided into two phases: propogation and weight update to generate output

Value(s) , Calculation of the cost Propagation of the output activations back through the network using the training pattern target in order to generate the deltas of all output and hidden neurons. For each weight, the steps must be followed: The weight's output delta and input activation are multiplied to find the gradient of the weight.

A ratio (percentage) of the weight's gradient is subtracted from the weight. This ratio gives the speed and quality of learning; called as the learning rate. The greater the ratio, the faster the neuron trains, lower the ratio, the more accurate the training is. The sign of the gradient weight indicates whether the error varies directly with, or inversely to, the weight. Therefore, the weight must be updated in the opposite direction, "descending" the gradient. Learning is repeated until the network performs adequately.

The following is pseudo code for a stochastic gradient descent algorithm for training a three-layer network (one hidden layer):

```
initialize network weights (small random values) do forEach training example named ex prediction = neural-net-output(network, ex) // forward pass actual = teacher-output(ex) compute error (prediction - actual) at the output units compute {\displaystyle \Delta w_{h}} \Delta w_h for all weights from hidden layer to output layer // backward pass compute {\displaystyle \Delta w_{i}} \Delta w_i for all weights from input layer to hidden layer
  // backward pass continued update network weights // input layer not modified by error estimate until all examples classified correctly or another stopping criterion satisfied return the network
```

The lines labeled "backward pass" can be implemented using the back propagation algorithm, which calculates the gradient of the error of the network regarding the network's modifiable weights.

III. EXISTING APPROACH

In India this process of checking the crack in railway track is still manually happening. This manual process will have many disadvantages. Railway department will provide well trained worker for this detection process, but sometimes worker cannot detect the missing crack because of his carelessness and many issues will be created. Sometimes because of field survey, in busy track, there is a chance of causing accident by hitting train. To avoid these types of incidents there is a need for automatic railway track defect detection system.

The first existing work addresses the problem of fault detection and isolation in railway track circuits. A track circuit considered as a large-scale system composed of a series of trimming capacitors located a transmitter and a receiver. A defective capacitor affects not only its own inspection data but also the measurements related to all capacitors located downstream .The outputs from local decision tree classifiers are expressed using the Dempster–Shafer theory and combined to make a final decision on the detection and localization of a fault in the system.

The ability of detecting imminent faults, it will be possible to prevent any breakdowns via preventive maintenance, which leads to improve Reliability and Accessibility and functional lifetime expansion of the system. In this paper, author employed Neuro-Fuzzy Network (NFN) for fault detection and diagnostics in a typical audio frequency track circuit, which is a combination of knowledge-based and data-based systems. It has the merits of both fuzzy systems and neural networks .Healthy and faulty data have been used for training the algorithm. When put to work, occurrence of fault modes or their imminence is detected and localized with a good precision by the algorithm.

IV.PROPOSED APPROACH AND SYNTHETIC DATASET

In photography, computing, and colorimetry, a gray scale or grey scale image the value of each pixel is a single sample. Images of this sort, are known as black-and-white or monochrome, are composed exclusively of shades of gray, varying from black at the weakest intensity to white at the strongest. Gray scale images are one-bit bi-tonal black-and-white images. Imaging are images with only two colors, black and white (binary images). Gray scale images have many shades of gray in between. It is the result of measuring the intensity of light at each pixel according to a particular weighted combination of frequencies or wavelengths, and in such cases they are monochromatic proper when only a single frequency (a narrow band of frequencies) is captured. The railway track images are converted to gray scale using following pseudo code.

ConversionFactor = 255 / (NumberOfShades - 1)

AverageValue = (Red + Green + Blue) / 3

Gray = Integer((AverageValue / ConversionFactor) + 0.5) * ConversionFactor

NumberOfShades is a value between 2 and 256.

Technically, any grayscale algorithm could be used to calculate AverageValue; it simply provides an initial gray value estimate -the "+ 0.5" addition is an optional parameter that imitates rounding the value of an integer.

A co-occurrence matrix or co-occurrence distribution matrix is defined over an image to be the distribution of co-occurring pixel values at a given offset. Another method that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM), also known as the gray-level spatial dependence matrix.

As there was no data set available for this particular work we have built our own data set from the scratch with the help of GLCM technique. In mat lab , gray co matrix(I) creates a gray level co-occurrence matrix (GLCM) from image I. It is also called as gray-level co-occurrence matrix or gray-level spatial dependence matrix. Also, the word co-occurrence is frequently used in the literature without a hyphen, co occurrence. gray co matrix creates the GLCM by calculating how often a pixel with gray-level (gray scale intensity) value i occurs horizontally adjacent to a pixel with the value j. (You can specify other pixel spatial relationships using the 'Offsets' parameter.) Each element (i,j) in glcm specifies the number of times that the pixel with value i occurred horizontally adjacent to a pixel with value j.

glcms = gray co matrix(I, Name, Value, ...) returns one or more gray-level co-occurrence matrices, depending on the values of the optional name/value pairs. Parameter names can be abbreviated, and case does not matter. [glcms, SI] = gray co matrix(___) returns the scaled image, SI, used to calculate the gray-level co-occurrence matrix. The values in SI are between 1 and Num Levels.

Using GLCM we created an array of 8*8 matrix then we used texture analysis function of MAT LAB to find the entropy of gray scale image using :- " e = entropy(I)"

Where, I is input image which is one of the features which we extracted from the image. Similarly, we obtained properties of gray-level co-occurrence matrix from 'gray co props' gray co props(glcm,properties) calculates the statistics specified in properties from the gray-level co-occurrence matrix glcm. glcm is an m-by-n-by-p array of valid gray-level co-occurrence matrices. glcm is an array of statistics for each glcm.

gray co props normalizes the gray-level co-occurrence matrix (GLCM) so that the sum of its elements is equal to 1. Each element (r,c) in the normalized GLCM is the joint probability occurrence of pixel pairs that defines spatial relationship having gray level values r and c in the image. gray co props uses the normalized GLCM to calculate properties. Which is used using command:- "s = gray co props(glcm)" The features which we obtained from above were:- Entropy, Contrast, Correlation, Energy, and Homogeneity.

Output was set to 1 for image having cracked rails and 0 for no cracks. This data was stored in an Excel Sheet for multiple images and this Excel sheet shown in figure was used as an data set for training Neural Network. The diagram shows the database sheet containing the extracted features of the images for the data set.

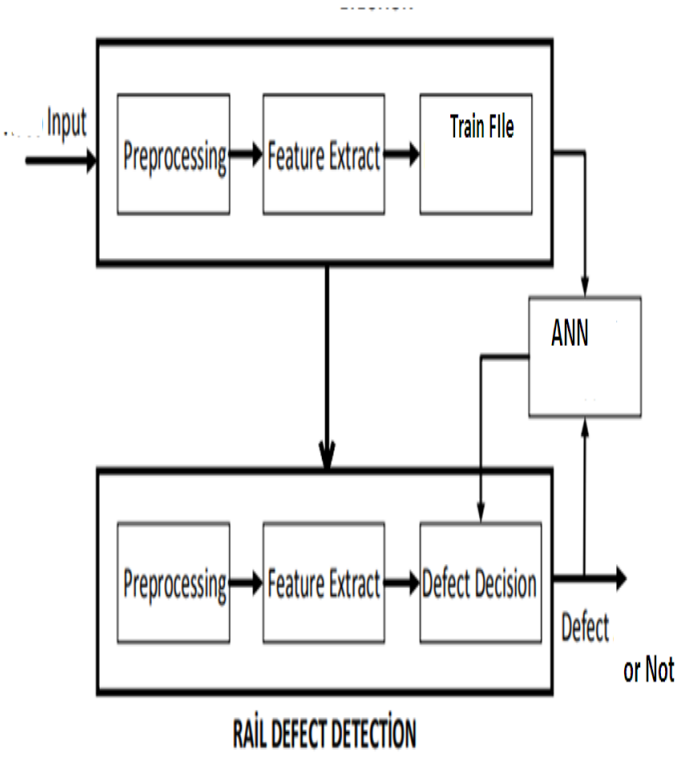


Figure 1: Proposed neural network based model for fault detection in railway tracks.

V.EXPERIMENTAL RESULTS

The neural network structure shown in figure 2 will have 5 inputs which are: Entropy, Contrast, Correlation, Energy, and Homogeneity. Figure gives us an idea of how the features(Extracted from input images) are given as an input to the neural network and absolute output is provided. A hidden layer which will be trained on the basis of dataset created and absolute output where, 1 will represent crack in the railway track and 0 will represent no crack.

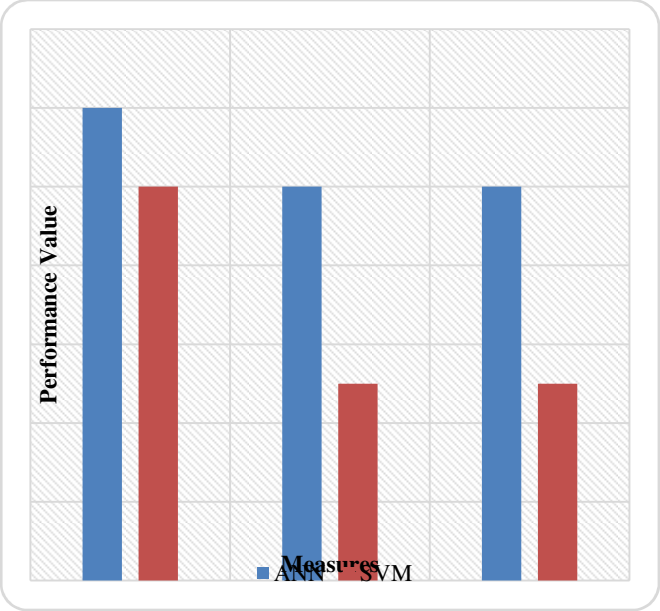


Figure 2: Created performance value for fault detection in railway tracks.

Figure 3 is the input image of a cracked track. This image is input to the Neural Network where the functions are performed on the image. This image is obtained from the cameras placed near the tracks. The image is converted to a Gray Scale Image. This is necessary for Feature Extraction since it cannot be performed on RGB image. It is done with the help of Image Processing Toolbox.

Figure 3: Input image and input image converted to grayscale.

Cropping function is used to crop the crack out of the track image. It is used to train the Neural Network. The features are extracted for matching and training the Neural Network. These specify the 100 strongest features of the image. Figure 4 shows the cropped image and marked features.

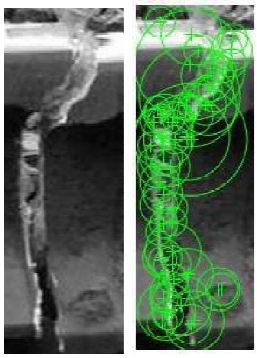


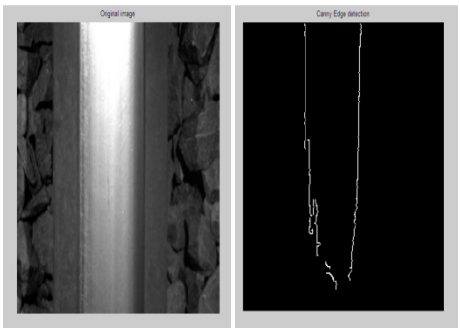
Figure 4: Cropped image and strongest features of crack image.

Feature matching of input image and stored image.

The features of the cropped image and input image are matched. Outlier and Inliers are matched with each other. The Outliers or the un required points are casted aside. Only the inliers are matched. Based on the matched points, the crack is detected in the input image if present. A box encircles the crack. Neural Network will go a step further and directly give us an absolute output of whether the crack is present or not and thus classify the images in two different stacks.

Crack detected in railway track.

In this system an image of railway track is given as an input to find whether there is fault or not. The features are **extracted of** that input image using same GLCM technique. These features are extracted from input image, will be given as input to the neural network i.e., entropy, contrast , correlation , energy , homogeneity and with the help of hidden layer which is trained using dataset absolute output will be generated. 1 will represent crack and 0 will represent no crack. In this way the neural network is used to detect crack in railway track.



VI. CONCLUSION

.

. The main intention of this work was to create a more feasible and robust method for the detection of the cracks on the railway tracks. We decided to go with Image processing since cameras are present everywhere and a single camera can cover a lot of track. Also there is no manual labor and the trains can work without any disruption even during the inspection. The cameras can inspect the tracks for 24 hours every day without disturbing anything since they don't need to be placed on the tracks or such. In this system, a method to detect cracks using image processing techniques. This technique replaces manual inspection by automatic . A video camera can be installed in each sections of the track to take images and provide as an input to the suggested system to detect any cracks in the track section. This will help to detect cracks immediately and reduce the possibilities of any miss happening. Since the system would be automatic and will require less manual intervention, the utmost efficiency of the system can be ensured.The future work of this paper is to include various other parameters like Metal Quality, presence of Water or Snow above accepted levels, etc of the The proposed approach provides much faster and reliable inspection of the tracks. The input image from a camera of the track is given to the Neural Network which is trained using the dataset of cracked tracks images. The neural network then performs the operations on it and gives us an absolute output of whether the track is cracked or not. Two separate data structures are formed to directly classify the cracked and non cracked images.Track and provide a more diverse output about the quality and usability of the track.

REFERENCES

[1] J. Chen, C. Roberts, and P. Weston, "Fault detection and diagnosis for railway track circuits using neuro-fuzzy systems," Control Eng. Pract., vol. 16, no. 5, pp. 585–596, 2008.

[2] K. Verbert, B. De Schutter, and R. Babuška, "Exploiting spatial and temporal dependencies to enhance fault diagnosis: Application to railway track circuits," in Proc. Eur. Control Conf., Linz, Austria, Jul. 2015, pp. 3047–3052.

[3] S. Ntalampiras, "Fault identification in distributed sensor networks based on universal probabilistic modeling," IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 9, pp. 1939–1949, Sep. 2015.

[4] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun. (2015). "Deep image: Scaling up image recognition." [Online]. Available: http://arxiv.org/abs/1501.02876

[5] L. Oukhellou, A. Debiolles, T. Denoeux, and P. Aknin, "Fault diagnosis in railway track circuits using Dempster–Shafer classifier fusion," Eng. Appl. Artif. Intell., vol. 23, no. 1, pp. 117–128, 2010.

[6] L. Oukhellou, A. Debiolles, T. Denoeux, and P. Aknin, "Fault diagnosis in railway track circuits using Dempster–Shafer classifier fusion," Eng. Appl. Artif. Intell., vol. 23, no. 1, pp. 117–128, 2010.

[7] Z. L. Cherfi, L. Oukhellou, E. Côme, T. Denoeux, and P. Aknin, "Partially supervised independent factor analysis using soft labels elicited from multiple experts: Application to railway track circuit diagnosis," Soft Comput., vol. 16, no. 5, pp. 741–754, 2012.

[8] M. A. Sandidzadeh and M. Dehghani, "Intelligent condition monitoring of railway signaling in train detection subsystems," J. Intell. Fuzzy Syst., Appl. Eng. Technol., vol. 24, no. 4, pp. 859–869, 2013.

[9] S. Sun and H. Zhao, "Fault diagnosis in railway track circuits using support vector machines," in Proc. 12th Int. Conf. Mach. Learn. Appl., vol. 2. Miami, FL, USA, Dec. 2013, pp. 345–350.

[10] Z. Lin-Hai, W. Jian-Ping, and R. Yi-Kui, "Fault diagnosis for track circuit using AOK-TFRs and AGA," Control Eng. Pract., vol. 20, no. 12, pp. 1270–1280, 2012.

[11] S. Ntalampiras, "Fault identification in distributed sensor networks based on universal probabilistic modeling," IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 9, pp. 1939–1949, Sep. 2015.

[12] M. M. Gardner et al., "Equipment fault detection using spatial signatures," IEEE Trans. Compon., Packag., Manuf. Technol. C, vol. 20, no. 4, pp. 295–304, Oct. 1997.