KSII Transactions on Internet and Information Systems

Structural Health Monitoring of railway tracks using IoT-based Multi-robot system

Journal:	*KSII Transactions on Internet and Information Systems*
Manuscript ID	TIIS-IA-2019-Mar-0244
Manuscript Type:	IoT & Applications
Date Submitted by the Author:	11-Mar-2019
Complete List of Authors:	T, Velmurugan Iyer, Srikrishna Mohankumar, Shashank galla, Hari S, Nandakumar
Keywords of your Paper:	Wireless Networks, LEACH, deep learning

Structural Health Monitoring of railway tracks using IoT-based Multi-robot system

Srikrishna Iyer[1], Hari Galla[2], Shashank Mohankumar[3], Velmurugan T[4] and Nandakumar S[5]

[1] School of Electronics Engineering
VIT University, Vellore, India
[e-mail : srikrishna.iyer2015@vit.ac.in]
[2] School of Electronics Engineering
VIT University, Vellore, India
[e-mail : hari.galla2015@vit.ac.in]
[3] School of Electronics Engineering
VIT University, Vellore, India
[e-mail : shashank.mohankumar2015@vit.ac.in]
[4] School of Electronics Engineering
VIT University, Vellore, India
[e-mail : tvelmurugan@vit.ac.in]
[5] School of Electronics Engineering
VIT University, Vellore, India
[e-mail :snandakumar@vit.ac.in]

Abstract

In a country whose railway system is more than a century old and used by millions of its inhabitants on a daily basis; it is essential to detect any deformities caused to the tracks in a more effective way without overly relying on manual inspection. This paper presents a way to detect the impairments and malformations autonomously using a multi-robot system and communicating them in real-time to the base stations. A hardware prototype is designed to implement a master-slave robot system capable of detecting rail surface defects namely, fractures, squats, corrugations and rust. The system incorporates ultrasonic sensors coupled with image processing using OpenCV and deep learning to classify a given fault that is detected. To eliminate manual inspection, the location and status of the fault can be conveyed to a central location enabling immediate attention by utilizing GSM, GPS and cloud-based technologies. The system is extended to a multi-robot framework designed to optimize energy utilization and the lifetime of individual robots. Thus, the LEACH protocol is implemented using a MATLAB simulation of 100 robot nodes and the corresponding performance metrics were obtained.

Keywords: Multi-robot system, rail surface defects, fractures, squats, corrugations, rust, image processing, deep learning, LEACH protocol.

1. Introduction

Indian railways provide a major mode of transportation for more than 25 million people every single day. Now the world's third largest rail network, structurally monitoring railway tracks have become an integral part of avoiding railway accidents and loss of life. While the number of railway mishaps has reduced over the last 10 years, the death toll due to the derailment has risen significantly. In 2016-17, 193 people died in these accidents out of which 78 of these accidents occurred due to derailment; bringing the count to an all-time high in the past decade. According to a study conducted, in the six-year period between 2009 and 2015, there were a total of 803 accidents in Indian Railways killing 620 people and injuring 1855 people. 47% of these accidents were due to derailment of trains making it the primary cause for major train accidents in India over the past decade, and this number has only worsened over the last couple of years where almost 53 percent of 586 train accidents that occurred were due to derailments. Therefore, we can say that these issues need our utmost attention. The main problem is caused due to old and antecedent tracks. The old pieces of steel and aluminum develop deposits of rust and are prone to cracks. Due to this, the rate of railway-accidents is higher in our country as compared to other countries in the world. The system relies mostly on excessive manpower and its development which leaves them behind in the race with the ongoing technological revolution. We need an efficient and effective way to handle this situation without using much manpower and make the system more autonomous [1-3]. Currently, the fault detection techniques are largely carried out by manual inspection, which is cumbersome and obsolete. This system of fault-checking requires plenty of manual labor, time and is highly inefficient. The dependence of this system on human judgment and external factors such as ambient light raises quite a few red flags due to a high likelihood of misdiagnosing or inadvertently ignoring some errors, making the system unreliable and in need of a desperate fix. The current research mainly aims to tackle the issues related to surface defaults such as peeling, deep fractures, folds and general deformities in the railway track [4].

Fig. 1. Rail surface corrugations, crack, squats and rust (From left to right)

Earlier work included detecting flaws using an ultrasonic testing method. This technique was effective in detecting minor cracks [5]. An autonomous vehicle using a PIC microcontroller with obstacle sensors setup was later proposed. This vehicle was capable of monitoring the location of the crack using a GPS module and sending text alerts using the GSM module [6]. Recently, researches have focused on machine vision system that use image-processing to detect defects in the morphology of the railway tracks [7]. Most of the research currently is concentrating on the overall accuracy of the system, but for utilization in real-world scenarios,

we need to shorten the computational time of the algorithm hence making the system robust and useable in the field [8]. Energy efficient communication protocol called LEACH was proposed for distributed wireless sensor networks which focused on optimizing energy consumption and throughput by dividing a large network of nodes into clusters [9]. To validate the performance of a LEACH network, several performance measures have been simulated and compared to an improved clustering algorithm namely stable election protocol (SEP) [10]. A Cluster head gateway switch routing (CGSR) protocol is proposed which describes a multi-robot communication system to coordinate multi-robot nodes through a cloud server [13]. In this paper, we have proposed a hardware implementation of a two-robot system; a master and a slave to structural monitor and detect faults as shown in **Fig. 1** on a railway track. RF communication protocol is used to coordinate the movement between the two bots i.e. the slave is designed to mimic the master bot movement. These bots consist of Ultrasonic sensors for detecting deep fractures, squats and corrugations. Also, the sensors are coupled with a pi camera to capture images to validate the presence of faults which include fractures, squats and rust. Such systems are essential for brisk, accurate and inexpensive railway track monitoring systems. Furthermore, we extended this solution by simulating a multi-robot environment using the LEACH ("Low Energy adaptive clustering") protocol - essentially a TDMA-based MAC protocol which helps us to minimize energy consumption while routing the information to the base station. The protocol was tested using MATLAB and conclusive results based on certain performance metrics were obtained to determine suitable network parameters.

2. Proposed Multi-Robot System

The Indian railway is one of the largest networks in the world spanning a track length of approximately 67,000 Km. A multi-robot system is proposed to establish a reliable distributed system that allows us to monitor the status and location of bots that are deployed. Hence, providing the flexibility to a random deployment of bots capable of identifying surface defects on railway tracks by communicating through a well-established routing mechanism. This mechanism is implemented using the Low Energy Adaptive Clustering Hierarchy (LEACH) protocol. Unlike multiple frameworks for each robot that makes the use of an IOT server, a multi-robot system incorporating the LEACH protocol can utilize a remote server simultaneously. In this paper, a framework is introduced for an IOT based cloud server for communication and coordination among robots. To reduce computation time, enhance battery life and improve bandwidth utilization, the robots are organized in pre-determined clusters. For every round, the cluster elects its own cluster head due to which the load is well distributed among the bots. The bot communicates to the closest cluster head for uploading real-time data onto a cloud server. Hence, reducing communication and computational costs. Since only the cluster head can communicate to the server, the LEACH protocol uniformly distributes energy utilization for each robot in a cluster. A wireless network implementing the LEACH protocol consists of two types of nodes namely, advanced and normal nodes. Due to the uneven distribution of energy, advanced nodes are considered to consume more energy than a normal node by a factor α. Consider a length of railway track required to be monitored, then three clusters and their respective cluster heads in a wireless robot network can be depicted as:

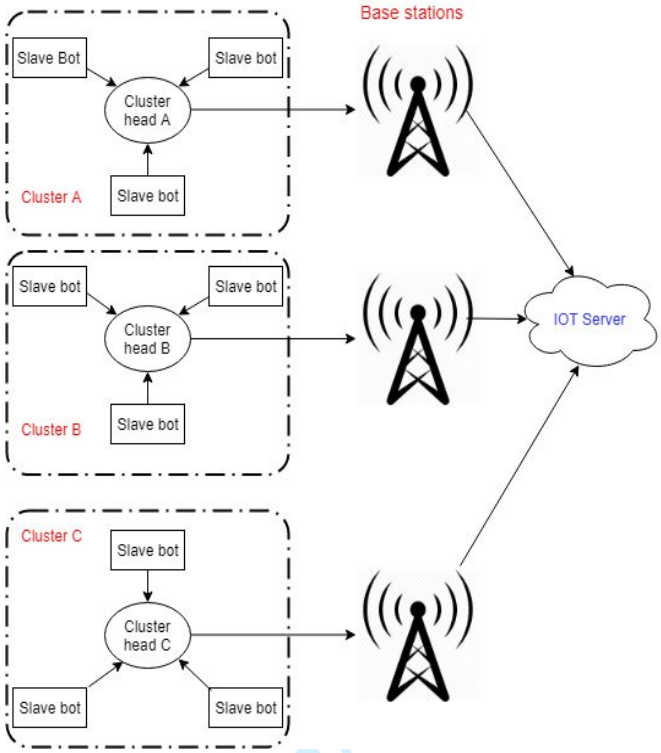

Fig. 2. Framework for a multi-robot wireless network system

Assuming that robot nodes are homogenous i.e. all nodes deployed have the same initial energy, the LEACH protocol ensures that each robot node becomes a cluster head exactly once $1/P_o$ iterations. Where P_o refers to the optimal probability that a robot node becomes a cluster head. The remaining nodes not elected as a cluster head belong to set G and the probability of a node in G to be elected as a cluster head increases steadily after every iteration. The process of selecting a cluster head begins by randomly selecting a number in the range [0, 1]. A node g ϵ G becomes a cluster head based on the following condition [9]:

$$CH = \begin{cases} Cluster\ head\ K, & t < T(g) \\ Robot\ node\ i \in G, & t \geq T(g) \end{cases} \quad (1)$$

Here T(g) is defined as:

$$T(g) = \frac{P_o}{1 - P_o \cdot \left(i \,.\, mod\left({}^{1}/_{P_o} \right) \right)} \quad (2)$$

Where i is current iteration number.

2.1 LEACH protocol: optimal clustering

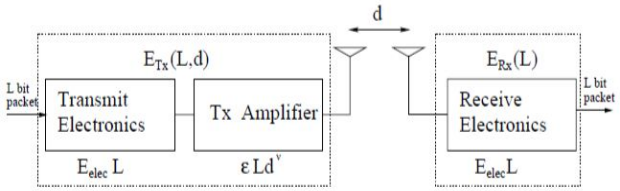

Fig. 3. 1st Order radio energy model

Optimal clustering allows uniform distribution of energy degradation for all robots, thereby minimizing total energy consumption by the entire multi-robot system. The energy dissipation model is described in **Fig. 3** [10]. The energy released by each robot system in a cluster of areaYxY, transmitting data packets of size L over a distance D given by:

$$E_{transmitter} = \begin{cases} L.E_d + L.\epsilon_{fs}.D^2, & D \le D_0 \\ L.E_d + L.\epsilon_{mp}.D^4, & D > D_0 \end{cases} \quad (3)$$

Where $E_d$is the dissipated energy per bit of a transmitter circuit on a robot node. ϵ_{fs}, ϵ_{mp} are constants whose values depend on type transmitter amplifier used in the circuit. D is the distance between a robot node and a cluster head and $D_0 = \sqrt{\epsilon_{fs}/\epsilon_{mp}}$. Assuming that distance $D \le D_0$, then energy utilized by a cluster head for one iteration is given as:

$$E_{clusterhead} = \left(n/C - 1\right)L.E_d + \frac{n}{C}L.E_{data} + L.E_d + L.\epsilon_{fs}.D^2 \quad (4)$$

Where n is the number of nodes in a cluster where C is the number of clusters. The energy dissipated by a robot non-cluster head is given as:

$$E_{non-clusterhead} = L.E_d + L.\epsilon_{fs}.{d_{NCH}}^2 \quad (5)$$

Finally, energy utilized by a cluster can be approximated as:

$$E_{total} = E_{clusterhead} + \frac{n}{C}.E_{non-clusterhead} \quad (6)$$

Hence, E_{total} is:

$$E_{total} = L[2nE_d + nE_{data} + \epsilon_{fs}(C.{d_{NCH}}^2 + n{d_{CHS}}^2)] \quad (7)$$

Where d_{NCH} is the average distance between cluster member and the cluster head robot and d_{CHS} is average distance between cluster head and sink. By differentiating (7) with respect to C and equating to zero, we get the optimal number of clusters CH_{opt}.

$$CH_{opt} = \sqrt{\frac{n}{2\pi}}\frac{Y}{d_{CHS}} = \sqrt{\frac{n}{2\pi}}\frac{2}{0.765} \quad (8)$$

$$d_{CHS} = 0.765\frac{Y}{2} \quad (9)$$

Where n=100 nodes, M=1000m and the optimal number of clusters CH_{opt} is six. The corresponding output of a NETSIM implementation of the proposed multi-robot system is presented in section 5.

3. Proposed System of Detection for Rail Surface Defects

The main objective of this paper is to design a multi-robot system that incorporates uses multiple sensor data and computer vision for monitoring and detection of rail surface defects. Each robot node uploads sensor data continuously to a remote server, which allows us to remotely monitor the condition of railway tracks. Furthermore, the on-system camera uses OpenCV software and machine learning to help classify the type of defect, which is notified to concerned authorities via SMS and a web server. The server also pinpoints the location of the defect on a map that may be detected so that maintenance authorities can initiate necessary action to rectify the same.

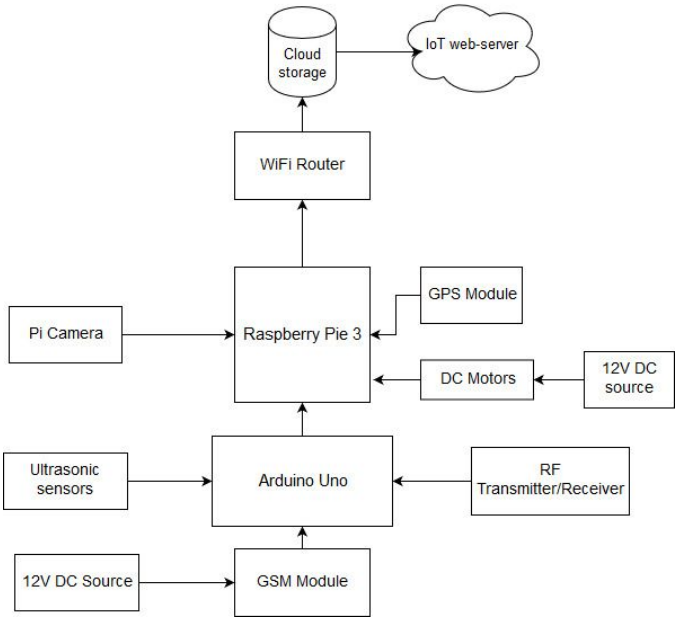

Fig. 4. The general framework of proposed system

3.1 Robot Design

Each robot consists of a Raspberry-pi 3 microcontroller capable of sending real-time data to an internet server. The Raspberry Pi is intended to automatically sync the condition of rail surfaces through sensor outputs, GPS data, images captured. Four ultrasonic sensors are fixed to detect faults above and on either side of a railway track surface as shown in **Fig. 5**. Each robot node has two DC motors for movement, which is coordinated to mimic each other through an RF communication module. Thereby, a master-slave mechanism is set up between the two robots designed in this paper. Furthermore, a Pi camera is interfaced to the raspberry pi, which performs video and image processing to help detect faults. The rail surface detection and classification system using image processing are outlined in **Fig. 7** and **Fig. 10**. Lastly, an

IOT web server is created to relay the position, status and extent of damage present in railway tracks. A general framework of the system is illustrated in **Fig. 4.**

3.2 Overall Working Mechanism

The general working of the proposed system for structurally monitoring, detecting and classifying surface defects is described using a flowchart as shown in **Fig. 6**. The workflow applies to each robot manually deployed on a railway track. This system can classify four types of rail surface defects, which include multi-directional fractures, corrosive rust on steel squats and rail corrugations. The entire system functionality can be subdivided into three parts A, B, and C to intimate authorized persons to take appropriate action based on the type of defect found. Part A consists of three ultrasonic sensors namely, A, B and C as positioned in **Fig. 5**. These sensors iteratively calculate the distance from the running and two vertical surfaces of a railway track. If any of the sensors detect distances greater than a threshold value (preset value based on a distance measurement between the sensor and the railway track surface), the 2 DC motors is programmed to halt for a delay of 10 seconds during which among several functionalities, an image is captured using the Raspberry pi camera module.

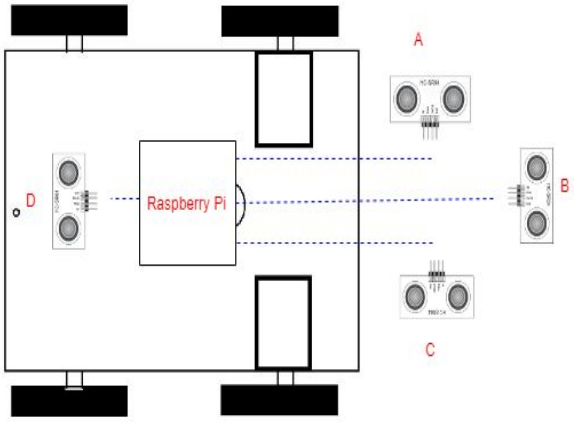

Fig. 5. Ultrasonic sensor placement

Consequently, image-processing techniques are employed as will be described in the next section for further validation. During this stoppage time of the robot, concerned authorities are notified through a text alert that a structural defect has been found. The text message includes an HTTP link of a web-server where additional information regarding the type of defect and its precise timestamp can be found. In addition, as depicted in section 5, railway authorities can view the crack remotely by downloading the image from the website. Furthermore, a web browser pinpointing the exact GPS location of a structural defect is displayed onto a computer screen. Hence, allowing maintenance personnel to arrive at the precise location with necessary tools for repair, maintenance or restoration work that may be required. Part B involves two ultrasonic sensors B and D whose distance measurement inputs are compared continuously. If the distances are unequal, the previously described alert mechanism is followed and authorities are intimated that rail corrugations are detected. Finally, the third part is solely based on real-time video processing of railway tracks to detect running surface fracture and rusts. The video processing is performed by extracting a 100 frames per second, where every

frame is processed using the same detection system as described for an image in the following section.

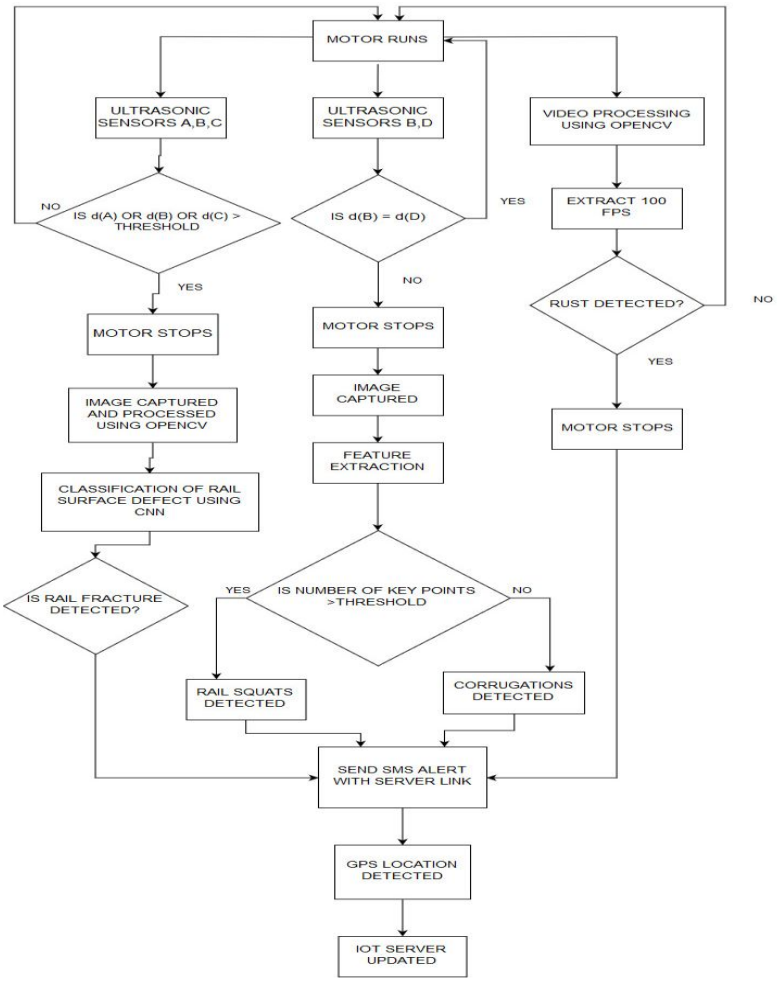

Fig. 6. Overall working of the proposed system

4. Rail Surface Detection System using Image Processing

4.1 Fracture Detection

The first part of the system as described in the previous section involves further validation of running surface fracture and crack detection through image processing. The general workflow is proposed in **Fig. 7,** consists of image acquisition, pre-processing, feature extraction, classification and information storage.

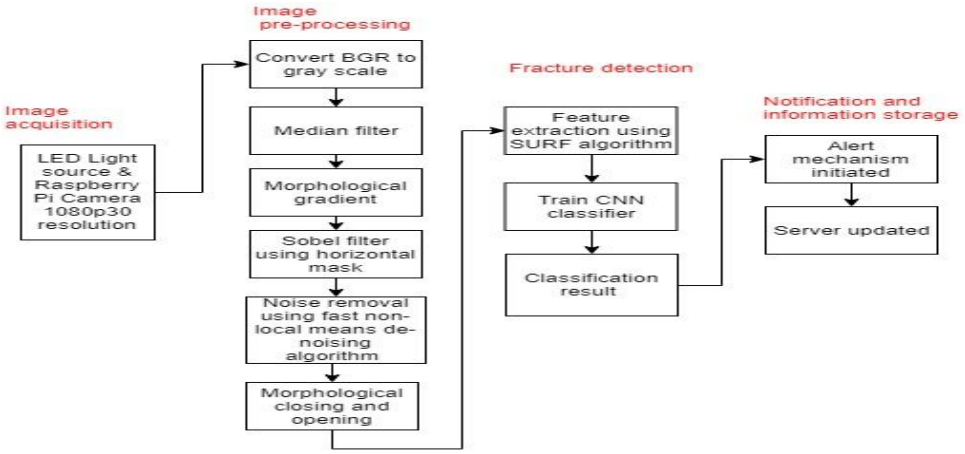

Fig. 7. Rail surface fracture detection system

During image acquisition, a noisy image is obtained due to a number of external factors that include unsuitable lighting conditions, vibrating robot body due to uneven terrain or interference from foreign particle occlusion. Thus, the image pre-processing removes noise that may affect the efficiency of fault detection. After the image is converted from BGR to grayscale, A 7X7 median filter is applied which sorts the pixel values in the ascending order for each window and the center pixel is replaced by a median value. From **Fig. 8**, we can observe that the morphological gradient gives us an outline of the fracture in the image by taking the difference between dilation and erosion of the resultant image. Next, to retain only horizontal edges of cracks, a horizontal Sobel filter mask as shown in **Fig. 9** is applied.

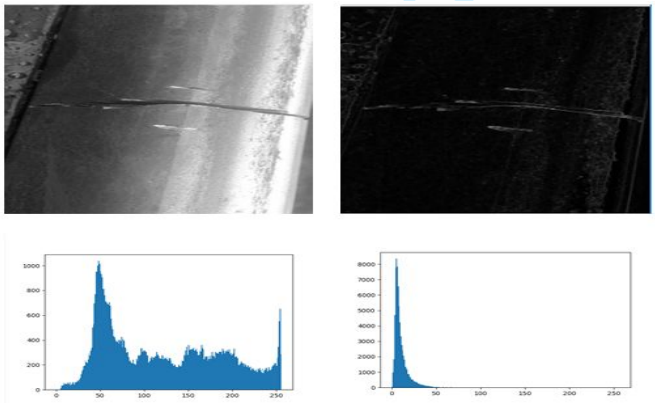

Fig. 8. Morphological gradient (right) of a rail surface crack (left)

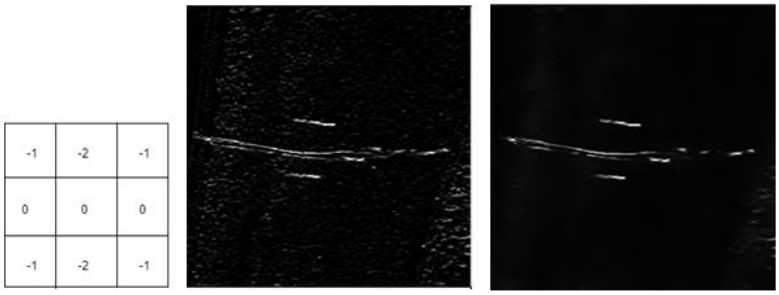

Fig. 9. Sobel filtered (center) image using horizontal kernel (left). Image de-noised (right) using fast non-local means de-noising algorithm.

Furthermore, to eliminate resembling noise pixels over the entire portion of an image, a fast-non-local means de-noising algorithm is incorporated [11]. **Fig. 9** shows the implementation of this algorithm. The image then undergoes morphological processing which utilizes a 2X2 rectangular structuring element to perform closing followed by opening operations. The opening operation allows us to remove background noise, if any, by performing erosion followed by dilation. While closing operation removed foreground noise by dilation followed by erosion of the image. For the next stage, key feature points of the crack from the image are extracted using the Speeded-up robust feature (SURF) algorithm [12]. This algorithm allows us to automatically detect feature points of distinctive objects in an image that may vary in terms of transformations, scale, rotation or abrupt changes in pixel intensities. Finally, a count for the number of key points is established based on which a decision is made whether a multi-directional fracture is detected. This classification is performed by training a CNN classifier whose results are described in section 5. Finally, the processed image is updated to the server for remote visualization of the railway line fracture. The process outlined in **Fig. 7** is followed for detection of squats in section 5.

4.2 Detection of Rust

From **Fig. 10**, this system utilizes an image pre-processing stage, which is different from the previous system of detection of railway line fractures. After the captured image is median filtered, it is converted into the HSV (Hue, saturation and value) color space.

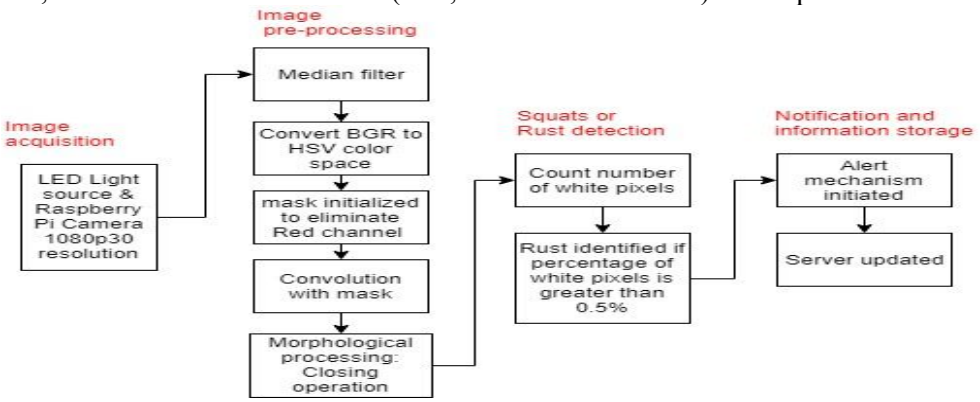

Fig. 10. Rail surface rust detection system

Hue value is the most accurate representation of different colors within range [0, 2π] degrees. For detection of rust, which is predominantly red in color, the H measure is tuned accordingly, while S and V values are kept constant for all intervals. The color space conversion is performed since Hue is independent at a particular value i.e. light intensity. Hence, it allows us to perform feature detection based on color and in the presence of poor or excess lighting conditions. The conversion of RGB to HSV color space is:

$$V = \max(R,G,B) \tag{10}$$

$$S = \frac{\max(R,G,B) - \min(R,G,B)}{V} \tag{11}$$

$$H = \begin{cases} \cos^{-1}\left[\dfrac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right] & B \le G \\ 2\pi - \cos^{-1}\left[\dfrac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)(G-B)}}\right] & B > G \end{cases} \tag{12}$$

From **Fig. 11**, we can clearly observe that the rusted parts of railway track is detected (in white) after a mask is convolved with the image. The mask is a 1X1 matrix created within a specific HSV boundary to eliminate different shades of red. Next, the image undergoes a morphological closing operation that helps smooth the transition between non-rust and rusted areas of the image. The resultant image is a binary image whose white pixels are counted. Any image with white pixels greater than 0.5% is classified as rust. Finally, the classification of the processed image is updated to the IOT server.

Fig. 11. A Rusted portion (center) detected from an image of a corroded metal (left) with white pixels representing corrosion (right).

5. Experimental Discussion and Results

5.1 System Implementation

The hardware system of implementation consists of a two-robot system which consists of a master and a slave robot. The slave robot is programmed and designed to mimic the master robot while it runs over a track to monitor and detect structural defects. RF communication protocol is established between them for the coordinated robot movement. The same has been

extended to simulate ZigBee communication for a 100 node multi-robot system using the LEACH protocol on MATLAB. Hence, allowing efficient path coordination, improved throughput and optimized energy utilization by transmitter circuits. The proposed system described in section 3, consists of a 433MHz RF communication module, GSM900 module for SMS alerts, a Raspberry Pi camera mounted on a servo motor, HC-SR04 ultrasonic sensors to detect cracks, corrugations and squats, neo-6M, GPS module, L293N motor driver IC, 12V DC motors powered by a 12V rechargeable battery and an Arduino UNO, all of which interfaced to a Raspberry Pi 3 model B microcontroller. The fault detection system prototype based on computer vision and ultrasonic sensor technology is shown in **Fig. 12**.

Fig. 12: Hardware prototype of the two-robot fault detection system

5.2.Simulation of the proposed wireless multi-robot system

A wireless sensor network containing 100 nodes as shown in **Fig. 13** is simulated in a *1km x 1km* space. The colored dots represent alive nodes, circles denote dead nodes, + are advanced nodes and the *x* marked at *(500,500)* is the sink. The node positions are randomly selected within the vector space. The 1st order radio characteristics used for our simulations are summarized in **Table 1**[10]. The simulation was performed on MATLAB R2016a.

Table 1: 1st order radio characteristics implemented

Parameter	Energy dissipated
Transmitter/Receiver circuit	$E_d = 50nJ/bit$
Data aggregation	$E_{data} = 5nJ/bit/round$
Transmitter amplifier If $d_{CHS} \leq D_0$	$\epsilon_{fs} = 10pJ/bit/m^2$
Transmitter amplifier If $d_{CHS} > D_0$	$\epsilon_{mp} = 0.0013pJ/bit/m^4$
Node initial energy	$E_0 = 0.5J$

The network was simulated, and outputs were compared for all nodes (normal nodes and advanced nodes) based on the fraction of advanced nodes present (m) and, a factor of the energy difference between normal and advanced nodes (α). The performance measures used to evaluate our simulation are:

- The number of nodes alive: **Fig. 13** shows the number of nodes that are alive for every round. It signifies the number of nodes that have not completely utilized their energy. Our simulations showed that by increasing the energy factor between advanced and normal nodes, more number of nodes were alive until round 5000.

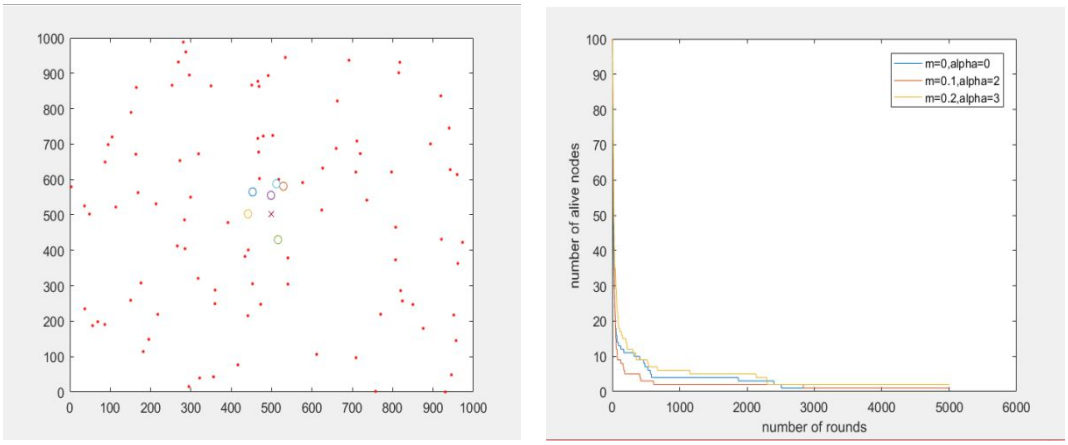

Fig. 13: The wireless sensor network containing alive and dead nodes(left) and number of nodes alive for every round(right)

- Number of cluster heads: It affects the number of packets transmitted by the cluster to the sink per round. Substituting M=1000m, d_{CHS}= 382.5m from (10) and n=100 nodes in (9), we get the optimal number of cluster heads, $CH_{opt} = 6$. from **Fig. 14** we found that with α=three, the optimal number of cluster heads are assigned more effectively for every round.

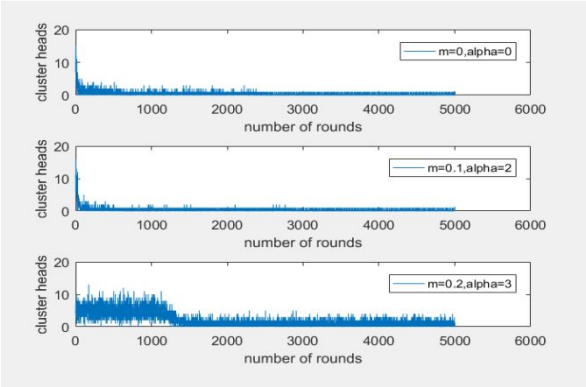

Fig. 14: Number of cluster heads per round

- Throughput: Rate at which the number of packets is transmitted from node to cluster head and cluster head to base station. The overall network throughput in **Fig. 16** shows that the network should be simulated with m=0.2 and α=3 to maximize throughput. The corresponding throughputs obtained for a node to cluster head and cluster head to the base station prove that the above condition maximizes the number of packets transmitted for every round.

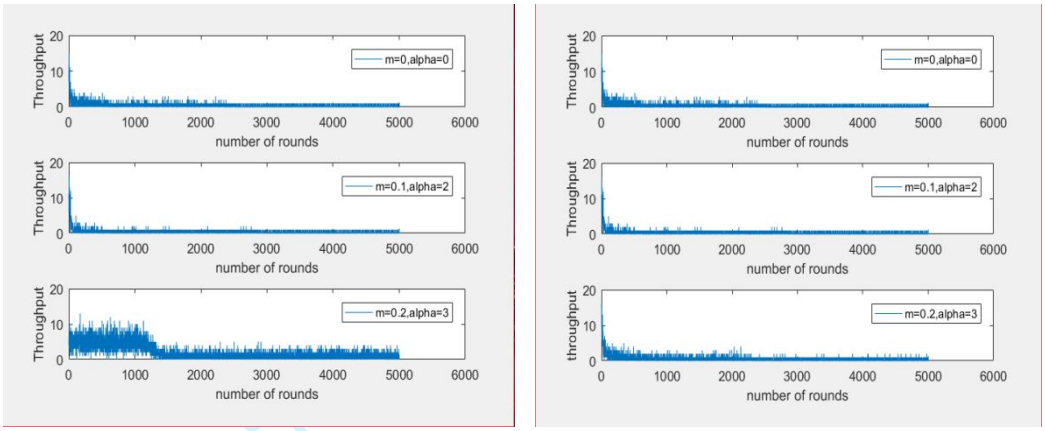

Fig. 15: Throughput for node to cluster head(left) and Throughput for node to cluster head(right)

To summarize, our simulation concluded that for a node incorporating radio characteristics as mentioned in **Table 1**, the wireless network must use m=0.2 and α=3 to maximize the number of alive nodes, network throughput and optimize the number of cluster heads created.

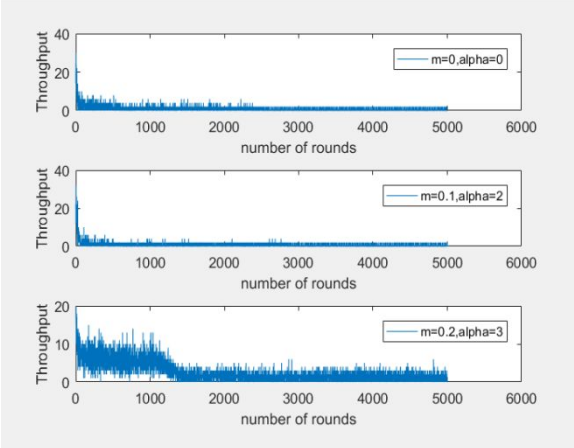

Fig. 16: Overall throughput of the entire network

5.3 Testing railway line fracture,sqauts and rust detector using image processing.

In this paper, we used OpenCV 3.3 for real-time detection and feature extraction of rail surface faults. **Fig. 17** shows the processed image of a railway line fracture and its key features detected using the SURF algorithm. The image is then fed to a Convolution neural network for further classification of crack. The CNN was trained using 403 Images of concrete crack images, out of which 203 were positive cases and the remaining, were negative cases of cracks.

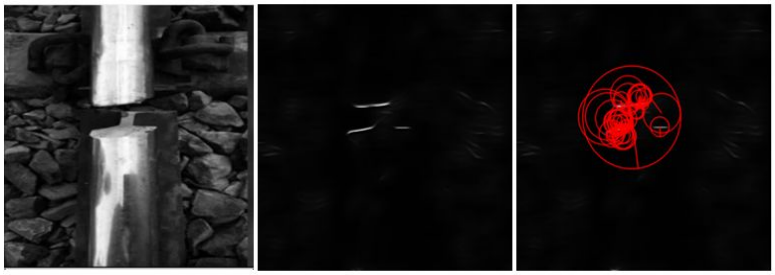

Fig. 17: Processed output depicting railway fracture (center) found in the input image (left) with key feature point extraction using SURF (right)

The CNN model was validated using 70 random test images from the training dataset. From **Table 2**, with almost negligible mean squared error values and high accuracy, R-squared and f1 scores, the model can predict a railway fracture from a pre-processed image accurately.

Table 2: Performance metrics of trained and tested CNN classifier model

Performance metrics	Training	Validation
R squared	0.95	0.935
Accuracy	95.04%	98.57
f1 score	0.987	0.977
Mean-squared error	0.012	0.014

High R squared values during training and validation indicate that the model was able to accurately describe the deviation of predicted outputs from the regression line(Threshold). Moreover, high f1 scores show that the model was able to predict 98.7% of classification outputs correctly during training while for validation, 97.7% of the outputs were obtained correctly.

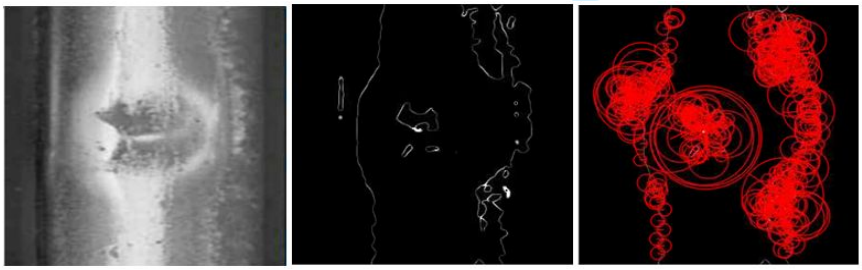

Fig. 18: Processed output depicting squats (center) found from the input image (left) with key feature point extraction using SURF (right)

Since ultrasonic sensors are used to detect both rail corrugations and squats, image processing was further used to validate if the captured image contained squats as shown in **Fig. 18**. Since there are no readily available datasets for classification of squats, count for the number of key points detected using the SURF algorithm was kept. The system detected squats if the count was greater than a preset threshold value. The rust detection system was applied to a single image and can be implemented for each frame during video processing. Since corrosion due to rust cannot be characterized by specific shapes or edges, our rust detection system extracts all shades of reddish coloration in the HSV color space.

Fig. 19: Rust detection program output (center) of rusted tracks (left) with white pixels representing corrosion (right)

Through extensive testing by hit and trial method, we found that the red component has two ranges in the HSV color space. The HSV range to detect reddish coloration of rust is defined for two ranges as:

$$Rust = \begin{cases} 1, & (0,50,255) \leq (H,S,V) \leq (21,50,255) \; or \; (175,50,255) \leq (H,S,V) \leq (179,50,255) \\ 0, & otherwise \end{cases} \quad (13)$$

S and V components are fixed between [50,255] for both ranges while [0, 21] and [175,179] are Hue intervals for first and second ranges respectively. After the image was thresholded using both HSV ranges to eliminate non-red color components, the resultant outputs were added, and binary thresholding was performed. **Fig. 19** depicts the rust area after binary thresholding. It was also found that 36.94% of image area consisted of rust.

5.4.Alert Mechanism

To relay information about the fault detected, an SMS alert is sent to a mobile phone. Also, when a fault is detected, the system is designed to trigger open a web browser with the GPS location pin marked onto google maps as shown in **Fig. 20.** Thus, allowing immediate attention and intervention of maintenance personals. Images captured and processed by the Raspberry Pi are sent to a Dropbox™ account (a cloud storage service). The images can be viewed and downloaded from a website linked to the cloud storage hence allowing railway personals to easily determine extent of the damage. A live Google spreadsheet logs in the status of real-time fault detection by both robots which can also be viewed on the website as shown in **Fig. 21**.

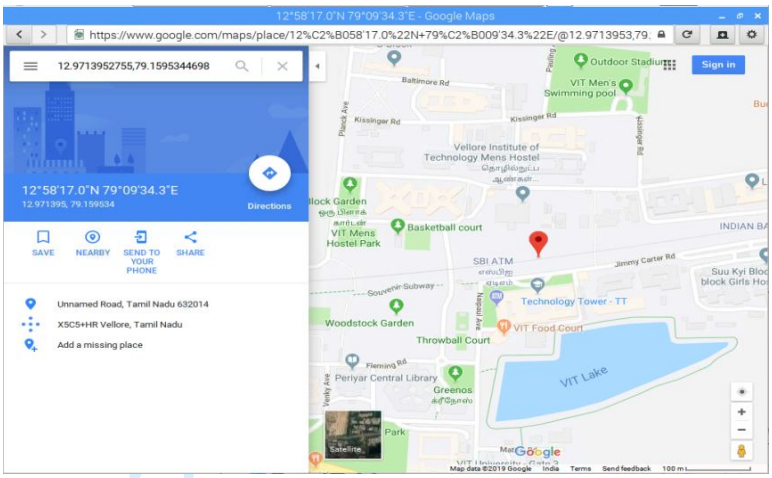

Fig. 20: Location of fault marked on Google maps

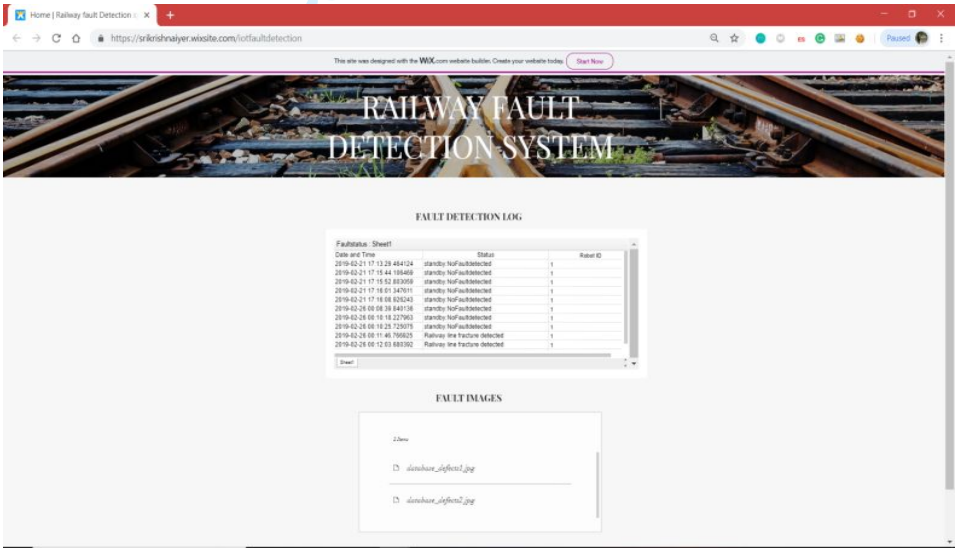

Fig. 21: Railway fault detection website

6. Conclusion

The paper aimed to propose a multi-robot system capable of monitoring and detecting surface defects on railway tracks which include fractures, squats, corrugations and rust. A 100 node multi-sensor environment was simulated using the LEACH protocol on MATLAB. Conclusive results showed that a network must incorporate a higher energy factor, α=3 i.e. advanced nodes with greater energy utilization than normal nodes to effectively optimize throughput and increase lifetime of every node. Thus, a multi-robot system can be realized to monitor one of the world's largest rail networks. A two-robot hardware prototype was implemented which utilized both ultrasonic sensor inputs and image processing to accurately classify faults detected and relay the information to a remote web server. Hence, eliminating the need for visual inspection as an automated alert mechanism allowed both visual and

location-based, real-time tracking of fault detection of railway lines.

References

[1] M. Singh, S. Singh, J. Jaiswal, et al., in Computational Intelligence for Homeland Security and Personal Safety, Proceedings of t2006 IEEE International Conference on. Autonomous rail track inspection using vision based system[a] (2006), pp. 56–59

[2] D Zhan, L Yu, J Xiao, et al., Study on track wear cross-section measurement utilizing laser-photogrammetric technique. J. China Railway. 36, 32–37 (2014) Min et al. EURASIP Journal on Image and Video Processing (2018) 2018:3

[3] G Tian, B Gao, Y Gao, et al., Review of railway rail defect non-destructive testing and monitoring. Chin. J. Sci. Instrum 37(8), 1763 –1780 (2016)

[4] L Xing, Research on Defect Characteristics and Classification of Higher Speed Rails (China Academy of Railway Sciences, Beijing, 2008)

[5] Lad, P., &Pawar, M. (2016, September). Evolution of railway track crack detection system. In 2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA) (pp. 1-6). IEEE.

[6] S.Sam,T.Joby Titus. Automotive Crack Detection for Railway Track Using Ultrasonic Sensorz. International Journal of Engineering Technology and Computer Research. (Dec 2016).

[7] Min, Y., Xiao, B., Dang, J., Yue, B., & Cheng, T. (2018). Real time detection system for rail surface defects based on machine vision. EURASIP Journal on Image and Video Processing, 2018(1), 3.

[8] J Lu, Z Ye, Huber fractal image coding based on a fitting plane. IEEE Trans. Image Process. 22(1), 134 –145 (2013)

[9] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences,* Maui, HI, USA, 2000, pp. 10 pp. vol.2

[10] Smaragdakis, Georgios, Ibrahim Matta and AzerBestavros. “SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks.” (2004).

[11] Buades, Antoni, BartomeuColl, and Jean-Michel Morel. "Non-local means denoising." Image Processing On Line 1 (2011): 208-212.

[12] Bay, Herbert, TinneTuytelaars, and Luc Van Gool. "Surf: Speeded up robust features." European conference on computer vision. Springer, Berlin, Heidelberg, 2006.

[13] Doriya, Rajesh, Siddharth Mishra, and Swati Gupta. "A brief survey and analysis of multi-robot communication and coordination." International Conference on Computing, Communication & Automation. IE

Srikrishna Iyer is an undergraduate student at Vellore Institute of Technology, Vellore, India. His research interests involve Computer vision, Image processing, machine learning, swarm intelligence algorithms and cognitive radio technology.

Shashank Mohankumar is currently an undergraduate student at The Vellore Institute of technology working towards receiving his Bachelors in Technolgy. His research interests include but are not limited to wireless sensor networks, wireless communication, and cognitive radio technology.

Hari Galla is currently an undergraduate student at Vellore Institute of Technology, Vellore, India. His research interests to name a few include computer networking, Image Processing and Machine learning.

T. Velmurugan received his B.Tech. degree in Electronics and Communication Engineering in 2000, M.Tech. in advanced communication systems from SASTRA University, Thanjavur India in 2002 and Ph.D. from VIT University, Vellore, India. He is currently serving as a Faculty in the School of Electronics Engineering, VIT University. He has published several papers in reputed journals. His research interest includes Handoff in Wireless Networks, Cognitive Radio Networks and IoT.

S.Nandakumar received B.Tech in Electronics and Communication Engineering and obtained Master's in Computer and Communication. He obtained his Ph.D in Heterogeneous Wireless Networks from VIT University, Vellore, India. He is presently associated with Vellore Institute of Technology as an Associate Professor. His research Focus on Next Generation Heterogeneous Networks, Cognitive Radio Networks, and Device to device Communication. He has teaching experience of 13 years.