QATAR UNIVERSITY

COLLEGE OF ENGINEERING

RAIL ROBOT FOR RAIL TRACK INSPECTION

BY

NOORA RASHID ALNAIMI

A Thesis Submitted to

the College of Engineering

in Partial Fulfillment of the Requirements for the Degree of

Masters of Science in Computing

June 2020

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Noora AlNaimi defended on 29/04/2020.

Dr. Uvais Qidwai
Thesis/Dissertation Supervisor

Dr. Mohammed AlMeer
Internal Examiner

Dr. Hashim Raza
External Examiner

Approved:

Khalid Kamal Naji, Dean, College of Engineering

ABSTRACT

ALNAIMI, NOORA, R., Masters: June: 2020, Masters of Science in Computing

Title: Rail Robot for Rail Track Inspection

Supervisor of Thesis: Uvais Qidwai.

Railway transportation requires constant inspections and immediate maintenance to ensure public safety. Traditional manual inspections are not only time consuming, but also expensive. In addition, the accuracy of defect detection is also subjected to human expertise and efficiency at the time of inspection. Computing and Robotics offer automated IoT based solutions where robots could be deployed on rail-tracks and hard to reach areas, and controlled from control rooms to provide faster and low-cost inspection. In this thesis, a novel automated system based on robotics and visual inspection is proposed. The system provides local image processing while inspecting and cloud storage of information that consist of images of the defected railway tracks only. The proposed system utilizes state of the art Machine Learning system and applies it on the images obtained from the tracks in order to classify them as normal or suspicious. Such locations are then marked and more careful inspection can be performed by a dedicated operator with very few locations to inspect (as opposed to the full track).

DEDICATION

For myself, my family, and Yalcin.

Thank you all.

ACKNOWLEDGMENTS

I would like to express my deep gratitude and appreciation to my supervisor Dr. Uvais Qidwai for his supervision, review and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

In transport systems, safety and reliability are the main factors that are always questioned, especially in railway transportation systems. Early inspection systems are crucial to maintain safe rail-tracks that will ensure safe journeys. Statistics show that 60% of railway accidents are due to derailment, and 90% are due to railway cracks [1]. Railway track cracks could be inspected by human personnel; however, this is not only time consuming, but also the accuracy is subjective since not all cracks are identifiable by naked eyes.

As Qatar Rail has launched the first train in Qatar in May 2019, it is very important to look for maintenance systems that suits Qatar's climate for railway track inspections. This demand requires inspection systems that will continuously inspect the status of all tracks over Qatar and issue immediate maintenance alerts to avoid accidents. Over the years, machine-driven inspection systems proved to offer a solution for faster inspection and maintenance. Such inspection systems are common in their ability in finding cracks in rail tracks, as well as the crack's location, which helps the maintenance team to reach and rectify the crack in lesser time.

However, available solutions vary in terms of being software-based solutions that are known as non-contact based solutions that apply computer vision technologies on recorded videos; or robotic solutions that are also known as contact-based solutions which are automated systems that are deployed on rail tracks and detect cracks using external sensors such as ultrasonic or IR sensors. Software-based solutions consume time to extract and analyze the images from recorded videos, and robotic-based solutions are limited in their ability in only detecting cracks using sensors without generating results or images.

To address these limitations, a novel automated system is proposed in this work that can be implemented on a robotic platform which performs inspection using non-destructive inspection (NDI) method based on visual inspection with local image processing and cloud storage of information that will consist of images of defected railway tracks only. Local image processing during the inspection is a novel inspection technique that will allow for faster inspection in parallel with cloud storage of information, which will only receive images of defected rail tracks.

1.1. Railway Track Defects

Railway track defects are divided into two main parts: Internal defects and surface defects [2]. These defects may exist in the head, weld or base section of the track. Figure 1 shows an example of an internal defect and Figure 2 shows an example of an external defects.

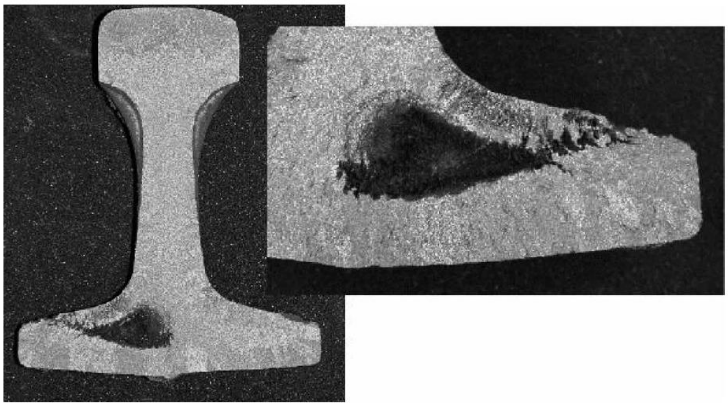

Figure 1. Example of an internal defect in a rail track [3].

Figure 2. Examples of surface defects [4].

The most common defects appearing in rail tracks are known as RCF (Rolling Contact Fatigue) as shown in Figure 3 which results from the friction in high-speed railways. Another common set of defects is the one resulting from the local climate condition and infrastructure peculiarities. High temperature and humid climate, as in Qatar, causes buckling and heat kinks - also known as sun kinks - in rail tracks. Defects like broken railway tracks or sun-kinks are more crucial than a loose ballast or growth of vegetation. Figure 4 shows some examples of sun-kinks and broken railways tracks.

Figure 3. An RCF defect [5].

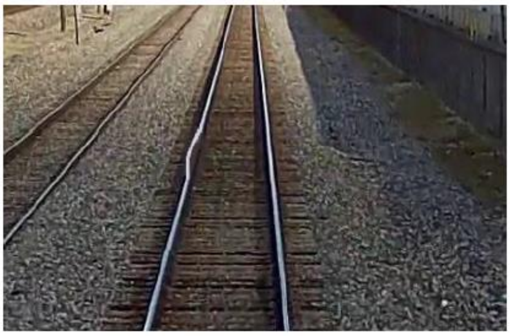

Figure 4. Example of a sun-kink [6].

1.2. Research significance

This research will be a pioneering work in the field of rail-tracks inspection due to the following reasons:

1. This research will consist of two main modules:
    a. Local image processing to analyze images while inspecting.
    b. Cloud communication storage for captured images that are analyzed as defected for further analysis.
2. The application of the rail robot with local image processing techniques combined with machine learning will replace the practice of overwhelming the cloud with all captured images. The cloud will

receive only images that are analyzed as defected with their location for further analysis; therefore, faster inspection and faster cloud processing.

3. All vision-based rail-track inspection solutions are based on acquiring images from recorded videos. This research is expected to be the first to directly send captured defect images to a cloud.

1.3. Research Questions

This research intends to answer the following research questions:

1. What are the best features to correctly classify the defects in the rail-track in the local image processing?
2. How would the cloud store received images?

1.4. Research Objectives

1. This research aims to provide a reliable cost-effective rail robot intelligence system for rail-tracks inspection based on non-destructive method by applying image processing.
2. This research also aims to apply local image processing techniques to detect the state of rail-tracks while inspecting. In addition, machine learning techniques will be applied to locally classify rail-track images as normal and defected, and send only defected images to the cloud.

1.5. Solution Overview

In this work, a novel non-destructive inspection method based on visual inspection is proposed. The block diagram of the proposed system is shown in Figure 5Figure 5.

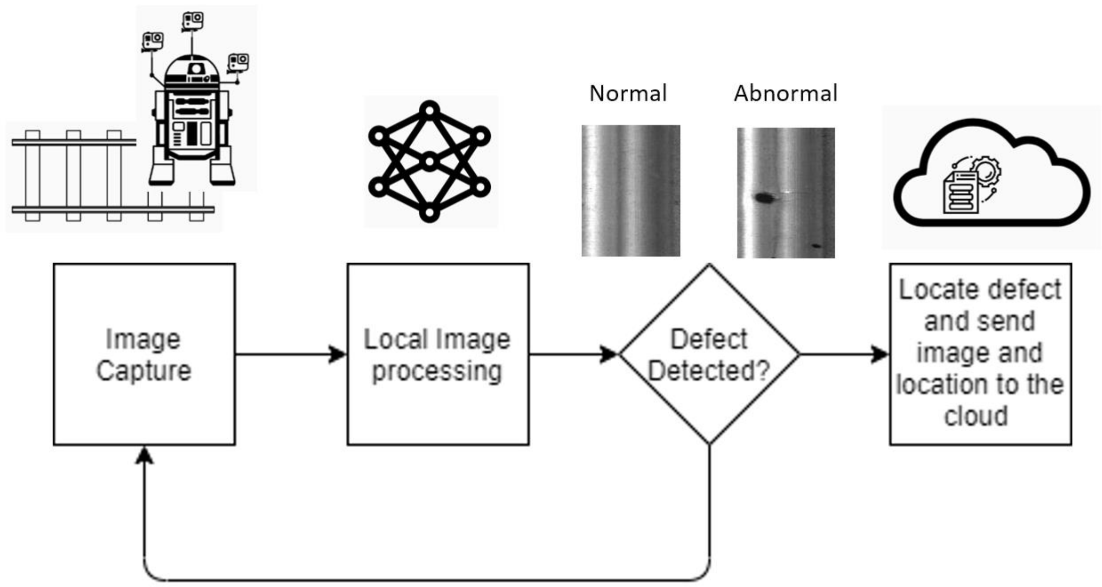

Figure 5. The block diagram of the proposed system.

The system consists of three main parts:

- A robot deployed on the rail-track that will capture images from both sides of the track.
- Local image processing module embedded in the robot.
- Cloud communication and storage of the defected rail track sections.

Most of the visual inspection solutions are based on image processing that is done after capturing the whole images of the rail-track, or after recording a video that is sent to the cloud or stored on local devices. That is, image processing is done at a control station away from the rail track. In this work however, image processing is done on the rail track while performing the inspection, and only defected rail track images are stored and sent to the cloud. This makes inspection faster and does not congest the cloud with unnecessary data. Figure 6 shows the flowchart of the system.

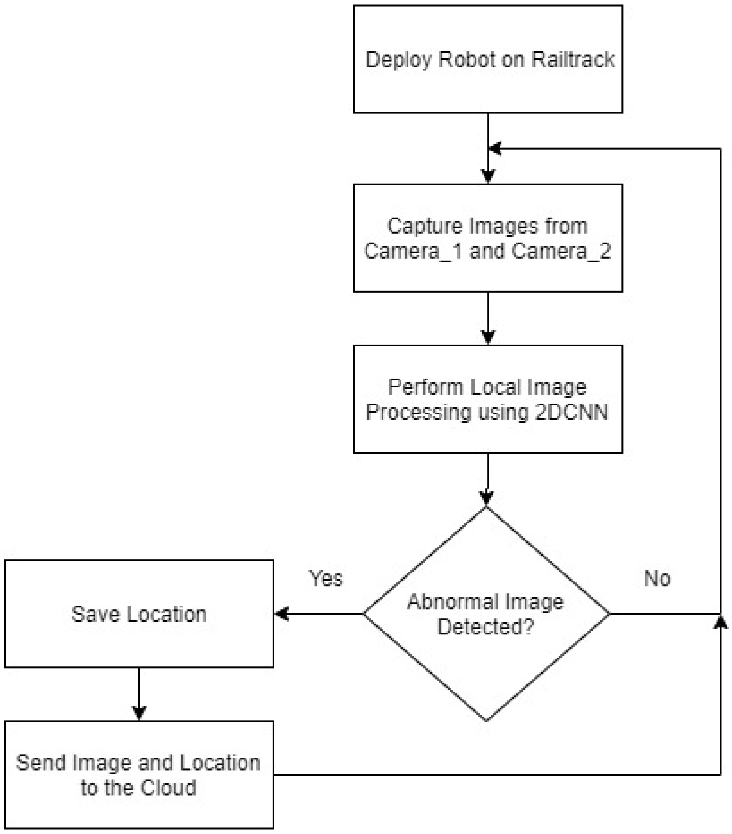

Figure 6. The flowchart of the propsed system.

Once the robot is deployed on the rail track, it starts capturing images of both sides of the track from the installed camera. The captured images are then processed by the image processing module that classifies the image as *normal* or *abnormal* locally. The binary classification of the captured images is done with a 2DCNN (Convolutional Neural Network). Once a defect is detected, the location of the defect is mapped and saved with the image to the cloud.

The following chapter gives a background on the machine learning algorithms used in this study. Chapter 3 describes the methodology and experimental procedures for binary classification and cloud storage. Chapter 4 presents the results and discussion, and finally Chapter 5 concludes this research, and provides some future directions for improvement and continuation of this study.

CHAPTER 2: BACKGROUND AND RELATED WORK

Machine derived inspection systems used over the years proved to offer a solution for faster inspection and maintenance. Machine derived inspection systems are common in their ability in finding cracks in rail tracks as well as the crack's location, which helps the maintenance team to reach and rectify the crack in less time. However, solutions vary in term of being contact-based solutions which are also known as robotic-based solutions, or non-contact based solution which are also known as software-based solutions. Contact-based solutions are automated systems that are deployed on rail tracks, and non-contact based solution are systems that apply machine vision technologies in rail track inspection. In this section, background information and available solutions in the literature are covered.

2.1 Railway Track Inspection Methods

Railway track inspection methods are either contact-based which are known as NDT (Non Destructive Testing), or non-contact based methods which is based on analyzing images or videos of the rail-track. Some examples of each type is:

2.1.1 Contact-based Methods

- Ultrasonic Inspection: this method can detect deep internal defects, but fails to detect surface and near surface defects [2],[7],[8],[9].
- MFL (Magnetic Flux Leakage): this technique can detect near surface defects such as RFC, but fails to detect deep internal defects [2].
- Eddy Current Inspection: this technique is based on magnetic fields; therefore, similar to MFL, this technique can detect surface defects, but fails to detect deep internal defects. To overcome this shortage, hybrid solutions combining both ultrasonic and eddy current inspections are available [9].

- Acoustic Emission Inspection: this method is common with steel rail-tracks, where it is used to detect crack's growth and accumulation as well as source of crack localization [2].

2.1.2 Non-contact-based Methods

- Visual Inspection: this technique is the most efficient technique used for surface defects detection. It is based on high-speed cameras that capture images of the railway tracks to be processed later based on pattern recognition of the captured images; therefore, it is economical and time saving, but requires higher computational time [2].

2.2 Contact-based Methods in Literature

An autonomous robot-based cost-effective railway crack detector is proposed in [1]. This system detects cracks and analyzes faults using an ultrasonic sensor, and alerts the control room through SMS with the crack's location. The architecture of the system, shown in Figure 7, consists of the Arduino micro-controller that connects the sensing parts (Ultrasonic sensors, GPS sensor and Optical encoder), with the actuating parts (Motors, GSM modem and LCD display, and the motor circuits that drive the robot with DC motors).

The ultrasonic sensors keep on reading distances to find cracks by sending sound waves and comparing the echoes received to a predefined threshold. If an echo is greater than the threshold, then a crack exists, and the GSM modem directly sends an alert message to the control room with the crack's location that is read from the GPS sensor. The LCD display shows real-time status of the system, and the Optical encoder is used to measure the robots' speed in RPM (Revolutions per Minutes), as a second check on the robot's location.

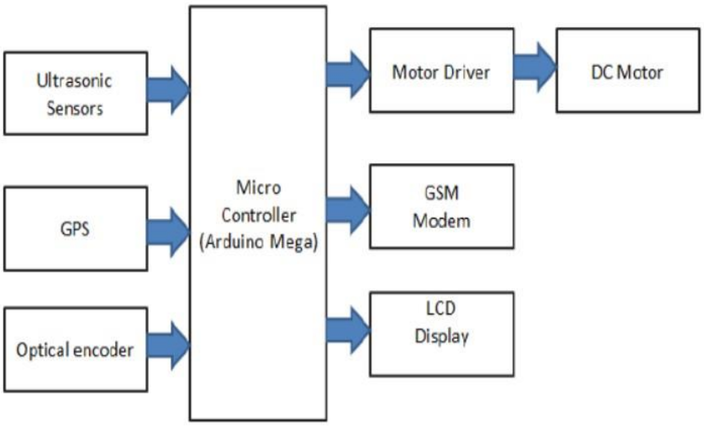

Figure 7. Block diagram of SCANOBOT [1].

Another ultrasonic based detection system is proposed in [8]. The authors proposed a hybrid feature extraction and selection method based on laser ultrasonic detected signals. The system combines WPT (Wavelet Packet Transform) which preprocesses data by decomposing different frequencies of the input signal into different frequency bands that are the feature sets, KPCA (Kernel Principal Component Analysis) which is used to reduce redundancy among the feature sets, and SVM (Support Vector Machine) which is used to classify the defects in the input signals. This method has achieved an accuracy of 98.73%.

Similar to [1], in [10] a robot-based system is proposed to detect cracks in rail tracks using IR sensor through voltage variations, and sends the detected location by GPS to the control station through a GSM module, which is then displayed on a map through .NET software. The architecture of the proposed system is shown in Figure 8 and Figure 9.

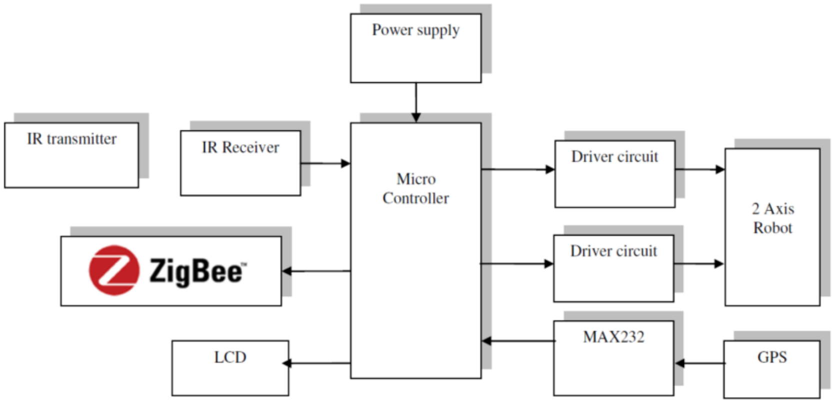

Figure 8. Block diagram of system [10].

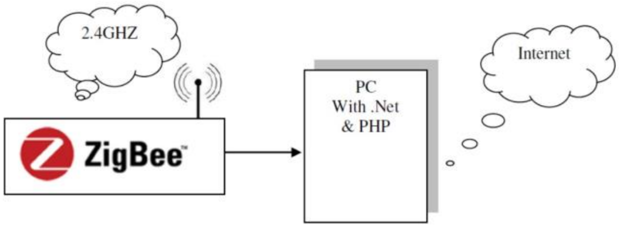

Figure 9. Block diagram of control room [10].

The PIC microcontroller detects cracks through reading the IR transmitter and receiver values, and sends the crack's location to the control station over ZigBee protocol. The microcontroller controls the robot's movement though the driver circuits, and the LCD Display Unit displays the IR sensors readings. The ZigBee protocol is also used at the control station to read the sent alerts from the robot. Once the robot is deployed on the rail, the IR sensors keeps reading to detect voltage variations in comparison with a predefined voltage threshold. If a crack is detected, the location retrieved by the GPS sensor is sent through GSM to the control room, in which the location will be mapped using .NET software.

The proposed systems in [1] and [10] are similar in the architectural model and functionality, varying only in the cracks detection sensor. Also, [10] provides mapped locations of all cracks, which would help the maintenance staff to directly reach the cracks and fix them. However, the location is provided as coordinates

Similar to [10], a robot based system that detects rail track cracks using an IR sensor is proposed in [11]. However, the authors have used a 5W solar panel to power up the system, which is an advantage over traditional systems that are powered by lithium batteries.

Finally, in [12], a robot based solution that can detect both internal and external defects, and communicate with the base station wirelessly using ZigBee protocol is proposed. The robot can detect external cracks with IR (Infrared) and Ultrasonic sensors, and can detect internal cracks with density sensor and locate the defect with GPS, however, no evolution metric was provided.

2.3 Non-contact-based Methods in Literature

Non-contact-based methods are based on analyzing images or video records of the rail tracks to detects faults or defects. This technique is based on visual inspection of the state of the rail track. Visual inspection based systems - which are also known as computer vision - are one of the most effective and important inspection tools for flexible automated rail monitoring [13]. Computer vision based techniques makes rail-track inspection possible from grayscale images only without additional sensors[14].

Computer vision based applications consist of two main steps: Image acquisition that consists of capturing images using a camera, and Image Processing which consists of improving image quality by reducing the noise generated due to illumination, climate factor, or shaking camera [15], [16]. Visual inspection techniques are the most efficient techniques used for surface defects detection. It is based on high-speed cameras that capture images of the railway tracks to be processed later based on pattern recognition of the captured images; therefore, it is economical and time saving, but requires higher computational time [2].

Computer vision based solution would not interfere with the train schedules; therefore, would not cause any interruption or delay. Once the image is captured, faults in track are localized by using detection methods such as Canny Edge detection method which locates discontinuities in the image. In feature extraction, single features do not provide enough information; therefore, multiple feature based systems are richer in useful information.

In [17], the authors proposed a computer vision based method by capturing images of the two neighboring rail tracks to determine the rail in the image and the distance between the rails. Images are captured by a camera placed on the train that captures the current main rail track of the moving train, and the neighboring rail track. This method detects pitch fault, contraction and expansion faults by comparing the image's pixel to a predefined threshold value by using canny edge detection, feature extraction methods and the closing morphological operation. The camera used in this system can capture 100 frames per second; however, the authors have used only 10 images per second to achieve fast processing.

In [9], the authors suggest an automatic inspection method of images captured by a digital scan line cameras installed on both sides of the rail track. Binary Image Based Rail (BIBRE) technique is used to extract the rail section of the rail tracks. The extracted rail sections are then enhanced with improvement techniques, and then faults in the rail are recognized using Gabor channels. The images are preprocessed using the Otsu algorithm, and the rail sections are detected using Canny edge detection and Hough change calculations.

In [16], an intelligent image processing algorithm that detects RCF (Rolling Contact Fatigue) is proposed. Region defects are detected in the images using adaptive histogram equalization, then segmented after detection by an adaptive threshold method. The authors created a 3D crack growth model using the COMSOL simulation software, designed by identified geometrical properties. The algorithm consists of multiple stages:

- The preprocessing stage that consists of removing noise from the image using the median filtering technique.
- The defect-processing stage in which segmentation thresholding is done to identify defects in the preprocessed images using the automatic iterative selection method.
- The defect post-processing stage where the images resulting from the previous step are further processed to remove any remaining noise and false defects by morphological operations.

The authors have also designed a crack growth detecting algorithm based on LEFM (linear elastic fracture mechanics). Although this system have a comparatively high accuracy; however, it is a complex and expensive system.

In [18], a vision based technique called TrcakNet is proposed that mitigates the false alarm rate such as vegetation, or birds droppings from captured images. This system is based on two deep learning networks that perform semantic segmentation of the captured the track, and binary classification to classify false and true alarms. Image segmentation is done by two paths which captures context and enables precise localization, and the neural network is trained using the Adam algorithm and the Binary cross entropy as the loss function. Faulty regions are then cropped from the segmented images which are then fed into the second neural network for classification that trains on 75% of the data. The proposed system was tested with ResNet and DenseNet neural networks and has achieved an accuracy rate of 88.8% and 90.3% respectively.

In [19], another computer vision based system is introduced that determines defected rail surfaces and identifies missing or defected rail components. The system is tested on images captured from two cameras fixed on a test rail vehicle. Images are processed by applying Canny edge extraction algorithm which is used for feature extraction, and a the rail track is extracted from the image by applying a search to obtain a 0 and 90 degree straight lines. This results in finding the rail track and its components in the image that are then inspected. Images are then converted to greyscale format and processed with padding, Average filtering and binariazation to check if a certain region is defected. This system was implemented on Visual Studio software using the Emgu CV library; however, evaluation metrics were not mentioned.

In [20], a computer vision based vehicle is proposed that runs on the rail tracks and captures images from a CCD (Charged-coupled Device) camera that are processed for crack image segmentation, crack identification, and parameter information extraction. Images are preprocessed by the Weighted Median filter algorithm for noise filtering, the Histogram Equalization algorithm for and image enhancement, and threshold segmentation for crack region extraction. Then Pixel Integral Projection is used to find the integral projection of the vertical and horizontal direction of the rack, and to obtain the parameter information of the crack by drawing the projection curve. The system was implemented on MATLAB; however, no evaluation metric was provided.

In [21] and [22], the authors have designed a rail defect detecting system with real time image processing in which the rail as well as the rail defects are detected. Rails are detected using Hough transform, Canny edge detection and morphological methods for feature extractions. Rail defects are detected using Robert's edge detection, Laplacian low pass filter and morphological methods for feature extractions. In this method, the system targets faults in head-checks, scours, fractures and undulation faults. This method achieved an average accuracy rate of 85.3%.

An automated video analysis based system is proposed in [23] that uses a vision-based system for rail track inspection to replace manual visual checks. This system detects rail track clips, which are metals that hold rail tracks to the ground, which if broken or missing may lead to accidents. The proposed system inspects rail tracks using images from a recorded video using an algorithm that can detect if the clip is old, broken, or missing.

The algorithm analyzes clips condition by applying color analysis such that a new clip is shaded by blue and an old clip is shaded by gray, which helps the maintenance staff later to inspect only the old (gray) clips instead of inspecting all clips at all locations. The system has two main parts: Image pre-processing and Clips locating.

The Image pre-processing consists of applying Gaussian Smoothing filter to reduce noise pixels in the image, Edge Detection algorithm to separate non-related regions from the targeted region - which is the track, clip and wheel, and Short Line Removal algorithm to remove all parts of the image that is not part of the target. On average, the system is 95.3% accurate in recognizing clips, and can detect gray clips with 86.5% accuracy, blue clips with 95.3% accuracy, missing clips from a track with 84.7 % accuracy. Figure 10 illustrates the system’s output.

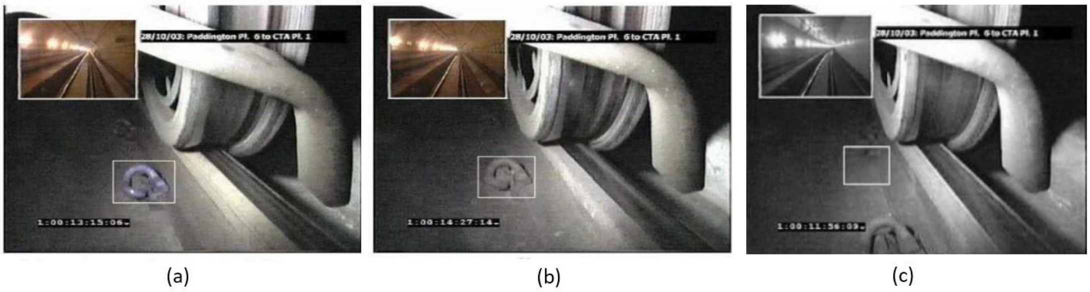

Figure 10. The system’s ability to detect blue (new) clips in (a), grey (old) clips in (b) and missing clips in (c) [23].

A deep multitask learning based rail track inspection system is proposed in [24]. This system is designed to detect cracks in rail tracks as well as rail fasteners – clips – using a single-view line-scan cameras. Unlike the system in [23] that detects rail track fasteners only, this system provides the possibility to also detect different material of the rail tracks such as wood and concrete, and classify them into material categories.

Rail track inspection is done through a customized software tool that provides the possibility of viewing and annotating data into separate boxes to avoid intra class variations in the neural network. The system allows the user to segment selected data as separate frames, control detection thresholds as well as select from various defined defect factors, which allows the user to inspect the rail tracks or track clips based on selected data segment. The system is shown in *Figure 11*.

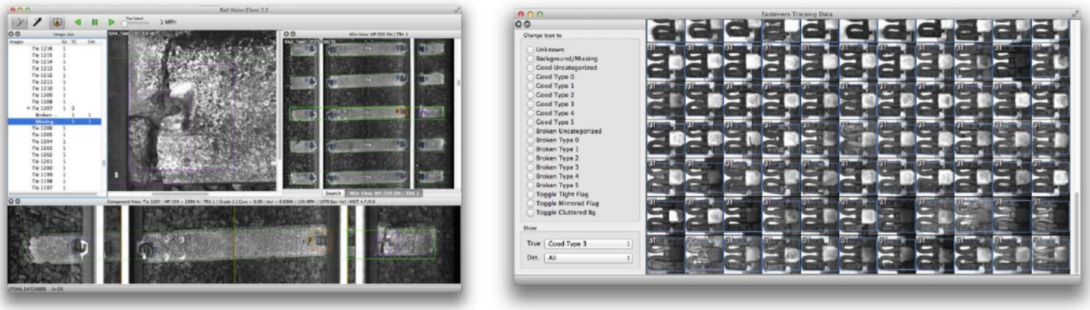

Figure 11. System’s GUI tool [24].Figure 11

A fully deep convolutional neural network is trained on 10 classes as well as the rail fastener classifier feature and a 32-bit SVM output channel. The system’s architecture shown in Figure 12 is divided as follows:

- Rail track material detection is done in 4 layers: conv1, conv2, conv3 and conv4_t. In this part, 10 score maps are generated that refer to the 10 material classes as shown in Figure 13 wood, and results accuracy are 95.02%.
- Rail track fastener detection (object detection) is done in 5 layers: conv1, conv2, conv3, conv4_f and conv5_f . In this layer, a fastener is searched for in a predefined Region of Interest (ROI) and then classified as missing or found. If found the image is further analyzed into defected, or good, which are assigned tags as illustrated in Figure 14. Layer conv4_f is trained to learn a model about the different fasteners, and layer conv5_f inherits this knowledge to distinguish between the different fastener parts.

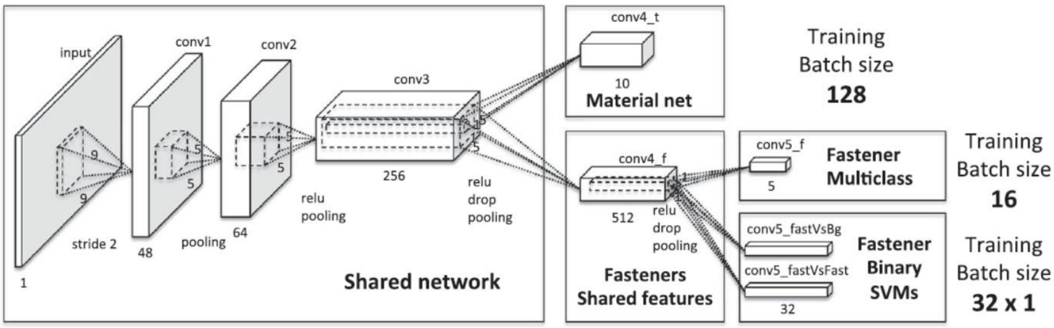

Figure 12. System's network architecture [24].

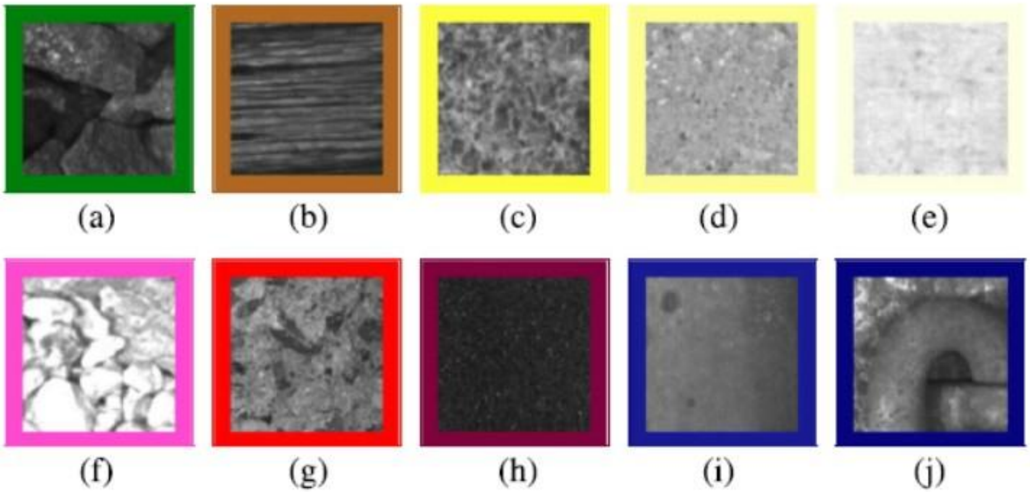

Figure 13. Material categories. (a) Ballast. (b) Wood. (c) Rough concrete. (d)Medium concrete. (e) Smooth concrete. (f) Crumbling concrete. (g) Chipped concrete. (h) Lubricator. (i) Rail. (j) Fastener [24].

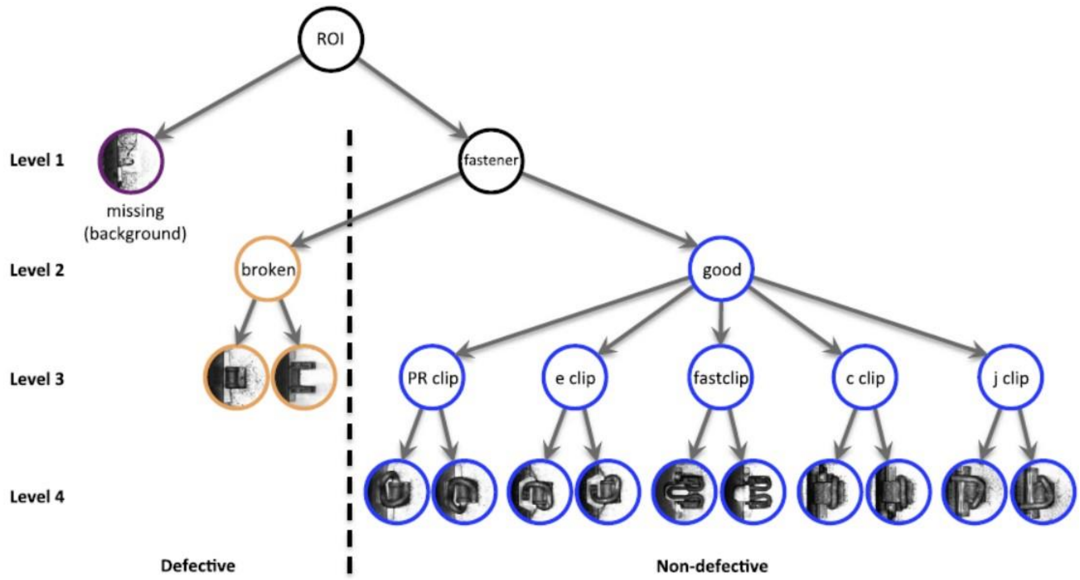

Figure 14. Object’s detection and classification [24].

The success and sustainability of transport systems demand continuous inspection and immediate maintenance. A basic inspection system requires the ability to detect cracks, locate them on the rail tracks and call for maintenance. The discussed systems offer solutions that vary in the method of cracks inceptions. Some systems use ultrasonic sensors or IR sensors that detects any abnormal condition change on the tracks, and other systems apply machine vision based algorithms that can detect abnormal conditions on data that resembles the rail tracks through cameras as well as deep learning.

With both systems, a continuous connection with a control room is required to locate the position of cracks on rails, as well as to issue an immediate maintenance call. Moreover, software-based solutions consume time to extract and analyze the images from recorded videos, and robotic-based solutions are limited in their ability in only detecting cracks using sensors without generating results or images, as well as having small low power systems that cannot be deployed on large rail tracks as in [1],[10] and [12]. Table 1 summarizes the rail track inspection techniques that were discussed in this research.

Table 1. Summary of Rail Track Detection Techniques

Inspection Method	Related literature
Contact-based methods	• Mahfuz et al. [1]: Robotic-based system that uses ultrasonic sensor and GSM modem. • Shekhar et al. [10]: Robotic-based system that uses IR sensor, GPS, GSM and Zigbee protocol. • Kasthuri et al. [11]: Robotic-based system that uses IR sensor and a 5W solar panel to power up the system. • Chittal et al. [12]: Robotic-based system that detects both internal and external defects using IR, Ultrasonic, and density sensor and communicates with the base station wirelessly through ZigBee and GPS. • Jiang et al. [8]: Robotic-based system that uses Ultrasonic sensor, WPT to decompose frequencies into feature sets, KPCA to reduce redundancy, and SVM to classify defects. This method has achieved an accuracy of 98.73%.
Non-contact-based methods	• Karakose et al. [17]: Uses Canny Edge detection, feature extraction methods and morphological operation to detect pitch fault, contraction and expansion faults. • Kumar et al. [9]: Uses BIBRE technique for rail section extraction and Otsu algorithm for image preprocessing. Rail faults are detected using Gabor channels, Canny edge detection and Hough change calculations. • Sambo et al. [16]: Uses adaptive histogram equalization, adaptive threshold, Median filtering, segmentation thresholding, and morphological operations are applied to detect RCF defects. A crack growth detecting algorithm based on LEFM is designed using the COMSOL simulation software. The system has a high accuracy; however, it is complex and expensive. • James et al. [18]: Uses two deep learning networks to perform semantic segmentation that are trained using the Adam algorithm and the Binary cross

Inspection Method	Related literature
	entropy as the loss function, and applies binary classification that is trained on 75% of the segmented images. The system was tested with ResNet and DenseNet neural networks and has achieved an accuracy rate of 88.8% and 90.3% respectively. • Tastimur et al. [19]: Determines defected rail surfaces and identifies missing or defected rail components using Average filtering, binarization, Canny edge extraction algorithm, and applies a search to obtain a 0 and 90 degree straight lines to find rail tracks in the image. Implemented using the Emgu CV library in Visual Studio. • Fu et al. [20]: Uses Weighted Median filter, Histogram Equalization, and threshold segmentation for crack region extraction. Also uses Pixel Integral Projection to obtain the parameter information of rail track cracks. • Tastimur et al. [21], [22]: Uses Hough transform, Canny edge detection, and morphological methods to detect faults in head-checks, scours, fractures and undulation faults. Achieved an average accuracy rate of 85.3%. • Singh et al. [23]: Video analysis based system that detects rail track clips. Applies Gaussian Smoothing filter, Edge Detection algorithm and Short Line Removal algorithm. The system is 95.3% accurate in recognizing clips, and can detect old clips with 86.5% accuracy, new clips with 95.3% accuracy, missing clips from a track with 84.7 % accuracy. • Gibert et al. [24]: Deep multitask learning based system that detects cracks, rail fasteners and rail track material . Uses a fully deep convolutional neural network that is trained on 10 classes, on the rail fastener classifier feature and a 32-bit SVM output channel. The system achieved a 95.02% accuracy.

CHAPTER 3: METHODOLOGY

3.1. Computation Procedure for Binary Classification

Local image processing is based on classifying captured images to normal or abnormal using a 2DCNN (Convolutional Neural Network) [13] [25] during the robots' inspection. Once an image is classified as abnormal, its location is recorded and sent along with the image to the cloud to be post-processed later by locating the defect in the image. In this section the details of the local image processing module are presented.

3.1.1. Preprocessing

The binary classification system was trained and tested on Type-I dataset [7][15] which consist of normal and defected rail-tracks grayscale images. Type-I dataset has 67 images captured from express rails. Most of the images in the dataset consist of different types of challenging defects; however, the availability of normal railway track images is very scarce. This small number of available images makes training and evaluating the neural network very challenging.

To solve this problem, the dataset was inflamed by cropping the original images of size 1000 x 160 to multiple images of size 160 x 160 as shown in Figure 15. Every image results in seven images of size 160 x 160. The seventh and last cropped part of the original image is of size 160 x 40. This image is always discarded to keep the size of all images consistent while training the neural network.

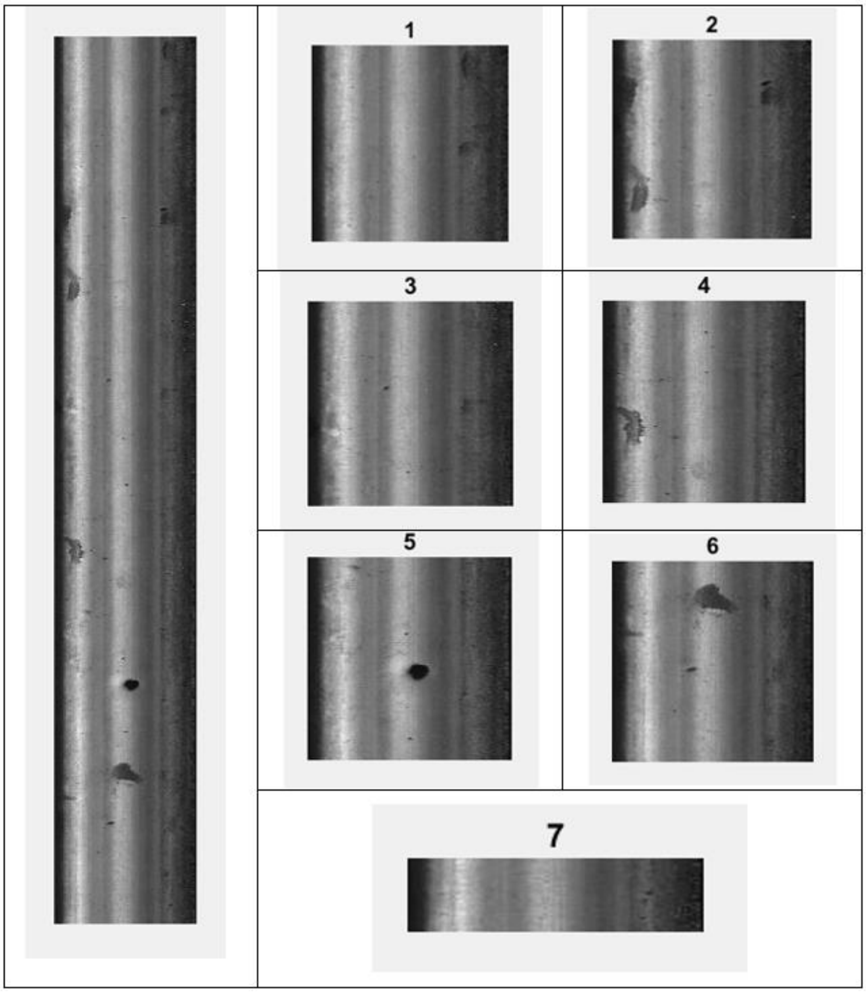

Figure 15. Data inflammation by cropping a single image into multiple images of size 160 x160.

When training the neural network, defects are detected in the image based on the pixel intensity; darker pixels are defects in the normal gray rail track image. As shown in Figure 15, the cropped images have a dark portion at both ends of the rail track segment that is misleading to the neural network. Therefore, the images are further cropped to remove the dark ends resulting in images of size 160 x 120 as shown in Figure 16.

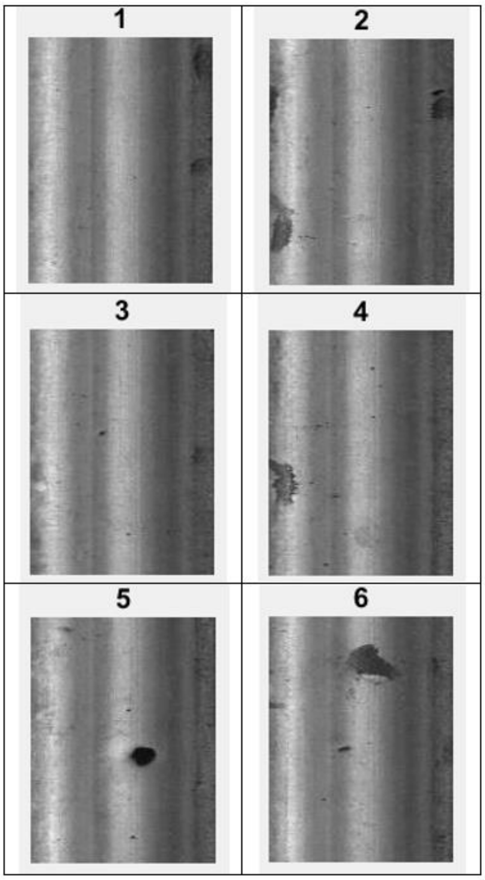

Figure 16. Croppping images and resizing to 160 x 120.

Finally, the brightness in the images was hiding some defects; therefore, image thresholding was necessary to adjust the brightness. After performing many experiments, the best threshold value found was 85%. Figure 17 shows the effect of image thresholding. Each cropped image is then rotated in 180 degree and mirrored as shown in Figure 18. This step resulted in a larger number of images to better train the neural network.

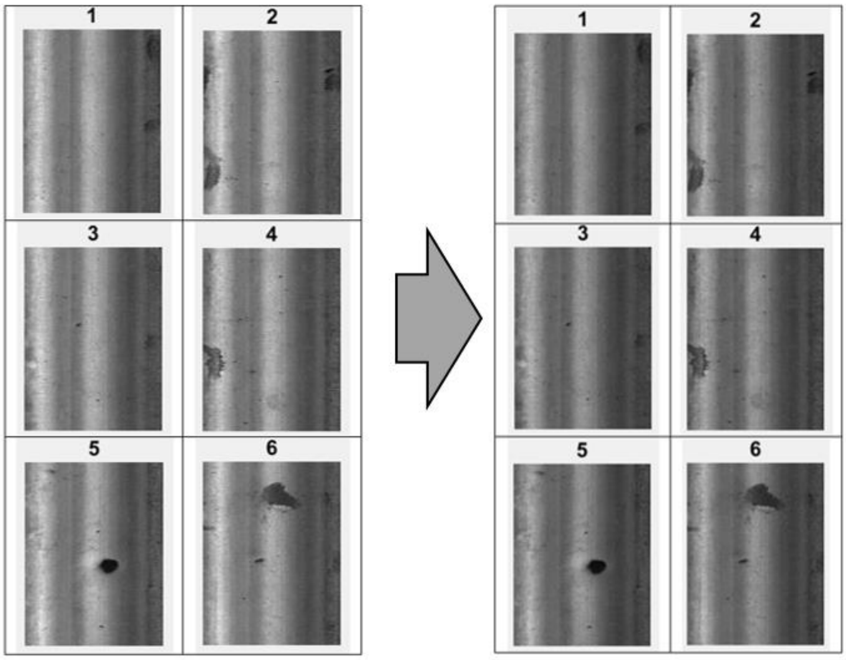

Figure 17. Image thresholding to adjust brightness.

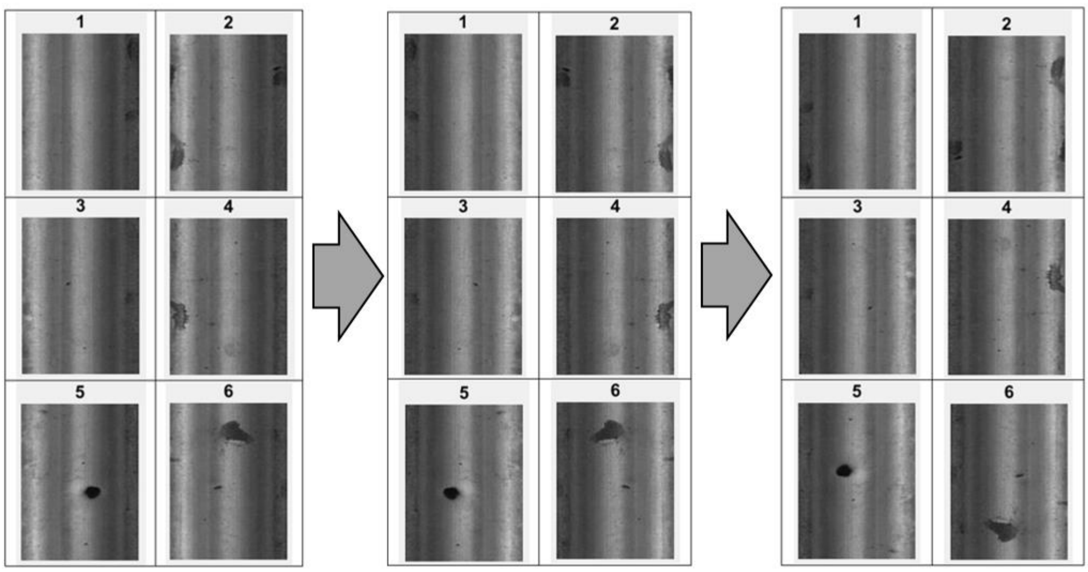

Figure 18. Data inflamation by mirrioning and rotation.

Images are then classified into two classes: *normal* and *abnormal* based on visual inspection. After this step, each classified folder is ready to be fed to the neural network. Table 2 shows the number of images in each class. Figure 19 and Figure 20 show some images from each class of Type-I dataset.

Table 2. Number of classes and images in Type-I dataset

Dataset Class	No. of Images
Type-I Normal	46
Abnormal	203

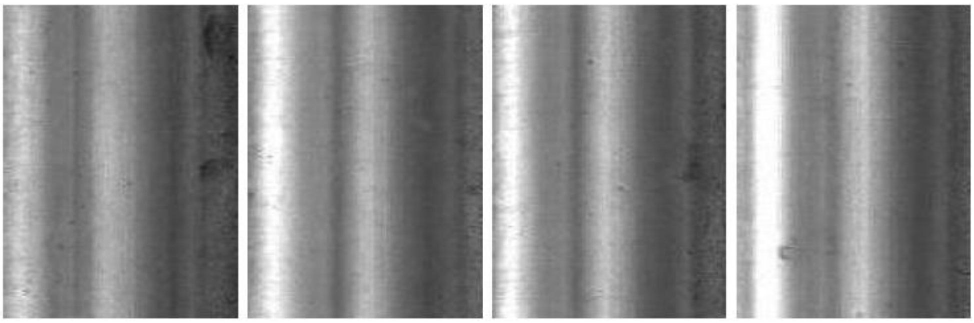

Figure 19. Type I normal samples.

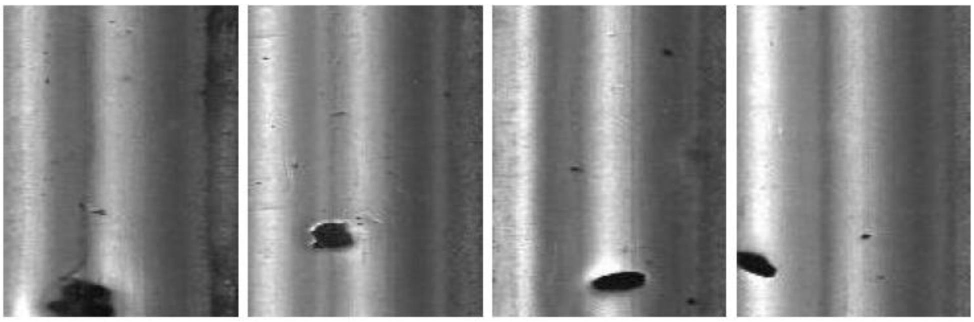

Figure 20. Type I abnormal samples.

3.2. Machine Learning Platform Setup

The binary classification system is implemented on MATLAB R2019. The designed 2DCNN consists of 24 layers as shown in Figure 21. The network has 5 convolution layers of size 5 x 5 each, in which learnables are increased at every layer. A batch normalization layer is introduced after every convolutional layer to standardize the inputs, the rectified linear unit (ReLU) was chosen as the nonlinear activation function, and the max pooling units of size 5 x 5 and a stride of 2 as the pooling layer.

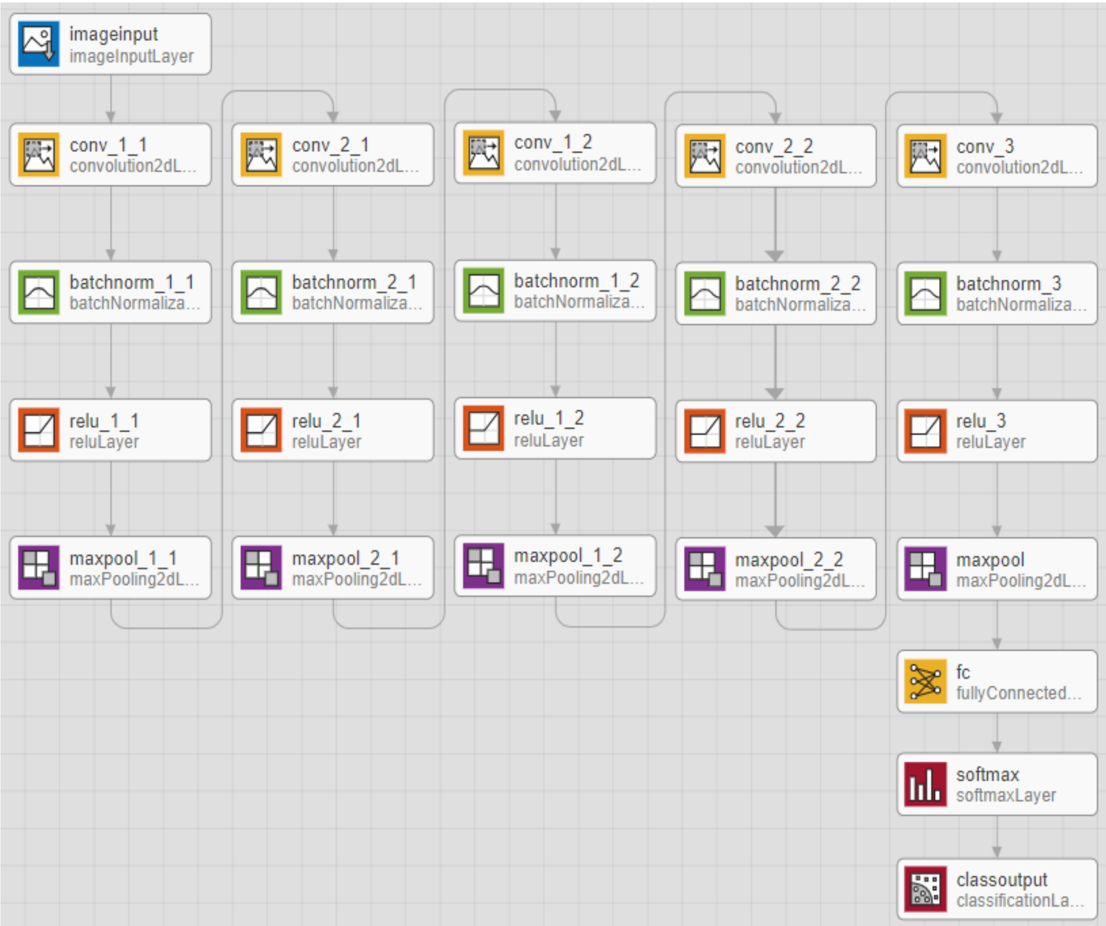

Figure 21. The neural network's architecture.

The input image of size 160 x 120 has a total of 19200 pixels. The first convolution layer conv_1_1 has 8 feature maps of size 5 x 5. The number of feature maps is increased in each convolution layer. The second convolution layer conv_2_1 has 25 feature maps, the third convolution layer conv_1_2 has 50 feature maps, the fourth convolution layer conv_2_3 has 75 feature maps, and the fifth convolution layer conv_3 has 100 feature maps. Filters in all convolution layers are of size 5x5. Figure 22 shows the learned features at the first layer which is based on pixel intensity, Figure 23 shows new recognized patterns, and Figure 24 and Figure 25 shows more detailed patterns. The final learned features by the high-level combinations of the features learned by the earlier layers is shown in Figure 26. Finally, the fully connected classifier layer classifies images into normal or abnormal as shown in Figure 27. Table 3 shows the activation details of the convolution and pooling layers of the network.

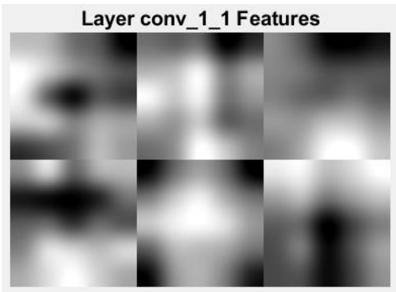

Figure 22. Learend features at the first convolutional layer.

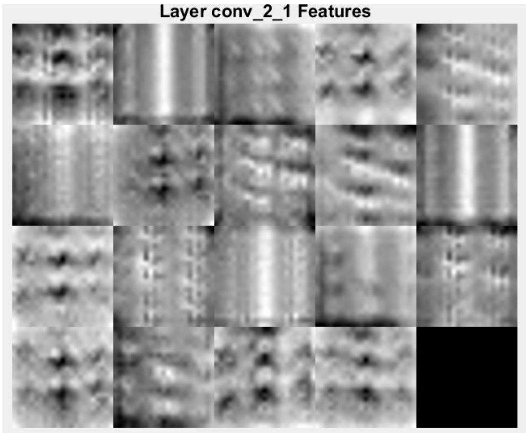

Figure 23. Learend features at the second convolutional layer.

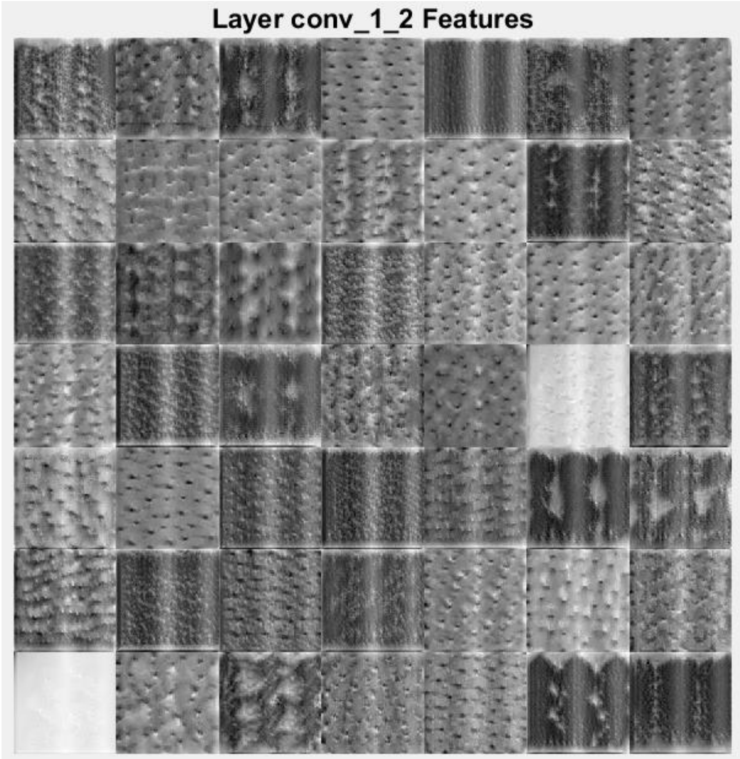

Figure 24. Learend features at the third convolutional layer.

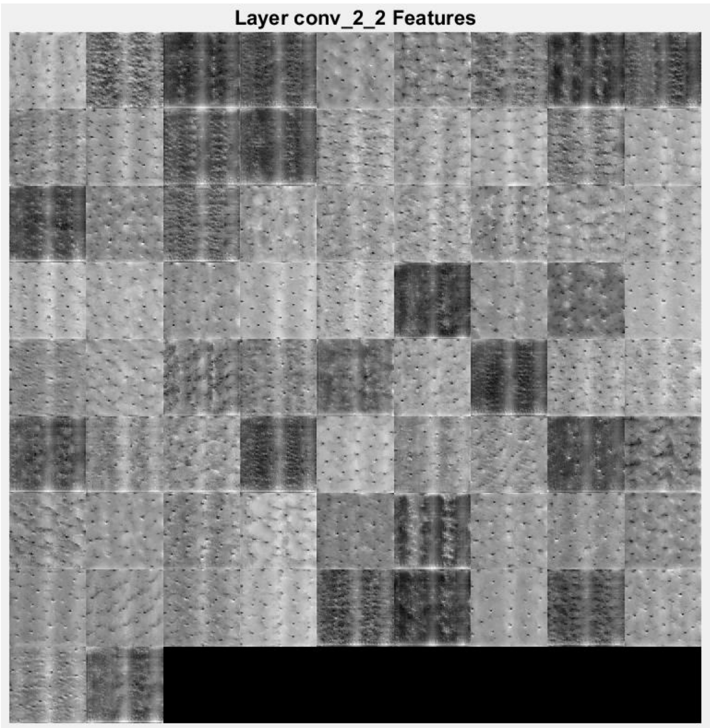

Figure 25. Learend features at the fourth convolutional layer.

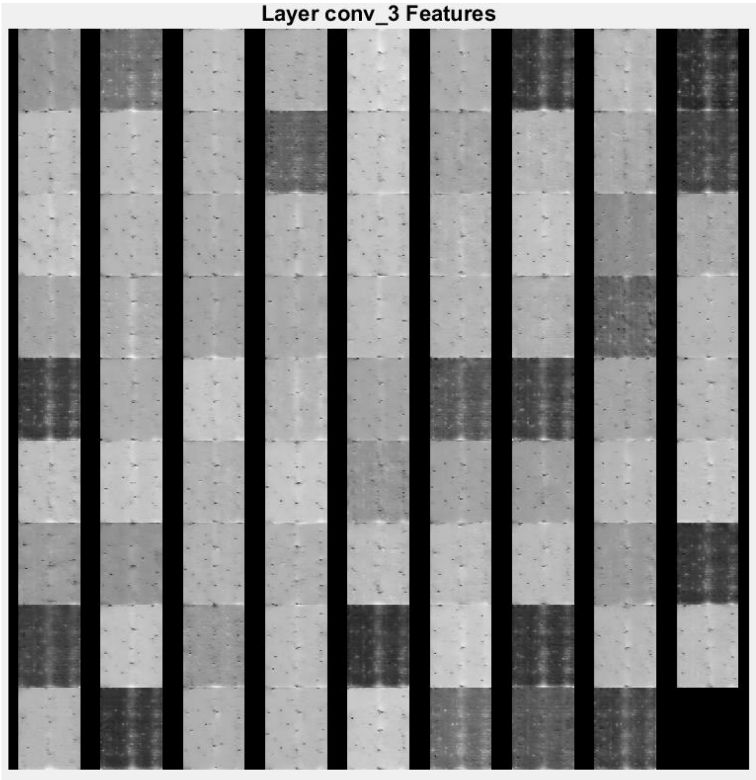

Figure 26. Learend features at the fifth convolutional layer.

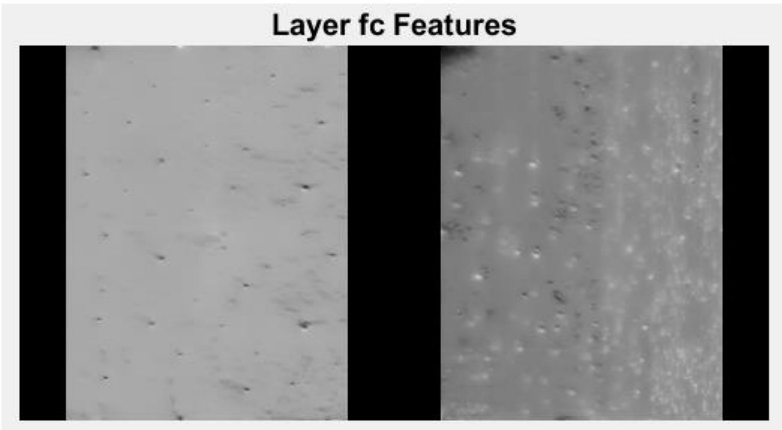

Figure 27. Learend features for classfication at the fullty connected layer.

Table 3. Convolution and maxpool layers' details

Layer	Activation Shape	Activation Size
Image Input	160 x 120 x 1	19,200
conv 1 1	160 x 120 x 8	153,600
maxpool 1 1	80 x 60 x 8	38,400
conv 2 1	80 x 60 x 25	120,000
maxpool 1 2	40 x 30 x 25	30,000
conv 1 2	40 x 30 x 50	60,000
maxpool 1 2	20 x 15 x 50	15,000
conv 2 2	20 x 15 x 75	22,500
maxpool 2 2	10 x 8 x 75	6,000
conv 3	10 x 8 x 100	8000
maxpool 3	10 x 8 x 100	8000

The neural network is trained with 60% of the data chosen randomly of the dataset, and validated with the other 40%. The neural network is trained with a learning rate of 0.02 in 128 epochs. The training has consumed 7 iterations. The mini batch loss has decreased over every iteration, and the mini batch accuracy has reached 100% on the fourth iteration. Training has consumed 12 minutes 17 seconds on a single CPU.

3.2.1. Evaluation Metrics

In this section, the evaluation metrics used for binary classification is defined.

3.2.1.1. Sensitivity

Sensitivity, which is also known as Recall or True Positive Rate, is the measure of actual positive cases predicted positive over all cases. This implies that there are some actual positive cases predicted incorrectly negative, which is known as False Negative. Therefore, Sensitivity measures the proportion of actual True Positives recalled by the classifier. Sensitivity is calculated by the formula:

$$Sensitivity = \frac{True\ Positives}{(True\ Positives + False\ Negatives)}$$

For rail track binary classification, True Positives are those images which are classified correctly as abnormal, while the False Negatives are the images that were misclassified as not abnormal.

3.2.1.2. Specificity

Specificity, which is also known as True Negative Rate, is the measure of actual negative cases predicted negative over all cases. This implies that there are some actual negative cases predicted incorrectly positive, which is known as False Positive. Therefore, Specificity measures the proportion of actual True Negatives predicted correctly by the classifier. Specificity is calculated by the formula:

$$Specificity = \frac{True\ Negatives}{(True\ Negatives + False\ Positives)}$$

For rail track binary classification, True Negatives are those images which are classified correctly as *normal*, while the False Positives are the images that were misclassified as not *normal.*

3.2.1.3. Accuracy

Accuracy is the ratio of correct predictions by the classifier to the total number of predictions made. Accuracy is calculated by the formula:

$$Accuracy = \frac{True\ Postives +\ True\ Negatives}{Total\ predictions\ made}$$

3.2.1.4. Precision

Precision is the number of correct positives out of all positives predicted by the classifier. Precision is calculated by the formula:

$$Precision = \frac{True\ Postives}{True\ Positives + Flase\ Positives}$$

3.2.1.5. Prevalence

Prevalence is the rate of all positive occurrences in the sample used. Prevalence is calculated by the formula:

$$Prevalence = \frac{True\ Postives + False\ Negatives}{Total\ predictions\ made}$$

3.3.Ad-hoc Cloud setup

Once the robot is deployed on the rail track, local inspection starts by capturing images from the cameras installed on the robot. Captured images are then classified into *normal* or *abnormal* through the binary classification system described in section 3.2. Once an abnormal image is detected, the image of the defect along with its location is stored in the cloud. On the other side, the maintenance team at the control room can view the image of the defects from the cloud, and can view the graphical location of the defect on the rail track. Based on the severity of the defect, the maintenance team can decide the required action. In this section the details of the ad-hoc cloud setup are described.

3.3.1. Experimental Setup

Cloud setup in this experiment is done over the Dropbox cloud. A Dropbox application is created over the cloud which is linked to MATLAB over the Dropbox API. Once a rail track defect is detected, the image and its location is sent to the DropBox application via the WEBWRITE function in MATLAB. The cloud connection is established via an API call done through a generated access token private to the application. At the control room, the DropBox folder is linked to the cloud using the same application's account.

To setup the experiment, 203 predefined defects from Type I dataset [7][15] are mapped to an experimental rail track plan using predefined coordinates. Figure 28 and Figure 29 show the rail track before and after mapping all possible defects. The image of each defect is named with its predefined coordinates.

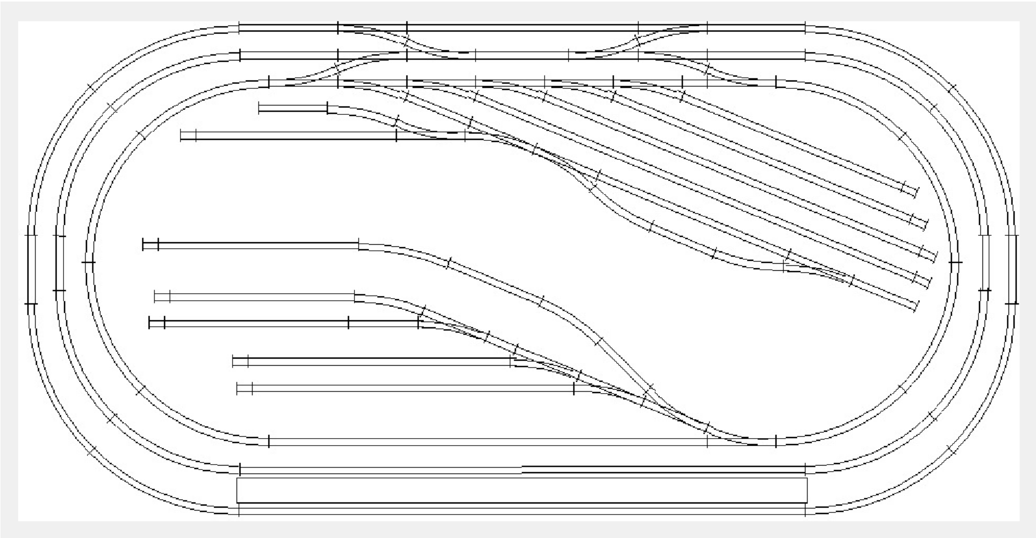

Figure 28. Rail track before mapping defects.

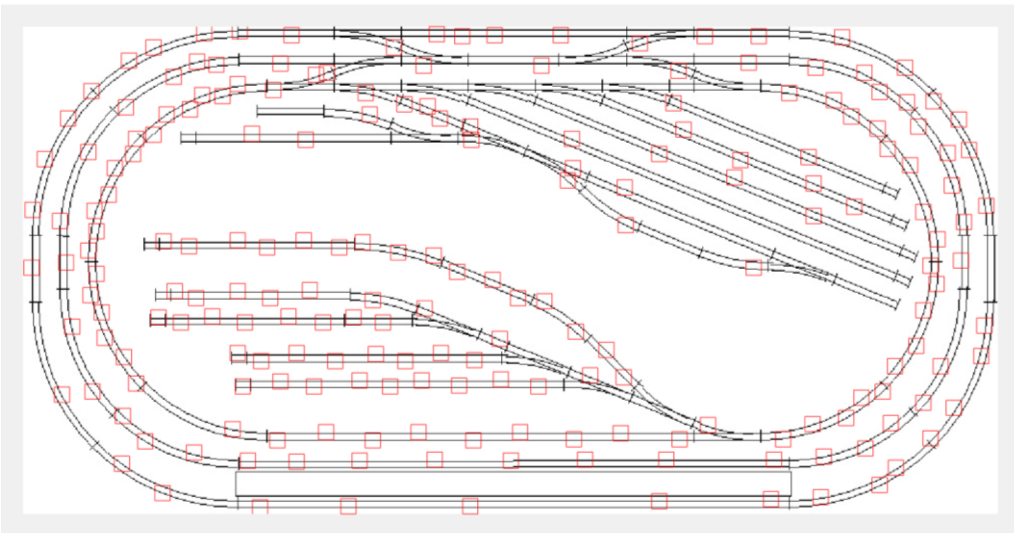

Figure 29. Rail track with predefind mapped defects.

3.3.2. Evaluation Metrics

The evaluation metrics used to evaluate the cloud connection and storage are the size of the cloud as provided by Dropbox, the ease of accessing the cloud from multiple devices at different locations, and the speed at which the data sent to and received in the cloud.

CHAPTER 4: RESULTS AND DISCUSSION

This section provides the results from the experiments on rail track binary classification and the ad-hoc cloud testing in Sections 4.1 and 4.2 respectively.

4.1 Rail track binary classification

The rail track binary classification system was tested on the Type I dataset. Training the model has consumed 12 minutes 17 second on a single CPU. The best test accuracy rate provided by the system is 97%. On average, the system consumes 0.166 milliseconds to classify one image. Figure 30 shows the training progress of the model.

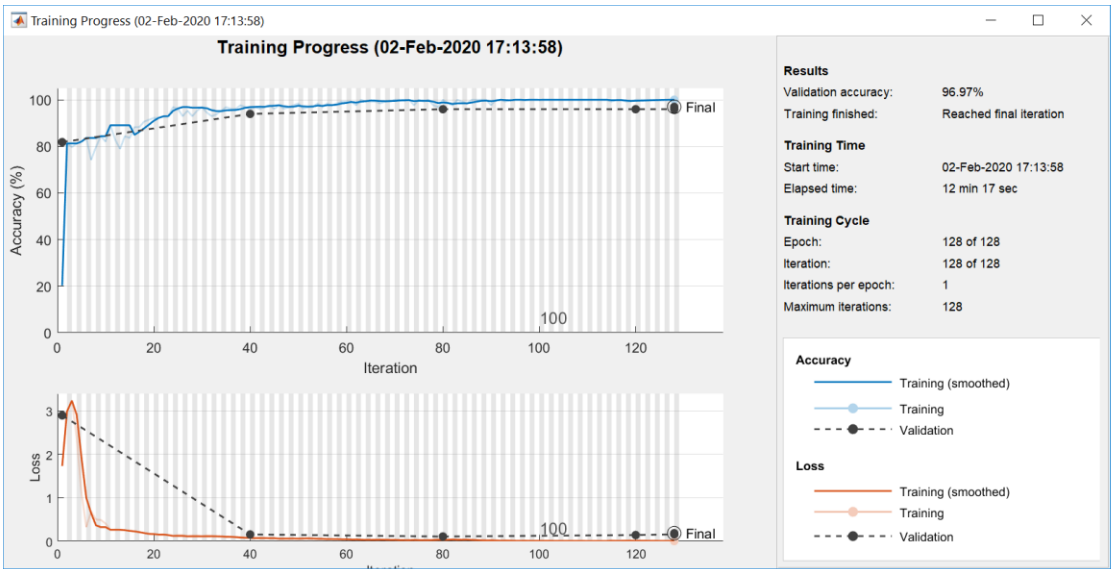

Figure 30. Training progress of the binary classification model.

The system has achieved a False Positive rate of 1%, False Negative rate of 2%, True Positive rate of 80.8%, and True Negative rate of 16.2%. For rail track binary classification, True Negatives are those images which are classified correctly as *normal*, while the False Positives are the images that were misclassified as not *normal*. Therefore, the system has achieved a significate low False Positive rate of 1%.

Figure 31 shows the result for some test data classified correctly with a significantly high accuracy rate. Table 4 shows the details of the evaluation results.

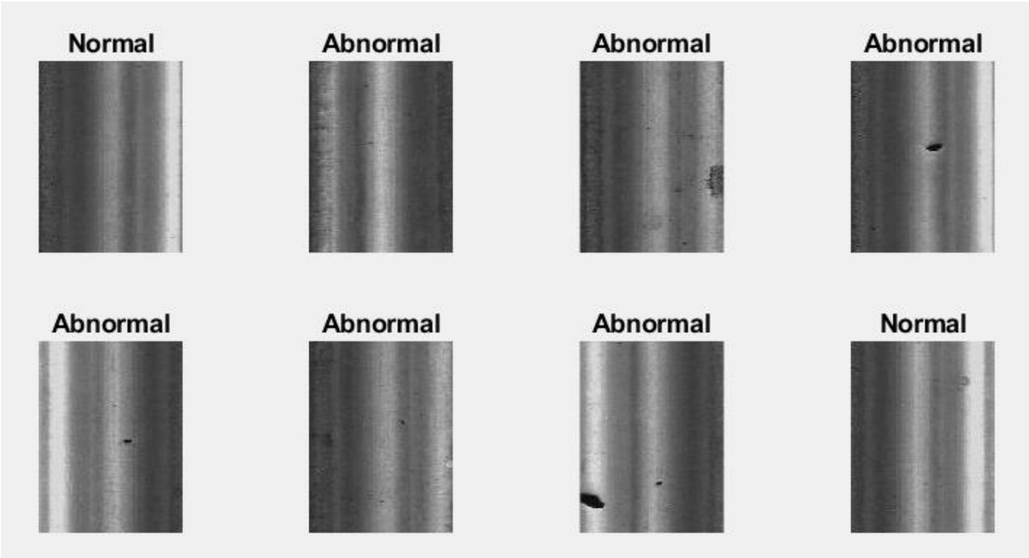

Figure 31. Sample result of the binary classification.

Table 4. Evaluation results

Evaluation Metric	%
Accuracy	97.0
Sensitivity	97.6
Specificity	94.1
Precision	98.8
Prevalence	81.8

4.2 Ad-hoc Cloud

The Dropbox cloud provides a storage space up to 2 TB and an ease of access from multiple devices be it a control room or a portable mobile device. In this experiment, data is uploaded to the cloud in an average time of 1.75 seconds, and a notification is received at the control room in an average time of 4 seconds. When an image is sent to the cloud, a notification is received at the control room indicating that new data is added to the cloud as shown in Figure 32; hence, the images and locations of the rail track defects are automatically stored in the cloud.

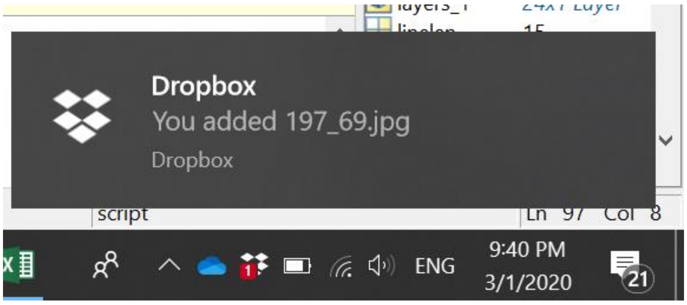

Figure 32. Dropbx notification on receiving a new data.

The cloud connection over Dropbox was established via an API call through MATLAB R2019. Once a defect is detected by the binary classifier, the defected image is stored in the cloud during an average time of 1.75 seconds. The notification shown in *Figure 32* received in an average time of 4 seconds shows how the image of the defect is stored in the cloud with the defect's location as the image's title.

Figure 33 and Figure 34 shows the data stored in the cloud from the a PC at the control room as well as from a mobile device. Once a defect is received at the control room, the location of the defect is extracted from the image's title through a MATLAB application that extracts the defect's coordinates and maps its location graphically to the rail track plan.

Figure 33. Dropbox content from a PC.

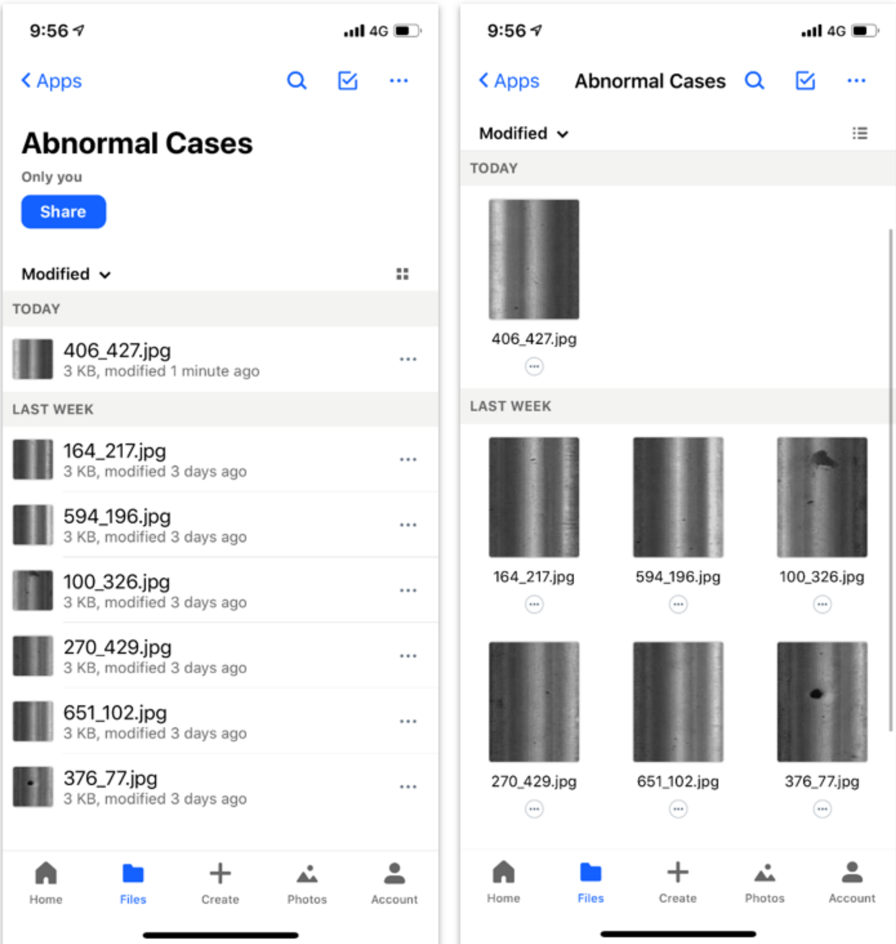

Figure 34. Dropbox content from a mobile device.

Figure 35 shows the MATLAB application that extracts the location of the defect from the images, and Figure 36 shows the defects mapped graphically to the rail track plan.

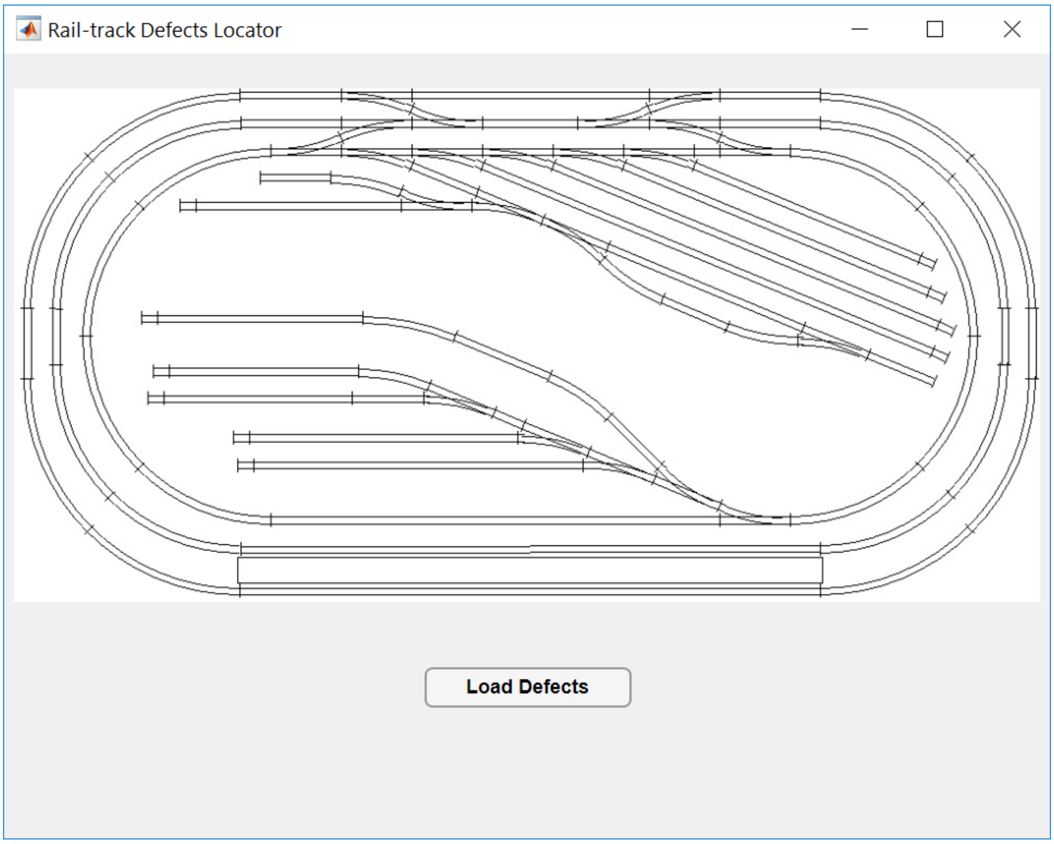

Figure 35. MATLAB GUI application at the control station.

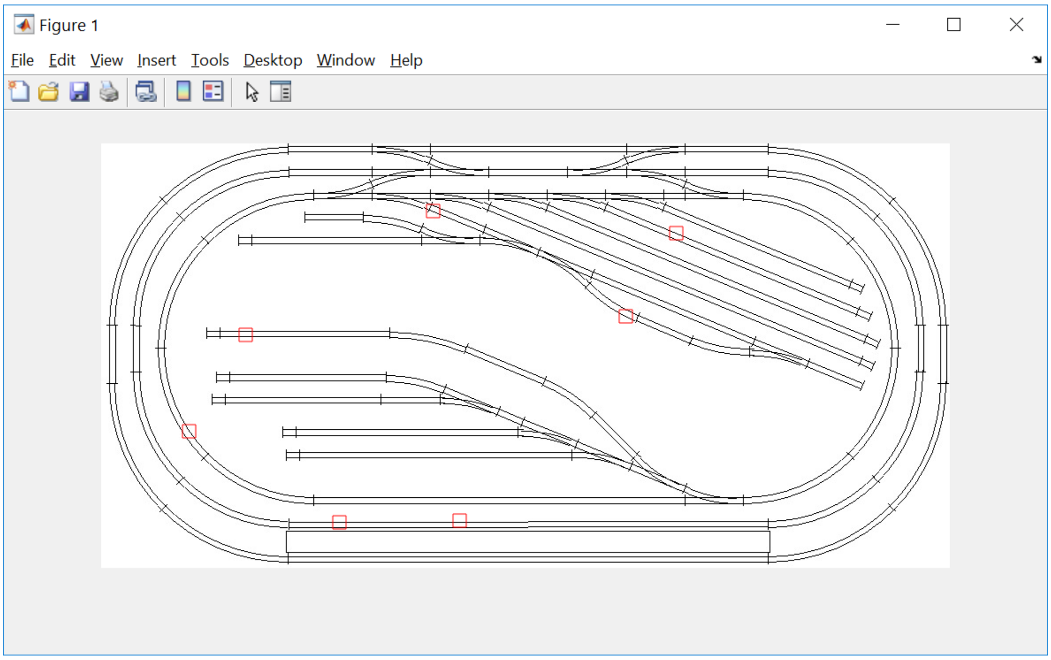

Figure 36. GUI output of defected locations of the rail track mapped to the rail track plan.

At the control room, the maintenance team can look at the defect at each location through the MATLAB application to decide which defect needs immediate maintenance. shows the locations of the selected defetcs mapped at the track, and shows the defects at the selected location.

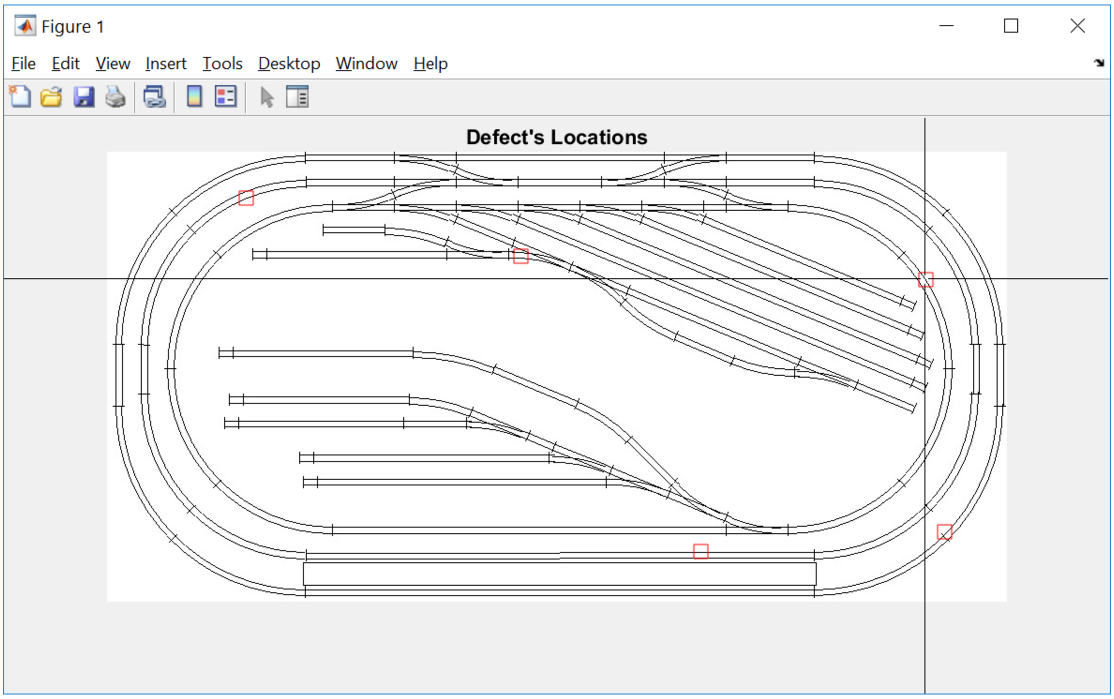

Figure 37. GUI option of viewing the defect at the selected location.

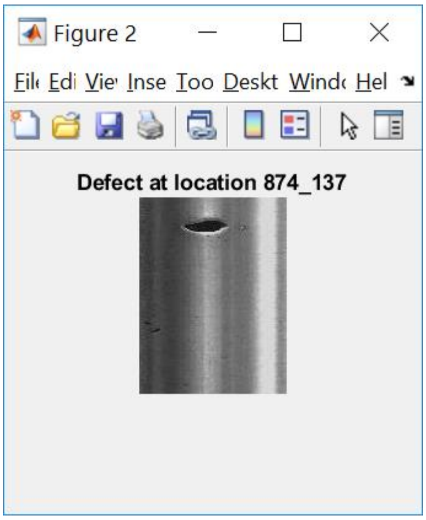

Figure 38. Defect at the selected location in Figure 37.

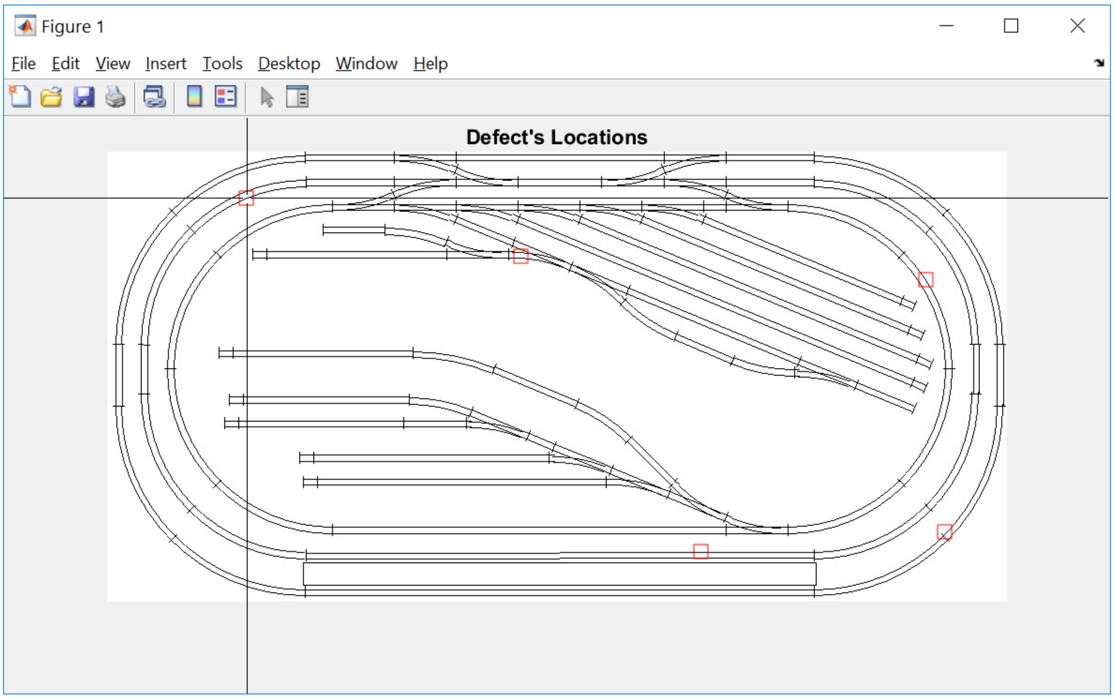

Figure 39. GUI option of viewing the defect at the selected location.

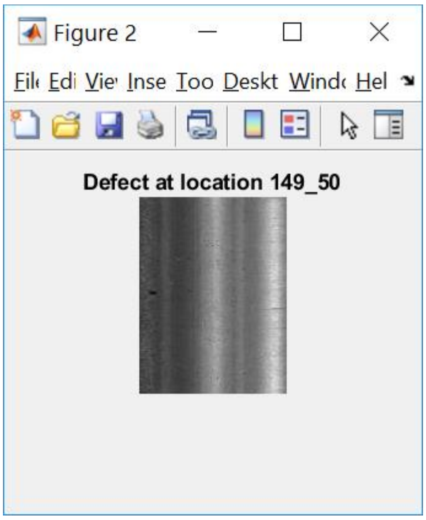

Figure 40. Defect at selected location in Figure 39.

CHAPTER 5: CONCLUSION

Rail tracks early inspection and immediate maintenance are the main major factors to ensure track's safety. Rail track defects are internal or surface defects, and most commonly results from high-speed friction and climate condition. Available inspection solutions are either contact-based that can be deployed on tracks and controlled from control rooms, or non-contact based solutions which are known as machine vision based solutions that can identify and locate cracks by analyzing images of the rail track.

This work has introduced a novel automated system for rail track inspection that integrates robotic platforms with visual inspection to detect and locate surface defects. The novelty of this work comes from providing a computer vision solution that provides local detection while inspection using 2DCNN. While the robot is inspecting, the captured images are sent to the neural network for classification and detection.

Once any surface defect is detected, it will be communicated directly to the cloud with the corresponding location for further inspection later. The proposed system has achieved an accuracy rate of 97%. In the future, more sensors will be added to detect internal defects, and the neural network will be trained on GPUs to speed up the training time and enhance the accuracy rate.

Rail transportation is new to Qatar; therefore; this research aids in providing a reliable cost-effective system that suits the climate of Qatar. The proposed system is cost-effective because it provides local processing that not only saves time but also does not congest the cloud with unnecessary data. The cloud will contain the images and locations of the defected areas only in the rail track; therefore, the dedicated operator would have very few locations to inspect as opposed to the full track. Moreover, more sensors will be added in the future to detect internal defects especially those that are caused by heat and humidity. Therefore, this research has provided a reliable cost-effective system with high accuracy rate that suits the climate of Qatar.

5.1 Related Publication

[1] N. AlNaimi and U. Qidwai, “IoT Based on-the-fly Visual Defect Detection in Railway Tracks” in *IEEE International Conference on Informatics, IoT, and Enabling Technologies*. (Accepted January 2020)

REFERENCES

[1] N. Mahfuz, O. A. Dhali, S. Ahmed, and M. Nigar, "Autonomous railway crack detector robot for Bangladesh: SCANOBOT," in *5th IEEE Region 10 Humanitarian Technology Conference 2017, R10-HTC 2017*, 2018, vol. 2018-Janua, pp. 524–527, doi: 10.1109/R10-HTC.2017.8289014.

[2] Q. Li, Z. Zhong, Z. Liang, and Y. Liang, "Rail inspection meets big data: Methods and trends," in *Proceedings - 2015 18th International Conference on Network-Based Information Systems, NBiS 2015*, 2015, pp. 302–308, doi: 10.1109/NBiS.2015.47.

[3] M. P. Papaelias, C. Roberts, and C. L. Davis, "A review on non-destructive evaluation of rails: State-of-the-art and future development," *Proc. Inst. Mech. Eng. Part F J. Rail Rapid Transit*, vol. 222, no. 4, pp. 367–384, 2008, doi: 10.1243/09544097JRRT209.

[4] C. Tastimur, M. Karakose, E. Akin, and I. Aydin, "Rail defect detection with real time image processing technique," *IEEE Int. Conf. Ind. Informatics*, vol. 0, pp. 411–415, 2016, doi: 10.1109/INDIN.2016.7819194.

[5] Z. Popović, L. Lazarević, L. Brajović, and M. Vilotijević, "The importance of rail inspections in the urban area -aspect of head checking rail defects," in *Procedia Engineering*, 2015, vol. 117, no. 1, pp. 596–608, doi: 10.1016/j.proeng.2015.08.220.

[6] S. Mittal and D. Rao, "Vision Based Railway Track Monitoring using Deep Learning," 2017.

[7] Q. Li and S. Ren, "A real-time visual inspection system for discrete surface defects of rail heads," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 8, pp. 2189–2199, 2012, doi: 10.1109/TIM.2012.2184959.

[8] Y. Jiang, H. Wang, G. Tian, Q. Yi, J. Zhao, and K. Zhen, “Fast classification for rail defect depths using a hybrid intelligent method,” *Optik (Stuttg).*, vol. 180, pp. 455–468, Feb. 2019, doi: 10.1016/j.ijleo.2018.11.053.

[9] M. Maneesh Kumar, M. S. Murali, M. Saranya, S. Arun, and R. P. Jayakrishnan, “A Survey on Crack Detection Technique in Railway Track,” in *Proc. IEEE Conference on Emerging Devices and Smart Systems, ICEDSS 2018*, 2018, pp. 269–272, doi: 10.1109/ICEDSS.2018.8544319.

[10] R. S. Shekhar, P. Shekhar, and P. Ganesan, “Automatic detection of squats in railway track,” in *ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems*, 2015, doi: 10.1109/ICIIECS.2015.7193128.

[11] “J.Kiruba Golda A.Mabel Rajakumari G.Kasthuri, E.Rosevangelin. Rail robot: Autonomous robot can clean and monitor your railway tracks data base to smart phone. - Google Search.” [Online]. Available: https://www.google.com/search?rlz=1C1CHBD_enQA884QA884&sxsrf=ALeKk01e6za1d1ViulBYy0TC5nDjBZyjVA:1583532149325&q=J.Kiruba+Golda+A.Mabel+Rajakumari+G.Kasthuri,+E.Rosevangelin.+Rail+robot:+Autonomous+robot+can+clean+and+monitor+your+railway+tracks+data+base+. [Accessed: 07-Mar-2020].

[12] K. Chittal, M. Nandhini, V. Krithika, I. N. Siyo, J. Adithyan, and M. Gowtham, “Autonomous rail inspection,” in *IEEE International Conference on Power, Control, Signals and Instrumentation Engineering, ICPCSI 2017*, 2018, pp. 2573–2577, doi: 10.1109/ICPCSI.2017.8392182.

[13] M. Chenariyan Nakhaee, D. Hiemstra, M. Stoelinga, and M. van Noort, “The Recent Applications of Machine Learning in Rail Track Maintenance: A

Survey," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2019, vol. 11495 LNCS, pp. 91–105, doi: 10.1007/978-3-030-18744-6_6.

[14] X. Gibert, V. M. Patel, and R. Chellappa, "Robust fastener detection for autonomous visual railway track inspection," in *Proceedings - 2015 IEEE Winter Conference on Applications of Computer Vision, WACV 2015*, 2015, pp. 694–701, doi: 10.1109/WACV.2015.98.

[15] J. Gan, Q. Li, J. Wang, and H. Yu, "A Hierarchical Extractor-Based Visual Rail Surface Inspection System," *IEEE Sens. J.*, vol. 17, no. 23, pp. 7935–7944, Dec. 2017, doi: 10.1109/JSEN.2017.2761858.

[16] B. Sambo, A. Bevan, and C. Pislaru, "A novel application of image processing for the detection of rail surface RCF damage and incorporation in a crack growth model," in *International Conference on Railway Engineering (ICRE 2016)*, 2016, pp. 12 (9 .)-12 (9 .), doi: 10.1049/cp.2016.0521.

[17] M. Karakose, O. Yaman, M. Baygin, K. Murat, and E. Akin, "A new computer vision based method for rail track detection and fault diagnosis in railways," *Int. J. Mech. Eng. Robot. Res.*, vol. 6, no. 1, pp. 22–27, 2017, doi: 10.18178/ijmerr.6.1.22-27.

[18] A. James *et al.*, "TrackNet - A Deep Learning Based Fault Detection for Railway Track Inspection," in *2018 International Conference on Intelligent Rail Transportation, ICIRT 2018*, 2019, doi: 10.1109/ICIRT.2018.8641608.

[19] C. Tastimur, O. Yaman, M. Karakose, and E. Akin, "A real time interface for vision inspection of rail components and surface in railways," in *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*, 2017, doi: 10.1109/IDAP.2017.8090267.

[20] S. Fu and Z. Jiang, “Research on image-based detection and recognition technologies for cracks on rail surface,” in *Proceedings - 2019 International Conference on Robots and Intelligent System, ICRIS 2019*, 2019, pp. 98–101, doi: 10.1109/ICRIS.2019.00033.

[21] C. Tastimur, M. Karakose, E. Akin, and I. Aydin, “Rail defect detection with real time image processing technique,” in *IEEE International Conference on Industrial Informatics (INDIN)*, 2016, vol. 0, pp. 411–415, doi: 10.1109/INDIN.2016.7819194.

[22] C. Taştimur, E. Akin, M. Karaköse, and I. Aydin, “Morfolojik Özellik Çikarimi Kullanan Görüntü Işleme Tabanli Ray Arizalarinin Tespiti,” in *2015 23rd Signal Processing and Communications Applications Conference, SIU 2015 - Proceedings*, 2015, pp. 1244–1247, doi: 10.1109/SIU.2015.7130063.

[23] M. Singh, S. Singh, J. Jaiswal, and J. Hempshall, “Autonomous rail track inspection using vision based system,” in *Proceedings of the 2006 IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety, CIHSPS 2006*, 2006, vol. 2006, pp. 56–59, doi: 10.1109/CIHSPS.2006.313313.

[24] X. Gibert, V. M. Patel, and R. Chellappa, “Deep Multi-task Learning for Railway Track Inspection.”

[25] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” Nov. 2015.