Novel Methods of Object Recognition & Fault Detection Applied to Non-Destructive Testing of Rail's Surface During Production

By

Qurrat-ul-Ain Malik

A thesis submitted in partial fulfilment of the requirements for Doctor of Philosophy in the School of Engineering

Manchester Metropolitan University

July 2013

Declaration

I hereby declare that the work has been done by myself and no portion of the work contained in this thesis has been submitted in support of any application for any other degree or qualification at this or any other university or institution of learning.

Moreover, no work has yet been published as per requirement of Tata Steel's Policy.

Qurrat-ul-Ain Malik

For my father, who never lets me give up!

Abstract

A series of rail image inspection algorithms have been developed for Tata Steels Scunthorpe rail production line. The following thesis describes the contributions made by the author in the design and application of these algorithms. A fully automated rail inspection system that has never been implemented before in any such company or setup has been developed. An industrial computer vision system (JLI) already exists for the image acquisition of rails during production at a rail manufacturing plant in Scunthorpe. An automated inspection system using the same JLI vision system has been developed for the detection of rail's surface defects during manufacturing process. This is to complement the human factor by developing a fully automated image processing based system to recognize the faults with an improved efficiency and to allow an exhaustive detection on the entire rail in production.

A set of bespoke algorithms has been developed from a plethora of available image processing techniques to extract and identify components in an image of rail in order to detect abnormalities. This has been achieved through offline processing of the rail images using the blended use of different object recognition and image processing techniques, in particular, variation of standard image processing techniques. Several edge detection methods as well as adapted well known Artificial Neural Network and Principal Component Analysis techniques for fault detection on rail have been developed. A combination of customised existing image algorithms and newly developed algorithms have been put together to perform the efficient defect detection. The developed system is fast, reliable and efficient for detection of unique artefacts occurring on the rail surface during production followed by fault classification on the rail imaging system. Extensive testing shows that the defect detection techniques developed for automated rail inspection is capable of detecting more than 90% of the defects present in the available data set of rail images, which has more than 100,000 images under investigation. This demonstrates the efficiency and accuracy of the algorithms developed in this work.

Acknowledgments

"All Praise belongs to ALLAH alone, Lord of all Worlds. Who created the heaven and the earth and all that is between the two and indeed in them there are many signs for those who seek."

(AI-Qur'an)

First of all, I thank Almighty God for giving me the knowledge, wisdom and self-confidence, and who endowed me with the potential and the ability to carry out this task with success.

I acknowledge the support, love, prayers of my parents who gave me this opportunity and who are behind every success of mine. Perhaps they are the major stimulating force that always facilitates me in achieving what I aspire.

I wish to express my deepest gratitude and appreciation to my commendable project advisor Dr. Nicholas Bowring. I am really thankful to him for his guidance, overall moral support, and his priceless time. His encouragement and invaluable advice was a source of inspiration for me without that any of this work would never have been possible, so my thanks go out to him especially.

I also want to thank all my colleagues in MMU, specially Dr Dalil Benchebra and Dr Christopher Johnson for their valuable time, help and guidance in all matters.

I am also very thankful to all the people who made my stay in UK memorable and made me feel like home. Special thanks to all of them for their assistance, guidance, wise advices and support.

Finally special thanks to my dear husband who stood with me all the time and helped me complete this work.

List of Acronyms

Acronym	**Meaning**
PCA	Principal Component Analysis
MVS	Machine Vision Systems
VISyR	Vision system for Real-Time Infrastructure Inspection
SIDP	Spectral Image Differencing Procedure
ANN	Artificial Neural Network
CCD	Charge-Coupled Device
RIS	Rolled in Scrap
TS	Tiger Stripes
AHE	Adaptive Histogram Equalisation
SUSAN	Smallest Univalue Segment Assimilating Nucleus
HSV	Hue Saturation Value
BPNN	Back Propagation Neural Network
JLI	Jørgen Læssøe Ingeniørfirma ApS

Table of Contents

Chapters

List of Figures

List of Tables

List of Equations

1 Introduction

1.1 An Introduction to Rail Inspection

Trains have become a convenient and popular means of travel throughout the world. In Europe alone millions of people depend on trains. Hence, safer and more secure trains are an important need of today. The rail network has a lot of areas to work on; in order to provide secure and reliable services to its users. Hence, the quality control of rail's design, production and maintenance is of utmost importance and concern. A well manufactured rail means reduced incidents and increased endurance that immediately results in more satisfied customers.

Due to the growing railway traffic, companies all over the world are interested in making an automatic inspection system to detect the defects on the surface of the rail, and also missing or damaged infrastructure of the railway tracks; such as missing or damaged sleeper fastenings. An automated system can increase the efficiency of the maintenance of railways and reduce the inspection time. Despite the fact that many advances has been made in rail engineering, production and inspection technology in the last few decades, threat to the safe operation of railways is still there due to its high usage and continuous increase in its service demands. Efforts are being made to check the rail manufacturing quality and understand the rail behaviour when submitted to the stress of hundreds of trains running on it. Serious research is being carried out to detect and treat the faults on the rail both during production and on site; before they become a threat. Researchers around the world are concentrating on different aspects to ensure more reliable rails.

An important part of the modernization of the rail industry is the capacity to use advanced techniques drawn from digital image processing to reduce the time and cost of both rail production and rail maintenance. In this context, rail inspection for defects on a production line has become a hurdle in the constantly evolving rail industry, as it still depends on humans who have not the capabilities for exhaustive, fast and accurate inspection. There is need of a tool for high speed assessment of rail quality and defect detection to complement the ageing methods. Such assessment would permit a complete inspection of the rails and increase the accuracy and the speed of defect detection. This thesis is a step towards the completion of such a tool.

In this thesis, a progressive approach has been used to investigate the surface defects that occur during the rail production process and develop adequate solutions to detect them. At first, the rail inspection system as existing on the production line has been characterized. It has been decided to use the existing camera acquisition system named JLI vision system, and provide software to analyse rail images. An exhaustive list of defects to be detected has been provided by Tata Rail Technologies – the sponsor of the project. There are many defects that require special attention both in production and maintenance. However, the manufacturing defects, which occur during the production process of the rails, need to be detected before the rails can be put on site, are the main topic discussed in this research. A set of the most frequent defects that can occur on rail surface, during production, have been identified and studied. Estimation by the experts in the field suggest that more than 95% of all the defects that subsequently lead to substantial threats can be identified into four types. Rolled in scrap, tiger stripes, gouged lines on the top of rails and wire defects, are the main production line defects. Detailed descriptions with the examples of these four faults have been presented in chapter 5. This thesis discusses these defects and a designed algorithmic tool capable of finding them.

A machine vision system in the production process could replace the visual assessment of the rail surface quality currently done by technicians. Currently, from images of rail rolling out of a manufacturing mill, only one out of five to ten rails are being checked by a human operator. Many defects might therefore be missed and a vision system capable of efficiently inspecting all rails would be a great step toward safe and secure rails during the manufacturing process. A machine vision system in maintenance would augment manual visual inspection by providing alternatives to laborious and labour intensive inspection that sometimes fail due to human operator task monotony. The rail inspection would not only be more reliable, secure and consistent, but also it would prevent rail workers from being in a hazardous and potentially deadly environment.

The role of the machine vision systems would be to process rail images in order to extract and classify anomalies both in production and maintenance. Rolling contact fatigue cracks, wheel burn marks, squats, corrugation, sleeper anomalies as well as missing sleeper fastenings and associated nuts and bolts; are the defects that need to be detected for track maintenance (Liu et al., 2010). Rolled in scrap, pin wire defects, line on the top and many other faults are the manufacturing defects (Mandriota et al., 2004). An automated machine vision system may have the potential to detect a comprehensive set of these types of defects. The algorithms used to detect these anomalies range from simple edge detection techniques to more complex feature extraction and automatic pattern recognition algorithms such as Artificial Neural Networks (ANNs) (Ruvo1 et al., 2008) and Template matching (Cole et al., 2002). Typical applications of such methods range from the recognition of missing rail objects in scenes of varying complexity to the detection of faults found on the rail surface and their classification (Sawadisavi et al., 2008)(Edwards 2009) (Resendiz et al., 2010).

The software being developed is to emulate the expert inspector in discriminating faults occurring from rolling mill to those occurring naturally at other stages of a rail's production. Prior studies using standard edge filters and object detection techniques such as Canny have been applied to rails (Babenko 2009). However, all those techniques when applied to the image types presented in this report, results in considerable false alarms (that may be referred as over detection). Hence, new approaches needed to be considered and tailor fitted to each fault /defect.

A comprehensive defect examination procedure is therefore necessary to facilitate the development of suitable algorithms. It has been done through the blended use of different object recognition and image processing techniques that include different edge detectors, filters, thresholding, Artificial Neural Networks, Principal Component Analysis and other possible effective imaging techniques. The diversity of faults such as shapes, textures, positions, etc. has required the use of existing digital image processing techniques as well as the use of novel algorithms for the detection and classification to be a reliable process, i.e. with few false positives and negatives. Hence, the thesis involves efforts done to improve the existing algorithms and developing better and more efficient new methods, to detect rail surface defects caused during its production; fit for the particular purpose described here.

1.2 Research Objectives.

The basic aim of this research is to create fast, reliable and efficient methods of fault detection for the unique artefacts produced during the rail manufacturing. The main purpose is to reduce the need for manual visual inspection in favour of a reliable and faster machine vision system. Hence, objectives are to find a more effective and efficient ways for locating and classifying defects on the rail system.

An automated rail inspection system is to be achieved for detection and classification of defects occurred on the rail surface while in production, from the images obtained from the mounted digital cameras (Stella et al., 2002). Another strand is that of defect detection and classification on rail structure on a production line using digital images available through a high speed line scan camera inspection system (R. T. C. limited 2006)(Deutschl et al., 2004).

The basic objectives of the research can be summarised as follows:

1. To test various existing object recognition techniques on rail images, comparing their performance, find drawbacks or problems with them, with a view to finding improved techniques.

2. Explore existing fault detection and identification techniques and assess their efficiency when applied to rail images.

3. To suggest new techniques or make effective changes in existing methods to produce better or more efficient results in both object recognition and fault detection/identification.

4. Ensure the new methods are robust for industrial use.

5. Exploration of the range of application of the new methods.

1.3 Structure of Thesis

The thesis has been structured as follows. Chapter 2 introduces machine vision systems and rail inspection. Chapter 2, the literature survey covers the relevant major developments and advancements in the field of rail inspection. It discusses common rail inspection methods in use as reported in the literature, concentrating mainly on machine vision rail inspection methods and technology in use or under investigation. Chapter 3 is the introductory chapter for the vision system installed at the site and the main defects under investigation. Defect detection software that has been developed for this research, makes use of the existing vision system setup, called JLI vision system. The first part of the chapter gives an insight into the JLI vision system, viewing camera locations, image saving formats, system analysis and limitations. The second part of this chapter describes rail defects, their important features, and the representative images.

Chapter 4 describes, discusses and evaluates the general image processing techniques that have been initially explored, tested and used at various detection stages of rail inspection. This chapter has been sectioned into image pre-processing, processing and post-processing methods. All the tested techniques, resulting images produced, their evaluation and significance have been covered.

Chapter 5 presents the description of developed detection algorithms of rail inspection. Individual defects, step by step detection methods and the results produced have been provided in detail in this chapter. A combination of customised existing image algorithms and newly developed algorithms have been put together to perform the defect detection. The developed algorithms are efficient and reliable.

Chapter 6 explains an alternate rail inspection method to the ones discussed in chapter 5. The alternate method is based on using Principal Component Analysis (PCA). This chapter concentrates on explaining an attempt to design an adaptable and trainable single technique for classification of all defect types. This chapter explains principal component analysis (PCA) features and their characteristics followed by classification of those features using neural networks.

Chapter 7 presents results from the algorithms described in chapter 5 and chapter 6. This chapter analyses and evaluates the performance of developed detection methods. This chapter also discusses processing time, detection success rate and the comparison of various algorithms used. A fast, reliable and highly efficient automated rail inspection system for the detection of unique artefacts found on the rail during its manufacturing process has successfully been developed.

Finally, chapter 8 summarizes and concludes the thesis and also highlights the future perspective of the current research done. The defects investigated in this research have never been discussed before and the devised solution has not been implemented before in any such company or organisation. The automatic detection of these defects is the basic novelty of the work.

1.4 Contribution to Knowledge

Fully automated defect detection software that makes use of the existing setup, and which has never been implemented before in any such company or setup has been developed. This is the basic novelty of the work. The developed automated detection system performs the detection using a bespoke combination of existing algorithms that have been customized specially for the purpose, and new image processing based algorithms for detection of the defects.

The objectives set at the initial stages of the research have been fully met. All the faults under investigation were thoroughly explored individually. Their feature characteristics were observed and studied in detail in order to detect them properly.

Various object recognition techniques were tested on the rail images; their performance was compared and evaluated. Final detection techniques have been developed based on the evaluation results obtained from the previous techniques. It was quite a lengthy process and most of the existing techniques failed to give satisfactory results. Hence, new techniques have been developed using a bespoke combination of effectively altered existing techniques and several newly developed methods to perform accurate defect detection.

Finally, the new developed methods were tested on a huge data set to confirm if they are robust for industrial use. This was again a very lengthy procedure as there were four different defects and for each type, there were so many images available that needed to be tested by the software. Random testing of the software was also performed at the site by the Tata steel experts on time to time basis during the research to confirm the effective working of the software.

Eventually fully automated system which is fast, reliable and efficient for detection of surface defects on the rail, that occur during production followed by fault classification on the rail imaging system have been developed.

2 Literature Review

2.1 Introduction

Significant research has been done to develop and improve automated applications to perform track inspection around the world. The growth in railway traffic has made railway maintenance operators interested in providing automatic methods of railway inspection; such as a Machine Vision inspection System (MVS). An MVS can be used to control both the quality of rail during production and on-site inspection of rail.

Rail defects during manufacturing or production process can include surface defects such as rolled in scrap, wire and line defects, tiger stripes etc. Meanwhile, on-site rail inspection looks for features such as sleepers and the missing rail infrastructure as well as inspecting the rail head for surface bound defects. Currently the most of the rail inspection both installed and during production is manual, which can be both dangerous and time consuming. Hence, an automated system can increase the efficiency of the maintenance of railways and reduce the inspection time. Machine vision for automated inspection using modern digital cameras together with high speed communications and very high processing power, extended the vision application to a commendable extent.

The research described here primarily is about the automated detection of defects mainly caused during the production. There is no specific literature available on rail inspection during the manufacturing process especially on the types of defects being considered in the current research. However, the machine vision methods explored and currently being used to detect on-site rail defects have been found to have significance for the detection of the defects being considered for the current research. Hence, the literature survey that has been done for the research covers the evolution of rail inspection by machine vision, current rail inspection technologies that are in use around the world and concentrates mainly on machine vision inspection of metals, mostly rails and rail parts.

2.2 Machine Vision Systems

> "Machine vision (MV) is the process of applying a range of technologies to provide imaging-based automatic inspection, process control and robot guidance in industrial applications" (Steger et al., 2008) (Graves et al., 2003).

MV may be explained as the analysis of images using highly efficient machines such as high resolution cameras, computers and image algorithms etc. to extract data for controlling a process or activity. Machine vision systems are currently in use or under development for different type of inspections in various industries including railways. Its advantages are: high speed, efficiency, reliability, consistency, ability to store, process and recall large quantity of data and no human dependency. Disadvantages can be the unusual or unseen situations that might occur at any stage in the future.

2.2.1 Fundamental Steps in Machine Vision Systems

Machine vision systems mainly have three basic sub systems or works in three main steps:

1. First step is the image or data acquisition system.
2. Image analysis or processing that has further two parts; image enhancement and feature extraction. In the image enhancement or pre-processing stage, edge enhancement and noise reduction can be done by applying specially designed filters to the image. In feature extraction or the image processing

stage, data are processed using various machine vision algorithms to detect and identify areas of interests. There is a detailed chapter on image processing later in the thesis.

3. Classification stage: The features extracted from the data during image analysis stage are classified in the final stage; which could provide basic information for identifying objects. Classification techniques have been discussed in detail in the later chapters.

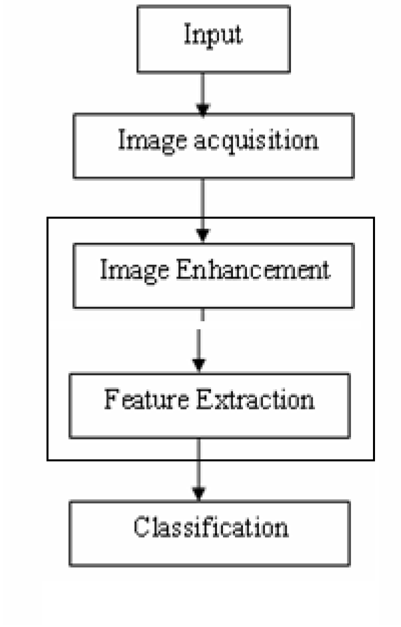


Figure 2.1: Components of typical image processing system

2.3 Machine Vision Cameras

Machine vision cameras acquire images in a video or still image medium and in the visible or infrared spectrum. There are mainly two types of machine vision cameras:

1. Area Scan Cameras: Area scan cameras read the image as a whole or frame by frame. Moving systems use this type of camera to avoid blurred images.

2. Line Scan Cameras: Line scan cameras need a continuous stream of data to acquire images. They are high resolution cameras used for very fast moving

systems.

2.3.1 Machine Vision Lighting

The quality and correctness of lighting are decisive aspects for any MV inspection system. The use of proper lighting is as important as the use of the right camera or the right lens for MV image acquisition. There are various factors that help to design a proper lighting system for MV. These factors include inspection environment, camera and lenses used, other light interactions etc. (Martin 2007). In an MV system the lights coming from the environment can be detrimental because of the variations in it. Controlled lighting is a recommendation for an MV inspection so as to highlight required features.

The importance of proper lighting is evident from the currently working machine vision systems using special illumination systems. The Vision System for Real-Time Infrastructure Inspection (VISyR) system is fully automatic and is FPGA-based vision system to inspect real-time infrastructure. VISyR is a patented real time Visual Inspection System for Railway maintenance uses a high resolution line scan camera to acquire images using a suitable illumination setup, equipped with six OSRAM 41850 flood light sources to overcome the effects caused by changes in the natural light. (Stella & Marino et al., 2007).

The Federal Rail Road Association America is using a machine vision based joint bar inspection system. The inspection system uses high speed, high resolution line scan cameras with a lighting system to capture the images of joint bars from the moving vehicles. The acquired images are then processed by a computer using imaging methods and defects are finally reported. The lighting system uses xenon lights, which are mounted on a beam outside the vehicle at a position where their emitted light pattern is in a line with a view of the camera. The lighting system provides uniform and consistent illumination and also overcome the effect of ambient light. The position and intensity of the lights have been set up in a way to provide most favourable lighting conditions for both the contrast and brightness. The system is efficient which is mainly due to the proper lighting system being used. (Berry et al. 2008).

2.4 Rail Inspection Research

Efforts are being made to check the manufacturing quality of the rail and understand the rail behaviour when submitted to the stress of large number of trains running on it. It is very obvious that the rail inspection is not something new, it started in the early 19th century. Researchers around the world are concentrating on different aspects to ensure more reliable rails. Many different image processing and other techniques have been tested and used at different times for the rail inspection.

Various form of literature is available on rail inspection ranging from visual, guided waves, ultrasonic, eddy current, magnetic induction and the new machine visions systems. Automated machine vision techniques and hybrid systems have been introduced recently for the high-speed inspection of rail tracks. Accelerometers, ground penetrating radar, light detection and ranging systems are among the emerging rail inspection systems.

However, It is noteworthy here that the defects currently under inspection in this research have never been researched or considered before. Hence, there is not any literature available specifically addressing on these types of defects that are being investigated for this work. Still there is related literature available for similar types of artefacts e.g. cracks, blobs on the metal surface in other areas, which has been reviewed in this section.

2.4.1 Common Rail Inspection Methods

Rail tracks are usually inspected for defects or missing components either visually, or by using ultrasonic testing, eddy current testing, magnetic flux and particle based methods. (Clark 2004) (Papaelias. et al., 2008)

Rail inspection has extensively applied non-destructive evaluation techniques, such as guided waves **(**Wilcox, P. et al., 2003) and ultrasonic waves (McNamara & Scalea 2002) (Kenderian et al., 2003) (Palmer 2005) (Rizzo et al., 2009) (Rizzo et al. 2010). The potential of guided waves and ultrasonic methods to inspect the defects on the rail has been vastly explored. Guided wave inspection has been found better than ultrasonic inspection as guided waves can propagate and penetrate to the longer lengths and are more sensitive to some cracks in a particular plane. (Wilcox et al., 2003). The use of guided waves was found to be quite reliable for the detection of both surface and internal defects; being fast at the same time. However, it is found that lower frequencies are unable to detect small defects. The ultrasonic based detection technique is found very effective but again ultrasonic fail to detect surface and the near surface defects. Poor resolution of ultrasonic's results in decreased efficiency.

At Jindal Steel & Power Ltd., India, an efficient and reliable defect detection ion system has been installed. The system uses laser technology to detect straight lines and edges of the railway track surfaces, eddy current and ultrasonic testing systems for defecting typical defects and also some internal imperfections on the railway tracks being manufactured in the company. (Raj et al., 2012)

Several alternative techniques have been researched, used and few are still under development for the rail inspection, such as electromagnetic induction (Clark 2004) (Edwards et al. 2006), eddy Current (Heckel et al. 2009), radiography and AC field measurement techniques. (Rose et al., 2002) (Gilchrist, 2006) (Innotrack 2008) and Acoustic emission (AE) techniques (Yılmazer et al., 2011)). However, an automated machine vision inspection system is the basic objective of rail inspection problem under investigation for this project.

2.4.2 Current MV Rail Inspection Technologies

Many machine vision systems are currently being used for various types of rail inspections and there are still many under development. ENSCO, Inc. developed a system for joint bar inspection. This system uses a high speed and high resolution line scan digital camera with some lights, all mounted under the vehicle to acquire images. Acquired images are processed by a powerful computer using some adaptive image processing algorithms to inspect the joint bar defect. The system finds external cracks in joint bars with 80% efficiency under ideal conditions. (Nejikovsky et al. 2005).

Another machine vision system, using area scan cameras, is under development by National Taiwan University of Science and Technology to detect elastic rail clips. The system pre-processes the acquired images to get noise free grey scale images.

Grey level variation identifies the rail position and wavelet transform finds the elastic clips. Finally, after thresholding and morphological processing, clips are identified using object recognition and other image processing methods. The results proved it to be a reliable and feasible visual recognition system (Hsieh et al. 2007).

A variety of machine vision systems are in use for rail and track inspection. Researchers in America have developed an automated system to inspect surface cracks in the rail, missing plates, fastening objects and improper gauge. Video data are gathered by area scan camera with strobe lights, lasers and sun shields (Resendiz et al. 2010). There is another system called Aurora system (Wamani & Villar 2009), which is mounted on high-rail vehicles, to inspect wood and concrete ties for rail seat abrasion, missing fasteners and improper gauge.

MERMEC Group, an Italian company for rail inspection and diagnostics has developed a machine vision system known as “Track Surface Detection System” to detect track defects. The system uses line scan cameras, works in three modules and is capable of functioning at speeds up to 160 km/hr. It can be installed on any vehicle and can inspect and classify tie type, fastening and surface defects, and ballast irregularities etc. (Sawadisavi et al. 2008) (Edwards 2009).

2.4.3 Automated MV& Hybrid Inspection Systems Survey

Automated vision techniques and hybrid systems have been introduced recently for the high-speed inspection of rail tracks. Mandriota et al., (2001) proposed a machine vision inspection technique based on texture analysis of rail surface. The texture analysis was performed using Gabor filtering to detect and classify a particular class of rail defects known as rail corrugation. Later on, they presented a machine vision system using PCA with neural networks for the detection and classification of the missing bolts from the rail. A line scan camera was used to capture the images which were then processed using wavelet transforms and principle component analysis to represent them in reduced number of coefficients. Artificial neural networks have been finally used for the classification (Stella et al. 2002). Filter-based feature selection was analysed further in the later years to perform texture analysis for the rail corrugation detection. Gabor filter, wavelet transform and Gabor wavelet transform were tested to extract the textural features from the images and results were compared. Gabor filter was found to be the best of the three approaches. (Mandriota et al., 2004). Another approach to detect rail defects using wavelet transformation was presented. Wavelet coefficients were extracted from the signals acquired by magnetic coil and Hall sensors. These coefficients were then used to differentiate healthy rail from defective rail (Toliyat et al., 2003). The results presented in the various papers show very good detection rate and closely related to current work, hence the approach can be very useful. Textural analysis approach and use of principal components and artificial neural networks all together can give a successful direction to the current research.

Shrutisagar and Hayagreev (2004) explored an automated dual track approach for detecting defects on surface and subsurface regions of a rail. A Canny Edge Detection algorithm was used on a real-time basis to detect the surface defects that are visually visible. Ground penetrating radar was used to observe the substructures such as thickness of the ballast and sub ballast layers, variations in layer thickness along the track, pockets of water trapped in the ballast etc. A line scan camera with a light source to get the video images, which are to be analysed using imaging techniques, gives a specific direction to the current research.

Deutschl et al (2004) developed an automatic vision based inspection technique for rail surface defects. Colour line-scan cameras and a special image acquisition method called spectral image differencing procedure (SIDP) followed by image processing methods were used to detect the rail surface defects. The image acquisition part consisted of a colour line scan camera with two monochrome light sources, blue and red, positioned before and after the camera in the direction of the movement. The resulting light reflection appears uniform on flat surfaces but different on non-flat surfaces. The final image was computed by a non-linear differencing function, that weakened the small differences but intensify others, resulting in an image called SIDP. The SIDP image is then segmented into defective and non-defective areas, using convolution filters and some morphological image techniques. The system produced reliable results even for heavily scaled surfaces and is already installed as an inline system at a rail manufactory. A special illumination system with post image processing is the key idea of the research. Convolution filters and morphological image methods used will be useful for this work.

Nejikovsky et al (2005) developed a very effective and feasible video system for automated joint bar inspection from a moving vehicle. The system uses high speed line scan cameras mounted under the vehicles, which capture high resolution images of the join bars. The images are then presented to an adaptive image processing algorithm that notifies the vehicle operator whenever it encounters any crack. The operator reviews the image of the crack and gives his confirmation or rejection on the issue. It is an automated machine vision system using series of imaging techniques and some adaptive approaches to detect cracks in the joint bar. The system is functional and works at a very high speed but false detection rate is high ("40% of detected cracks were confirmed and 60% were rejected by the system operators").

Frayman, Zheng, and Nahavandi (2005) used a combination of image processing algorithms for the automatic inspection of surface defects in aluminium die casting. A web cam and ring light in a box are used for image acquisition module. Image information processing is performed using genetic algorithm to extract parameters for optimal structuring element and thresholds for segmentation and noise removal. These parameters are then used for morphological operations to perform detection. The system is 99% accurate and being used in industry as part of normal production. However, the results can be improved by improving image quality simply by using high resolution cameras instead of a web cam. The system gives an insight into variety of image processing algorithms and their usefulness. Genetic algorithms and morphological operations used can be useful for the current study and might give good results. Moreover, the importance of uniform lighting is also evident from the system.

Singh, et al., (2006) proposed automated machine vision system for railway track inspection. Their work concentrated on detecting missing and replaced clips. A video camera records the condition of the railway track and the video produced is analysed using imaging techniques to perform the detection. A Gaussian smoothing is used to remove noise from the images; canny edge detector detects the edges; and finally lines of required length are stored while rest are ignored. As clips are found on a well-defined area on the track, the edges on those images are analysed and values greater than some threshold value determine the absence or presence of the clips.

Gaussian smoothing and canny edge operator are quite useful and are tested for detection of defects under investigation in this work.

Yella and Gupta (2007) presented an approach for automating human condition monitoring procedure for wooden sleepers. They proposed automatic testing procedures using two methods; impact acoustic analysis and vision analysis. Acoustic examination gives sharp sound when struck with an axe or hammer. A hammer was used to strike the wooden beam and the signals were recorded on the computer as a WAV file. Then using the signal processing techniques; the signals were interpreted and features were extracted. In the machine vision method two sets of images were acquired by the help of the camera placed on the inspection vehicle. Three features (number of cracks, average crack length and width) were extracted from the first set and a measure of how far the plate had gone into the sleeper from the second set of images. All the features were fused to form a single vector which was then passed to pattern recognition stage. However, there was destructive testing and disagreement on certain detections. But the suggestion made for the use of sensors on an intelligent vehicle that runs and collects the data, processes it and notifies the operator at the same time for the future work, was useful Marino and Stella (2007) developed a fully automatic and configurable FPGA based vision system for real-time infrastructure inspection called ViSyR. The ViSyR works at a very high velocity and is highly flexible, configurable due to an FPGA-based hardware implementation. It is capable of detecting the defects of the rails and at the same time presence or absence of the fastenings accurately. It acquires images using line scan camera, camera link protocol and a frame grabber. Image acquisition was done under a lighting setup to reduce the effect of variable natural lighting conditions. Here, the system performs rail defect detection and tracking in two phases. Firstly, Principal Component Analysis (PCA) reduced the data, which in the second phase was fed to a neural network to perform the classification. The detection of missing fastening objects was performed in three phases, a prediction phase that predicts the areas that might have the bolts missing, followed by a data reduction phase which used 2 dimensional wavelet transform, and finally a neural network based supervised classification phase, which showed the missing bolts. A very helpful research about rail defect detection based on PCA and neural networks that will be useful for the current work.

Edwards et al., (2007) investigated the use of machine vision to inspect railcar safety appliances using combination of machine vision algorithms. It was concluded that use of machine vision enhances both the performance and speed of inspection and reduces the cost at the same time. Later on, J. Riley Edwards and Steven Sawadisavi (2008) started a project to use machine vision to detect irregularities and defects on the rail tracks and fastening objects. The objective was to perform the inspection according to the regulations of American Federal Railroad Administration. They used edge detection and texture information to detect rail, ties and tie plates. They proposed that in addition to machine vision, interim approaches to automated track inspection will surely make the inspection system more effective and efficient. This was to be achieved by using vehicle mounted cameras for capturing digital videos and performing image enhancement procedures. The work was aimed to improve the existing algorithms and the hardware. It provides very useful information related to the automated machine vision inspection of rails together with image processing procedures, as machine vision inspection no doubt enhances efficiency and speed of inspection.

Papaelias, Roberts, and Davis (2008-2009) researched on various non-destructive evaluations (NDE) techniques for inspecting internal and surface defects of the rail. They presented an idea was to use automated optical cameras and a current sensor to detect the surface defects damage and to use ultrasonic inspection for internal defects. Later on in 2009, they applied ACFM (alternating current field measurement) technique to detect the induced notches that successfully detected the majority of the abnormalities. The paper suggests use of automated camera and various techniques for rail inspection which forms basis of the current research.

Jie. et al., (2009) presented a vision based inspection technique for detecting particular kind of rail head surface defects, which are called Rolling Contact Fatigue (RCF) defects. Automatic detection system constituted of pre-processing, defect location and identification, followed by a post-processing subsystems was implemented. They also proposed a simple and fast algorithm to locate the defects which was found to give better results and higher precision. It involved some geometrical analysis adopted directly on a grey-level histogram curve of the smoothed rail head surface image. The paper gives a good insight into negative effects of noise in the images and how they can be reduced to make better detection possible.

Liu, Wang, Zhang and Jia (2010) proposed automated machine vision system for rail surface defects inspection. They mainly worked on spalling of rail head and surface cracks. The system involves noise removal by applying mean filters followed by extraction of an accurate defect region. The defects are then identified by dynamic threshold and feature matching. Finally, the detected defects are evaluated by calculating percentage of wear of rail head and length of cracks in surface. It is a very recent useful paper in reference to noise removal and feature matching procedures that will be helpful for the current work.

Li et al., (2011) proposed an automated machine vision inspection and monitoring system for a railroad company. The proposed system has four modules for data acquisition, track condition monitoring, defect severity analysis and long-term predictive assessment. Data acquisition captures videos of the track using the cameras mounted on a moving vehicle. The data are transferred to the next module for monitoring; which detects important rail components and determines any abnormality existing in those components or their location. The results are transferred to the next module which determines the severity of defects detected by the previous module. Finally the appropriate maintenance and measures are taken to solve the severe problems; and also long term assessment is performed after examining the condition of the working components so far. The proposed system was theoretical and different imaging techniques were yet to be tested and also a use of artificial illumination was being considered. They claimed good performance in their preliminary study and the overall system is comparable to the one that has to be developed for current work. Later on, based on the proposed system, an automatic machine vision based rail inspection system was presented last year (2010) to detect important rail components, with main focus on detecting anchors. Image acquisition is performed by four cameras with special settings; connected at a specific location on a moving bus, to guarantee maximum coverage of the track. The main imaging techniques used by the system were Hough transform for detecting straight lines, and Sobel operator for the edges. Learning-based Adaboost discriminative classifier, using multiple cascade classifiers, has been used to detect anchors. These techniques in combination with some post mathematical operations have been applied to perform the detection. The system is accurate and efficient and currently better than other advanced rail inspection systems in anchors detection. (Trinh et al., 2012)

Another machine vision system has been proposed recently, using imaging algorithms to identify surface defects of the thermal-state heavy rail. The system uses 6 CCD (Charge-Coupled device) cameras installed at different angles, some high frequency fluorescent lamp for illumination and image processing work stations. The images are taken by the cameras and transferred to the work stations. Enhancement and noise removal is performed using the image algorithms of colour adaptive median filtering and histogram equalization. Edges are detected on enhanced images of the hot rail using common operators of Sobel, Roberts, LoG and Canny. Finally some mathematics were applied to detect the defects using the contrast obtained from the shape of former and last pixel lines of the rail images. The algorithm can effectively and rapidly identify several types of surface defects on hot heavy rails. However, it was concluded that the algorithm is not capable of detecting few of the specific type of defects. The paper provides useful information regarding image enhancement, noise removal and edge detection. All the applied imaging techniques have also been used at several stages for the inspection algorithms developed for the current project. (Xue et al, 2012)

2.5 Metal Surface Inspection

There is not specific literature available for the type of images that needs to be inspected for the current work. Moreover, the defects to be detected occur during manufacturing process, hence, can be considered as metal surface inspection to some extent. Therefore, literature related on machine vision inspection of metal surface has also been studied in detail and useful information from it has been summed up in this section.

Wu and Hou (2003) proposed an automated visual inspection of metal surfaces. Co-occurrence matrices are the basis of the proposed method, as they play an important role in texture analysis. Feature descriptors such as difference Moment and Entropy are computed from the reduced grey level co-occurrence matrices of a metal image. These features are finally analysed; and if they fall in some specific predefined interval they are classified as defects and vice versa. It is a good statistical approach with reliable detection rate to some extent, which can further be improved by using modified grey-level co-occurrence matrices. The rail images provided for the current research work are also grey level images and have been tested using simple statistics such as standard deviation, variance and mean to perform the surface inspection.

Bonnot, Seulin and Merienne (2004) searched the defects on the metallic industrial parts. The defects searched mainly are due to scratches and lack of machining. Particular surface features are acquired using images taken in annular lighting in bright field and rotating lighting in a dark field. The lighting device was designed to make an efficient revelation of the scratches. Dark field images are then processed individually and combined to get one image revealing most of the defects. The resulting image was then segmented using a threshold into the defects classes. A trained classification based on discriminant analysis, was finally performed to recognize the different classes of defects. The system implemented on an experimental industrial production line gives very little rate of false detection. The images of the defects show that they are somewhat similar to the defects under consideration here. The paper suggests that considering the defects under different lighting conditions from various angles can help enhancing different defects.

Xie (2008) surveyed and dealt the visual surface inspection of different types of surfaces such as steel, stone, textile, wood etc. as a texture analysis problem. It was a comprehensive study covering and classifying all the previous work done in the field. It focuses on texture analysis based image processing methods that perform visual surface inspection. Surface defects are either textural irregularities or tonality deviations. The study emphasized on first type of defects that is local abnormality or texture irregularity detection; which covers most of the problems. Survey suggested that the current methods to perform texture feature extraction and analysis can be divided in four categories; statistical approaches (histogram statistics, co-occurrence matrices, local binary patterns and autocorrelation), structural approaches (primitive measurement, edge features, skeleton representation and morphological operations), filter based methods (special and frequency domain analysis) and model based approaches (fractal models, random field model, texem models). A very useful survey concluding statistical and filter based approaches to be the most useful and

Tonality defect detection as a considered approach. It also suggests that when good prior knowledge of normal and defective samples is available, then supervised defect classification can be used as in the current work. The study also suggests localising the defective regions to understand the formation and nature of defects more, rather than classifying the whole surface as a defect. Finally, it concludes that real time performance is highly desirable, which is the important objective of this research as well. The research work in short gives several effective directions for the future study.

Blackledge and Dubovitskiy (2008) developed a new machine vision system for a steel surface inspection based on Euclidean and Fractal geometric of an object in the image, to monitor the quality of steel sheets during production. The system uses a Weiner filter to remove noise and light flecks and then detects edges of the defects and localise them using an automatic iterative algorithm. Euclidean and Fractal geometry based features of the localised objects are computed which are finally classified using fuzzy logic. The system is believed to be accurate with some innovations of using fractal analysis and fuzzy logic. The noise filter used and fractal analysis can be tested for the future study and might give some good results. Moreover, fuzzy logic or more generally Artificial Neural Networks can be implemented for defect classification in the future work.

A machine vision system based on Gabor filtering and adaptive threshold to detect scarfing of slabs has been developed by Choi, Jeon and Kim (2011). Scarfing results by non-uniform ignition produced by a scarfing machine, which during rolling develops into serious defects. These defects appear in the form of discontinuous bright bands and borderlines. Gabor sine and cosine filters are used with high and low threshold values. These threshold values are determined adaptively using the mean and standard deviation values. The results presented in the paper shows its effectiveness. Gabor filters are widely being used for texture analysis and hence, can be useful for the future study.

2.6 Concluding Remarks

The rails are being inspected currently for surface and internal defects using many different non-destructive evaluation techniques all around the world. Visual inspection having the lowest speed has been speeded up by automated visual inspection methods, but the process has still a low inspection speed. Ultrasonic have been used mostly for rail inspection, but it fails to detect surface and near surface defects. Eddy currents are reliable for surface defects detections and also has high testing speed. However, they can detect up to certain depths and fail to detect deep defects. They are also sensitive to electromagnetic noise and irregularities present. Guided waves can efficiently detect both surface and internal defects at high speed but their lower frequencies sometimes miss small defects. Hence, none of the current detection method is 100% efficient and fully automated. However, it is clear from the literature survey that nothing has been done till now on the types of defects being dealt in the current research work. Most of the literature found is on detecting missing rail objects, rolling contact fatigues or cracks on joint bars. Moreover, automated detection of such defects is the basic novelty of the work.

Current rail inspection extensively uses high speed cameras for image acquisition along with image processing methods and techniques to detect different faults and missing objects on the rails, which form the basis of the current research. Hence, the image acquisition methods, and pre and post processing image techniques, imaging procedures and fault detection and classification methods used in the above studies provide useful grounds for future work. Especially various image processing techniques such as edge detection methods, morphological image processing, convolution filters, feature extraction methods, PCA and neural networks previously been used will be the starting point for this work.

3 Description of Rail Manufacturing Computer Vision (JLI) System and Main Defects

3.1 Introduction

An industrial computer vision system (JLI) exists for the inspection of rails during production at a rail manufacturing plant in Scunthorpe. The JLI system has been designed by JLI Vision for Tata Steel and has been implemented at the rail production line. The system consists of 6 cameras and powerful halogen lighting units positioned around the path of rails during the cooling process, giving 6 complementary viewing aspects. Videos of each rail rolling pass the camera are recorded and each frame is converted to a single image.

Each rail is 110 m in length which in images is 440 images on average, by each camera per rail. The time between successive rails while in heavy production is typically 1 min 30 sec. Currently, a human operator in the control room checks the images for defect on the rails, but due to the amount of images that needs to be seen;

only few rails are randomly assessed. Hence, to perform real time analysis of the rail, the software has to be capable of processing a complete rail within approximately a minute.

There are many types of serious defects found on the rails of which rolled in scrap, tiger stripes (a pattern of defects that is stripe like in appearance), a line on the top of rail and wire defect are noteworthy and have mainly be investigated with in the current research. The following chapter describes the main properties of the said defects, a description of the JLI system in detail, type and properties of images taken by it and also the camera angles that acquire particular defects.

3.2 JLI Vision System

The first industrial vision system was developed in 1980, which led to the establishment of a company called Jørgen Læssøe Ingeniørfirma ApS. The company is developing, manufacturing and installing the advanced and dedicated computer vision systems for industries and laboratories, since 1985. Jørgen Læssøe Ingeniørfirma ApS was renamed in 1998 to JLI vision a/s (JLI).

JLI's vision systems provide reliable and flexible control for the production for years and results in high quality product. Some important characteristics of the system are as follows:

1. It is independent and a customised system, designed especially for the problem.
2. Each JLI system is designed after a careful study of the installed production lines to ensure the necessary functionality.
3. All the systems are fully automatic using digital cameras and customised software and come with internet connection.
4. Designed to perform quality control tasks and gives excellent reliability.

3.2.1 JLI Vision Rail Inspection Setup

The JLI Company has developed and installed a customised inspection system for Corus, which is now known as Tata Steel. The JLI system is found at the Scunthorpe site of Tata Steel. The installed JLI system has six cameras and floodlights, which are mounted on a circular holding ring through which the rails and sections pass at speed up to 4 m/sec. A set of six cameras are available at the site, as seen in Figure 3.1.

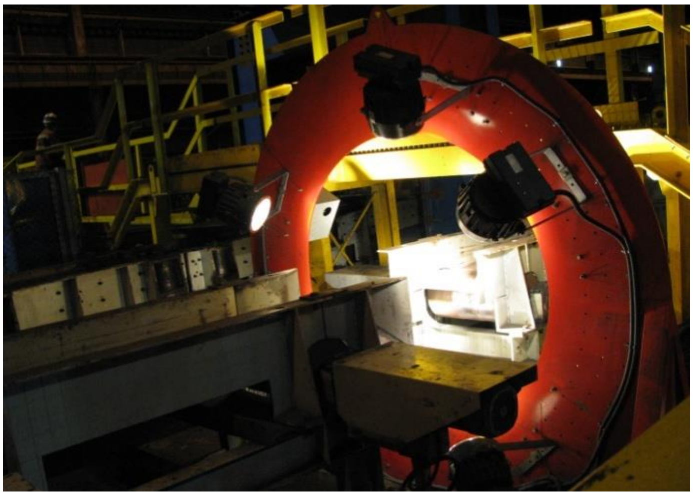


Figure 3.1: On-site area scans cameras and high intensity lighting.

The images from all the six cameras are recorded electronically as the rail enters the ring, at a frame rate of 16 frames/second on each camera. The installed flood lights are driven by special electronic drivers to make steady illumination in order to avoid shaky or blurred images.

The images of the entire rail which are approximately 2000 images per rail are compressed and successively saved on a 500 GB hard disk. The system has the ability of hot swapping to a new drive if the current one is full or if the current images are to be retained. The images are in JPEG format and resolution varies for different camera angles. Usual resolution of the images is 1380 x 760 for most of the defects and 1380 x 400 for Tiger Stripe type defects.

3.2.2 Current JLI Inspection and analysis

The images saved by the cameras in the hard disk are then inspected by an operator in an office at his/her own speed and not by the production flow. All images are inspected one by one and the defective ones are marked. The serial number and distance of the defective rail from the start of the rail is saved in a report on a computer. The marked rails and sections are further scrutinised after cooling and if a fault is identified, the operators examines the nature and complexity of the defect and might figure out the underlying cause. For example, the mill might have been causing a defect which necessitates stopping the production and correcting the fault. Defective mills roll collars or uneven timing for cooling water results in defective rails. The fast and automatic storage of the data makes the early detection of faults possible and ensure a high production rate and quality assurance.

3.3 Main Defects and their Description

There are 4 types of defects that, according to Tata Steel who are experts on the subject, cover more than 95% of all defect found on rails during production. It is the algorithms for the effective detection of the defects that form the substance of this thesis.

The defects have been referred over the years as rolled in scrap, as can be seen in Figure 3.2 (a), a line on top of the rail, as can be seen in Figure 3.2 (b), tiger stripe, as can be seen in Figure 3.2 (c), and a wire defect, as can be seen in Figure 3.2 (d). All of the mentioned defects appear on the top view of the camera system and on the bottom view and are very difficult to detect by established computer vision techniques.

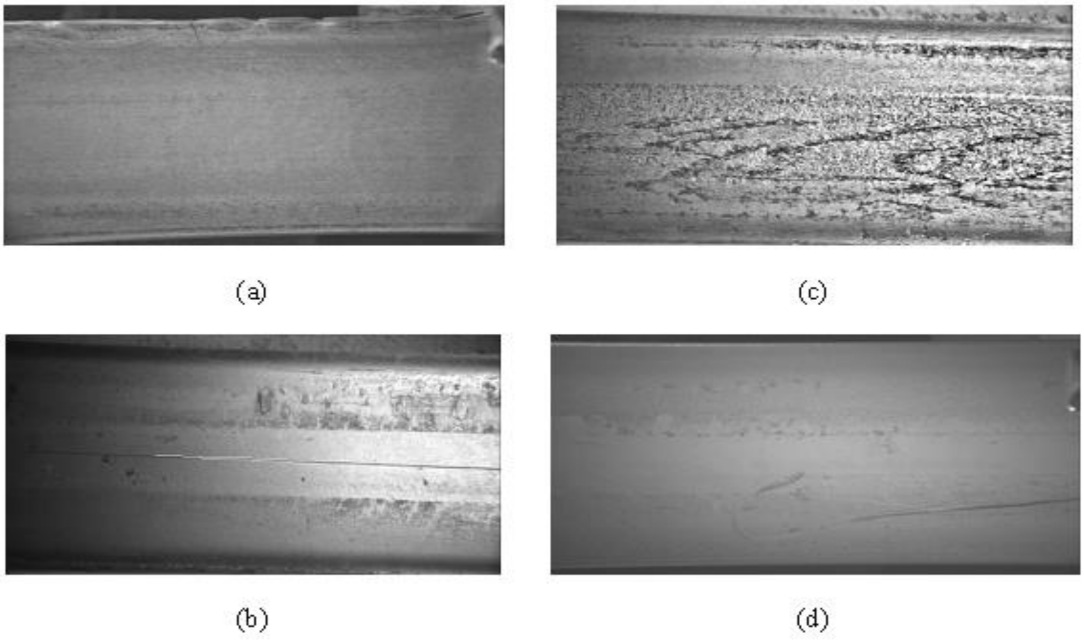


Figure 3.2: Types of defect, (a) Rolled in scrap, (b) Line on top, (c) Tiger stripes, (d) Wire defect

3.3.1 Rolled in Scrap

The rolled in scrap defect is a type of defect that appears when the rail exits the furnace. It appears on the side of the foot of the rail. It is due to the side of the rail being scratched by any sharp surface or material coming out of mill while passing, leading to protruding and/or extruding material on the edge of rail's foot. It can be detected by measuring any irregularities at the bottom of the rail edges i.e. deviation from a straight line.

Rolled in scrap can usually be seen from a particular camera viewpoint as indentations at the edge of the rail. Isolating them involves the detection of the edge of the rail followed by detection of non-straight lines or derivatives in the edges, in other words the detection of the defect.

Rolled in scrap can be seen clearly in Figure 3.3. The uneven edge of the side of the foot of the rail is clearly visible, in this image.

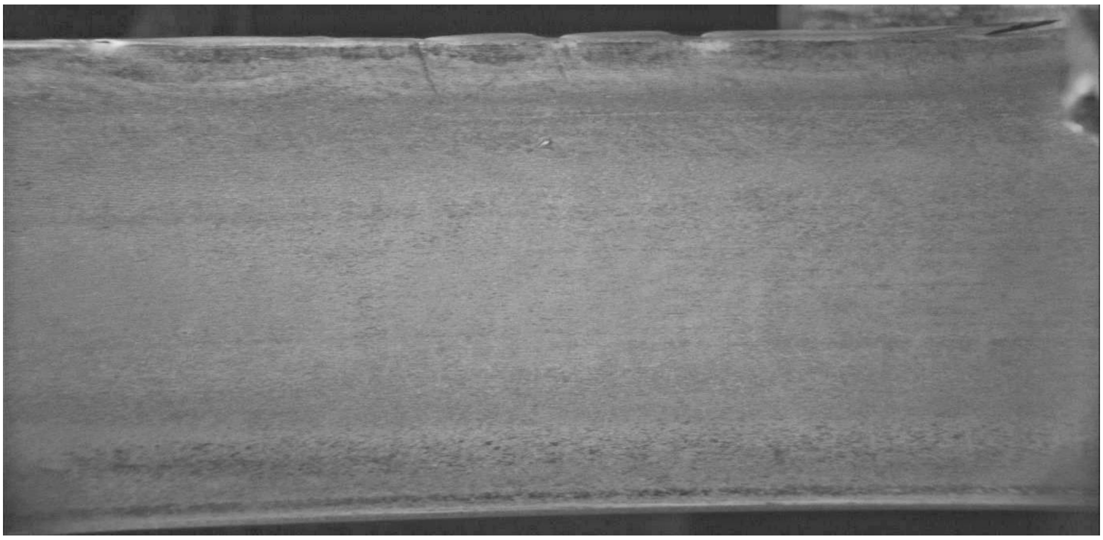


Figure 3.3: Rolled in scrap captured by camera 0

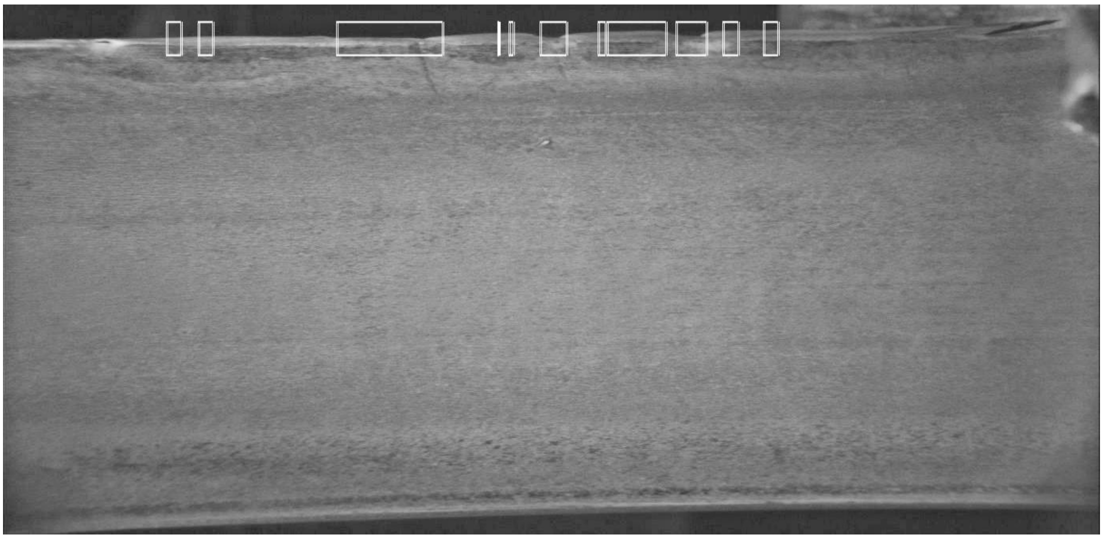


Figure 3.4: Irregularities marked for rolled in scrap manually

3.3.2 Line on Top of Rail

A line on top of rail appears as a straight line at the top of the rail and can have various origins. They are repetitive and short pitch; typically less than 400 mm. They are horizontal straight lines right in the middle of the rail surface. One of the causes might be the worn vertical entry roller guide, in one of the finishing mill stands. It can be avoided by replacing the roller with a clean set and adjusting to give more clearance if possible.

Figure 3.5 and Figure 3.6, represents line on the top of the rail defect, example images. It can be seen from the example images that lines on the top are horizontally running lines, which may break after a certain length and might not be present on the successive images. The random lines present on the rail's surface need to have some reasonable length to quality to be a line defect. The lines have to be present across most of the consecutive images to be a valid defect.

I

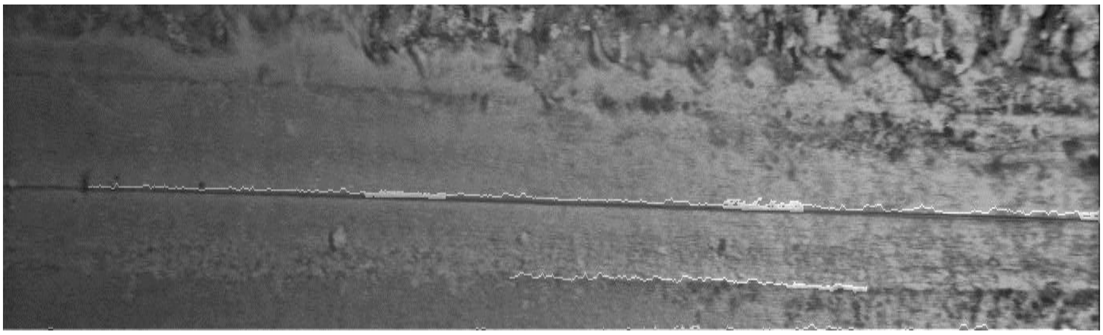


Figure 3.5: Original rail image showing Line on the top of rail type defect highlighted with while color.

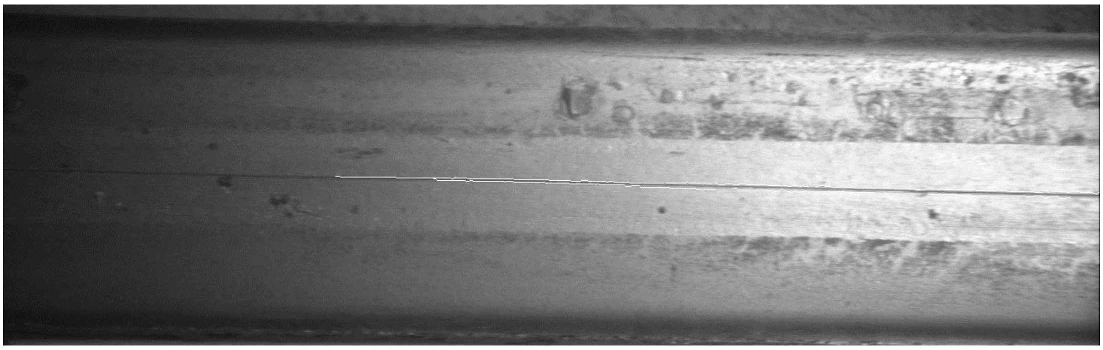


Figure 3.6: Another example image of a rail with Line on the top of rail type defect.

3.3.3 Tiger Stripes

Tiger stripes appear as dark areas suggesting patterns on a tiger. They occur when the rail is left for a longer time under the cooling water than required. They occur at the top of the rail as known as crown of the rail. Hence, they are mostly captured by the top camera which is camera 4 or the side ones; camera 3 and 5.

They are also referred as “flow marks” and are actually uneven distribution of metallic outer surface. They might be caused by unequal or uneven spray pattern of cooling water. Even the tilted or too close spray of cooling water on the hot rail surface can cause the tiger stripes to appear on the surface.

The marks are of variable sizes depending on the cause, which makes them difficult to detect. They can either be as large as shown in the Figure 3.7 or as small as shown in Figure 3.8. However, they have characteristic shape, which appears like a roughly drawn less than “<” sign of various shapes and sizes. Moreover, they are generally present on many successive images. These features help in Tiger Stripes detection.

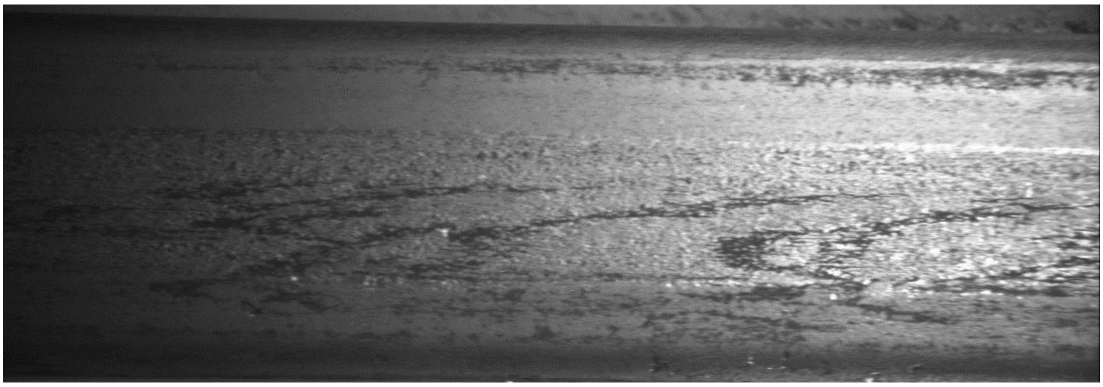


Figure 3.7: Large tiger stripe pattern – Original image

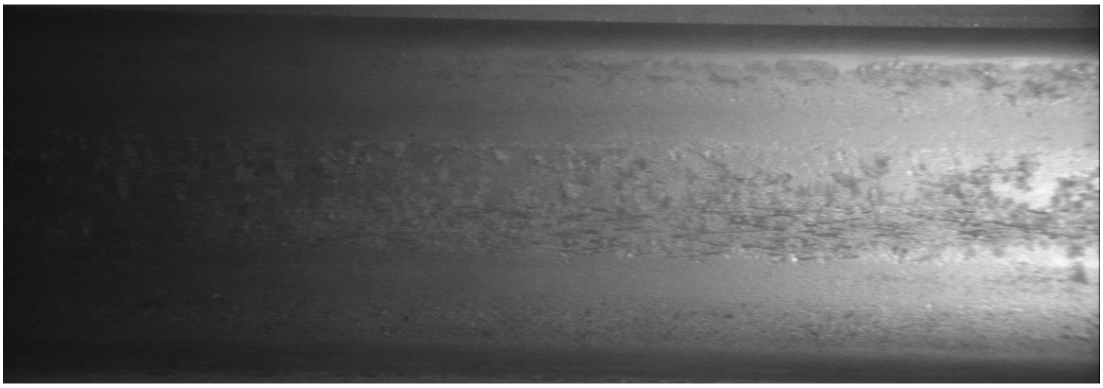


Figure 3.8: Original rail image with small tiger stripe type defect pattern

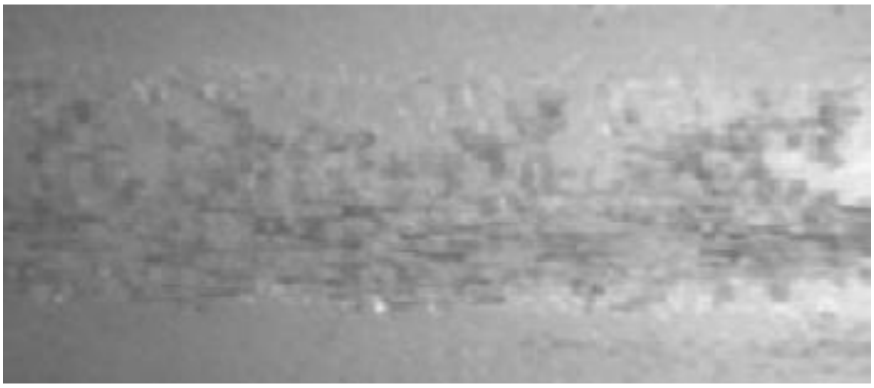


Figure 3.9: Small size tiger stripe pattern – Close view of the image displayed in figure 3.8.

3.3.4 Wire Defect

A wire defect is a line type defect that affects the base of the foot in a central region. It appears under the rail and is the most difficult and complicated defect of all to detect from a computer vision aspect. This defect can appear as straight lines, curves or even random arbitrary shapes, with varying size.

Wire defect is caused by materials forcing out through the roll collars. Its maximum frequency is 4 instances per rail and it can be up to 0.5 mm deep. This defect can be avoided by rectifying the mill. Its early detection and quantification is necessary to reduce the cost and safety by very large margins. However, the variation in this type of defect made them extremely hard to detect, while maintaining an acceptable level of false positives.

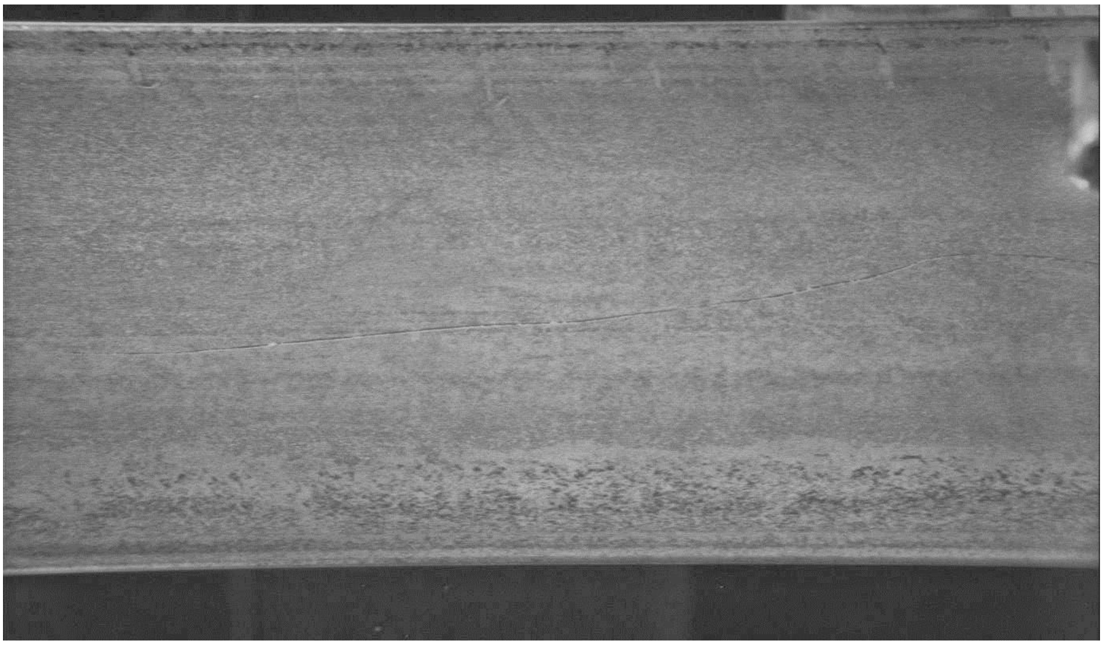


Figure 3.10: Original rail image with wire defect captured by camera 1.

3.4 Significance of Image Names

Images taken by the JLI system are in grey scale and saved in JPEG format. Each image taken by the JLI system is saved with a unique name. The name of each Image has different parts in it and includes distance from the start, camera number, date and the time it has been taken. For example there is an image named “0000000_1 19-05-2009 at 08.52.49.jpg”. The first 8 bits show the distance in meters from the start. The example shows that this is the first image of the rail, starting point of the head of the rail. Then the number ‘1’ shows that it has been taken by the camera number one. There are 6 cameras and their numbering starts from 0 to 5. The next part of the name is the date followed by the time at which the image was captured.

Example: Image Name = 0000259_1 19-05-2009 at 18.14.39.jpg

1. 0000259 – Image is at the distance 259 meters away from the rail head.

2. "_1" – The camera number one captured this image.

3. "19-05-2009" – The image has been taken on 19th may 2009.

4. "18.14.39" – The image was captured at 18:14:39 hours.

5." Jpg" – The image has been saved as jpg format.

3.5 Defects and Viewing Cameras

All the four defects are at different locations on the rail. Hence, they are all captured by different camera angles. Figure 3.11 shows the schematic of the camera positioning around the rail. The cameras have been numbered from 0 to five, started from the bottom left camera.

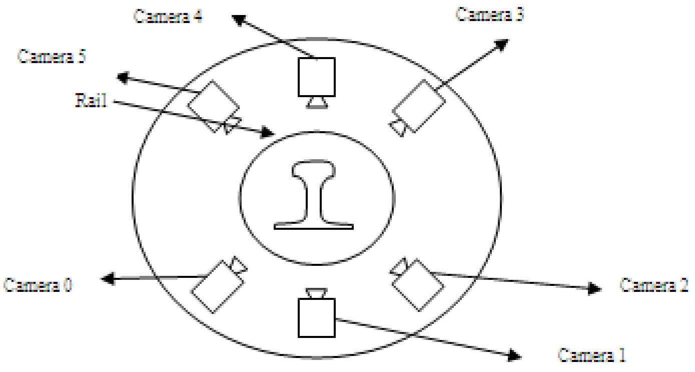


Figure 3.11: Schematic of the camera positioning around the rail

Moreover the name, with which each image is saved to the memory, clearly represents the camera it has been captured from. All the defects with their location on the rail and the camera numbers capturing them have been presented in Table 3.1.

Table 3.1: Defects, their Location on the Rail and Respective Detecting Camera Number

Defects	**Location**	**Camera Number**
Rolled in Scrap	Side of Foot of the Rail	0 and 2
Wire Defect	Base of the Foot in Central Region	1
Tiger Stripes	Top of the Rail	3, 4 and 5 but usually camera 4
Line on Top of the Rail	Top of the Rail	3, 4 and 5 but usually camera 4

3.6 Limitations with JLI System

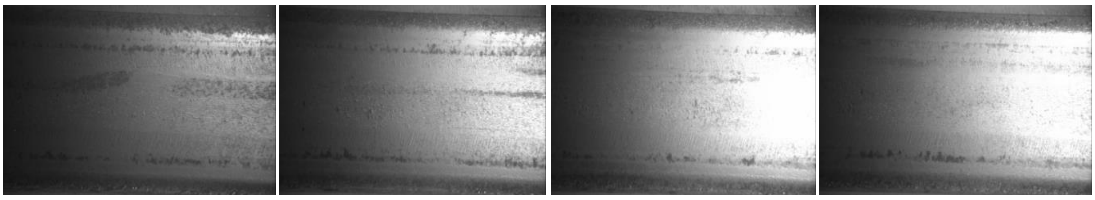
The JLI system installed at the site had been designed for human visual inspection and lacks some characteristics, such as optimal lighting or camera positioning for optimal defect detection. The importance of lighting and camera positioning is evident from the rail inspection literature available so far.

Marino and Stella used a special illumination setup, equipped with six flood light sources to overcome the effects caused by changing natural light. The system has been discussed before in chapter 2 (Marino et al. 2007). The Federal Rail Road Association America is using a joint bar inspection system with a lighting system made of xenon light, which provides uniform and consistent illumination and also overcome the effect of ambient light (Berry et al. 2008). Deutschl et al (2004) presented a rail surface inspection technique called spectral image differencing procedure (SIDP). Two monochrome light sources, blue and red, positioned before and after the camera in the direction of the movement, have been used to facilitate detection of defects on the rail surfaces.

Mandriota et al., (2001) proposed a machine vision inspection technique based on texture analysis of rail surface. A line scan camera DALSA with 512 pixels of resolution was installed below a diagnostic trolley with an appropriate illumination setup using six OSRAM 41850 FL lights, to reduce the effect of day light.

However, in the JLI Vision system installed at the site, there is no special attention given to the design for an optimal lighting for optimal image processing, or camera positioning for better defect viewing, as had been done by some successful working systems around the world. Hence, the requirement of the inspection system to be developed was to be easily implemented on-site with minimum disruption of the existing setup, which made the task more difficult.

The setup results in images of uneven lighting, as can be seen in Figure 3.12 and with varying level of blurring as shown in Figure 3.13. The uneven lighting condition is firstly due to the use of non-diffuse light sources (spotlights) that causes the appearance of some very bright areas of the image while other areas appear very dark, as can be seen in Figure 4(a), and secondly to rail positioning not being constant, leading to varying light between successive images of the same view as can be seen from Figure 4(a) to Figure 4(d). An extra varying level of blurriness caused due to a deposit of dust, that occurs over time as the rails are produced, can also be observed. Human intervention to physically clean the lenses is required to avoid such noise. An air blowing system is fitted but has limited success with keeping the lenses clean. The algorithms for detecting the type of defects identified above have to be robust to the change of light and the clarity of the image.



(a) (b) (c) (d)

Figure 3.12: Light variation on successive images.

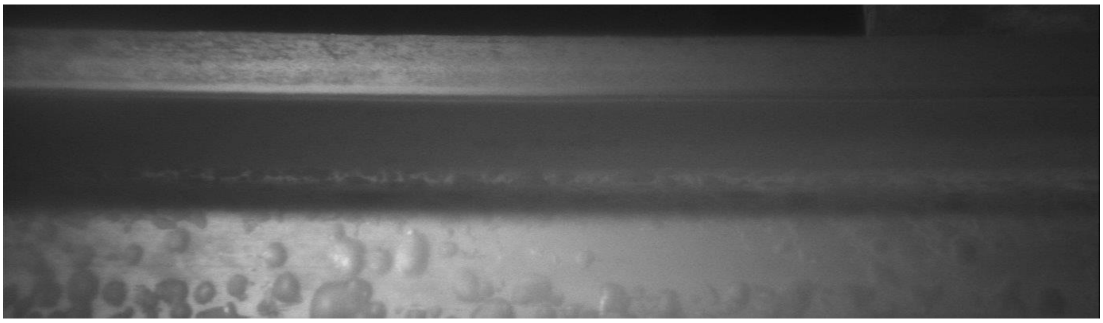


Figure 3.13: An example of blurred rail image.

A set of four highly probable and critical defects that would require dressing, chopping or even destruction of the rail have been selected by the technical managers at Tata steel. The critical aspects of the faults reside in their return and rejection rate by the rail customers, i.e. the companies that purchase the rail from Tata steel, and in their rate of appearance, as combined they represent more than 95% of the entire defects library observed over the years by the highly qualified personnel.

3.7 Summary

Description JLI system found at TATA Steel Scunthorpe site has been explained in detail in this chapter. The system components, working, features and limitations have been summed up. Moreover, the main defects that are to be investigated in the current research work have been discussed in detail. Each defect's properties, main feature, naming convention and its representative images have been provided and explained in detail. Next chapter gives a review of general image processing techniques that have been tested in the research process and those methods that are also used in the defect detection techniques that have been developed for the current work.

4 A Review of General Image Processing Techniques

4.1 Introduction

This section presents the algorithms tested, studied and used for the detection of each defect. These are generally a bespoke combination of existing image processing techniques and new algorithms developed specially to be fit for purpose. There are many image processing methods that have been studied, explored and tested. Moreover, few novel methods have been developed as well to perform the detection. This chapter contains the brief overview of the image processing methods that has been used at different stages of the defect detection.

The detection process for all the defects has the following distinct phases.

- Image Pre-Processing stage that includes noise removal and enhancement followed by image thresholding.
- A processing stage that includes edge detection, straight line detection or feature detection depending on the type of defect.

- A classification stage to classify the detected fault to a particular class of defects.

4.2 Image Pre-Processing

Most of the images have lot of noise due to dust, rain and other climatic factors. Sometimes they also have non-defect scratches and marks caused for various reasons. This noise needs to be removed before any further processing, to avoid false detection. Moreover, in the current scenario the most important aspect that needed correction is the uneven light. All such non-defects abnormalities need to be ignored before performing actual defect detection as the noise often seems similar to the defects and can cause confusion in their detection. Image pre-processing is a major part of the research as it helps to get better detection results. It involves:-

1. Noise Removal,
2. Image Enhancement ,
3. Light Adjustment - critical for the current inspection.

All the image processing techniques tested to enhance the image quality for better detection have been discussed in detail with the result images in this section.

4.3 Image Filtering

Filtering is a technique for smoothing, sharpening, de-blurring, removing noise, and enhancing edges in an image. It can be used to apply various effects on the image. A 2D filter matrix, termed as mask or kernel, is used to filter a 2D image by different filtering methods. Filters with convolution or correlation either in the spatial or frequency domain is relatively simple or is more commonly used.

Convolution is performed using a mask, which can have an arbitrary shape, depending on the application. Standard masks generally have odd number (3, 5, 7, etc.) of elements in each dimension, and at every location have their centres aligned with the current pixel. Moreover, sum of all elements of a mask should be one in order to have the same level of brightness as the original image. The mask is applied to each pixel and its neighbours within an image. Any small size of mask is chosen depending on the requirement and is moved \ slides over the whole image, generally starting from the top left corner. The function to be performed is repeated for every position of the mask. Each position of the mask corresponds to a single output pixel. For convolution filtering values of the mask and the corresponding underlying image pixels are multiplied and finally added to get the output value.

There are different types of filters available to perform different tasks. Matlab has some predefined filter masks and few in built functions such as ‘imfilter’ and ‘filter’ to apply those masks on the images. There is another Matlab function called ‘fspecial’ used to create predefined 2D filters. Several types and sizes of filters were tested to remove the noise, enhance the defects and reduce the non-defect detections.

4.4 Noise Removal / Image Smoothing

Noise in the rail images during production is mainly due to environmental factors. The presence of dust and marks left by cooling water that is sprayed on the rail appear similar to the defect marks. There is burn marks that form during the moulding process and scratches caused due to material being forced out from the mill while the rail is still hot. All these marks and deposited materials can appear similar to a defect on rail images. Moreover, the metallic grey colour of the rail with the same comparatively darker shade of background and poor lighting adds to the noise. The image showing defect and defect similar noise is presented in Figure 4.1. Wire type defect has been marked with red in the image and non-defect noise areas with blue. It can be seen that how similar both the areas look, leading to subsequent confusion in their detection and differentiation. Hence, this noise needs to be removed before performing any test for accurate defect detection. A range of different types of filter has been tested for the reduction and removal of noise from the image. Some of the noise reduction methods have been found useful and are used during the pre-processing stage of the defect detection.

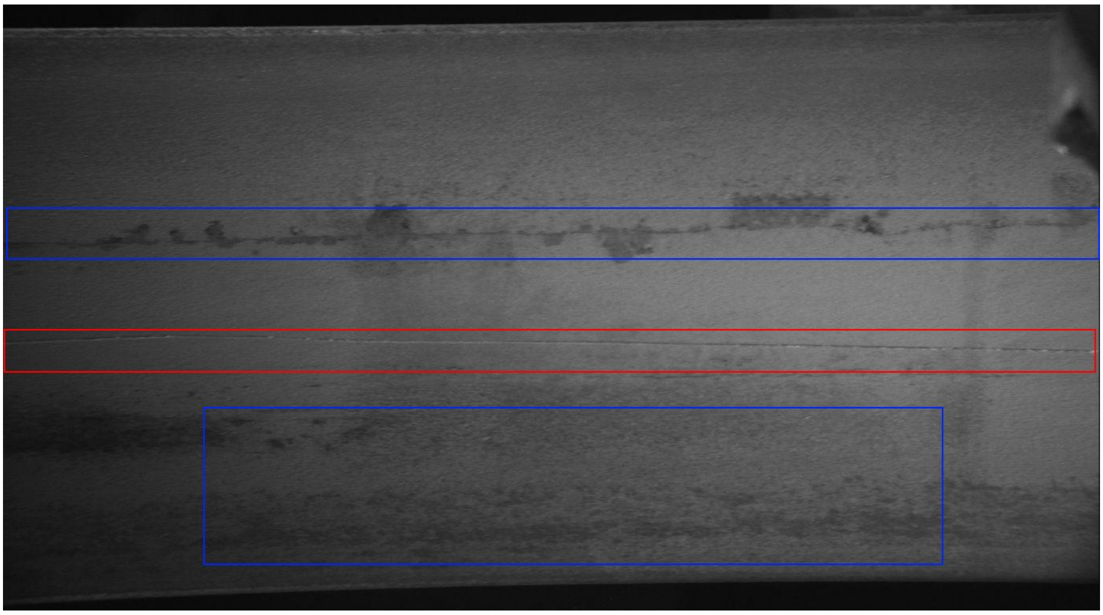
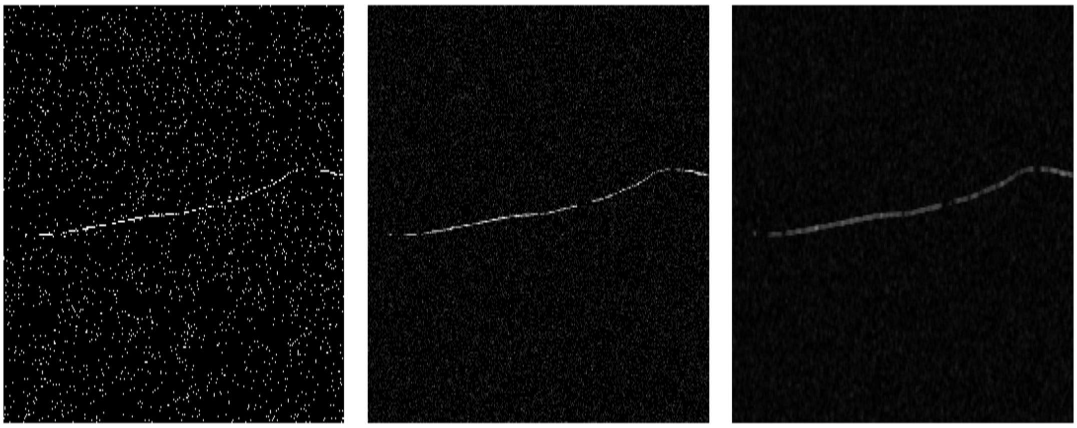


Figure 4.1: Original image showing wire defect marked in red and non-defect noise appearing similar to the defect as blue.

4.4.1 Noise Removal by Average Filter

Average or mean filter replaces all the values under the sliding window by one average. Average filter smoothens the image data hence removes the noise from the image. It performs special filtering on each of the pixel in an image using a square or rectangular window surrounding the pixel. It is the simplest way of removing the noise or in other words reducing the variation difference between the neighbouring pixels.

The use of large size filters results in increased computation time. The result of average filter tested on image with artificially added noise image has been shown in Figure 4.2. The image used to test the filter has been manually drawn similar to the wire defect lines that appear on the rail. This image has been referred as handcrafted image in this chapter, and has been used only for the testing purposes. Although, the large mask size remove noise more efficiently but also blur the edges in the image at the same time, which is obvious from the Figure 4.2. Hence, the direct application of the average filter didn’t give good results. It was tested at various stages of detection and has been found very useful for the particular form of rail images, in very few cases. However, modified form of an average filter has been used to perform image equalization in one of the developed detection method.



(a) (b) (c)

Figure 4.2: Noise removal results using average filter. From left (a) image with manually added noise (b) Noise removal result image obtained using 3x3 average filter (c) Noise removal result image obtained using 10x10 average filter.

4.4.2 Noise Removal by Median Filter

Median filter is another type of filter available to remove unwanted noise for an image. Median filter takes the median value of all the pixels in the mask window. It sorts all the value under the mask in ascending order and assigns the output pixel to the middle value. Masks of various shapes can be used which help to save line structures. The results of median filters applied on manually added noisy defect image have been shown in Figure 4.3. The filter was tested on various images and results were comparatively better than the average filters as can be seen in the figure below. However, it failed as such to remove the desired amount of noise keeping the image sharpness intact.

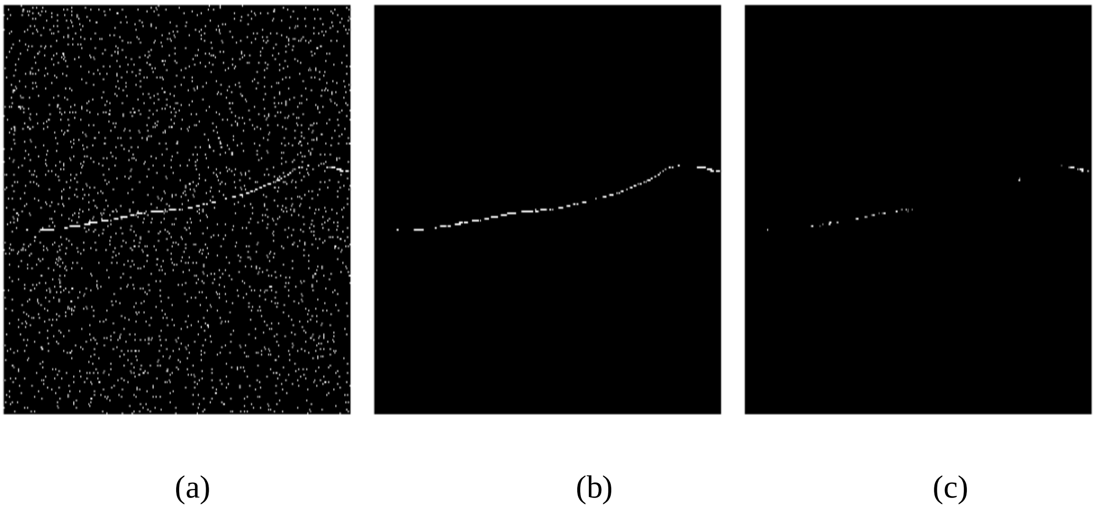


Figure 4.3: Noise removal result by median filter. From left (a) Handcrafted defect image with lot of noise (b) Result of 3x3 median filter (c) Result of 6x6 median filters.

4.4.3 Noise Removal by Gaussian Filter

The Gaussian smoothing operator or Gaussian filter is also used to remove details and noise from an image. It is similar to previous filters discussed and it is applied by the same convolution process but it uses a different type of filter. The Gaussian kernel (filter) which is used for smoothing has a shape of a Gaussian bulge and has special properties. The result of Gaussian smoothing when applied on handcrafted defect image with manually added noise has been shown in Figure 4.4.

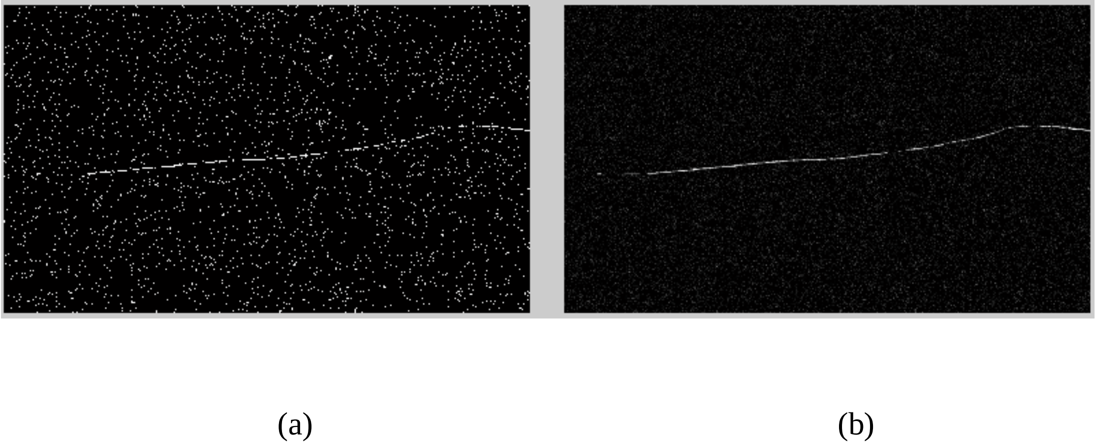


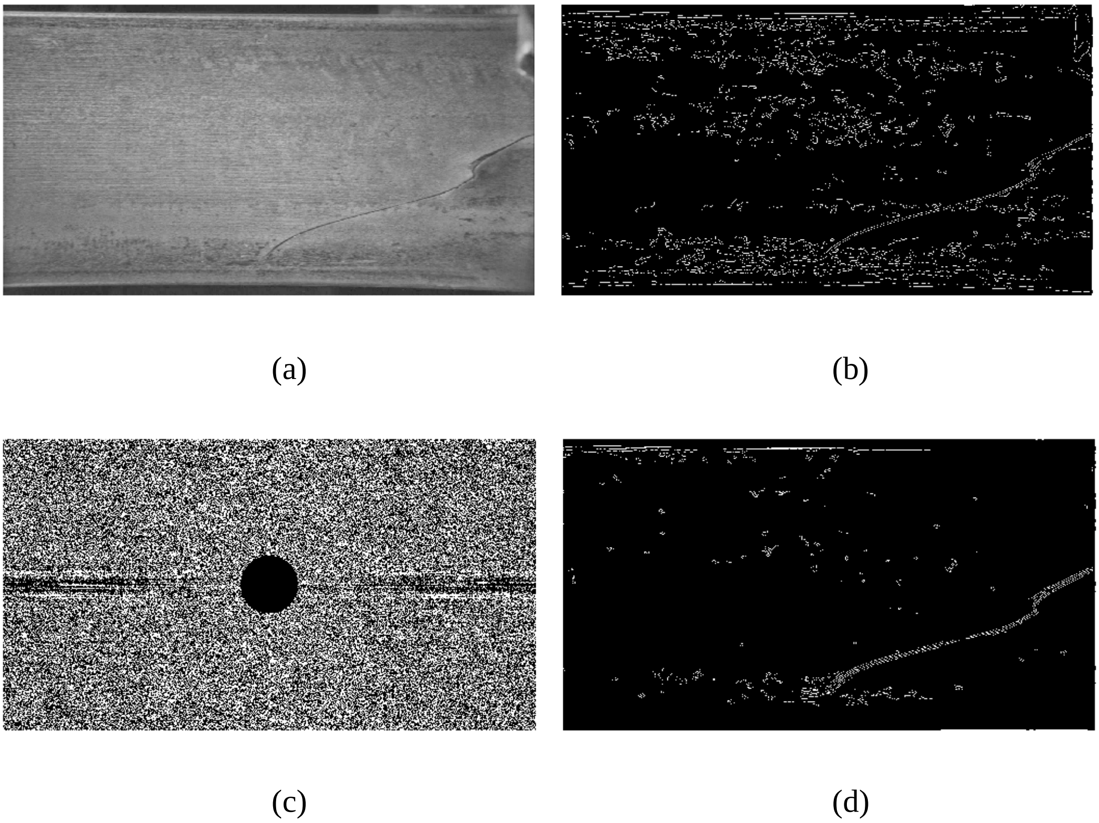
Figure 4.4: Noise removal result by Gaussian filter. From left (a) Handcrafted defect image with lot of noise (b) Result of 3x3 Gaussian filters with sigma equals to 2.

Gaussian smoothing is comparatively better than the previous filters tested for noise removal. However, it also failed to remove the desired amount of noise while keeping the defect details in a rail image intact.

4.4.4 Noise Removal by Fourier Transform

A Fourier transform is a mathematical operation that breaks a function into its constituent frequencies. The initial wave form is time dependant, so it is referred as time domain representation. The resulting frequency spectrum is in the frequency domain. In image processing, the Fourier Transform is an important tool that splits the image into its sine and cosine component functions. The input image is in special domain and the resulting image is in frequency domain.

The transform has been used to remove both the noise and any unwanted edges in the defect detection research. The image is converted into its frequency domain and particular frequencies are removed and then inverse Fourier transform is applied to get the original image without the noise or with the edges highlighted. The results are shown in Figure 4.5. The technique manages to remove some unwanted non-defect edges and marks; however, it is found that using the Fourier Transform method also loses some important information such as the defect lines.



(a) (b)

(c) (d)

Figure 4.5: Result of noise reduction using Fourier transform. From left (a) Original image with wire defect (b) Original image after canny edge detection (c) Fourier transform of the original image with central frequencies of diameter equal to 75 pixels removed in frequency domain(d) Canny edges image result after removing central frequencies.

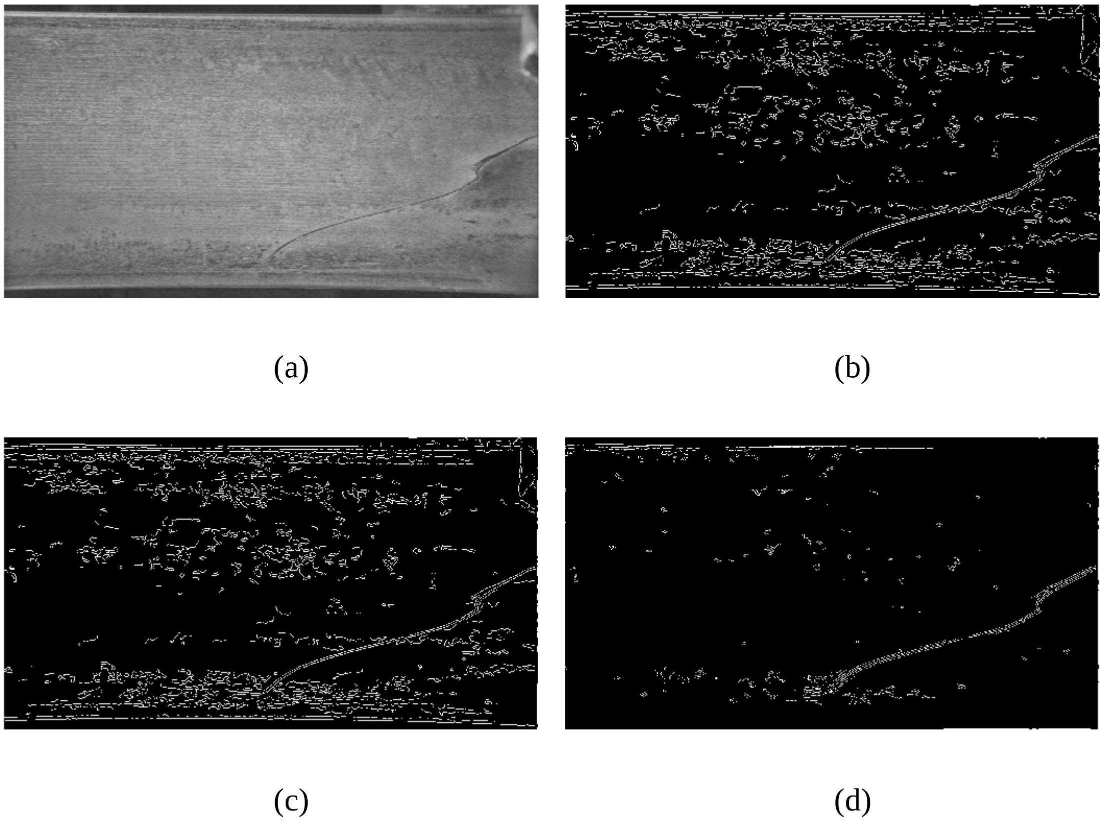
4.4.5 Evaluation of Noise Removal Techniques

The noise removal techniques tested has been compared in *Table 4.1***.** All the noise removal methods had their own advantages and disadvantages. As mentioned earlier the defects and the noise appears similar on the images, hence it was very difficult to find any technique that removes all the noise, but keep all the defects with their sharpness intact.

Table 4.1: Comparison of noise removal techniques tested

Noise Removal Techniques	**Advantages**	**Disadvantages**
Average Filter	• Easy and simple technique • Efficiently smoothens the image data and removes the noise	• Large size filters increase the computation time • Large mask size though efficient, but blur the edges in the image.
Median Filter	• Masks of various shapes can be used which helps to save line structures	• Cannot keep the desired sharpness of the image
Gaussian Smoothing	• Reduces noise efficiently • Reduces edge blurring	• It takes time • Reduces details
Fourier Transform	• Efficiently manages to get rid of unwanted noise	• Loses some useful image information with the noise as well

The images produced by all these methods have been shown in Figure 4.6. The Fourier Transform method gave the best results but at the cost of losing some desired information such as few defect lines and their parts. Hence, the final detection methods were designed in a way so as to reduce the effect of noise without losing any useful information. For this purpose modified forms of these mentioned methods were tested and used at various detection stages.



(a) (b)

(c) (d)

Figure 4.6: Comparison of tested noise removal methods. From left (a) Original wire defect image (b) Noise removed by average filter of size 5x5 (c) Noise removed by median filter of size 3x3(d) Noise removed by Fourier transform with central frequencies of diameter equal to 75 pixels removed from the image.

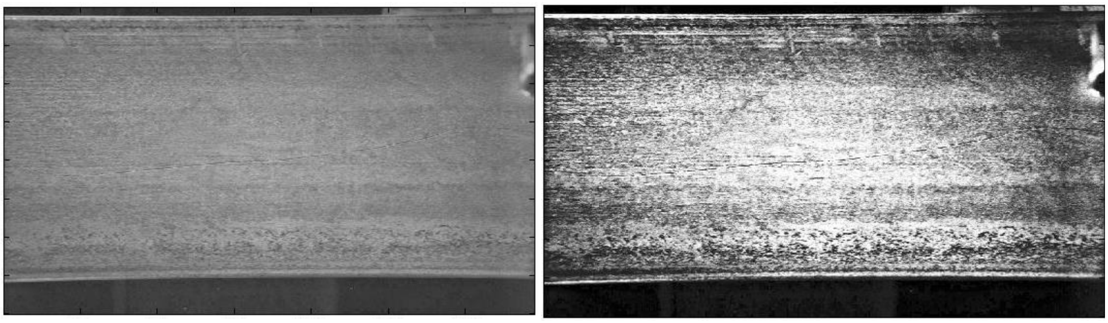
4.5 Image Light Adjustment & Enhancement

There is an uneven lighting condition in the rail images of JLI system due to the use of non-diffuse light sources (spotlights) and inconstant rail positioning which results in varying light between successive images. Lighting limitations and problems provided by the JLI system have been discussed in the last chapter in detail. Inadequate lighting results in light variation and varying level of blurriness in successive rail images. This causes bright and dark regions with in an image or successive images. Darker defective areas of the rail are comparatively more difficult to detect. Therefore, uniformly bright images are an important requirement for performing efficient and accurate fault detection. For light adjustment purpose, various techniques were tested and the findings were quite useful. Image equalization methods were also tested and to adjust the lighting on the images and also to enhance the features at the same time. The tested and used methods and their results have been summed up in this section. Hue saturation value (HSV) adjustment and Gamma correction methods were also tested for the purpose and their details and findings have been annexed at the end of report.

4.5.1 Histogram Equalization \ Normalization

Equalization is a method of contrast adjustment using histogram or other similar methods. It is done to enhance the image or to improve the shades or lighting.

Histogram equalization defines a mapping of grey levels into other grey levels such that the distribution of other grey level is uniform. This mapping stretches contrast for grey levels or in other words expands the range of grey levels near the histogram maxima. Since contrast is expanded for most of the image pixels, the transformation improves the detection ability of many image features. The main advantage of this method is that it is very simple and straightforward and is an invertible operator. However, its disadvantage is that it is indiscriminate, i.e. it cannot differentiate between noise and the defects for the type of images being investigated for this project. There is a possibility that it may increase the noise and un-sharp the desired defects to be detected instead of doing the vice versa. The results of histogram equalization produced by inbuilt Matlab function called “histeq” have been shown in Figure 4.7. It was not found to be useful for the particular kind of rail images under investigation as can be seen in the figure. However, a customised histogram method using average filter has been developed, which is presented later in the section.

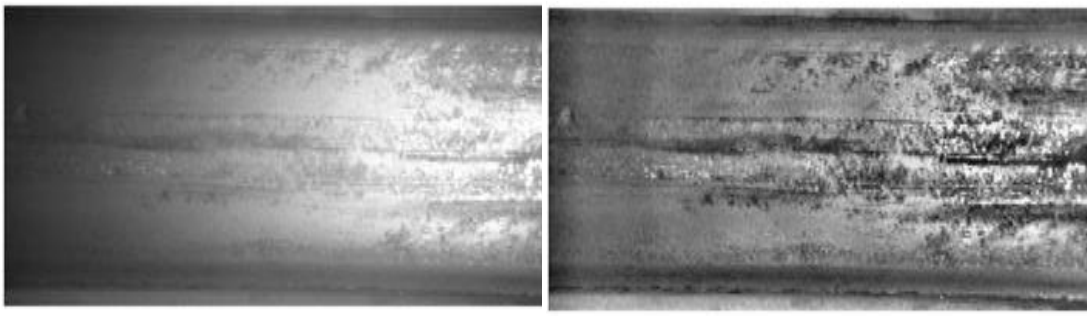


(a) (b)

Figure 4.7: Histogram equalization result. (a) Original image with wire defect (b) Image after Histogram equalization

4.5.2 Adaptive Histogram Equalization (AHE)

Adaptive Histogram Equalization (AHE) is an extension of the conventional Histogram Equalization technique. It is a more effective and broadly applicable contrast enhancement method. It works by dividing whole image into several small regions and computes the histogram of each of these sub regions. Bilinear interpolation is used to combine the neighboring regions of the image to remove pseudo boundaries. However, it has very slow speed and can also enhance noise. The results of AHE have been shown in Figure 4.7. It can be seen that it effectively enhanced the contrasts, but also the noise in the image, which is not the requirement of the images being investigated. Hence, this method has not been used for the final detection.

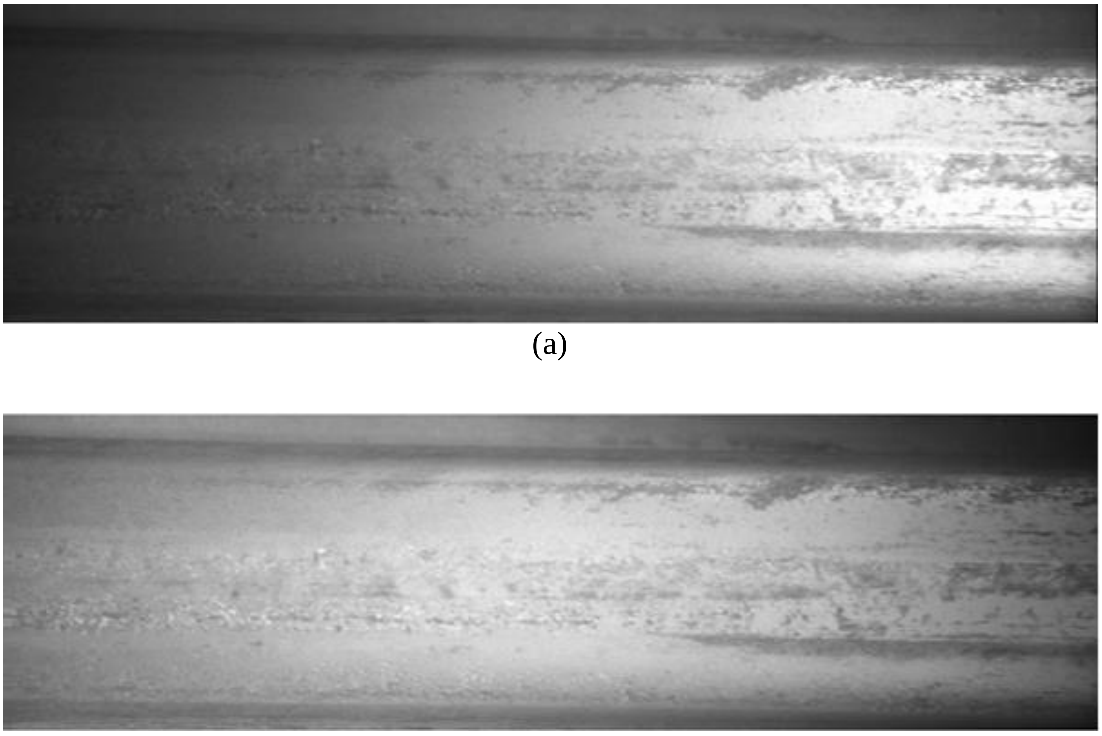


(a) (b)

Figure 4.8: Adaptive histogram equalization result. (a) Original image with wire defect (b) Image after Adaptive histogram equalization.

4.5.3 Intensity Scaling

A general observation of most rail images provided is that their left part is darker and their right sides appear to be brighter. This occurs due to the positioning of the spot lights installed on the JLI system. Hence, the images require some intensity to look uniform. Requirement was to brighten the darker part and make the brighter part little dark for the image to appear uniform. Hence, for every image the intensity scaling factor was different depending on the frequency of increasing or decreasing light intensity. Therefore, intensity scaling was tested and has been used. The scaling factor was calculated by taking the differences of intensity levels from right to left (bright to dark) and successively adding incrementing scaling factor to the column's pixel values. The results produced by self-written code can be seen in Figure 4.9. The resulting image looks quite uniform and facilitated the detection process later on.



(a)

(b)

Figure 4.9: Result image from intensity scaling. From left (a) Original image (b) Image after non-linear Intensity scaling from right to left.

Intensity scaling factor is determined using the mathematics shown below in Eq. (4.1). Average pixel value of first column of the image which appears the darkest is subtracted from the average value of pixels from the last column. The result is divided by total number of columns to find the scaling factor for a particular image. The scaling factor is incremented and added to the successive columns throughout the width of the image using the Eq. (4.2), shown below. For each column (x), the same scaling factor is added to all the values. The scaling factor is incremented from right to left for successive columns.

$$\text{Scaling Factor} = \frac{\frac{\sum_{y=0}^{n}(\mathrm{I}(x_m, \mathrm{y}))}{n} - \frac{\sum_{y=0}^{n}(\mathrm{I}(x_0, \mathrm{y}))}{n}}{m} \qquad \textit{Eq. (4.1)}$$

$$\mathrm{I(x,y)} \;=\; \sum_{\mathrm{x=0}}^{\mathrm{m}} \mathrm{I(x,y)} + \mathrm{ScalingFactor} \qquad \textit{Eq. (4.2)}$$

Where,

I = Image of size m x n

m = total number of columns in the image

n = number of rows in the image

Scaling Factor = Intensity scaling factor computed from the image

4.5.4 Evaluation of Light Adjustment and Enhancement Techniques

Light adjustment and enhancement techniques discussed in this section have been compared in Table 4.2. Few other techniques such as hue saturation value and gamma correction were also tested and their details have been given in appendix B at the end of the report. All the evaluated light adjustment techniques have their advantages and disadvantages. However, none of them was found to be ideal or perfect for the type of rail images under consideration. Intensity scaling was found to be a simple and effective method that brightens the dark areas of the image, without disturbing or altering important image information. Hence, has been used for the images with poor lighting contrast, to adjust their brightness before performing the actual defect detection.

Table 4.2: Comparison of light adjustment and enhancement techniques tested and used

Techniques	Advantages	Disadvantages
Histogram Equalization	It is simple and straightforward and an invertible operator	It is indiscriminate, cannot differentiate between noise and features to be enhanced.
Adaptive Histogram Equalization	It Improves the local contrast and enhance the details	slow computation speed and can over enhance the noise
Intensity Scaling	It is simple and effective for the current type of grey scale rail images	Scaling factor is different for each image for each location, which needs to be computed every time.

4.6 Image Equalization Performed for Tiger Stripes Using an Average Filter

Image equalization is applied at the pre-processing stage in one of the detection algorithm for Tiger stripes, as lighting is non-uniform even on the successive images. Hence, to get uniform effect of lighting and a more smoothed image, equalization has been applied. A range of standard histogram equalisation algorithms were previously unsuccessfully applied, both local and global, before the customized method was developed.

Equalization has been performed using specially made method of finding the background lighting of the image. The algorithm then uses that background lighting to correct the original image. An average filter with a length equal to the height of the original image and a width of one pixel is applied to the original image. The result is an image representing the lighting, which is then inverted and added to the captured frame. The result of equalisation performed on Tiger stripes has been shown in Figure 4.10 and is explained in more details in the Methods chapter.

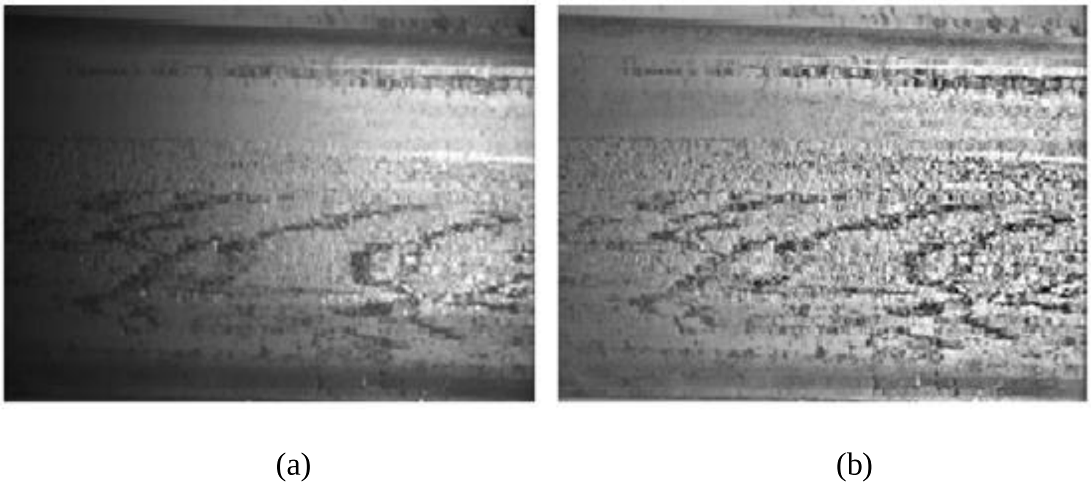


Figure 4.10: Image equalization result (a) Original Tiger stripe image (b) Tiger stripe image after equalization.

4.7 Binarization \ Thresholding

The operation that converts a grey scale image into a binary image is known as binarization that is performed by applying a threshold on the image. Thresholding perform this segmentation (binarization) on the basis of the different intensities or colors in the foreground and background regions of an image. In addition, it is often useful to be able to see what areas of an image consist of pixels whose values lie within a specified range, or band of intensities (or colors). The input to a thresholding operation is typically a greyscale or color image. In the simplest implementation, the output is a binary image representing the segmentation. Black pixels correspond to background and white pixels correspond to foreground (or vice versa). In simple implementations, the segmentation is determined by a single parameter known as the intensity threshold. In a single pass, each pixel in the image is compared with this threshold. If the pixel's intensity is higher than the threshold, the pixel is set to, say, white in the output. The result of thesholding can be seen in Figure 4.11.

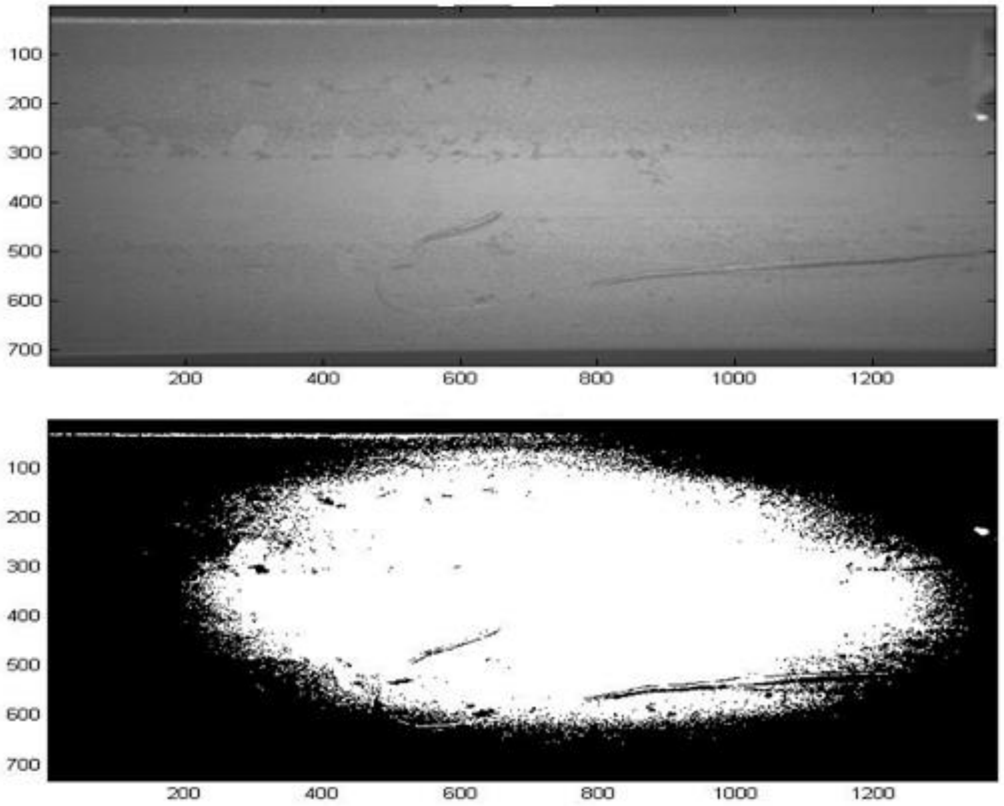


Figure 4.11: (a) Original rail image with a wire type defect (b) Rail image after thresholding.

The faults under consideration are all types of marks or lines. Thresholding is used in all the detection algorithms developed in order to segment out the regions of interest from the backgrounds. The threshold value is different for different defects and is provided by the user in all the developed algorithms.

4.8 Edge Detection

Edge is defined as a jump in intensity of that image. Edges are placed in the image with strong intensity contrast or a strong illumination gradient, which can be highlighted often by calculating the derivatives of the image.

Various methods have been developed to ease the process of edge detection. The methods can be classified into two main categories, Gradient methods and Laplacian methods. The method involving first derivatives is a gradient method and the method involving second derivatives would be a Laplacian method.

Gradient edge detectors have been used in this research. There are many types of gradient edge detectors of which Sobel, Canny, Prewitt, Robinsons and Roberts have been tested and results have been presented in the next section. However, the Robinson Compass and then Susan edge detector came out to be the best ones for detecting the edges from the images used in the research. They have been used to detect edges for the rolled in scrape, line on top and wire type defects.

4.8.1 Robinson Edge Detection used for Rolled in Scrap Detection

Compass Edge Detectors are the alternative to the differential gradient edge detection. The template closest to the local area of the pixel is used to determine the magnitude and orientation of the edge. These detectors have eight kernels each, which are produced by taking one kernel of the respective type and by rotating its coefficients circularly. Each of the resultant kernels is sensitive to an edge orientation. An input image is convolved with these kernels to calculate the gradients along different directions such as South, South-East, East, North-East, North, North-West, West and South-West directions. The maximum gradient value gives the magnitude of the edge (Sudeep K C and Dr. Jharna Majumdar 2011).

The Robinson Compass is one of the types of the kernel that has been used in the research. Robinson’s compass masks or kernels are as follows:

$$N = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad W = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad S = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad E = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}$$

$$NW = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad SW = \begin{bmatrix} 0 & -1 & -2 \\ 1 & 0 & -1 \\ 2 & 1 & 0 \end{bmatrix} \quad SE = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad NE = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

Figure 4.12: Robinson’s compass masks

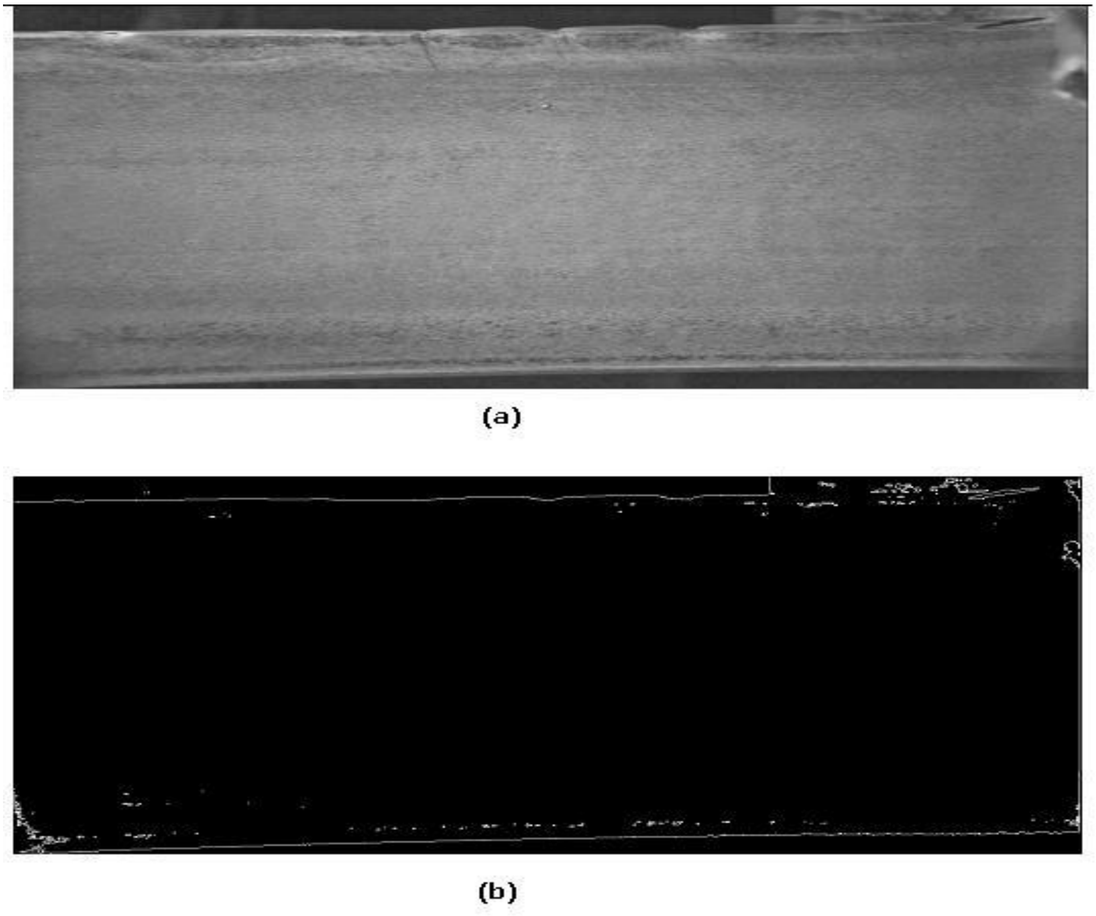


Figure 4.13: (a) Original image (b) Edges detected by Robinson edge detection.

4.8.2 SUSAN Filtering

In some applications the results produced by SUSAN are found better than those from Sobel and Canny. Hence, it was tested for detection process and found very useful at various places. This algorithm performs corner and edge detection without needing any sort of noise reduction or calculation of image derivative.

SUSAN is an acronym for Smallest Univalue Segment Assimilating Nucleus. SUSAN is a digital approximation of circular masks, either with constant or Gaussian weighting to get isotropic responses. The mask is passed throughout the image. At every placement of the mask a central point called nucleus is defined and the brightness of all other points within that mask's placement is compared with the brightness of the nucleus. After that process, an area of the mask with same brightness can be marked. In SUSAN each point of an image is associated with a local area of similar brightness. This association forms the basis of SUSAN and that area of similar brightness is called Univalue Segment Assimilating Nucleus, which contains most of the structural information of the image as shown in the figure 1.13. The USAN area is at a maximum when the nucleus lies in a flat region of the image surface, it falls to half of this maximum very near to a straight edge and falls even further when inside a corner (S.M. Smith 1995).

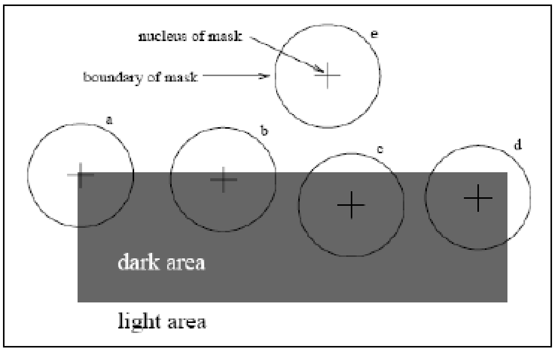


Figure 4.14: Susan functioning (Smith 1995).

Susan algorithm was tested on wire defects with reasonable results. Different customized input masks were tested and applied. The masks tested and results obtained from Susan filtering have been given and explained in wire defect chapter. The result of Susan filtering has been shown in Figure 4.15.

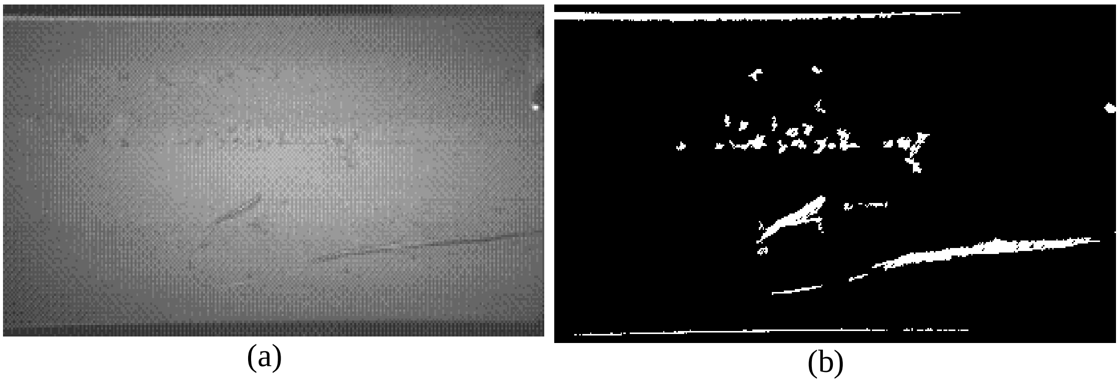


Figure 4.15: Result of Susan edge detector. From left (a) Original image with wire defect (b) Susan edges detected

4.8.3 Comparison of Different Edge Detectors

The results of various edge detectors when applied on original rail image have been shown in Figure 4.17. The original image is presented in Figure 4.16. It is obvious from the images presented that Robinson and SUSAN edge detectors are the best working edge detectors for the type of rail images under investigation. All the other edge detectors either completely failed to detect the defect or picked up lot of noise and non-defect marks from the rail.

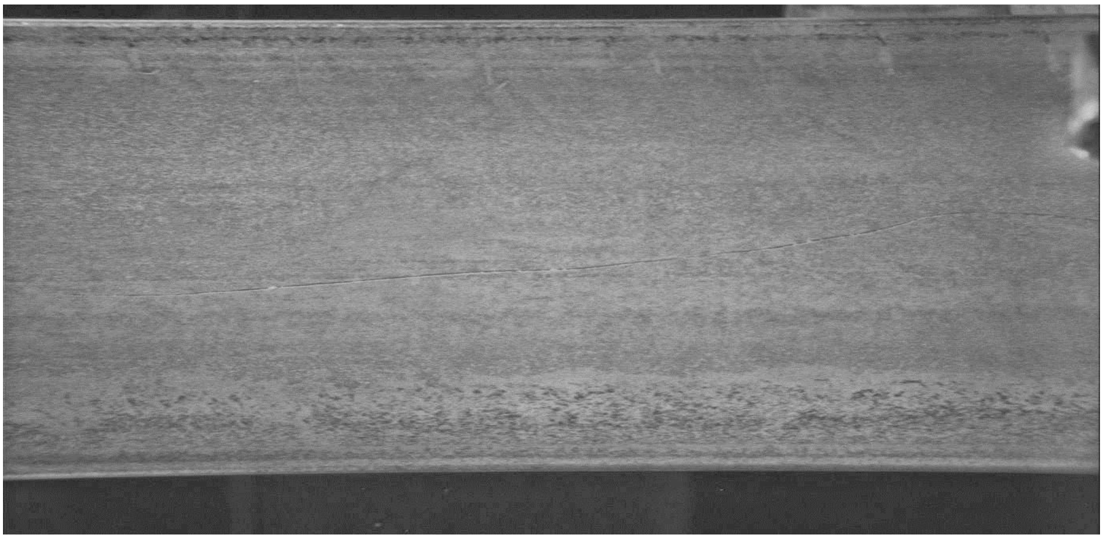


Figure 4.16: Original rail image with wire defect

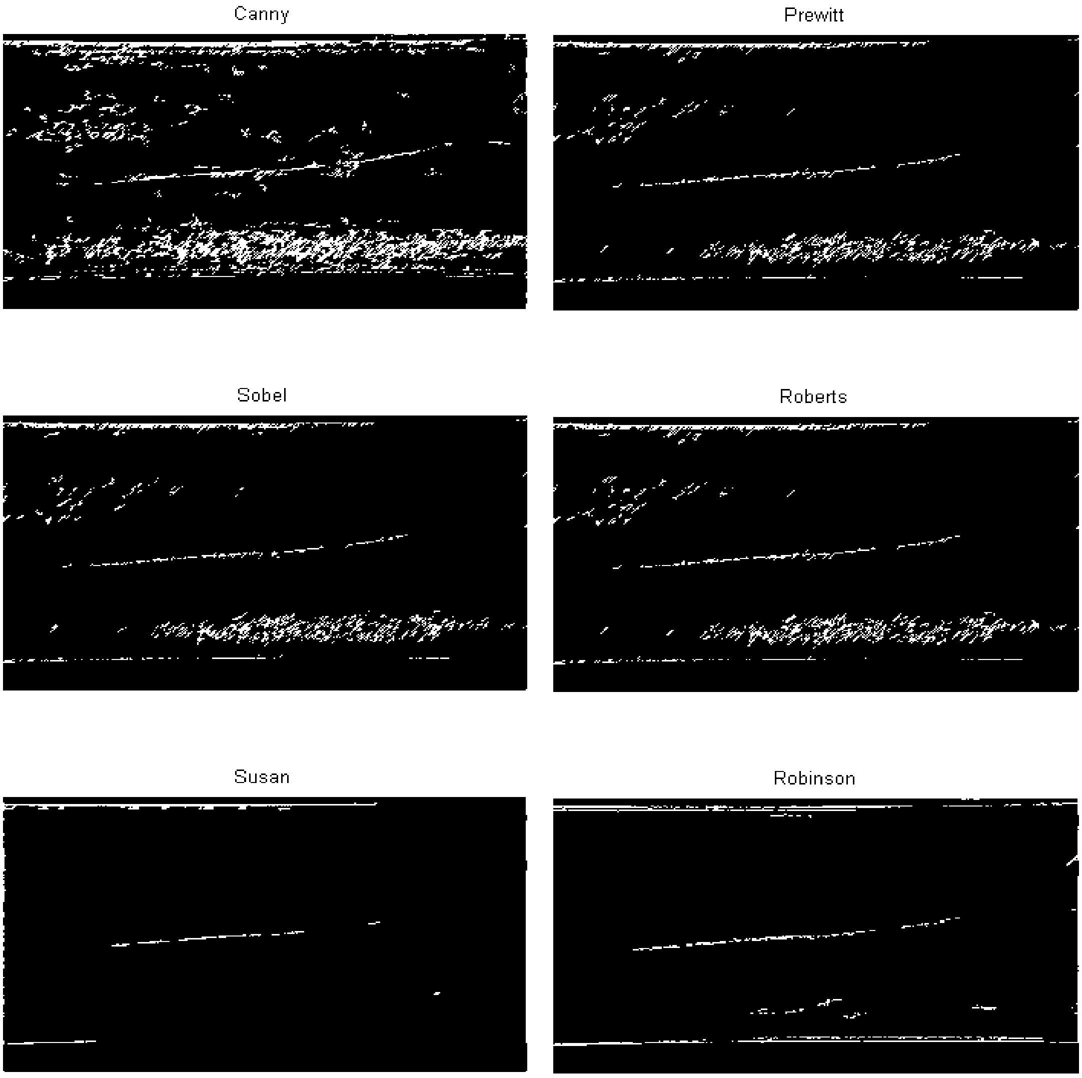


Figure 4.17: Comparison of the results produced by various edge detectors.

4.9 Straight Line Algorithms

There are many different methods and image analysis techniques to find the straight lines in an image. The edge / line tracking algorithms were reviewed and applied on the test images before making an attempt to write a new method. Hough and Radon transform are an example. Hough transform was tested but was not found very effective due to the presence of random similar non-defect lines on the rail. A very simple line detector algorithm was required that followed only the horizontally spreading lines. Hence, own code has been written, that has been explained below.

4.9.1 Hough Transform

There are an infinite number of possible potential lines that can pass through any point, each with different orientation. The standard Hough Transform determines which line, out of these, passes through most features in an image. A Hough Transform is used for detecting lines automatically. It works by determining which edge point belongs to which line. It may be called as a global method for detecting edges. In the process first step is to acquire an image and detect edges by applying any edge detector like Sobel, Canny etc. Hough transform is then performed on edges found to make a histogram. Peaks are detected from the Histogram and the peak point in a Hough space is accumulative value of all the sinusoids, using which straight lines can be defined in the input space. Finally the prominent lines are extracted (Hamarneh et al. 1999).

4.9.2 Specially Made Straight Line Algorithm Used for Lines on Top of the Rail

A simple line detection algorithm has been designed and used for line defect detection. This algorithm detects the high pixels in the columns, as the defect to be detected occurs in the form of horizontal lines. Hence, if the pixel is found high next to it, it starts storing lines. In this way, it stores all the possible connected lines in the horizontal direction in an array. The algorithm was tested to detect line on the top and wire defects. The results obtained are shown in Figure 4.18.

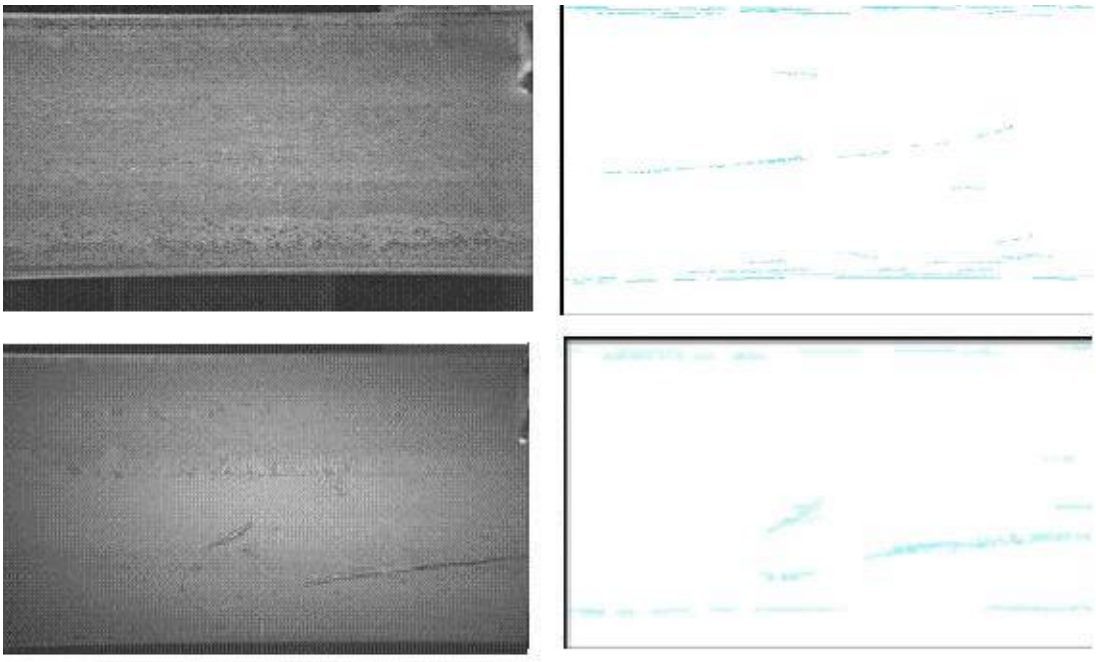


Figure 4.18: Results of wire defect detection using straight line algorithm

However, it was found more useful for detecting lines on the top than wire defect. As can be seen from the figure above, the algorithm was picking irrelevant features as well. Moreover, the wire defects have arbitrary shapes and mostly appear in the form of curves. Hence, the new line detecting algorithm has been used only for line type defects. The results have been shown and discussed in detail in the next chapter.

4.10 Principal Component Analysis

PCA involves a mathematical procedure that transforms a number of possibly correlated variables into a smaller number of uncorrelated variables called principal components.

PCA involves the calculation of the Eigen value decomposition of a data covariance matrix or singular value decomposition of a data matrix, usually after mean centring the data for each attribute. The results of a PCA are usually discussed in terms of component scores and loadings. Often, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data (Mudrov´a and Proch´azka 2005).

The most important use for PCA is that once a pattern or a feature is known in any data, it can be compressed efficiently without any loss of important data.

PCA is the next step for the current research. The plan is to improve the current detection by the use of PCA. It has currently been tested on line on top, tiger stripes and wire type defects. Their application method and results have been given in the later chapters.

4.11 Post Processing

Once the edges, lines and features or in other words defect areas have been extracted from the images, the next step is to classify them. However, before the classification stage, the images need to be processed to get rid of any non-defect areas to avoid the false detection in the latter stage. For this purpose post processing is applied in order to drop all or most of the noise or non-defect areas that have been picked up during the previous processing stage, in order to reduce the false positive detection. Morphological operators have been used for the purpose.

4.11.1 Morphological Operators

Morphology deals with processing the images based on their shapes. Morphological operators usually take binary image and some structuring element as their input. Binary morphology is indifferent to the value of grey level or colour of the pixel. The input image is combined with some structuring element to produce an output image of the same size. Set of operators such as intersection, union, difference, reflection and complement are used to combine input image with the structuring element (Wamani and Villar 2009).

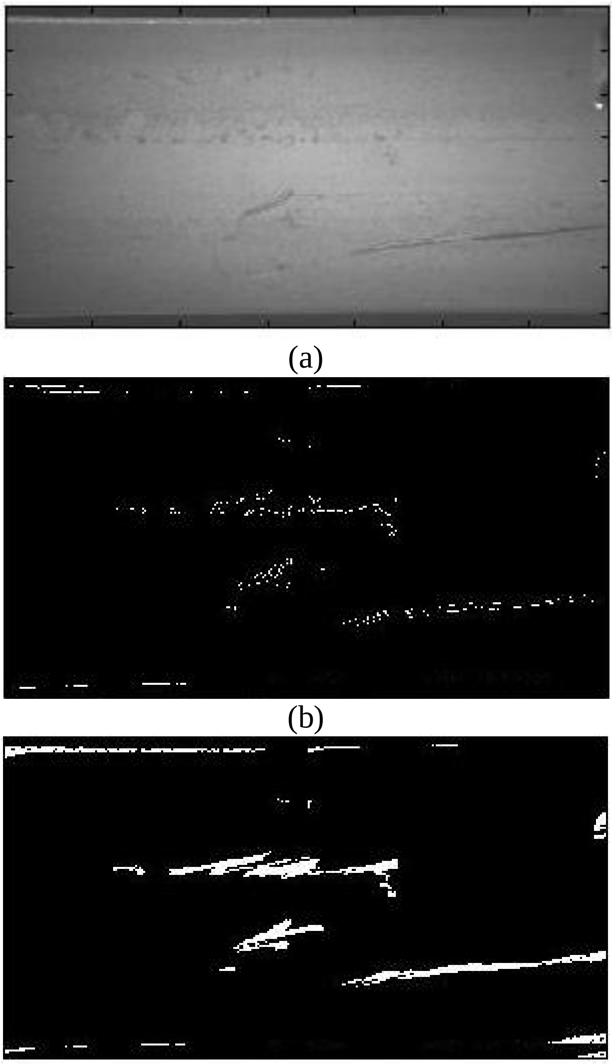
Morphological operation is the comparison of each pixel in the input image with its neighbours. Size and shape of the neighbourhood and shape sensitive morphological operation can be selected in the input image.

Dilation and erosion are the basic morphological operations. Dilation expands the object in the image by adding pixels to its boundaries while erosion removes the boundary pixels from an image object. The size and shape of the structuring element used determines the number of pixels to be added or removed from the objects in an image[1].

[1] http://www.mathworks.co.uk/help/toolbox/images/f18-12508.html

Morphological operators have been used at various stages of detection to enhance the defects and to reduce the back ground noise detected at the same time. Inbuilt functions for open, close, dilate and erode are already available in Matlab and OpenCv library to perform different morphological operations. The example result of dilation operator has been shown in the Figure 4.19. Morphological operators have the ability to join the shapes or areas of interest in an image.

It can clearly be seen that the dilation operation function joined the broken edges and made the defect clearer.



(a)

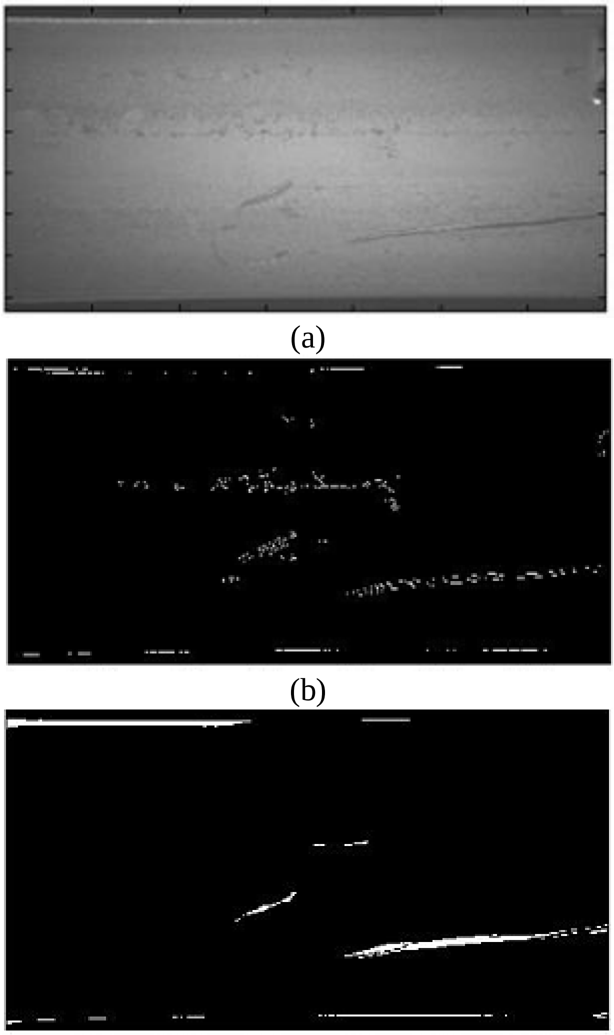
(b)

(c)

Figure 4.19: (a) Original image (b) Canny edges (C) Canny edges after dilation operation

4.11.2 Morphological Operation Performed at Post Processing Stages

Inbuilt functions to perform morphological operations exist in both Matlab and OpenCv and it removes all the blobs (white connected dots) that are >= “size". That's very useful to clean up images. This OpenCV version also fills in gaps smaller than the user defined “size", within larger white areas that are smaller. The function has been modified and used with customized variables to achieve the best results in almost all the detection algorithms developed. The example result image has been shown in. It is very obvious from the result shown that the method helps to get rid of unwanted non-defect detection.



(a)

(b)

(c)

Figure 4.20: (a) Original rail image (b) Canny edges (c) Canny edges after morphological operations.

4.12 Classification Stage

The final stage is the classification stage. The resulting images after all the processing are finally examined for the presence of any type of defect out of the four defects. The defective image is then classified to a particular defect type.

Different classification procedures have been used for different defects. Template matching has been tested on various defects for classification. Neural networks are the next plan to perform the classification. The existing neural network functions have been applied and tested. To date, more work need to be undertaken to complete this work. The future work might be based on neural networks depending on the quality of results achieved in near future. Currently they are under consideration and being researched on.

4.12.1 Template Matching

Template matching is a technique in image processing where an image is segmented or its areas of interests are separated out from the image using a small template. It may also be defined as comparing the parts of the image against each other. The template or the mask or the sample image is matched by moving it to all possible locations on the source image and where ever it finds something similar, it is marked out. The result is numerical index of the highly similar locations.

Template matching can either be feature based or template based. Feature based method uses features such as corners or edges to do the search. Template approach uses the entire template to perform the matching. Matching is performed pixel by pixel and fast template matching can be performed by using cross correlation method (Brunelli 2009) (Lewis 1995).

Inbuilt template matching function of Matlab was tested on wire defects to get the results shown in Figure 4.21. The result image clearly shows extra noise and non-defect detection at the same time. Hence, template matching technique isn’t of much success for the purpose.

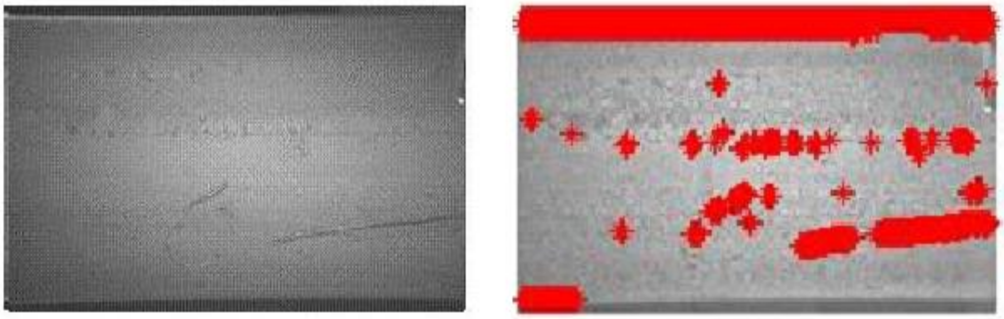


Figure 4.21: Results of wire defect detection using template matching

4.12.2 Artificial Neural Networks

Neural networks are an advanced technique of computer programming. They are efficient at pattern recognition and solving other such problems. An Artificial neural network works in the same way as biological nervous system (brain) works. It is made up of large number of interconnected units, called neurons that are capable of processing information. These neurons work in coordination with each other to solve some problem. As human brain learns by experiences or examples, artificial networks are also configured for a specific application through a learning process.

A neuron is the basic processing unit of a neural network. They receive input in the form of information from some source, process them according by using the learned behaviours, and output the results.

Input layer neurons get input from some real source and the output layer provides the output. All the other processing neurons are hidden. Single neuron is not really helpful, but their combination into multilayer structures, called neural networks, is a solution to many problems.

Neural network has been used in combination to edge detection algorithms and morphological operators in the research. The detection results produced by neural networks have been presented and discussed in the next chapter.

The Feed Forward Back-Propagation architecture is currently the most efficient, simple and popular neural network solution for complex, multi-layered networks. It is used in many applications, as provides workable non-linear solutions to ill-defined problems.

The typical model has an input layer, at least one or more hidden layer(s) and an output layer. Theoretically, there is no limit on the number of hidden layers but typically they are one or two. Feedback is the basic characteristic stage of this type of network. In this stage the output of one layer is directed back to a previous layer. The information flows through in and out layers during recall. The process of simulation of a network with an unknown set of data to get the answer is called recall. Back- propagation occurs only during training and is not used during recall.

4.13 Summary

This chapter has outlined all the image processing techniques that have been tested in order to develop an efficient detection method for the defects under investigation in this research. It has also covered the methods that have been used as part detection in the final developed algorithms. The chapter has been sectioned into pre-processing, processing, post processing and the classification stages for the better understanding of the imaging processes. Next chapter explains the defect detection algorithms developed for the current research in detail.

5 Developed Defect Detection Algorithms

5.1 Introduction

This chapter presents the description of the algorithms developed specifically to process the JLI images together with some of the results of the defect detection being produced on individual defect images provided by Tata Steel, for each type of defect. The offline images were to be treated by the software and a report of the detection was shown on screen for experts to assess.

The software was run on an Intel Core I7 CPU 920 at 2.67 GHz, with 8 GB of RAM, Windows XP 64 Bits Professional. The software was developed using the OpenCV image processing library encapsulated in EMGU for C# (Microsoft Visual Studio Professional). The running time for a rail of on average 110 m long with on average 440 images per camera view is on average 55 s. The time between each rail while in heavy production is more than 1 min 30 seconds, making the software capable of analysing rails in real-time.

5.2 Rolled In Scrap (RIS)

During the manufacturing process, the rail gets rolled in scrap when it exits the furnace on the side of its foot. Rolled in scrap occurs due to the scratching caused by some sharp objects (e.g. corners) left out of the mill's surface while the rail is passing through it. The defect appears in the form of protruding and/or extruding material on the edge of rail's foot. It can be detected by measuring any irregularities of the rail edges. The defect can be seen in the Figure 5.1.

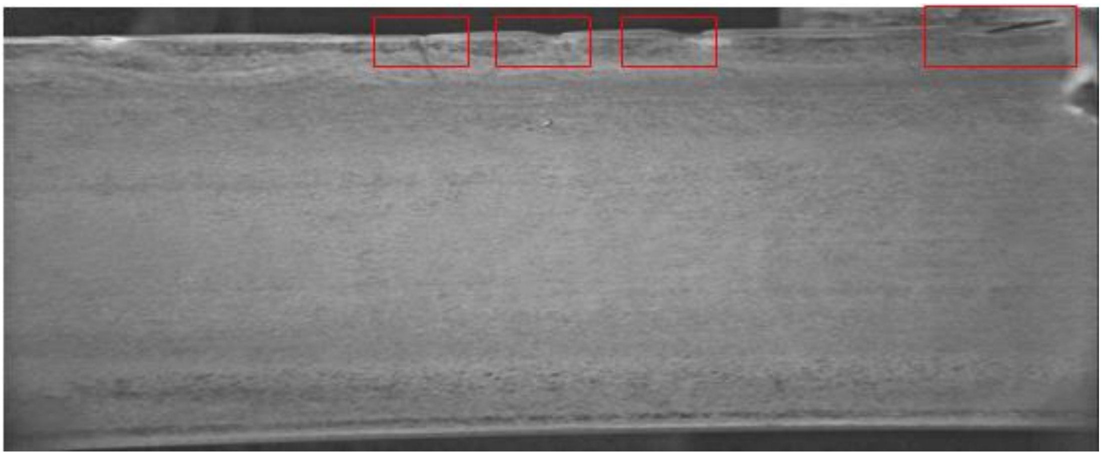


Figure 5.1: Rail image with rolled in scrap

5.2.1 RIS Detection Methodology

The RIS defect appears only on the side of the foot of the rail. They can usually be seen from a particular camera viewpoint as indentations at the edge of the rail. Isolating them involves the detection of the edge of the rail, followed by detection of non-straight lines on the edges. Finally, calculating the depth of the cavities appearing in the processed images due to presence of the defect, which are classified as a defect if exceed certain threshold. An example of RIS has been highlighted in Figure 5.2.

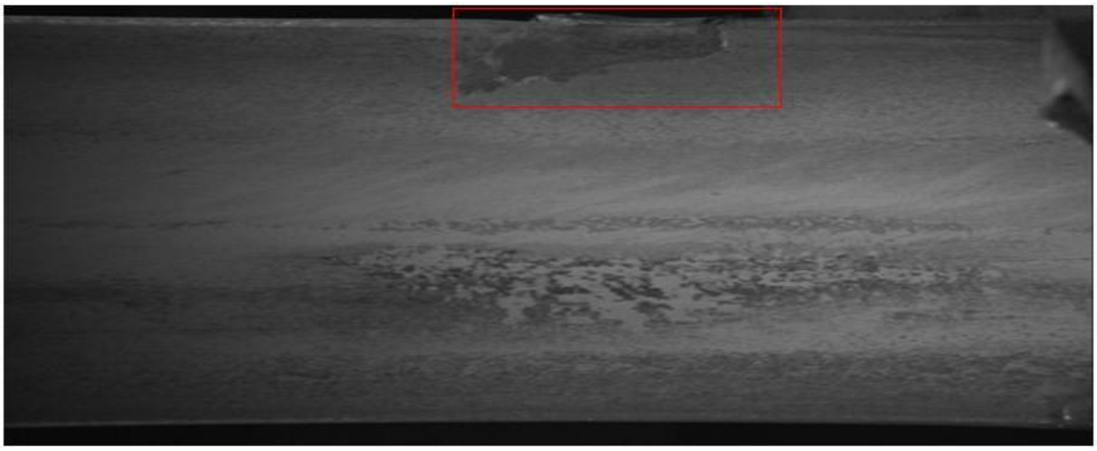


Figure 5.2: Rail image with rolled in scrap defect highlighted by a red rectangle.

Figure 5.3 shows a typical Image after rolled in scrap detection; this represents the initial detection of the defect on the edge of the rail, performed by the developed detection technique which explained in the later section 5.2.3 in detail.

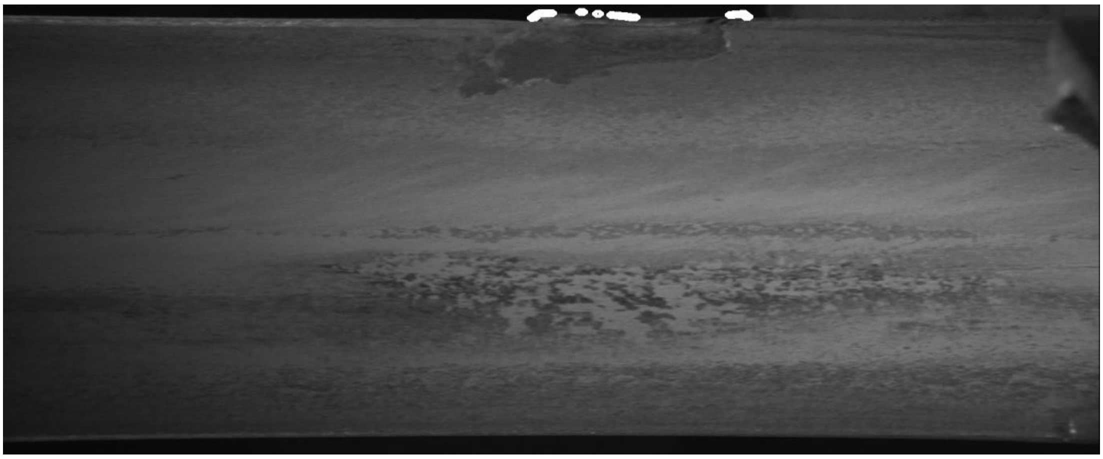


Figure 5.3: Image after RIS detection

The basic logic or methodology behind the detection of RIS can be divided into three basic steps that have been summarised below:

1. A threshold is applied to the image to obtain its shape or in other words separate the edge of the rail from its back ground. The background of the rail in all the images appears to be very dark when compared to the rail that appear grey / lighter and hence, simple thresholding results in black and white image shows the exact shape of the rail. This action is independent of the lighting conditions. This step is shown in Figure 5.4 .

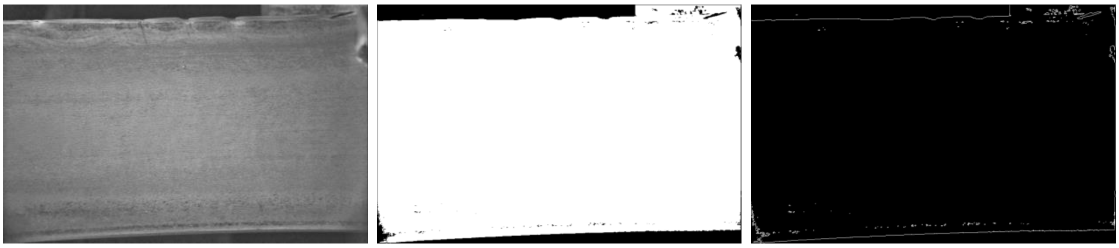

(a) (b) (c)

Figure 5.4: Step for rail edge detection, (a) Original image, (b) After threshold, (c) After edge detection

2. The second step of the algorithm is a standard edge detection using the Robinson kernel to find the edge of the rail, as can be seen in Figure 5.4 (c). Many other edge detectors were also applied and tested, their comparisons have been provided in section 0. However Robinson kernel for the edge detection was found comparatively better. The edge detection process and the Robinson kernel have been explained before in detail in section 4.8.1. The Robinson Compass is one of the types of kernel that came out to be the best for detecting the edges from the images used in the research because most of the defects appear to be in the form of horizontal lines. Hence, the Robinson kernel detecting horizontal spread of edges has been found quite useful and been used.

3. The final step is to find the distance of the edge from the edge of the image in terms of total number of pixels in between.

5.2.2 Parameters Required for RIS Detection

There are few main parameters that are required by the algorithm to perform detection, which can be determined by the user at run time in order to make the system more flexible. These are:

1. **Threshold:** A low threshold value is determined by the user to produce the binary image. This threshold is used to separate the rail from its background.

2. **Gradient:** It determines the minimum length of a difference or slope of the edge found to qualify as a defect

3. **Length of Deviation (L):** Deviations of some certain lengths are classified as defects. Length parameter helps to determine the minimum length required by the anomaly to qualify as a defect.

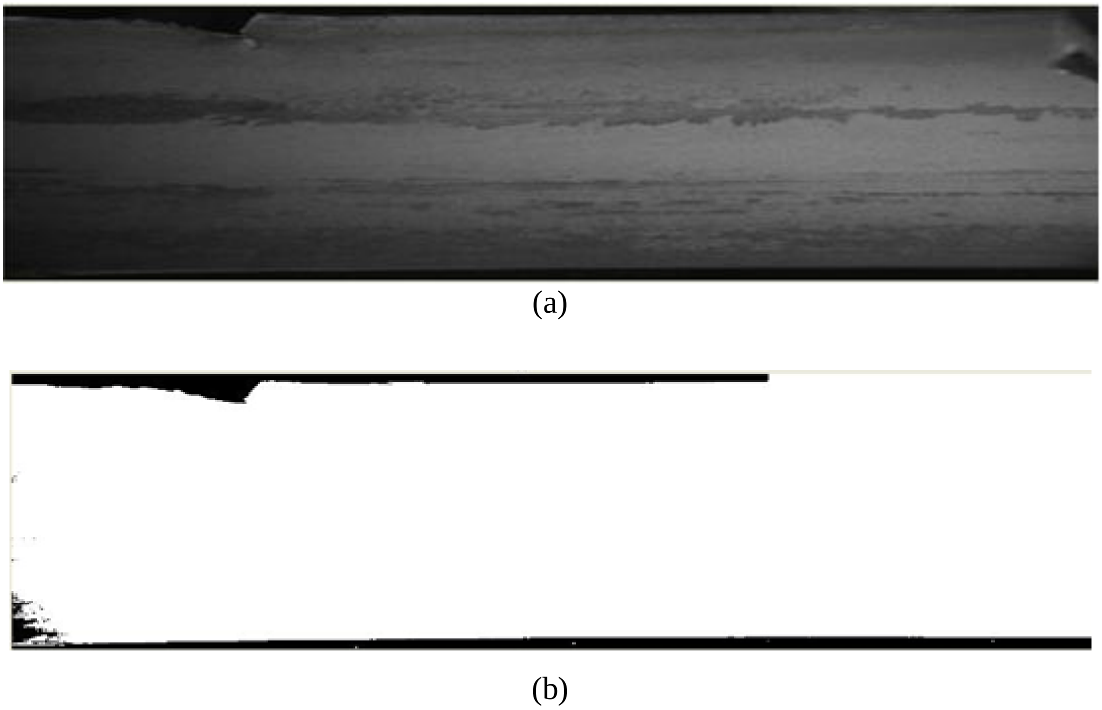
4. **Gap:** It determines the distance between two consecutive pixels for deviation to be computed. Usually the Gap is empirically chosen to be 50 pixels, which means that the length of each pixel from the top or the bottom is compared with the length of another pixel, 50 pixels away from it.

5.2.3 Algorithm Implementation for RIS

The detection algorithm for Rolled in Scrap has following steps:

1. **Thresholding:**

User defined threshold value is applied to an input, which is a grey scale image to turn it to its binary form. All the pixels having value greater than the threshold level are assigned maximum value which is 255 or 1 for white. While all other values which are less than the threshold value are assigned 0 for black. The result of the thresholding operation can be seen in Figure 5.5.



(a)

(b)

Figure 5.5: (a) Original image with RIS defect (b) Image after applying threshold

2. **Morphological Operation:**

   A morphological close operation is then applied to close the broken edges and get better view of the binary image. The functions usually need parameters for structuring element, operation specification (close or open etc.) and number of time the operation is to be performed (iteration). A rectangular structuring element of size 5x2 has been used to perform morphological close operation for RIS detection. The operation makes the edges look smooth and continuous in the binary image form of a rail and is performed only once

3. **Edge Detection:**

   RIS can appear at either top or bottom edges of the rail. Hence, top and bottom edge lines need to be searched. The gradient edge detection has been performed using a bespoke algorithm based on the Robinson gradient edge detector method. Edges pixels are at local maxima of gradient magnitude and gradient direction is always perpendicular to the edge direction. This observation has been used to develop an algorithm that finds the maxima of gradient magnitude and classifies them as TopLine and BottomLine. TopLine refers to the top edge of the rail and BottomLine refers to the bottom edge.

   A Robinson edge detector has 8 kernels in different directions as explained in chapter 4, section 4.8.1. For each location in the image, the gradient is calculated using neighbors of that pixel, for each kernel. The maximum gradient is assigned to that image location, which eventually gives the edges of the image. However, the kernels detecting horizontal lines have mostly been used according to the requirement in most cases and in some cases the two or three kernel results have been added together to get better edges. The result of edge detection is presented in Figure 5.6.

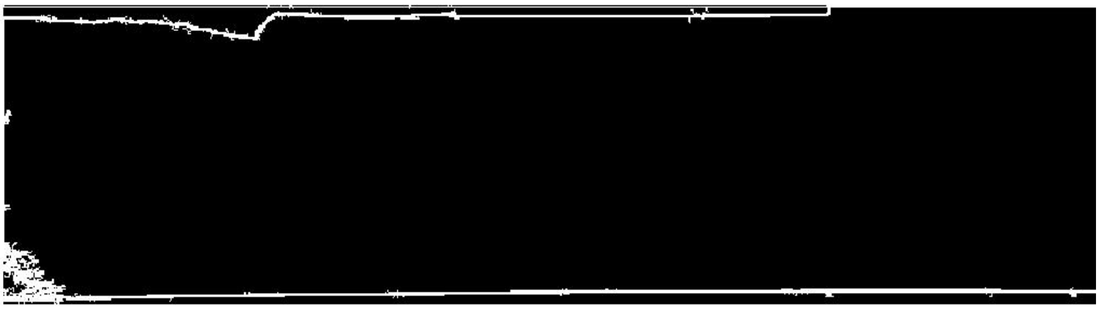


Figure 5.6: Result of edge detection step using Robinson kernel for RIS defect

4. **Top & Bottom Line Distance:**

   Once top and bottom edge lines are marked, their respective distances from top and bottom far edges of the image are calculated respectively. Distances for each edge pixel in both the top and bottom lines are stored in their respective arrays. This can be written as follows:

   Top Distance Array = Distance of each point of top edge line from the top edge of the image in terms of total number of pixels.

   Bottom Distance Array = Distance of each point of bottom edge line from the bottom edge of the image in terms of total number of pixels.

   These expressions can be represented in general equation form as shown below:

$$p(i) = \sum_{i=0}^{k} \mathrm{d(i)} \qquad \textit{Eq. (5.1)}$$

   Where,

   p = Array containing distances of all the points on the top or bottom line from their respective image edges.

   d = Total number of pixels from any point 'i' at the top or bottom line from respective image edge.

   k = Maximum number of pixels in the top or bottom line.

   The graphical representation of the above equation is shown below in Figure 5.7.

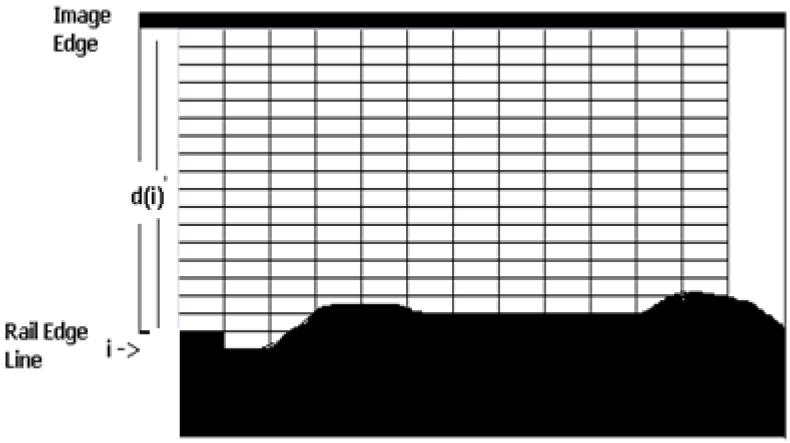
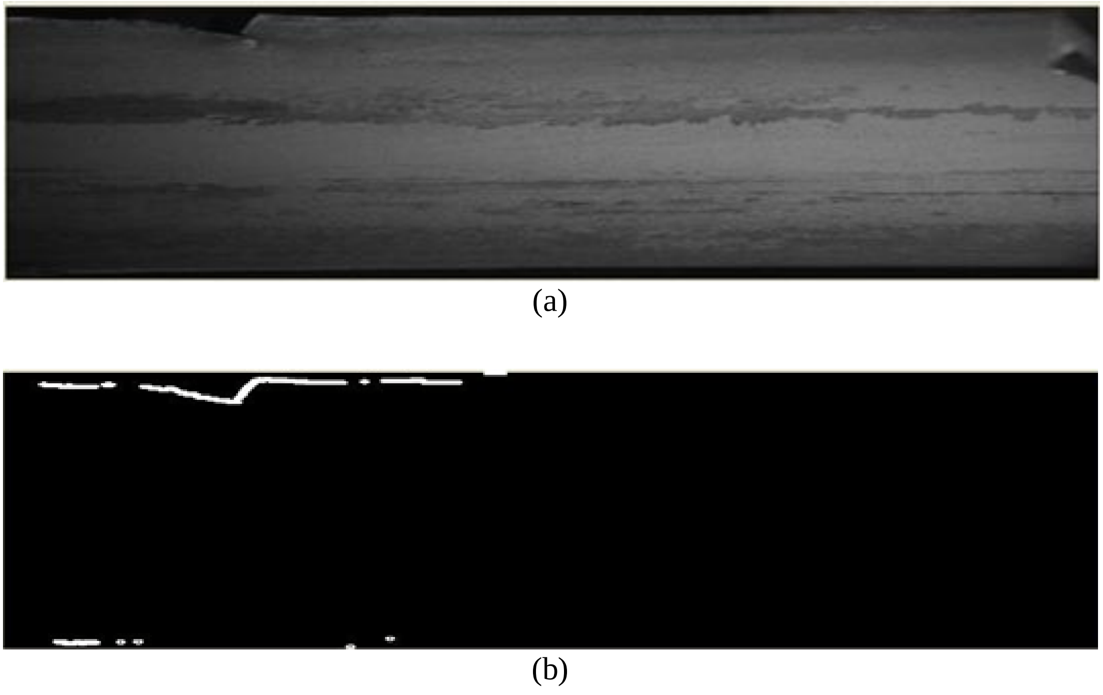


Figure 5.7: Graphical representation of distance

Top and bottom lines with significant deviation are marked or drawn on a blank copy of the image as shown in Figure 5.8.


(a)

(b)

Figure 5.8: RIS detection result (a) Rail image with RIS (b) RIS top and bottom lines marked

5. **Gradient \ Deviation Calculation:**

   The defective edge areas are then marked or classified as defects by calculating deviation of the distances of the rail edges. The deviation is the absolute difference between the deviations (in terms of number of pixels) calculated in the last step some G gap apart. The deviation is assessed over k successive pixels, as follow:

$$D(i) = \sum_{i=0}^{k} |\mathrm{p}(i) - p(i + G)| \qquad \textit{Eq. (5.2)}$$

Here,

D(i) = Deviation between distances at point i and i+G

p(i) = Distance from the edge at point i

G = Distance between two consecutive pixels for deviation to be computed

k = Total number of pixels or length of the area to be scanned in an edge line

In the above Eq. (5.2), p(i) represents the distance from the edge of the image and G represents an arbitrary gap between the comparisons of the pixel values. G is defined as the width at which a deviation becomes significant, i.e. at the position of the extrusion / protrusion. The deviation between two successive pixels where there is defect is close to the deviation of two successive pixels where there is no defect, whereas the deviation between an area with no defect and an area with defect is high. The resulting deviation, D provided a close overview of the condition of the edge of the rail and each of its values can be then assessed as being part of a defect or not, the higher the value, the bigger the defect. The parameter k allows the variation of the length of protrusion / extrusion to be detected, whereas the parameter G is for the depth.

5.2.4 Classification of RIS

For rolled in scrap detection, the final step is to find the degree of unevenness calculated in the previous steps, which is found out using a specially developed algorithm. The whole process can be summarised as finding the distance of the edge of the rail from edge of the image in number of pixels followed by calculating the deviation of the each distance value from another value G pixels apart. Finally, in the classification stage, the areas of a certain length where the deviation values are all greater than predefined gradient value are marked as a defect. The bigger the area, the larger the defect and the higher the value of the deviation, the deeper is the defect. This is done for both top and bottom line of the rail.

The classification process is done by comparing the value of deviation computed in the last step, with a gradient value defined by the user. Hence, every deviation value greater than gradient triggers the program to start counting the occurrences of deviation for which it is greater than the gradient value, from that pixel point onwards.

Finally, if the number of consecutive occurrences of deviation values greater than gradient equals or exceeds the Length of Deviation (L), the area is marked as a defect.

$$\mathrm{N} = \sum_{i=0}^{k} \mathrm{D(i)} > X \qquad \textit{Eq. (5.3)}$$

$$\mathrm{N} \geq L => Defect \qquad \textit{Eq. (5.4)}$$

Where,

N = Number of consecutive occurrences of deviation values greater than gradient

D(i) = Deviation of distances between two points on the edge

L= Length of Deviation

X = Gradient value defined by the user

k = Total number of pixels or length of the area to be scanned in an edge line

The gradient value of 2 pixels has been found to be the best empirically and has been normally used for detection. The minimum length of deviation, again decided empirically is usually 5 pixels. Hence, every deviation found at the edges which is more than 5 pixels wide and 2 pixels deep is classified as a RIS defect. A typical result from Rolled in Scrap detection has been shown in Figure 5.9.

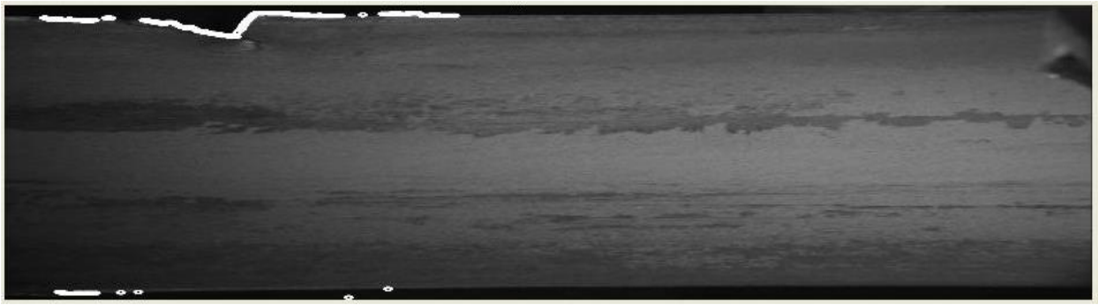


Figure 5.9: RIS defect detected and marked on original image

5.2.5 Detection Rate

The rolled in scrap detection rate is very high with almost 100% positive. The results have been concluded after performing successful detection on the data set of rail images only with RIS type defect, provided by TATA steel. The available data set has over 1000 images with the RIS type defect. All the rolled in scrap type defects that were present on the set of images submitted by Tata Steel were successfully detected, with a small amount of false positive. False positive RIS detection rate is only 5% for the current algorithm which occurs due to dust particles appearing on the images close to the rail edge, incorrectly detected as faults.

5.3 Line on Top of the Rail

Line on top of the rail appears in the form of horizontally running lines on the middle of the top surface of the rail. It is usually captured by the top middle camera of the JLI system. Detection of line on the top of the rail is a different challenge that involves edge detection, using a customized edge detector, followed by detection of straight lines. Once again standard image processing routines failed to detect the line. Hough transform and radon transform for line detection were considered and Hough transform was even tested, but were not found effective. The testing details have been provided in section 4.9. A representative image for line on the top of rail is shown in Figure 5.10. The defect line has been highlighted for the better viewing with a white colour line.

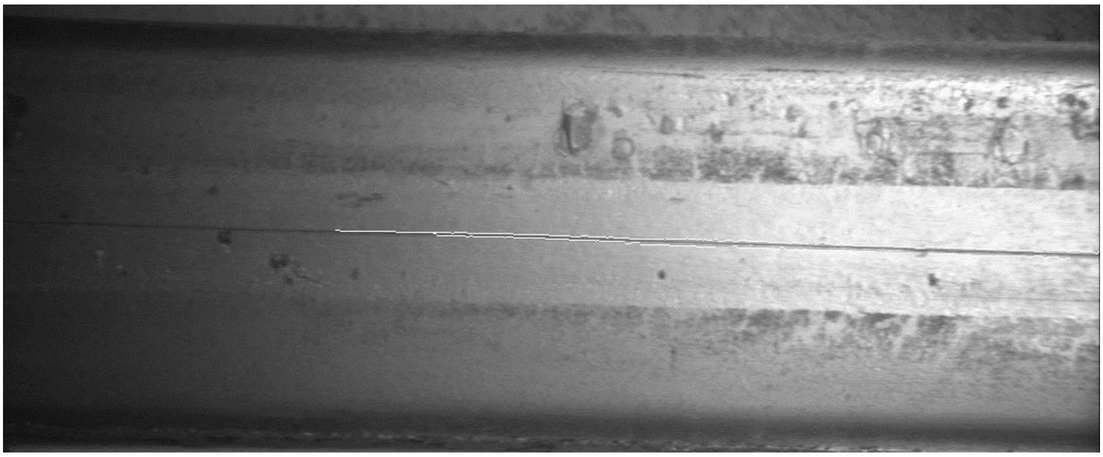


Figure 5.10: Original Image with Line on the Top of Rail Defect

The general observation according to the information provided by Tata is that these defect lines are present right in the middle of top surface of the rail; and run on successively on consecutive images. This observation gives a clue as to where to look for the defect or what part of the rail is to be examined closely and also the minimum length of any detected line, on that special area, to be classified as a defect.

5.3.1 Line Detection Methodology

There are various existing line following or line detection algorithms such as the Hough and Radon transforms. Many of them were exhaustively tried on the line defect images, but none of them gave any satisfactory results (details provided in section 4.9). Actually, the defects mainly under investigation had horizontal lines in them, so a relatively simple line detections algorithm was needed, which has been designed and used.

The methodology for detection of lines has following basic steps:

1. Find all the possible lines in the horizontal direction, only at the middle part of the rail’s tip surface by filtering. This is done by detecting high pixels in each column and other high pixels next to it in the horizontal direction. If the adjacent connected pixels are found high, the algorithm starts storing lines. In this way an accumulation of all the possible connected lines in the horizontal direction is stored in arrays.

2. Compare the lengths of the detected lines to a pre-determine length of the

line defect.

3. Any of these lines with a length equal to or greater to the predefined length and also runs on successive images is classified as a defect.

5.3.2 Parameters Required for Line Defect Detection

The parameters that are required by the algorithm to perform detection can be determined by the user at run time. The detection algorithm developed here uses these parameters:

5. **Threshold:** A threshold value is determined to separate all the visible lines from the average value on the rail surface.

6. **Minimum Length:** The minimum length parameter for detecting straight lines is used. Only the lines with the lengths greater than this minimum length are classified as defects.
7. **Threshold for Morphological Operation**: Threshold value to perform morphological operation is used to clear blobs and noise from the image. In other words filter out lines and places of significant lengths and areas.

5.3.3 Algorithm Implementation for Detection of a Line on the Top of the Rail

The line on top of the rail is again detected in two steps. The algorithm consists of edge detection coupled with a straight line finder algorithm.

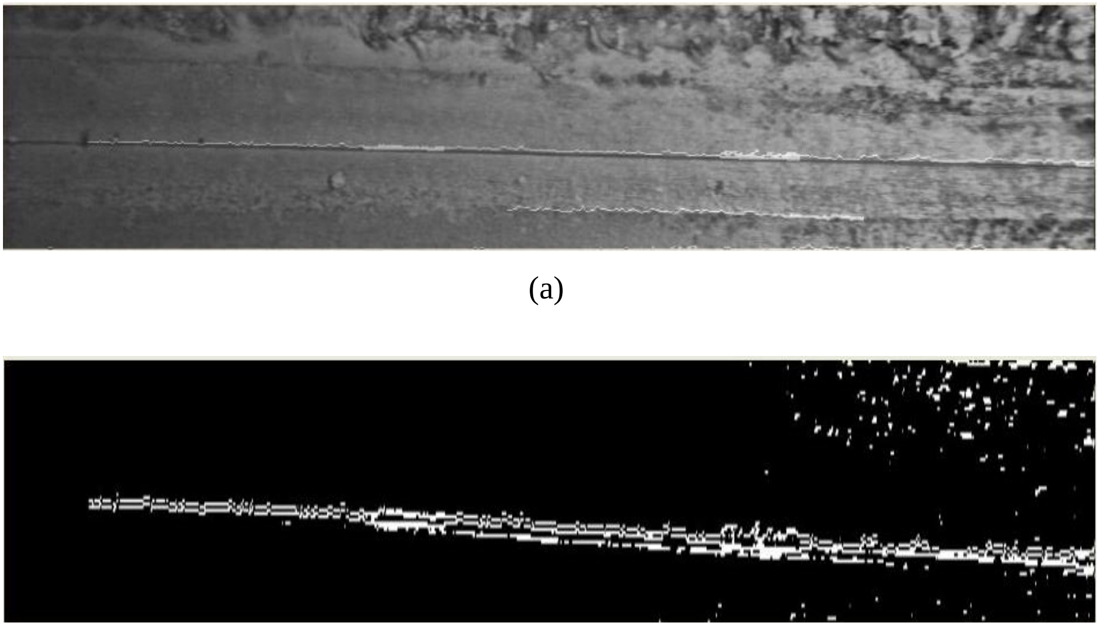
1. Filter Horizontal Lines \ Horizontal Edges:

In this step of detection all the horizontal lines existing in the image are filtered out using a kernel. The edge detection process has been explained in detail in the previous chapter. The mask is convolved with the original image and gradient is calculated at every point in both x and y directions to detect both vertical and horizontal lines. However, here only horizontal lines are to be detected. There are inbuilt functions available both in Matlab and OpenCV to detect the edges. However, to apply the mask mentioned above, a bespoke C# algorithm has been written. The code convolves the mask with the original image to find edges/horizontal lines, and is based on simple logic using loops and conditions.

An input of grey scale image is convolved with a kernel in order to filter out all the horizontal lines in the image. The example of the kernel used is shown below.

$$k=\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The above kernel filters out the horizontal line in an image only. The resulting convolved image contains all the horizontal lines and the example of the image obtained is shown in Figure 5.11.



(a)

(b)

Figure 5.11: Result of horizontal line detection (a) Original Image (b) Image with Lines Filtered Out in Horizontal Direction

2. **Threshold to Binary:**

Grey level slicing is used to highlight a specific range of grey levels in an image. The images being investigated for the research are all grey level images. The defect lines appear at different grey levels than rest of the back ground of the rail. Hence, grey level slicing has been used to separate background from the defect using a certain threshold. User defined threshold value is applied to slice the image resulted from the last step to generate a binary image. All the pixels having the value greater than the threshold level are assigned maximum value which is 255 or 1 for white. While all other values which are less than the threshold value are assigned 0 for black. For this specific type of defect the best working range of threshold slicing values empirically calculated is between 60 and 70. Level slicing graph has been shown in Figure 5.12: Grey-Level Slicing of Image.

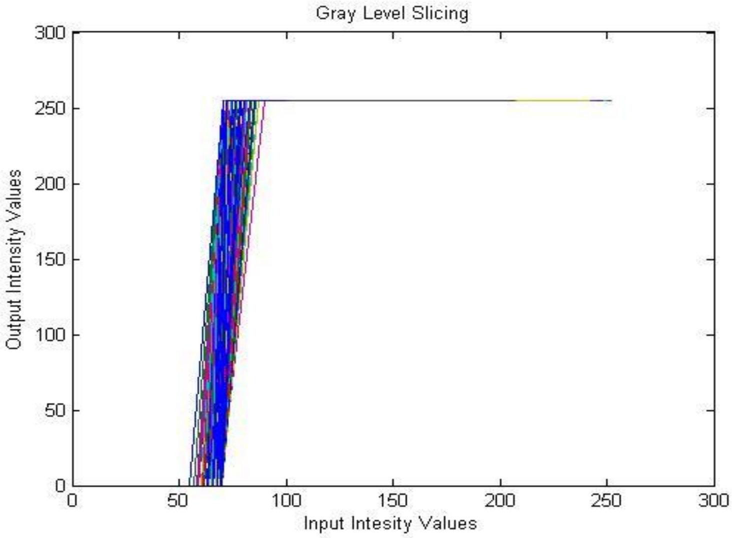
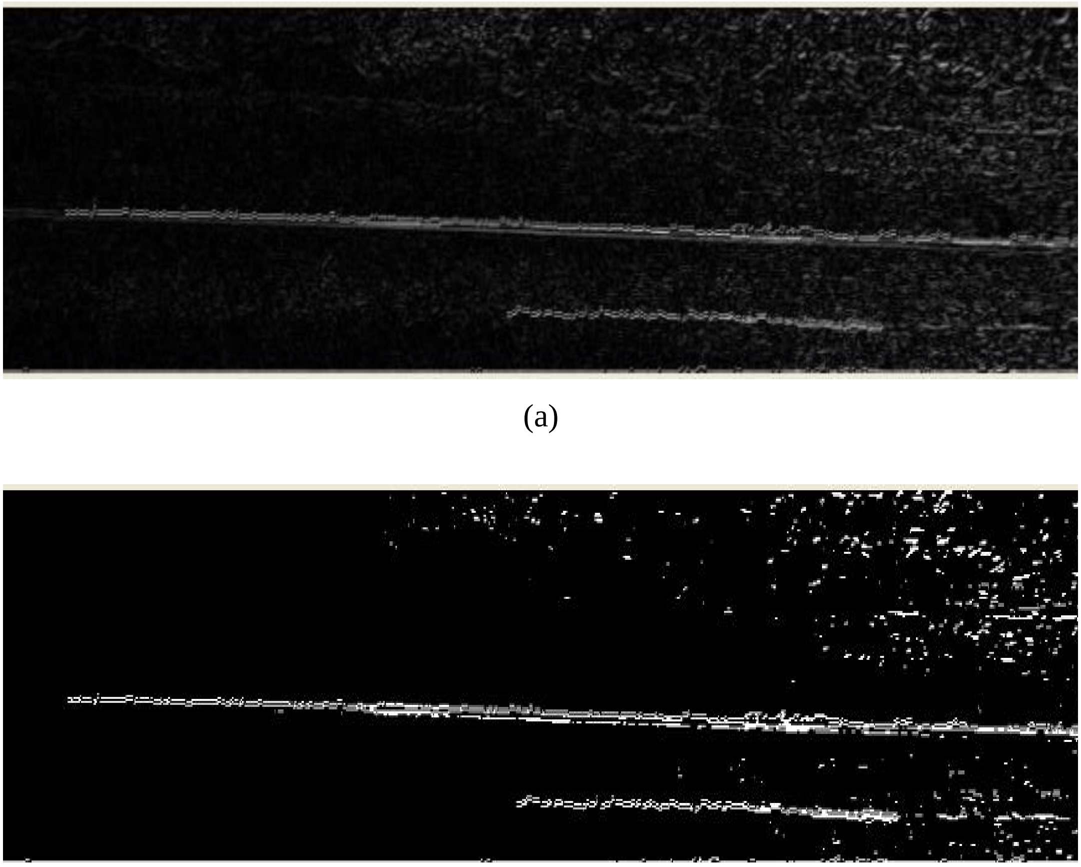


Figure 5.12: Grey-Level Slicing of Image

The resulting image obtained after thresholding or grey level slicing has been shown in Figure 5.13.



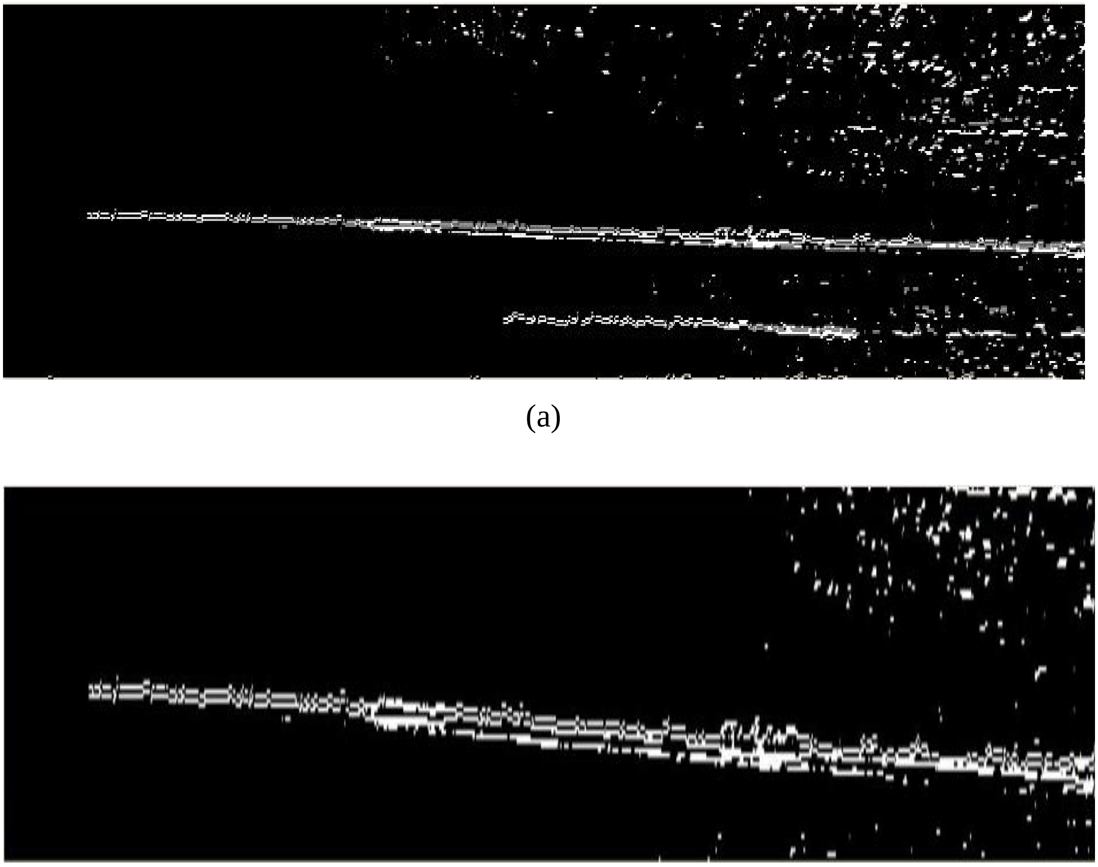
(a)

(b)

Figure 5.13: Threshold result (a) Image after filtering horizontal Lines (b) Thresholding Result.

2. **Select Middle Area:**

   Once all the horizontal lines are known, the region of interest is set to the middle area. Hence, only that part is processed for the later investigations, rest of the image is ignored. This is done in order to reduce the computation to be done as line defect only appears \ runs on the middle area of the rail. The result of region of interest selection is shown in Figure 5.14.



(a)

(b)

Figure 5.14: Middle area selection result for Line on top of the rail defect (a) Binary Image with Horizontal Lines (b) Region of Interest set to middle area

3. **Removing Non-defect Lines and Noise:**

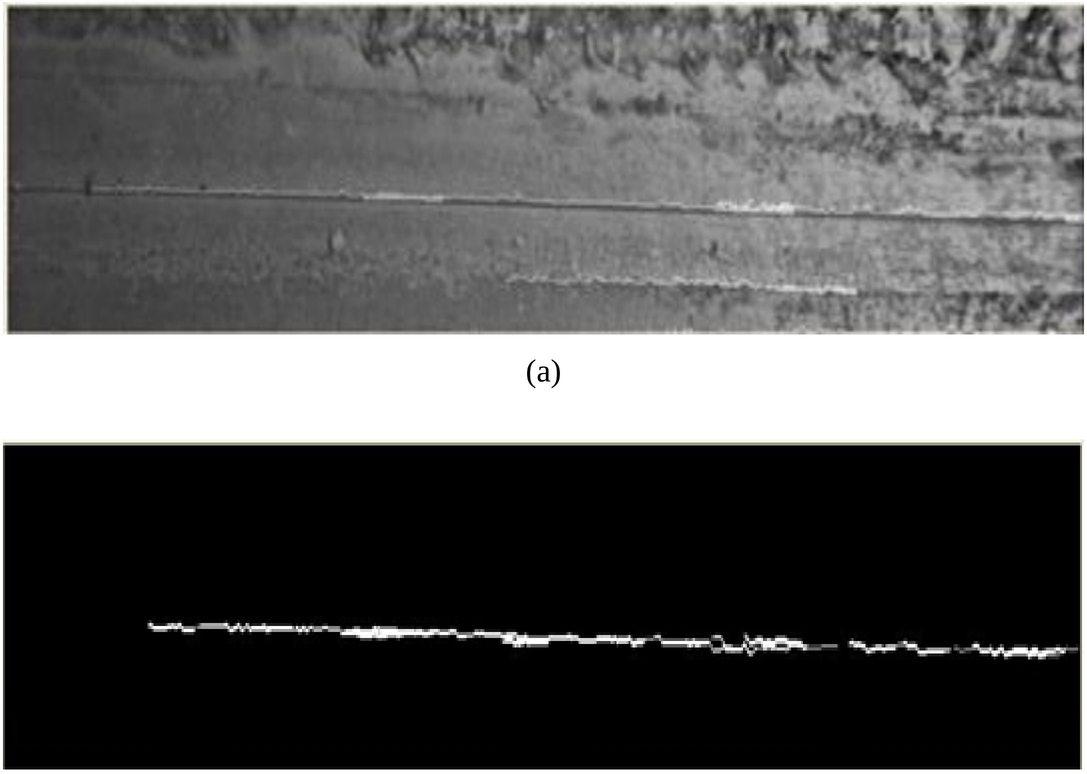
   The lines with some minimum area or length are the significant points; the rest are noise that needs to be removed from the image obtained at the end of last step. Modified form of morphological operation method is used to draw only the contours that have length greater than some predefined threshold value, all other lines or edges are dropped. Typically BWAreaOpen method is used for the purpose, which is available in both Matlab and OpenCV. The method accepts two parameters, an image and a threshold value. It removes all the objects that are smaller than that given threshold. However, a customized method for morphological operation has been defined in C# and used. The method uses three parameters, an image, and threshold and thickness values. The method gets an image and finds all the edges \ contours in it and redraws the ones that have area greater than the threshold provided; on new blank copy of the image. The edges drawn have width equal to the thickness value provided. Hence, the function generates an image with size exactly equal to the original image (image received by the method) but containing only those significant edges \ lines that has length \ area greater or equal to the threshold value provided; and pre-defined thickness. In other words the resulting image contains only the lines that possibly are a defect.

5.3.4 Classification \ Storing Horizontal straight lines

The next step is finding straight line/lines; this is again a specially made algorithm. This part includes eliminating all the lines that are not growing or spreading horizontally and are smaller than a certain length. The whole image is scanned column wise and any pixel found high is stored in an array. That ever pixel stored in that array is checked horizontally. Every pixel stored in the array when taken from left to right must have its successive pixels on successive columns of the image, to be a line defect. Lengths for all the lines are saved this way and finally filtered by using the length parameter. The code for the purpose have been written using combination of simple programming routines such as arrays, loops with simple checks such as high pixel value check etc.

A small recursive routine has been written to follow all the lines horizontally, originating anywhere in the selected middle region of the image. For each high pixel stored in the array, that routine is called to follow the pixel's horizontal spread. The routine is called recursively until the line ends and saves the length of the horizontal spread. Number of time the routine is called is actually the length of the line found, which is compared with threshold.

The final step is to compare the length of all the detected lines with minimum length parameter. The lines with certain length are finally classified as defects and marked on the output image, while all other lines are dropped. Example of line defect detected by the software is shown in Figure 5.15.



(a)

(b)

Figure 5.15: Result of Line on the top of rail type defect's detection (a) Original Image (b) Line Defect Detected

5.3.5 Detection Rate

The line on top detection rate of current algorithm is of 92%. There are 8% of the lines not detected by the system. The results have been concluded after performing successful detection on the data set of rail images only with Line on the Top type defect, provided by TATA steel. The available data set has more 1000 images defected by Line on the Top type defect. However, those are faint lines that according to expert are not as critical as the deep appearing defects detected by the algorithm. There is a very small percentage of false positive, which is found to be only 1% that is negligible.

5.3.6 Neural Networks Testing for Line on the Top of the Rail

An alternative approach has been investigated that involves the use of neural networks. This section covers the method and results obtained by the application of neural network method for the defect detection. A Feed Forward Back propagation neural network has been used to get the results.

Neural networks are an advanced pattern recognition technique of computer programming. They are proving to be very efficient at solving other similar problems. Hence, it is worth trying to use them for defect detection. The results obtained so far, are very encouraging and expected to be improved to a greater extent in future.

Different operations and filters are applied to digital images by convolution with a mask. A certain size of mask is chosen empirically and is moved over the whole image. The function to be performed or the calculation is repeated for every position of the mask. A range of statistical measures have been tested and are used mainly to represent the image. Statistical measures has been used as the rail images provided by Tata Steel are noise and textured. The basic statistics mainly used in the research are as follow:

- Mean
- Variance
- Standard Deviation
- Median

The Mask size can be chosen at run time but most of the experimental results listed here are produced by mask size of 6 by 6 or 11 by 11.

The mask is moved by a step of one or more over the whole image with the statistical information stored in arrays. The combined statistics array has every of these statistic as one of its row. All or combination of the statistics have been tested to get the results.

<u>Implementation:</u>

For training purpose, the part of the image containing the defect is taken as the target. The statistical values of this area are stored into an output array. The input array contains the statistical values calculated by masking the whole image. The network is then trained on this data.

The trained network is then used to classify the statistical values of any image, to detect any similar defect/statistics it has been trained on initially. This process is called simulating the network.

The results from neural network based detection of line on the top of the rail are shown below in Figure 5.16.

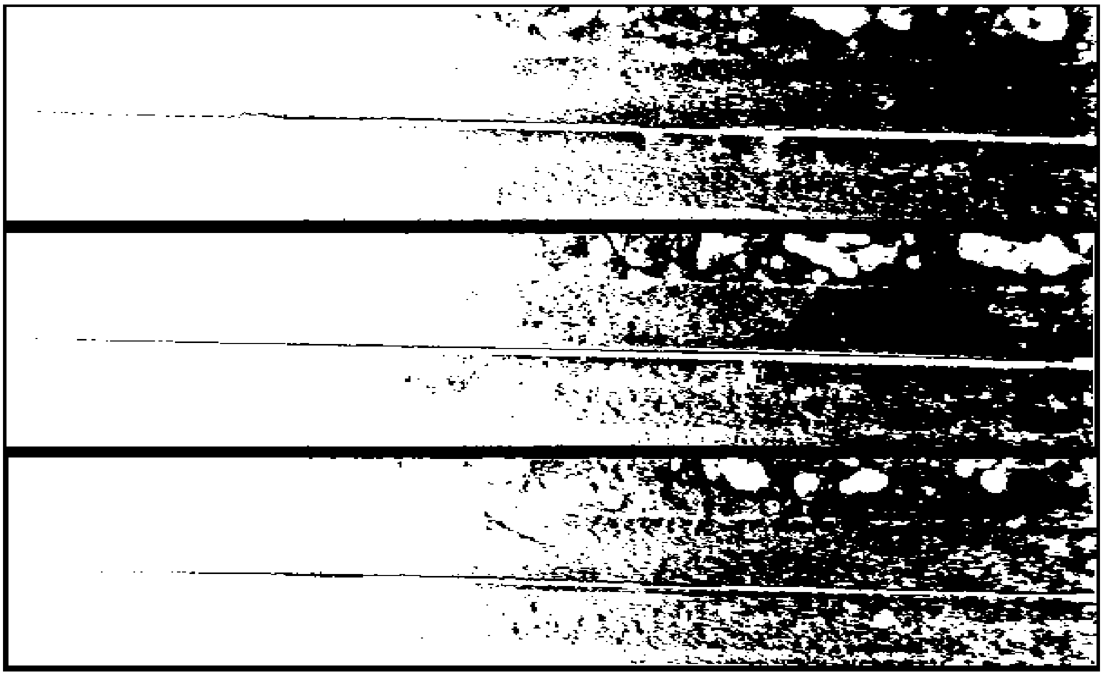


Figure 5.16: the detection of a Line on the top of the rail detection using neural networks

The detection results obtained from the neural networks testing contains lot of noise detected as well, as can been seen in the Figure 5.16 above. The results are not very satisfactory as compared to the detection results produced by the previous method explained before in this section (results provided in section 5.3.4).

5.4 Tiger Stripe (TS)

“Tiger stripes” are dark areas that appear on the rail top surface when the rail is left under the cooling water longer than required. They occur at the top of the rail as known as crown of the rail. Hence, they are mostly captured by the top middle camera and sometimes also by the right and left cameras on the tops edge.

The marks are of variable sizes but with a characteristic shape, which appears like a roughly sideways drawn zigzag pattern of various sizes. Their successive occurrence on many images together helps in their detection. However, their detection is not straightforward as last two defects, due to their largely different sizes. An example image is shown Figure 5.17.

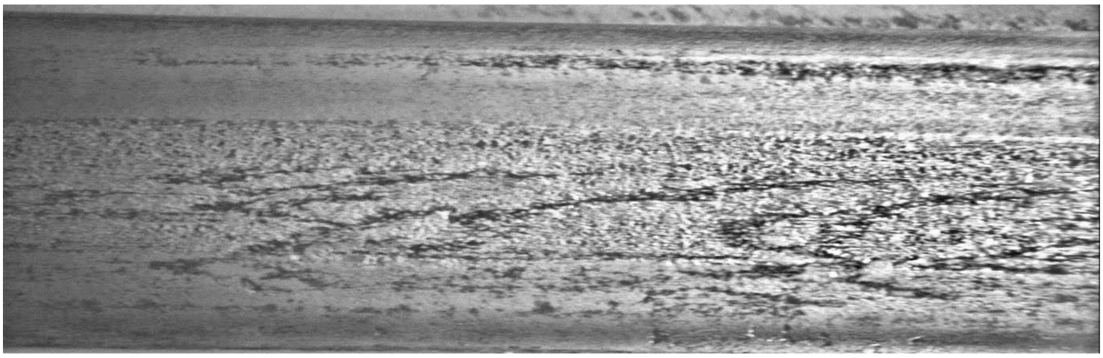


Figure 5.17: Large tiger stripe pattern – Light adjusted image

The TS can be as large as shown in the figure above and can be as small as shown in Figure 5.18 below. Both the example images presented here are contrast adjusted in Matlab to suggest the clear \ better view of the defect.

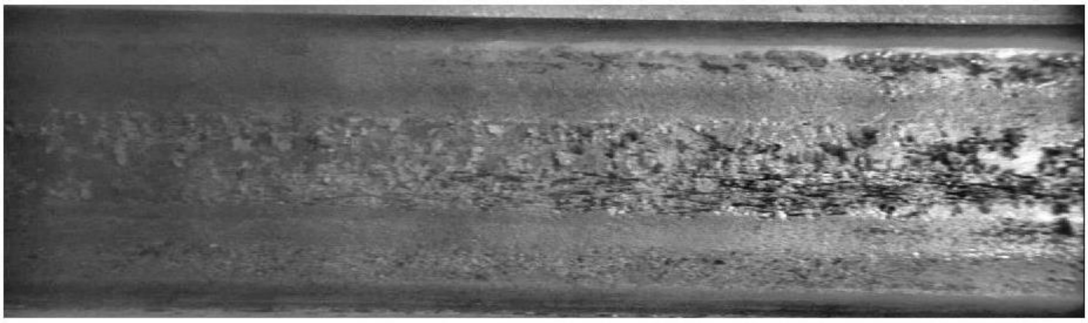


Figure 5.18: Small tiger stripe pattern histogram adjusted

5.4.1 Light Adjustment

The use of non-diffuse light sources (spotlights) and inconstant rail positioning has resulted in non-uniform lighting conditions for the images, which results in changing brightness within various parts of the same image and also between successive images. A poor lighting condition on the images with tiger stripes makes the defect detection almost impossible. Image shown in Figure 5.19 is an example of unprocessed image with TS defect. It can be seen that not only whole image is dark but left part is darker to an extent that the defect seems not to be present on that side.

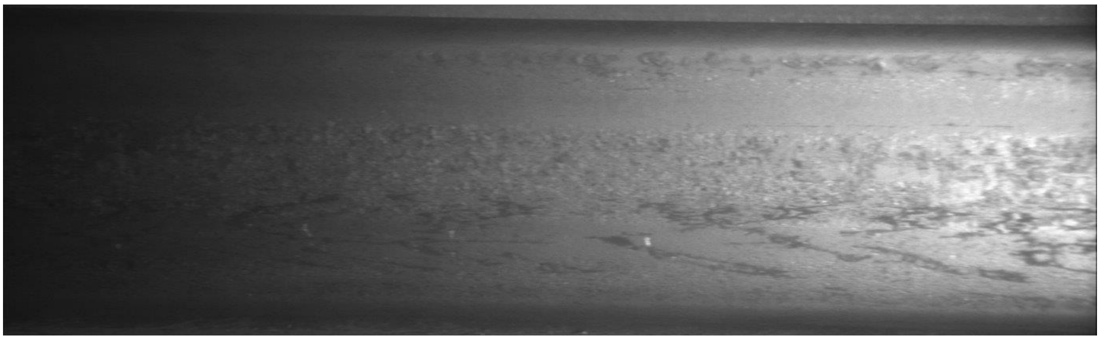


Figure 5.19: Original tiger stripe image

An average filter has been used to adjust the lighting of the images. A range of standard histogram equalisation algorithms were previously unsuccessfully applied on both local and global, necessitating the generation of an new, effective average filter.

The Image is equalised using a method of finding the background lighting of the image, which has been used to correct the original image. An average filter with a length equal to the height of the image and a width of one pixel is applied to the original image. The result is an image representing the lighting, which has been shown in Figure 5.20.

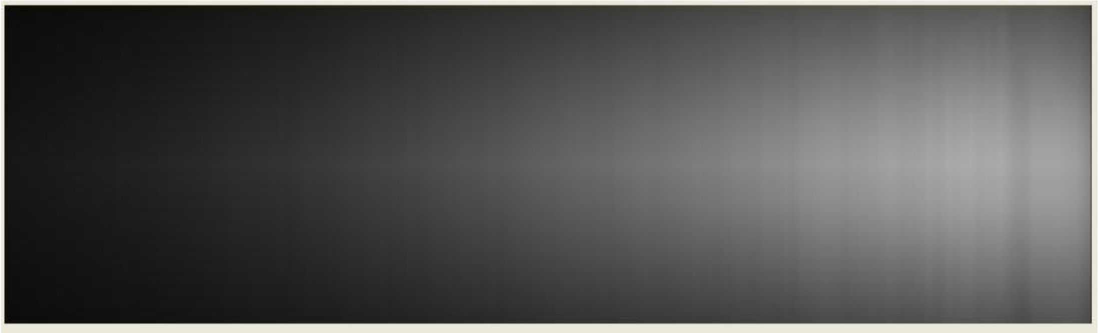


Figure 5.20: Extracted light field of the image in Figure 5.19.

The light field image is then inverted and added to the captured frame. Image smoothing uses an average filter and has been found very effective as can be seen in Figure 5.21. The equalization/averaging process and filters have been explained before in the section 5.4.1.

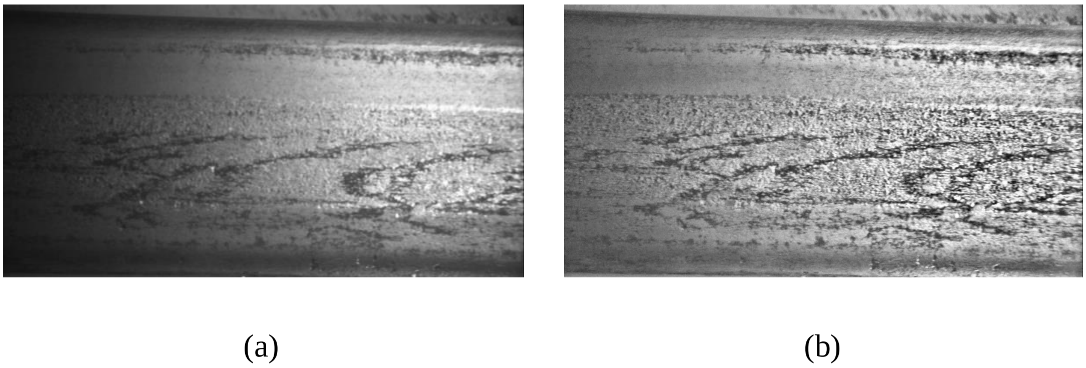


Figure 5.21: Result of Tiger stripes image equalization (a) Original image with tiger stripes (b) image after equalization.

5.4.2 TS Detection Methodology

Detection of TS artefacts involves the normalization of the image intensity followed by detection of the dark zones.

The tiger stripe detection is performed in three steps:

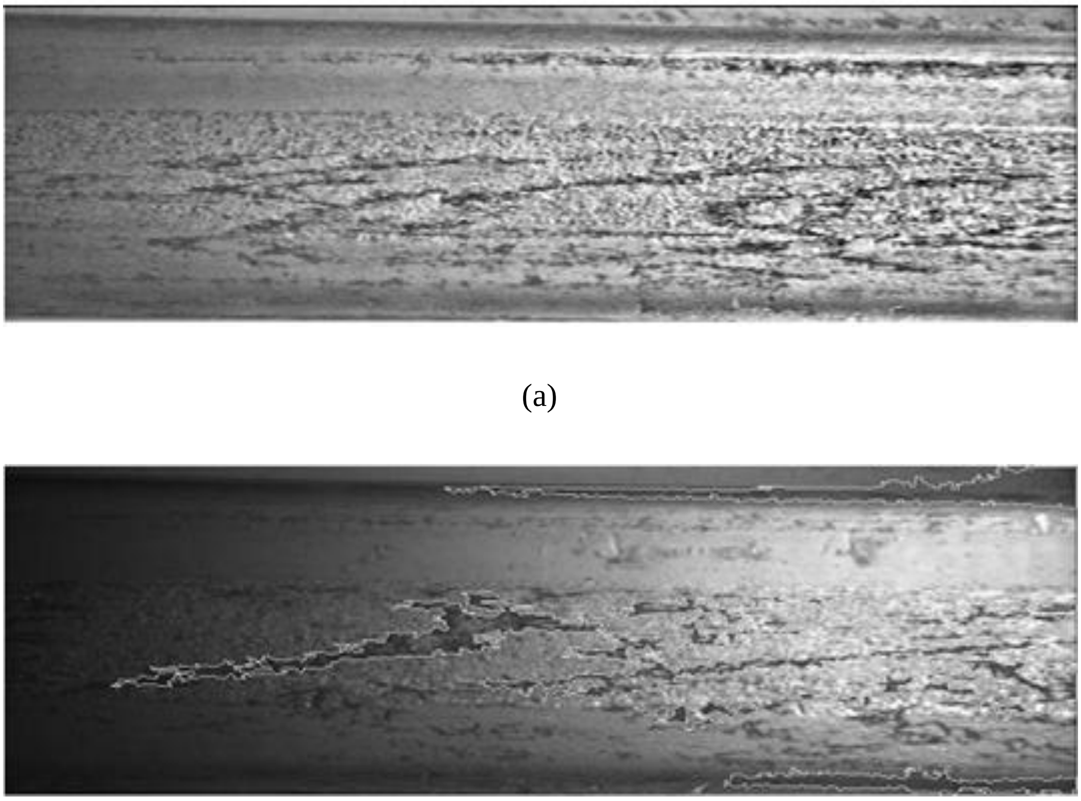
1. Image equalization: Image is equalised using a combination of an average filter and image subtraction technique explained in the last section to make the image lighting uniform.

2. Thresholding: A simple threshold is used to extract the dark areas or in other words all the edges in an image from the equalised image of the last step.

3. Classification \ Area size selection: A parameter has been empirically defined for the size of dark zone. This parameter is used for the classification of defects. Dark zones of some distinct size and their presence on successive images; is characteristic of this defect that helps in its detection.

5.4.3 Parameters Required for TS Detection

There is only one main parameter that can be controlled externally by the user, which is the size of the dark zone. The number of pixels covered by the detected dark zone is compared with the size variable provided and if the size is found equal or bigger than that, it is classified as a defect. The selected dark zone can be seen in Figure 5.22.



(a)

(b)

Figure 5.22: Example image for Tiger Stripe type defect (TS) detection (a) TS original image (b) TS dark zones selected

5.4.4 Algorithm Implementation for TS

The detection algorithm for Tiger Stripes can be summed up in the following steps:

- Input image is again a grey scale image. The first step is image equalization to adjust the lighting on the image, which is applied using specially developed algorithm. The method works in the following steps:

- A convolution border is added to the original image. Image is padded on its top and bottom sides only, with 0's. The zero borders have size equal to that of an image. In other words it can be said that a black copy of the image is padded on top and bottom of the image, to serve as convolution border. Hence, the resulting image has same width as that of orginal image but three times the height.

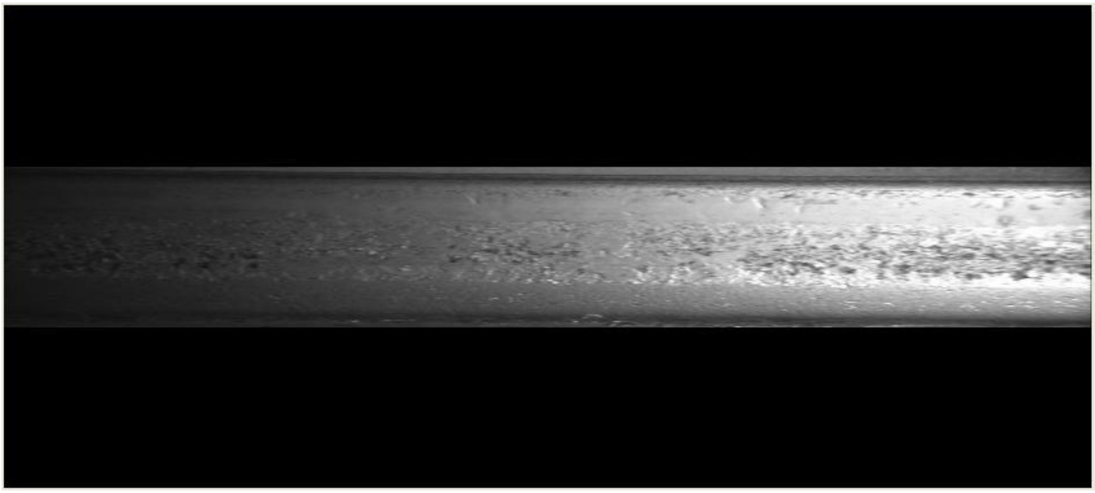


Figure 5.23: Convolution border added to the original image

- A filter with a length equal to the height of the image and a width of one pixel, having all 1's in it, is convolved with the bordered image obtained after padding, in the last step. Convolved image is subsequently scaled by a factor of 1 / (height*width) of the filter used. Convolution with an average filter results in an image representing the lighting. Convolution operation shown in Eq. (5.5) has been performed to extract the above light field.

$$(\mathrm{f} * \mathrm{k})(x, y) \quad = \frac{1}{h * w} \sum_{i=x-w, j=y-h}^{x+w, y+h} \mathrm{f}(\mathrm{i}, \mathrm{j}) \mathrm{K}(\mathrm{x}-\mathrm{i}, \mathrm{y}-\mathrm{j}) \qquad \textit{Eq. (5.5)}$$

f = Convolution filter used

h = Height of the filter f

w = Width of the filter f

k = Image to be convolved with the convolution border padded.

x = width of the image k

y = Height of the image k

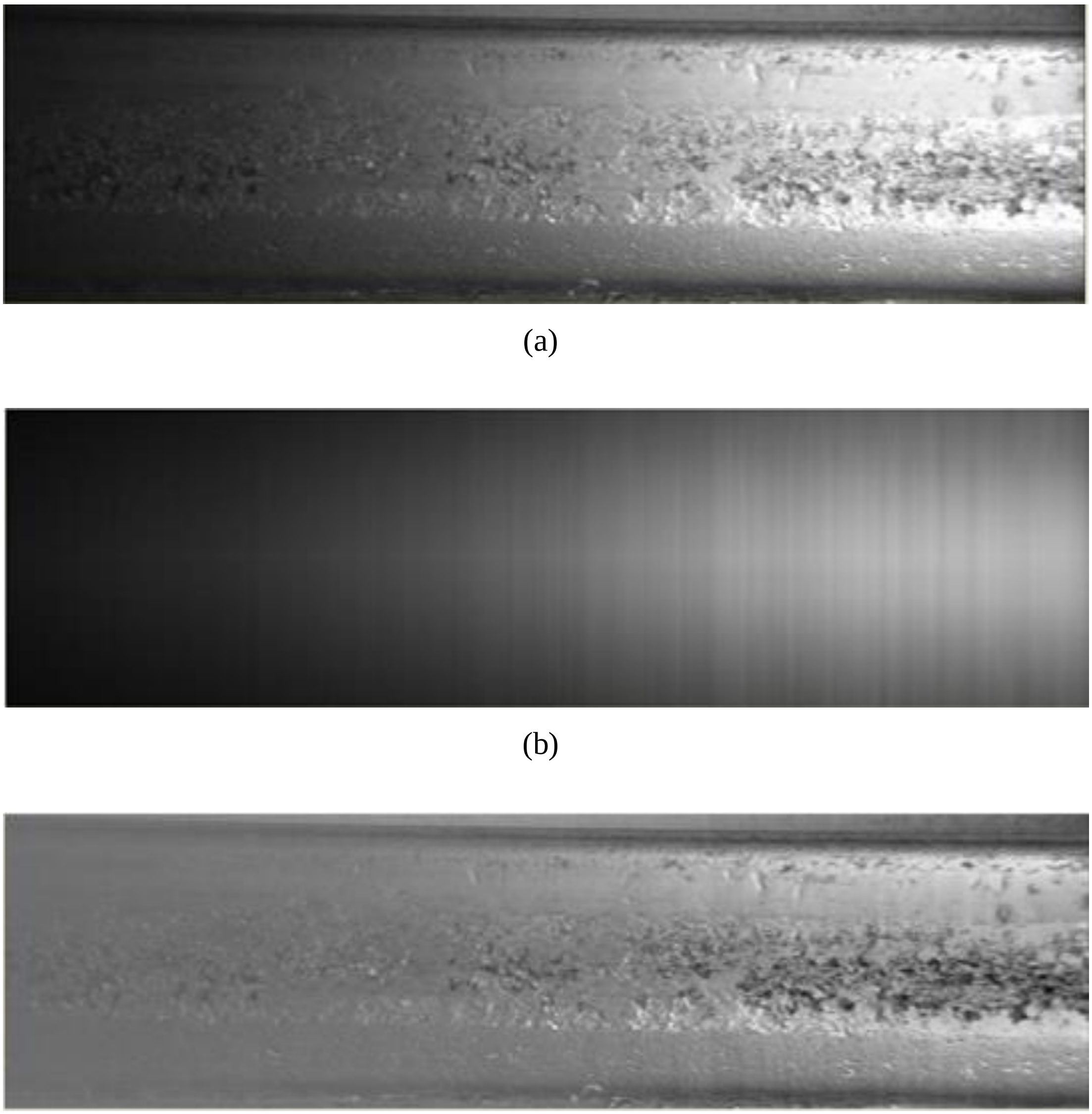
- The resulting image representing the lighting is then inverted. Image inversion is done using a typical grey scale reversal process. For example, if there is a grey scale image I with values ranging from 0 to 255 then inverted Image for I is given by

$$Iinv(x,y) = 255 - I(x,y) \qquad \textit{Eq. (5.6)}$$

  I_{inv} = Inverted Image of original image I(x,y)

  I(x,y) = Original Image

- The inverted image is finally subtracted from the captured frame to make the image look uniformly bright everywhere. Original image, the light field obtained after using the average filter and finally the adjusted image obtained after subtracting the inverted light field from the original image has been shown below in Figure 5.24.



(a)

(b)

(c)

Figure 5.24: Light adjustment result for tiger stripes (TS) (a) Original image with TS (b) Light field extracted for TS image (c) Light adjusted for TS image

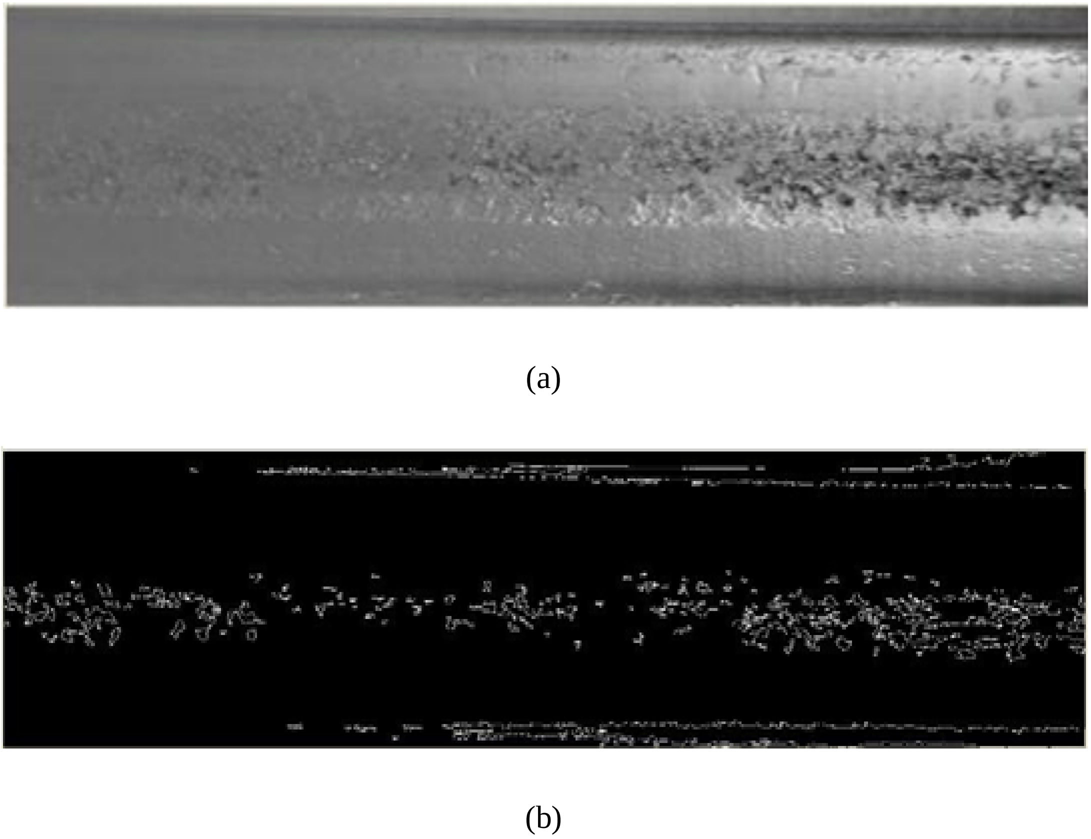
2. The resulting image is then binarized using threshold to binary conversion method. The inverse thresholding method has been used that works in inverse way of the normal thresholding method explained in the last section. In the inverse method if the pixel value is greater than threshold; it is assigned as 0 rather than 1 as done in typical thresholding method. Image obtained after thresholding has been shown in Figure 5.25.

Figure 5.25: TS image after thresholding

3. Next step is to find dark zones on the image. There is a small method defined to help in finding the dark zones. The method accepts 3 parameters. These parameters are for Image, threshold and the thickness. The function works in three steps:
    - First it makes the blank copy of the image.
    - The function then finds all the contours in the original image that have area greater than the threshold value provided.
    - The next step is to draw all the contours on the blank image created with thickness provided in the arguments.
4. The final step is classification based on the size of the dark zones detected. It is explained in the following section.

5.4.5 Classification Method Used for Tiger Stripes

A special classification technique has been developed. Size variable is used to perform the classification. The variable is defined by the user and the dark zones detected, as a result of pre-processing and thresholding stages, compared with the variable. If the size of the dark zone is found bigger than the defined variable, they are marked as defects. The final classification is based on the presence of similar size dark zones on successive images. This is the key and the useful point for the defect detection. The result of tiger stripe detection is shown in Figure 5.26.



(a)

(b)

Figure 5.26: Result of Tiger stripe detection (a) Light adjusted original image with TS (b) TS defect marked / detected

5.4.6 Detection Rate

The tiger stripe detection rate, using the explained algorithm, is 80% only, when tested on almost 1000 images with tiger stripes. Varying size of the defect and some cases where the stripes are faint; the algorithm fails and is still critical to the quality of the rail itself. However, there is not false positive detection, which means all marked rails have tiger stripes on them are detected. Several other techniques have been tested to improve the detection rate, which have been explained in the later sections.

5.5 Wire Defects

A “wire defect” is a line type deep scratch, found at central area of the base of the foot. The variation in this type of defect made them extremely hard to detect from computer vision aspect, while maintaining an acceptable level of false positives. They have various shapes and sizes and can appear as straight lines, curves or even random arbitrary lines of different sizes.

However, it has been observed that the wire defects have occurrences of dark / light lines next to each other. This unique property of the defect has been helpful for its detection, which is done by measuring deviation of intensity of dark and light lines within a window. Many other techniques were also tried on wire defects less successfully, which have been explained with their results later in this chapter (section 5.6). An original rail image with a wire defect can be seen in Figure 5.27.

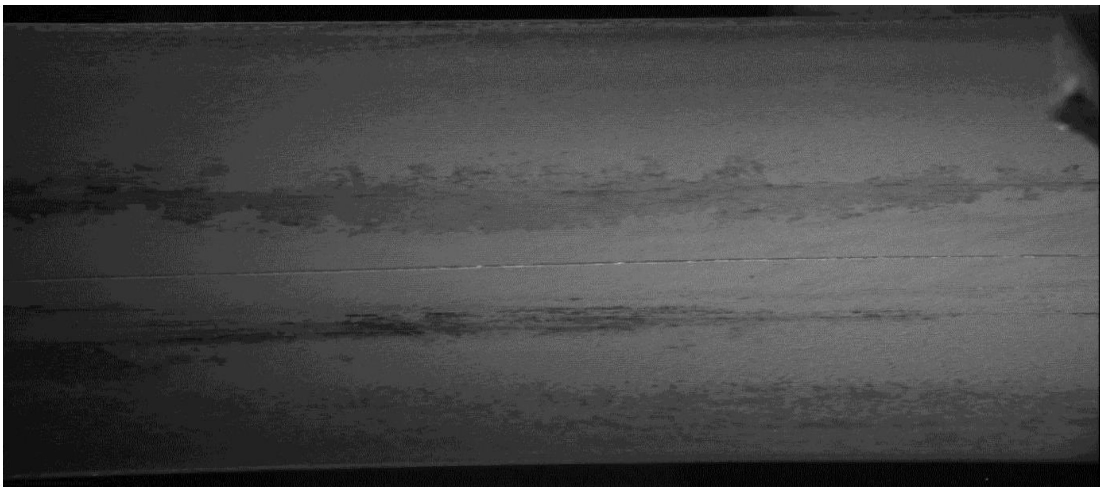


Figure 5.27: Original rail image with wire defect.

5.5.1 Wire Defect Detection Methodology

The defection of the wire defect is based on a characteristic of the defect's appearance with the original lighting. This type of defect has some special characteristic that have been used to detect it. An empirical observation is that it always appears as a dark line contiguous to a light line on the entire set of provided images and on all type of shapes. The algorithm for detecting this type of defect takes advantage of the continuity of dark and light lines. An XY plot of a line drawn vertically through the fault, showing the adjacent high and low intensities is shown below in Figure 5.28. It is very obvious that how intensity varies at the cross section of a wire defect, showing the presence of dark and light lines next to each other.

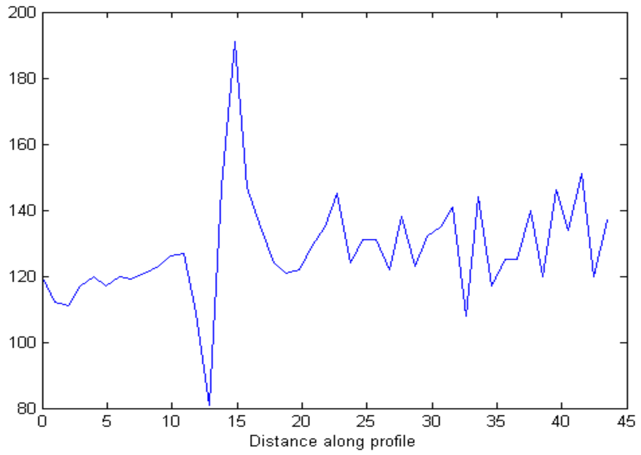


Figure 5.28: XY plot of a line drawn vertically through the wire defect

A sliding window was used to detect the local variation and deviation of a pixel's grey values, a defect is considered to be present using two criteria, the deviations of grey level should be significant (i.e. above a manually determined threshold) compared to the average pixel value in the window, and the light and dark line should be contiguous. The result of the detection can be seen in **Error! Reference source not found.** for line type of wire defect. A post-processing filtering that involves morphological operations, is necessary to remove all detected blobs and marks which are too small to be a defect, i.e. with a 'defect area' too small or a 'defect length' too short. Consultations with experts enabled the setting of the threshold, necessary for this algorithm to work.

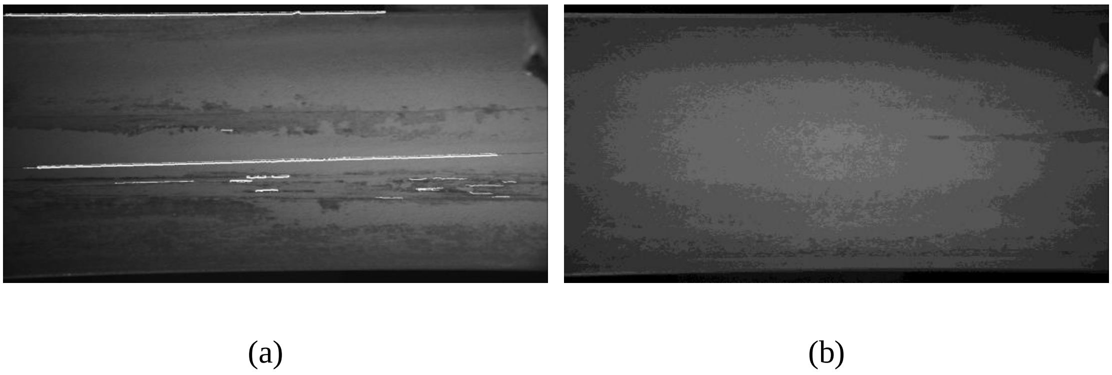


Figure 5.29: (a) A successful detection of a line (b) A successful non detection of an artifact

5.5.2 Parameters Required for Wire Defect Detection

The defined detection algorithm for wire defect uses three main parameters:

1. **Window Size:** Size of the average window, which is to be convolved with the image to perform average filtering, is required and can be manually determined by the user. This average window can have variable number of rows and columns.

2. **Low Threshold:** A low threshold value to determine dark lines during detection is needed.

3. **High Threshold:** A high threshold value to find light \ bright lines in an image is required.

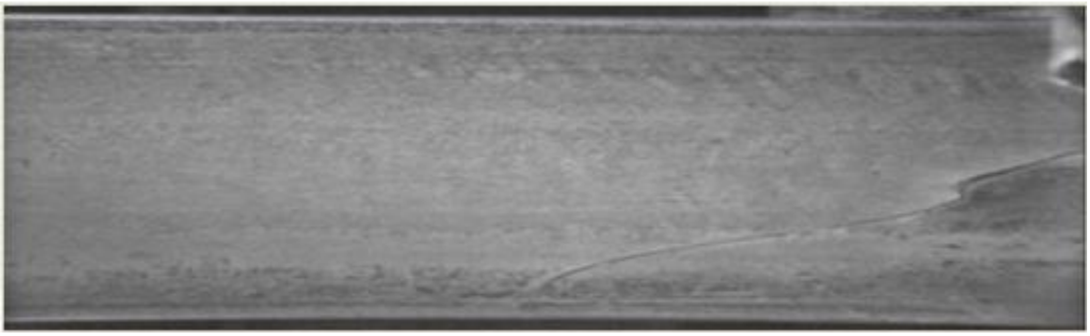
4. **Detected Zone Size:** Minimum size required by a detected dark and bright line \ zone is used. The zones having some certain size which is greater than this parameter value are assigned as defects, remainder are all ignored.

5.5.3 Algorithm Implementation Steps for the Wire Defect

The detection algorithm of wire defect detection has following steps:

1. **Extract Edges by Un-sharp Masking:** An average filter of size equal to the window size variable; with all its values as 1's was created. The filter was convolved with the original image to produce a smoothed image. Average filtering was used to remove noise by removing pixel values from an image which are different from their surroundings. The result of smoothing was shown in Figure 5.30.

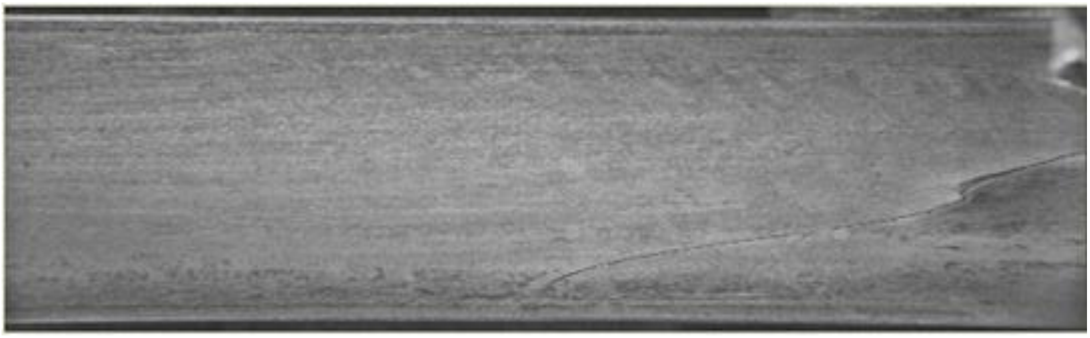
(a)



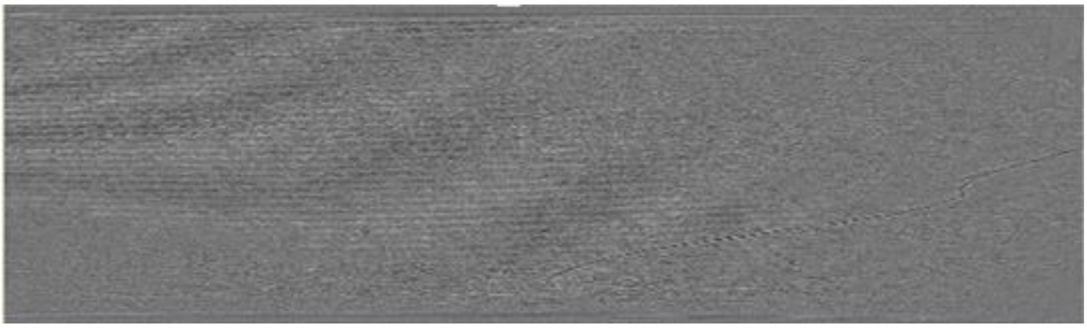
(b)

Figure 5.30: Result of image smoothing step for wire defect detection (a) Original image with wire defect (b) wire defect image after smoothing

The filtered smooth image is then subtracted from the original image to obtain the edge image. The resulting edge image can be seen in Figure 5.31.



(a)



(b)

Figure 5.31: Result of edge extraction using un-sharp masking for wire defect (a) Original image with wire defect (b) Edges extracted from original image by un-sharp masking technique.

The process is called un-sharp masking which is a sharpening operator used to enhance the edges and other high frequency components in an image. It has been used to extract all the edges from an original image. The process of un-sharp masking has been shown in Equation Eq. (5.7).

$$I_E(x,y) = I_i(x,y) - I_{smooth}(x,y) \qquad \text{Eq. (5.7)}$$

Where,

I_E = Edge image of the input image

I_i = Input image

I_{smooth} = Smoothed version of the input image I_i

Deviation of Pixel's Grey Values: Average value of the smoothed image is calculated and is used to find the deviation or variation of pixel's grey value. There is some manually determined threshold value, which is used as cut off band. All the pixel values in the edge image are compared to the average value. The edge values that lie in the cut off band are all dropped, being their insignificant deviation from the average value. This step results in an edge image with only significant pixel value deviation from the average value. The image obtained is shown in Figure 5.32.

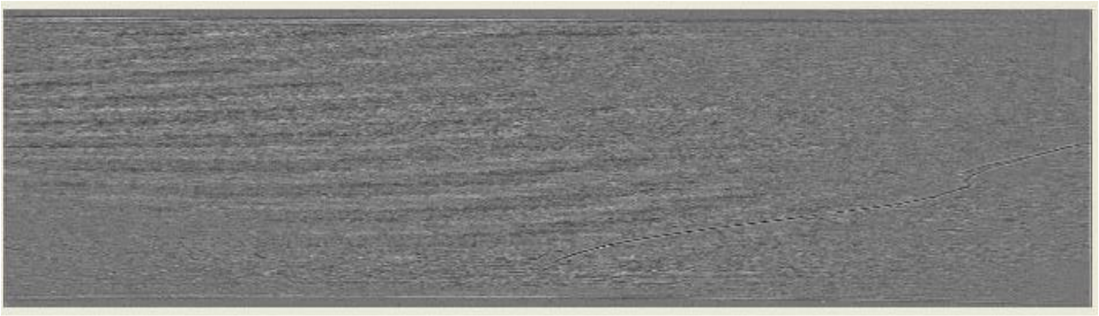


Figure 5.32: Image of significant deviation edges

The above explained process has been expressed in.

$$I(x,y) = I_E(x,y) > (Avg + \varepsilon) + I_E(x,y) < (Avg - \varepsilon) \quad \textit{Eq. (5.8)}$$

Where,

I = Output image

Avg = Average value of I_{smooth}

ε = Threshold limit for deviation

I_E = Edge image of the input image

2. **Detection of Bright & Dark Lines \ Edges:** The image resulting from the last step is then scanned for all the dark and bright edges present in it. All the edges with pixel values greater than '0' are assigned to positive image and all the edges with values less than '0' are assigned to negative image. The result is two images, one with all the bright \ light edges and one with the dark edge lines separated from input image, that came from step 2. The resulting images are shown in Figure 5.33.

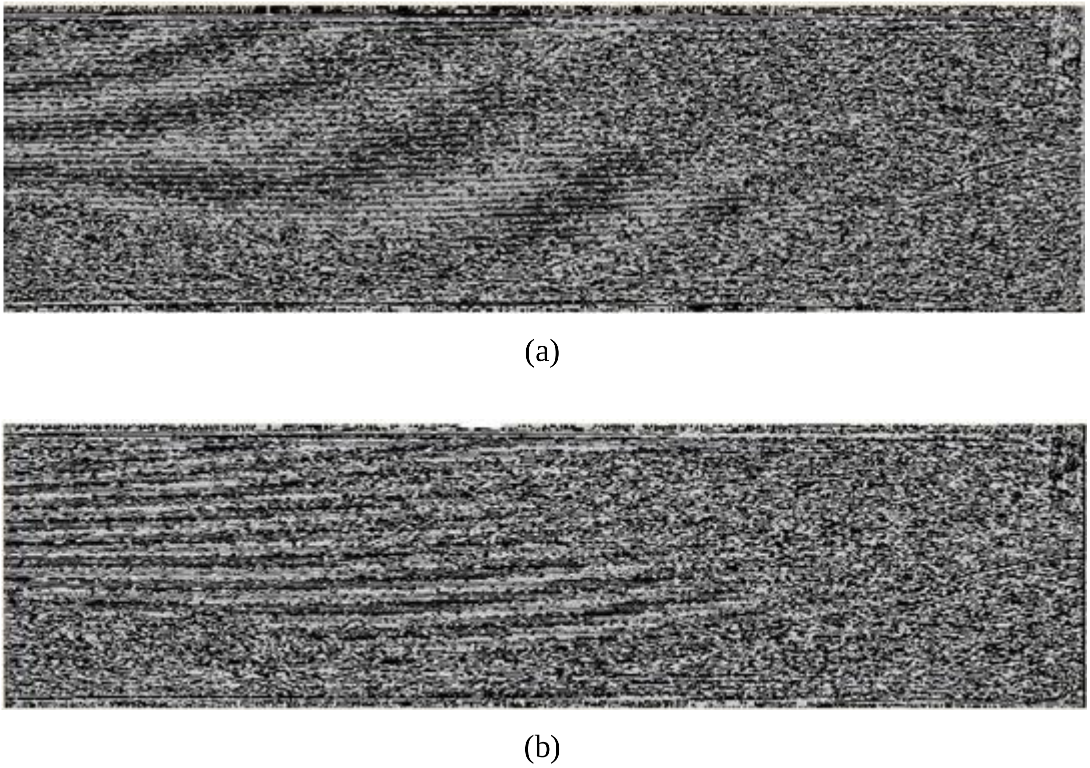


Figure 5.33: Detection result for bright and dark lines in a wire defect image (a) Image showing positive edges of an image with wire defect (b) image showing negative edges of an image with wire defect.

The process can be expressed by the following equations.

$$I_{pos}(x,y) = I_{in}(x,y)>0 \qquad \textit{Eq. (5.9)}$$

$$I_{neg}\,(x,y)= I_{in(}x,y)<0 \qquad \textit{Eq. (5.10)}$$

Where,

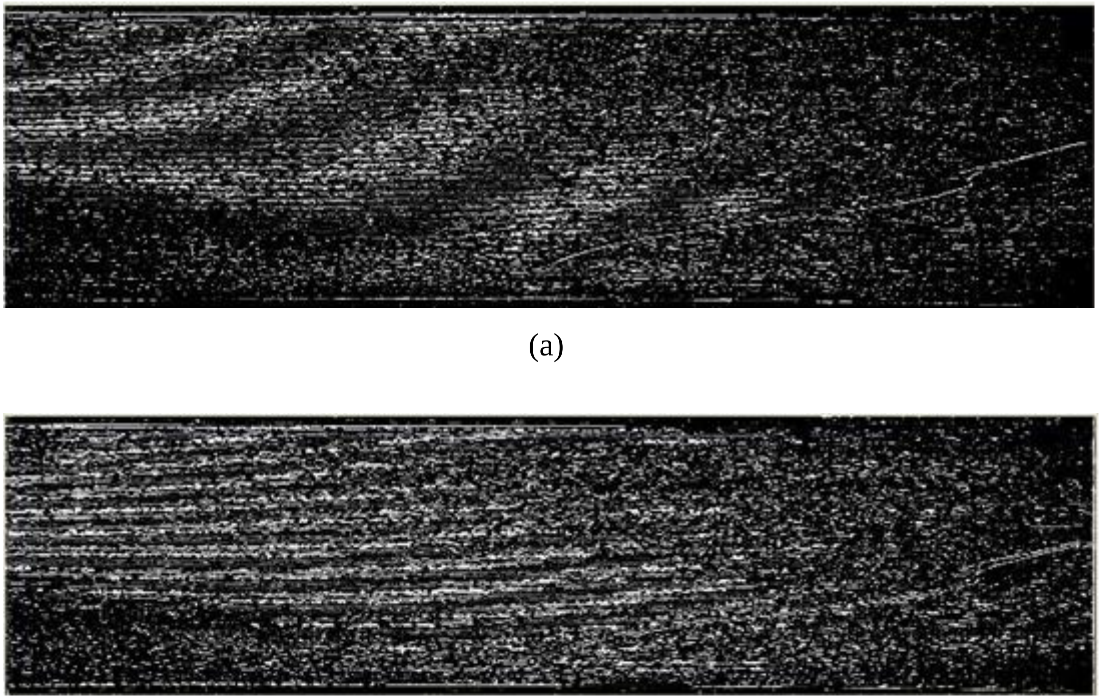
I_{in} = output of step 2 which is I

I_{pos} = All the edges with pixel values greater than ‘0’

I_{neg} = All the edges with pixel values less than ‘0’

3. **Threshold to Binary:** Resulting images from the last step are then binarized using the thresholding to binary method. I_{pos} is binarized using high threshold value provided and I_{neg} is binarized using low threshold value provided. Two threshold values, one for low and one is for high, are used keeping in view that the defect occurs in the form of light and dark lines next to each other.

Hence, image is binarized using the two threshold values. The resulting two images can be seen in Figure 5.34.



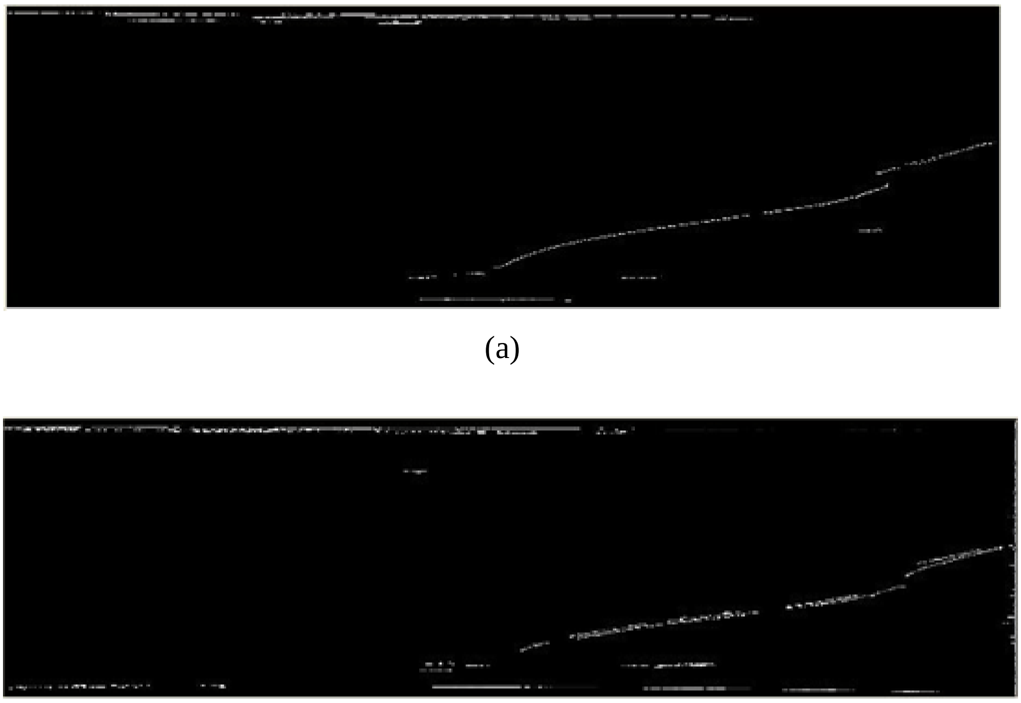
(a)

(b)

Figure 5.34: Threshold result showing bright and dark edges for wire defect image (a) Image showing only dark edges/lines in the image (b) Image showing only bright edges detected in the image

4. **Morphological Operations:**

   A morphological operation is then applied to both images in order to get better view of their binary image forms by clearing unnecessary small edges and objects from the images. A function has been defined for this purpose, which makes a blank copy of the input image. It then finds all the edges that have area greater than the value of size zone provided in the input image and redraws those areas on the blank copy. Hence, clears small and unnecessary blobs and objects from the input image. The clean images produced containing only areas of interests that can be seen in Figure 5.35.



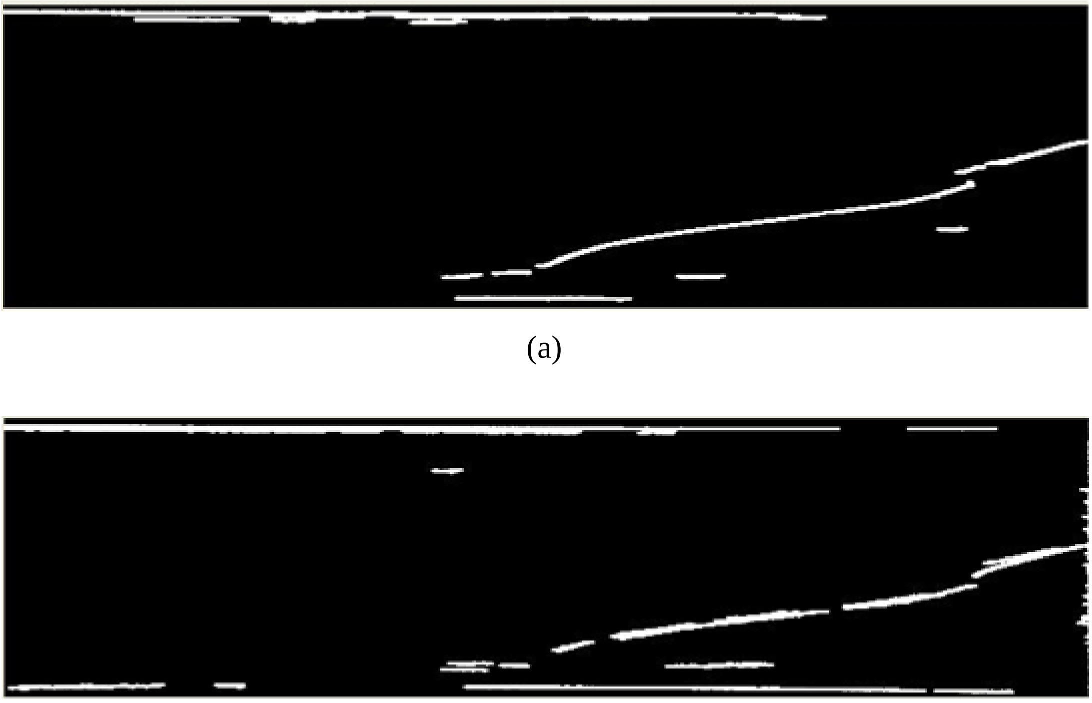
(a)

(b)

Figure 5.35: Result of wire defect negative and positive edge images after morphological operations applied. (a) Clean positive edge image (b) Clean negative edge image

Another morphological operation of Dilation is applied to the two images which causes bright regions (edges in this case) to grow. The elliptical shaped structuring element of empirically defined size has been used to perform the dilation operation on both images. Hence, both the negative and positive edges found as bright regions are dilated to connect any broken edges \ areas.

This step results in binary images of I_{pos} and I_{neg} containing only significant size areas of interests, further dilated to make them more significant. The final images can be seen in Figure 5.36.



(a)

(b)

Figure 5.36: Result of edge dilation step for wire defect detection (a) Clean positive edge image after dilation (b) Clean negative edge image after dilation

5. The resulting two images from step 5 are fianlly multiplied with each other to find if the positive \ bright edges are located right next to negative \ dark edges. The multiplication process outputs ‘0’ if either of the inputs is ‘0’. Hence, the resulting image will have only the edges that are present in both I_{pos} and I_{neg} at the same locations. The result of multiplication between the images from step 5 can be seen in Figure 5.37.

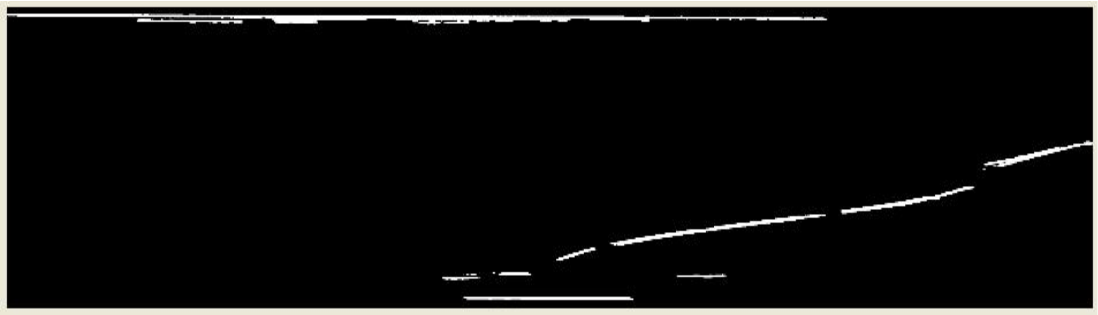


Figure 5.37: Multiplication result of negative and positive edge images

In other words, the output image will have only contiguous light and dark lines on it. The process has been expressed in the following equation form.

$$I_{out} = I_{pos} \times I_{neg} \qquad \textit{Eq. (5.11)}$$

Where,

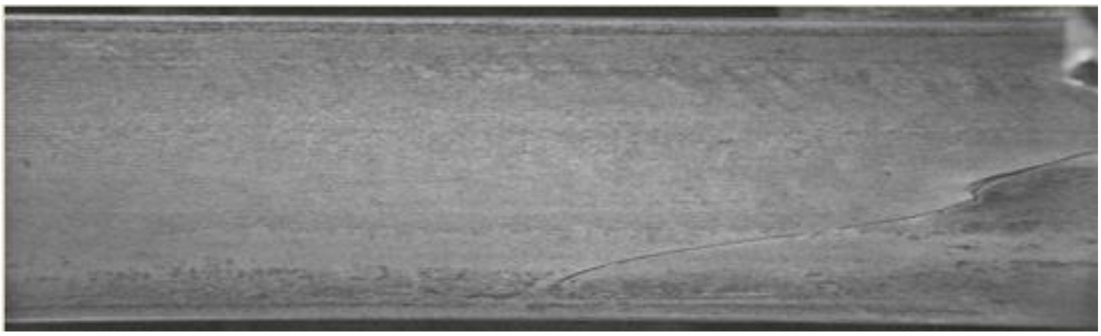
I_{out} = final output image

I_{pos} = morphologically processed image with all positive pixel values \ bright lines

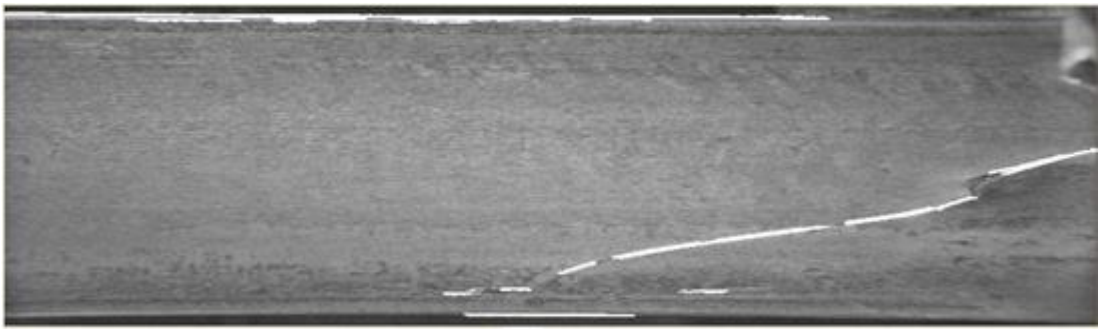
I_{neg} = morphologically processed image with all negative pixel values \ dark lines

5.5.4 Classification of Wire Defect

The classification of the defects depends on the length of the wire defected and hence, uses a size variable for comparison. The function explained in step 5 of the algorithm's implementation draws only the contours/edge lines that have size greater than the detected zone size parameter provided; all other lines or edges are dropped. Hence, the two images (negative and positive edge images) are formed containing only the contours of specific lengths. Finally, both the images are multiplied together to get the image with only those lines which are present in both the images. Hence, if there is anything present on the final product, it will be classified as a defect. The final detection result can be seen in Figure 5.38 and Figure 5.39.



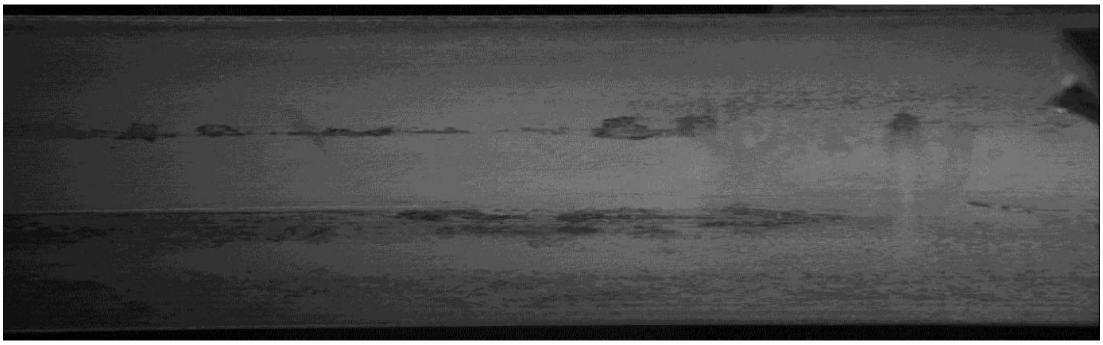
(a)



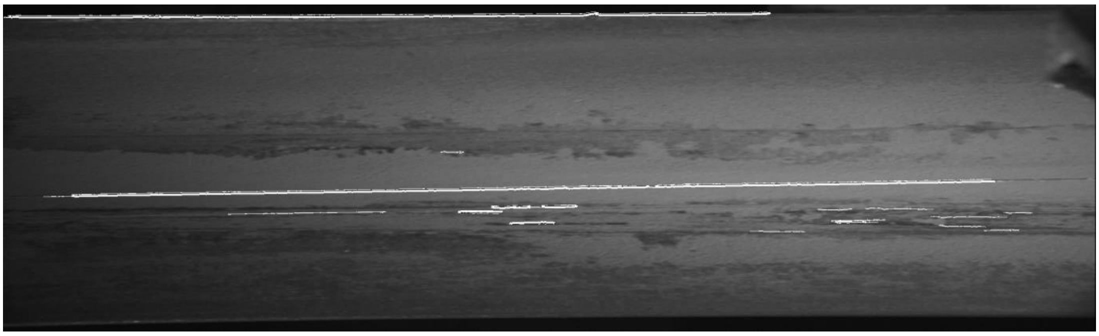
(b)

Figure 5.38: Detection result for wire defect (a) Original image (b) Wire defect marked on the original image

Another example of wire defect detection can be seen Figure 5.39.



(a)



(b)

Figure 5.39: Detection result of wire defect example 2(a) Original image (b) Wire defect marked on the original image

5.5.5 Detection Rate

The algorithm for wire defect has a perfect detection rate of 100% of all the wires.

The success rate has been computed after performing successful detection on the data set of rail images having wire type defect. The available data set has over 1000 images of wire defect. The algorithm does have 5% of false positives due to occasional water drop that are incorrectly labelled as being faults. However, the false detections can always be assessed by a post-viewing as being safe.

5.6 Alternate Techniques Tested for the Wire Detection

This fault was the most difficult defect to detect, with a comprehensive range of methods exhaustively tested and applied to the wire defect images. Many algorithms and image analysis techniques such as Fourier transform and Susan Filtering, Straight line algorithms, neural networks, Principle Component Analysis were tested, with generally negative results. All these algorithms detected non-defect marks and noise also. There is a very fine line between the defect and other lines and marks present on the track that are not defects. So there had to be a technique to detect only the defects. The other techniques applied and tested on the wire defects and the results produced by them have been provided in this section.

5.6.1 SUSAN Function

The details of SUSAN filtering and how it works; have been explained in the previous chapter in detail in edge detection section. Matlab code has been written manually to get the results. All the parameters and the kernel used are user defined. Various masks with their empirically determined sizes and angles are used, which are slided over the whole image using ‘nested for loops’ to perform the SUSAN operation.

The best combination of optimised parameters produced the results shown in the images in Figure 5.40. It can be seen in the images that they are detecting the wire defect lines, but also picking up extra noise and dust of the back ground, which made it difficult to differentiate defect from non-defect.

SUSAN filtering combined with other techniques was also tested to get some improved detection, which has been discussed in the next section.

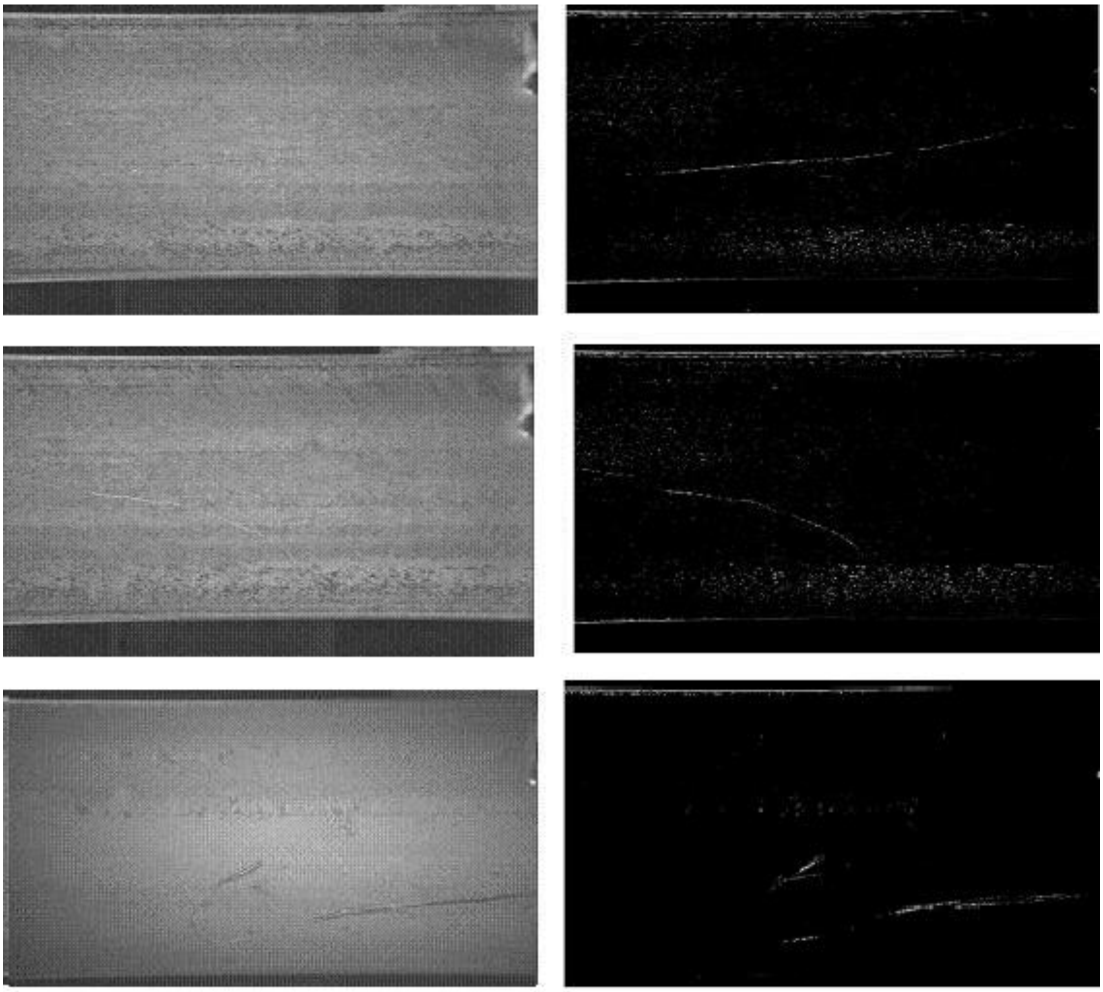


Figure 5.40: Results of wire defect detection using SUSAN function

5.6.2 Fourier Transform and SUSAN Filtering

As seen in the Figure 5.40, noise picked up by the SUSAN filter was a problem. Hence, Fourier transform was tested and used along with the same optimised SUSAN filtering as discussed before, to get image with reduced noise.

Fourier transforms and the steps to remove noise from an image using it have been explained in detail in the previous chapter 4 in the noise removal section (section 4.4). Fourier transform of the image is taken and circular area containing the noise is removed. The SUSAN filtering is applied after taking the inverse Fourier transform of the resulting image. It can be clearly seen from the results shown in Figure 5.41 that still a significant number of non-defect areas are being detected. Hence, it cannot be classified as a good detection method.

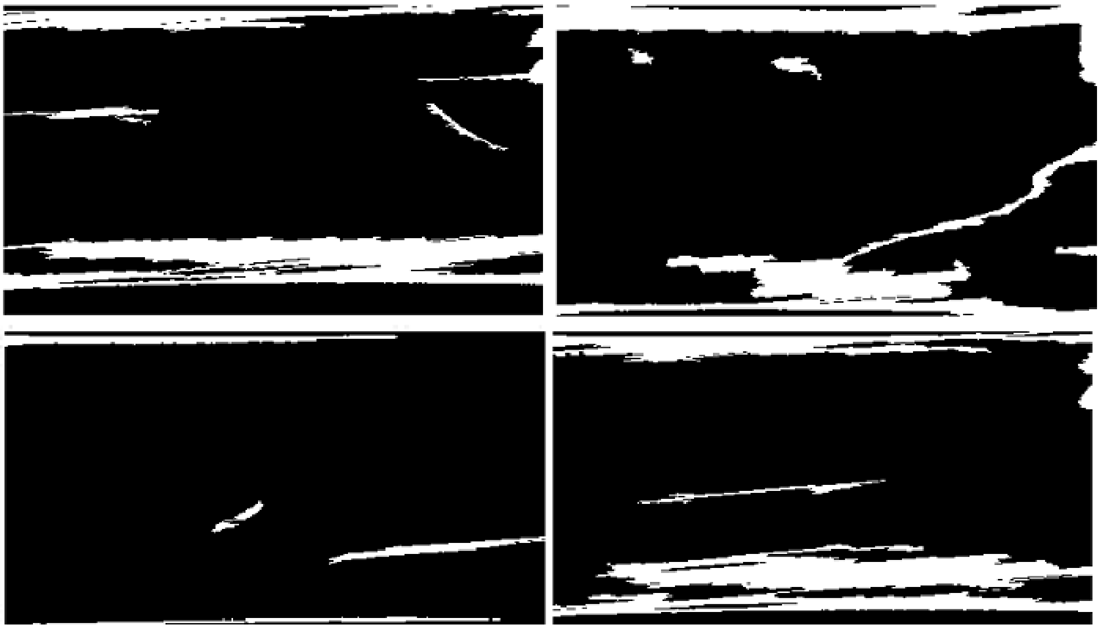


Figure 5.41: Results of wire defect detection using Fourier transform with SUSAN filter

The above results have been obtained using Matlab. “fftshift” and “ifft2” functions have been used to take the Fourier and its inverse after removing certain frequencies. SUSAN filter has again been applied using the manually written code in Matlab.

5.6.3 Neural Network Wire Defect Detection

A back propagation neural network has been used to process the wire defect images shown in Figure 5.42. A small size of mask is chosen and is moved over the whole image; to calculate statistical measures such as standard deviation, mean and variance of the area under the mask etc. Statistics calculated from the defective area were used to train the network. After training, the trained network is simulated by the statistic values of any image to detect any similar defect. The details of implementation steps have been provided at the end of the report in appendix A. The results from neural network based wire defect detection are shown in Figure 5.43.

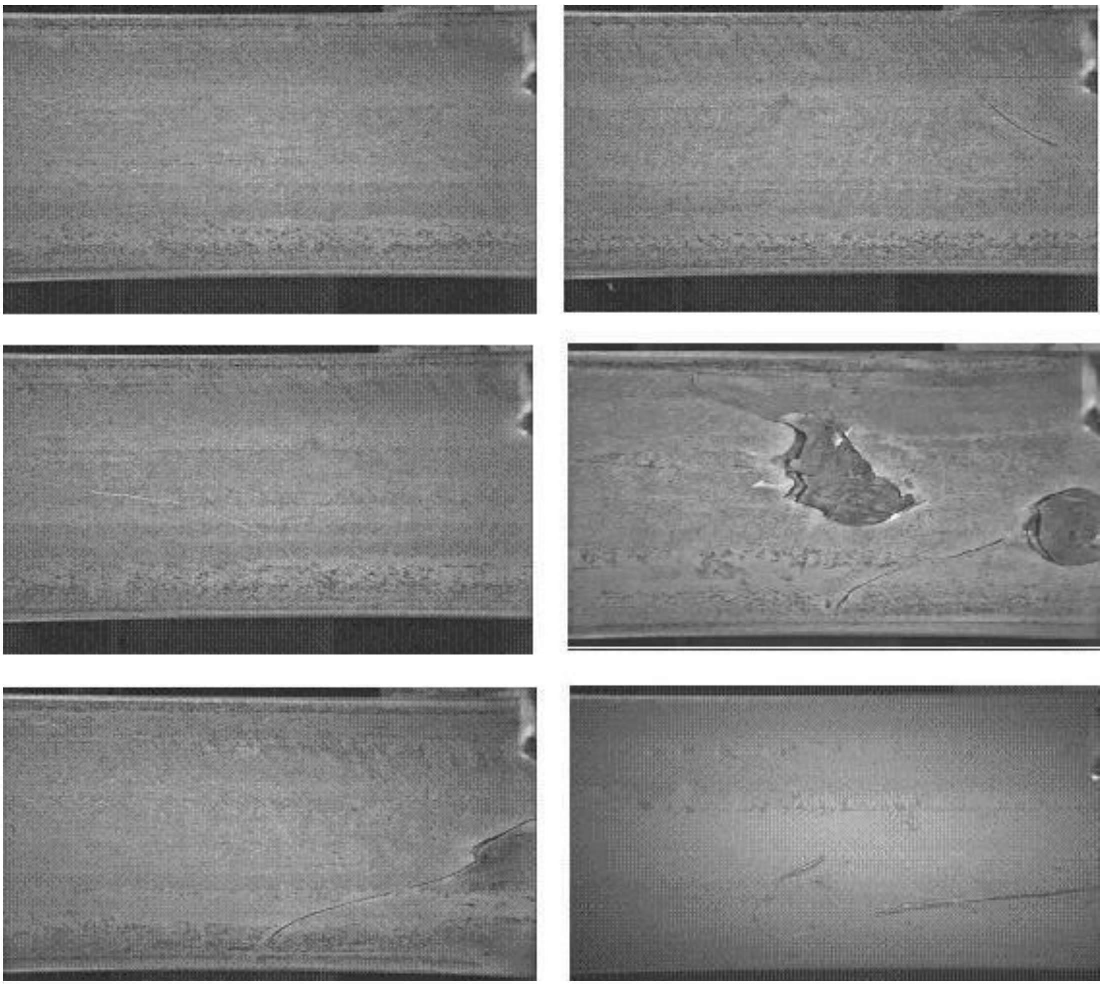


Figure 5.42: Original images showing wire defects

The results obtained are better than other techniques previously tested but still the neural network is struggling to identify the defects form the non-defect marks \ edges on the rail, which can clearly be seen in the images shown. Neural networks are still being tested in different training sets and in combination with other detection methods to get better results. Matlab existing function called “newff” has been used for the testing purposes. The function creates feed forward back propagation neural networks.

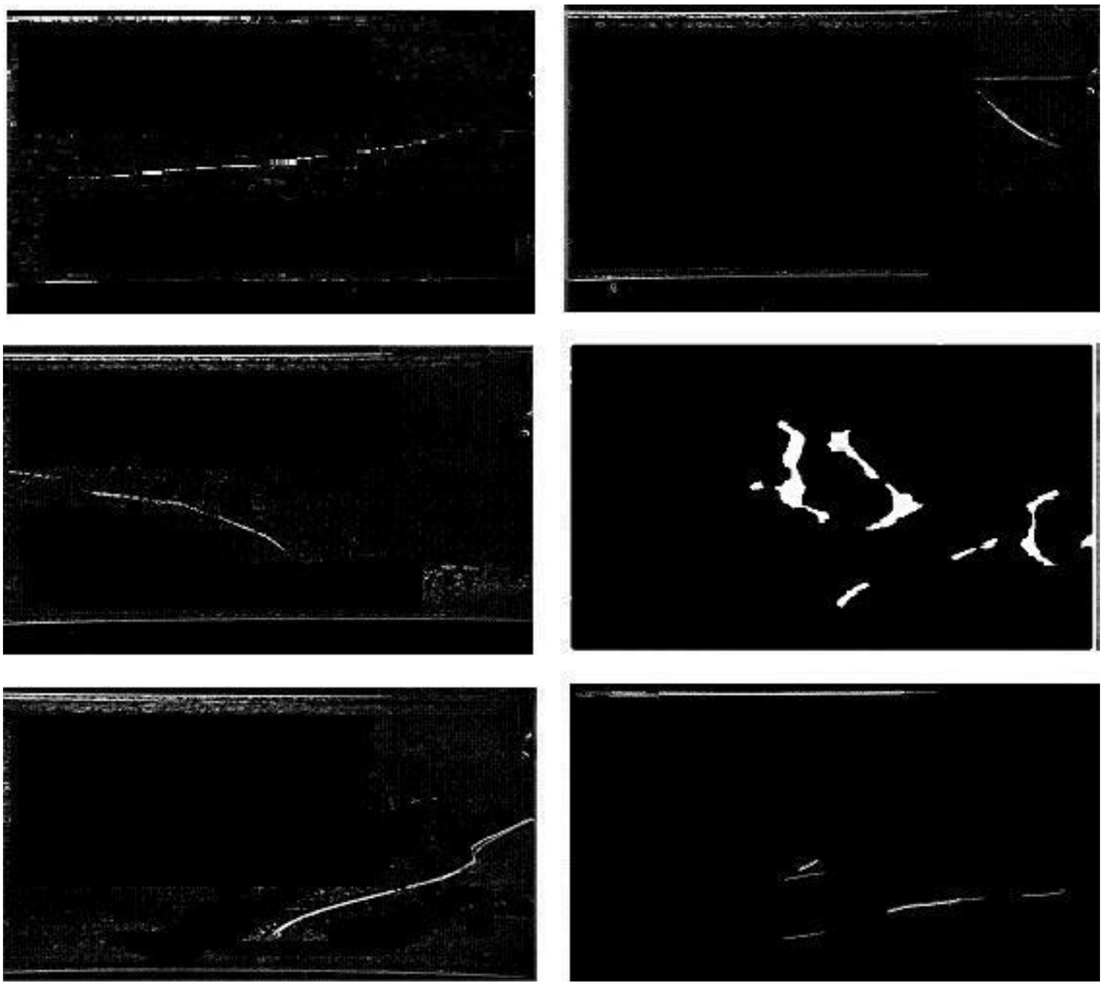


Figure 5.43: Results obtained for wire defects using neural networks

Back propagation networks are very efficient at prediction and classification (segmentation) but they have few practical consideration. They are comparatively slow only to train, however once trained they are a lot quicker. There is no explanation about the learning of the network available i.e. a learned network does not contain any information about its learning. Anything can be fed on network and it becomes a problem, even the irrelevant data. For example, noise on the images cause a problem. Sometimes overtraining of the network usually occurs that results in error. Considering industrial application data might change over change, and the network working today might not work tomorrow. Hence, these things have to be evaluated before using neural networks in industry. Although the results are good and comparable to the results produced by previous method explained in section 5.5, however, the neural network implementation needs some further testing before this method can be implemented in the field which needs more time.

5.7 Summary

The list of adapted existing imaging procedures, and also lists the special algorithms that have been written for the detection have been provided in Table *5.1*. For each defect specific logic and detection steps were defined and have been implemented using the combination of existing imaging methods, their adapted forms and specially written algorithms.

Table 5.1: List of Developed & Adapted Algorithms

Algorithms	**Advantages \ Comments**
Logic Detection Steps	• For each defect specific logic and detection steps have been defined. The logic has been implemented using the combination of existing imaging methods, their adapted forms and specially written algorithms.
Image Equalization	• Image Equalization has been applied using Average Filter for Tiger Stripes to adjust the lighting. • The code has been written manually to perform equalization
Straight Line Algorithm	• Helps find the defects that appear as straight lines by detecting all the lines in the image that have horizontal spread. • Code has been written manually to find the lines with horizontal spread for line on the top of rail type defect.
Adapted Morphological Operation for Tiger strips	• A code has been written manually to perform the morphological operations. Various inbuilt morphological function exist in Matlab and OpenCV operation. But has been modified and re-written for the specific use for the detection of tiger stripes.
Rolled in Scrap Finder	• A code has been written manually to find the deviation of the edge of the rail to find rolled in scrap type defect.
Wire Defect Finder	• A sliding window is used to detect local variation and deviation of pixel grey values. A code has been written manually to perform the operation.
Classification Procedures	Classification algorithm routines have been written that • Compares the size of dark zones in tiger stripes, • Compares line size in Line on the top, • Compares wire size for wire defect • Compares deviation calculated for rolled in scrap All with a defined parameter in a specific manner to classify them as defects for each case.
Statistics for Training Neural networks	• Methods have been written to scan the whole image and computer statistics of mean, variance, standard deviation.

Optimised edge detectors, thresholding methods and other image processing methods have been applied to achieve the current results. Neural Networks which are very flexible, efficient and simple to handle and implement. Their training is found to be much faster as compared to the self-organizing maps or any other neural networks presently being used for image processing.

Detection of each defect involves specific algorithmic steps that have been defined using the best combination of existing imaging methods, their adapted forms and specially written algorithms. The results obtained, success rates, detection speed and performance evaluation of the developed detection algorithms has been described in detail in the next chapter.

6 PCA Method for Defect Detection

6.1 Introduction

This chapter describes an alternate rail defect detection method based on Principal Component Analysis technique to those methods, described in the last chapter. A single adaptable, trainable and efficient detection method for the detection of all the rail defects under investigation has been explained in detail in this chapter. Chapter covers the implementation steps of PCA based detection method, analysis and comparison of PCA statistics computed followed by classification of those features using back propagation neural networks.

6.2 Principal Component Analysis (PCA)

“Principal Component Analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.” (Yinglin and Li, 2011)

PCA is a method of identifying similarities or differences in a data, in other words it can be referred as a technique that recognizes patterns present in a data. PCA reduces dimensionality of data, retaining the useful information. This technique has been chosen for rail images because of two of its main characteristics; it is known useful for handling high dimension data and extracting useful features of the data. That is the basic need for inspecting rail images, as rail image data under investigation is very large and all the defects to be detected have specific features.

PCA technique is tested with an aim of developing a generic detection technique for the different types of defects found on these manufactured rails. The major objective is to design an adaptable and trainable single technique for detecting the different types of faults. A system is to be developed that can learn and adapt itself to the changing types of faults with the time. It is hoped to develop a more generic approach rather than the series of bespoke and tailored algorithms presented so far in Method-1 (chapter 5), using different new and developing techniques such PCA and neural networks. Idea is to find the PCA features of defects and clear rails and train the neural network in order to classify them. The defect detection steps using PCA method have been presented in a form of a flow chart in Figure 6.1.

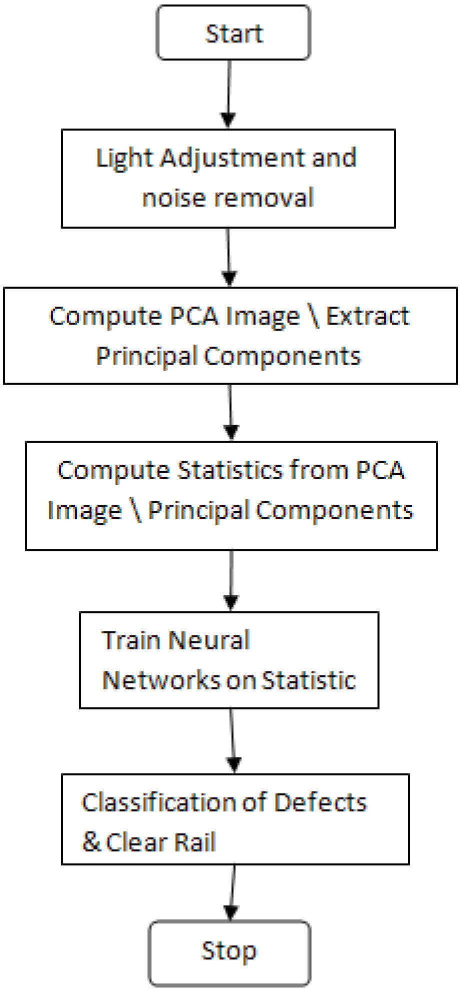

Start
Light Adjustment and noise removal
Compute PCA Image \ Extract Principal Components
Compute Statistics from PCA Image \ Principal Components
Train Neural Networks on Statistic
Classification of Defects & Clear Rail
Stop

Figure 6.1: Flowchart of steps for PCA based Defect Detection Method

6.2.1 Principal Component Analysis (PCA) Calculation

PCA calculation involves the computing of covariance matrix of any given data. Covariance is to find out that how different variables change together, with respect to each other. It is calculated using the following steps:

1. **Centralise the data:** The mean of each dimension of the data is subtracted from each element in that dimension. This process results in a data set which has mean equal to 0.
2. **Covariance:** Centralised data is used to calculate covariance matrix. For example, if (x,y) is a 2 dimensional data set, its covariance is calculated

using the following formula:

$$\mathrm{Cov}(x, y) = \frac{\sum_{k=0}^{n}(\mathrm{x}-\bar{\mathrm{x}})(\mathrm{y}-\bar{\mathrm{y}})}{\mathrm{n}-1} \qquad \textit{Eq. (6.1)}$$

Where,

x = data in X dimension

y = Data in Y dimension

$\bar{x}$ = Mean of x

$\bar{y}$ = mean of y

3. **Calculate Eigen vectors and Eigen values**: Eigen vectors and Eigen values of the covariance matrix are then calculated using matrix algebra. Eigen vectors are certain vectors which when multiplied by any matrix change only their magnitude, and direction remains either unchanged or exactly reversed. The factor by which the magnitude changes is called Eigen value, which is positive if vector's direction remains unchanged and is negative if the direction is reversed after multiplication. Eigen and Eigen values are related in a following way.

$$\mathbf{I * E_{vector} = E_{value} * E_{vector}}$$

I = Matrix of any dimention

Evector = Eigen Vector

Evalue = Eigen Value

If I is a matrix of any dimension, then E_{value} is an Eigen value of I, if they are related to each other by a non-zero Eigen vector E_{vector}.

4. **Select Eigen vectors with largest Eigen value:** In the final step Principal components are chosen. The Eigen vectors with highest Eigen values are the principal components of the data.

6.2.2 Computation of PCA Projection Image

An in built OpenCv PCA library functions \ methods are available which have been tested on the rail images. The said functions work in similar steps explained above to perform PCA analysis and computes Eigen vectors, Eigen values and projects the principal Eigen vectors which are known as Principal Components. Specific region of interests from the image can be inputted to the system in matrices form, which computes covariance matrix using the covariance equation from Eq. (6.1). . Eigen vectors and Eigen values are then calculated from that covariance matrix following the steps explained in section 6.2.1. Finally, the vectors with highest Eigen values are assigned as PCA features or Principal Components of that particular matrix.

A code has been written that slides a window mask of some defined size over the whole image and PCA features of area covered by the mask at every location are calculated. The PCA result for the mask size 6 has been presented in Figure 6.2(b).

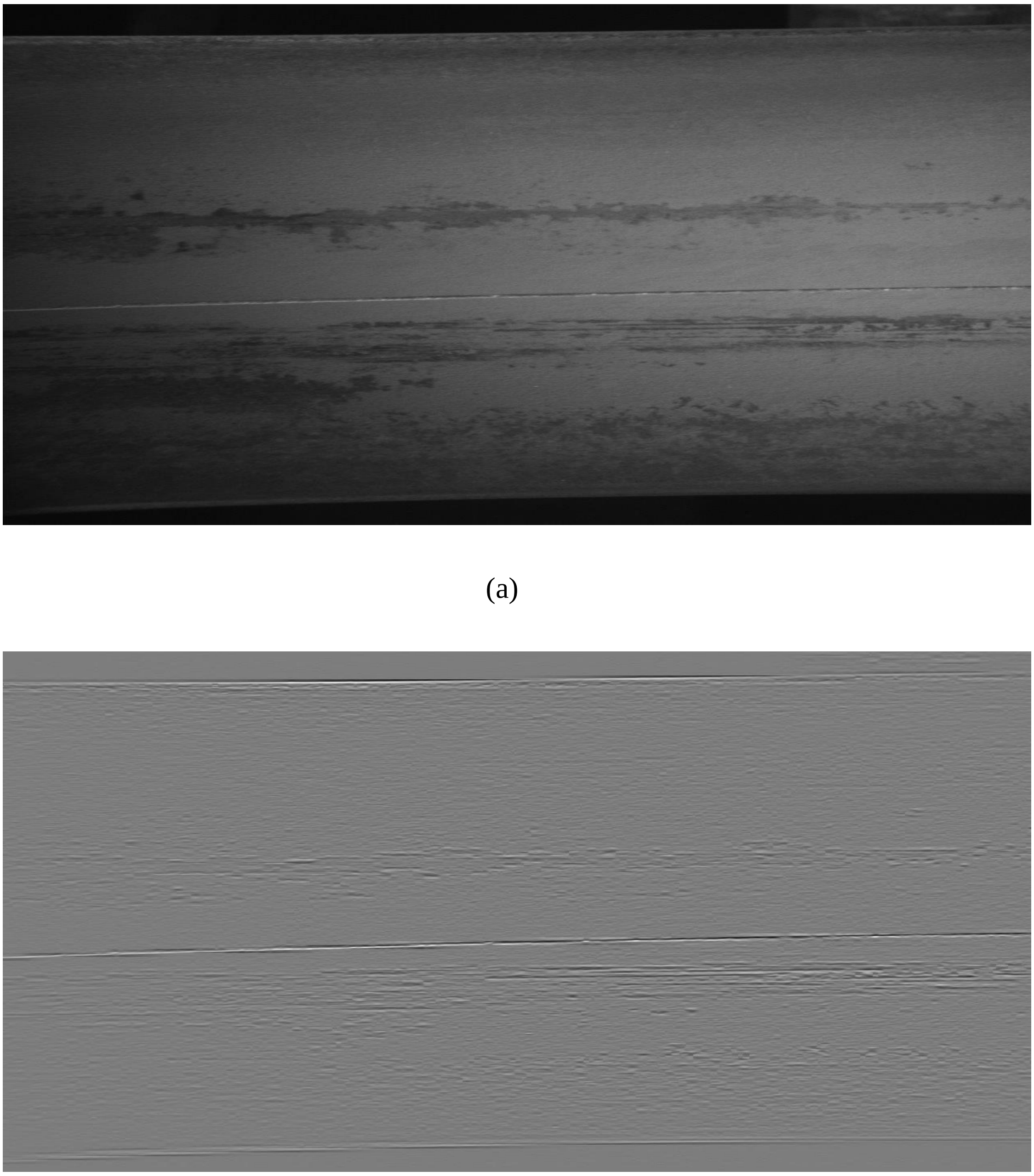


Figure 6.2: PCA result of rail image with a wire defect (a) Original rail image with wire defect (b) PCA Projection image obtained by simply plotting the principal components computed for each location of a sliding window of size 6, from a complete image.

The PCA image from the original image is computed by the following steps.

1. Grey scale input rail image is enhanced if needed for light adjustment and noise removal, using the procedures explained earlier in chapter 4 only if any pre-processing is required.

2. A small mask \ window size is defined, which is slided over the whole image and principal components of each location under that window are computed. Mask sizes of 4, 5 and 6 give the best results. Further increase of the mask size picks the background noise as well. Possible explanations for better result for smaller mask sizes are the method of compression used to save the image and the nature of the light source on the rails.

3. Variables to store input vector, average vector, Eigen vectors, Eigen values, Eigen vectors projections or decomposition coefficients (referred as PCA features as well) are defined.

4. A PCA method is called which performs PCA analysis of the input vector set provided to the method. First, average vectors are calculated followed by the computation of covariance matrix. Finally Eigen values and Eigen vectors are calculated and then all the values are stored in array vectors. The details of the PCA computation have been explained in the previous section of Principal Component Analysis Calculation, section 6.2.1.

5. The computed Eigen vectors are arranged in descending order of their respective Eigen values. Therefore, the first column of the Eigen vector are those vectors with maximum Eigen values, these represent the Principal Components.

6. Eigen vectors are finally projected into vector's subspace using another OpenCV method. This method computes the decomposition coefficients matrix. In other words, the input vectors are projected into subspace of the first column of Eigen vectors \ principle components.

7. These PCA projections of Eigen vectors, for every position of the sliding window over the whole image, are assigned to the corresponding location of a blank copy of the image. Resulting PCA image after the complete computation is displayed. The window sliding step can be chosen. However, for best results the step size is one pixel in horizontal direction and window

size in vertical direction. Therefore, only the first column of the Eigen vectors that have maximum Eigen values are chosen. The result of the procedure up till this step can be seen in Figure 6.2. PCA projections have been referred as PCA features in the chapter.

6.2.3 Statistic for PCA Projection Image

The algorithms steps explained in the last section allow the user to calculate PCA representative image, which is the projection of principal Eigen vectors or principal components of the image. Defective and clean areas of rail can be clearly seen on this PCA projection image shown in Figure 6.2. Representative areas for the defects and clear rails can be marked and selected on this image. Statistics such as mean, variance and standard deviation (STD) of the selected areas can then be calculated for the selected regions of the rail. PCA representative image like the one presented in Figure 6.2 is used to compute the statistics for either the whole image or for the defective and clear area of the rail that can be selected manually by a user. The software allows selecting any area of the image for which whole statistics needs to be computed, using a mouse. This can either be a rectangular region from the image or only an area around a selected point from the image. The processes by which these statistics are computed for complete image are listed below:

1. **S**tandard deviation, mean, variance and other statistics of these PCA features are computed using a defined method. This method scans the whole PCA projection image by sliding a small vertical window of defined length and width of one pixel. The width is chosen as one pixel as most of the defects are in the form of horizontal lines, therefore standard deviation usually varies vertically and not horizontally. The window is convoluted over the whole image with step size of 1 in both columns and rows direction.
2. A blank copy of the input image as saved for each statistical mask. Statistics that are calculated for each location of the mask are assigned to their respective location on their blank copies.
3. At the end, each statistical test has its representative image saved for further processing or classification purposes.

In case of selected regions of the rail image, the selected \ marked area is stored in an array. The statistics of that small area are computed sliding the similar vertical window of defined length and width of one pixel over the region, with step size 1. Statistics for each location of the window are finally computed and saved in respective arrays.

The calculated statistics are then plotted against each other to see the variation clearly. The next section covers the PCA statistics plots, their comparison and evaluation for each type of defect.

6.3 PCA Statistical Profile Analysis

The Eigen vectors, Eigen values and PCA projection vectors \ features extracted from the clear and defective rail images have been analyzed. Their values and statistical relationships between these values, such as variance, standard deviation and mean for defects and clear rails, were computed and described in the last section. These statistics have been compared and plotted together on the same graph each other to find the relation that best describes the data in this section. These plot obtained have been presented and analysed in this section. This has been done to differentiate the profiles of defective and clear parts of the rail. The plots have number of points on X axis and the PCA features, their variance and standard deviation respectively on graph b, c and d on Y-axis. Defects are plotted in blue color lines and values from clear areas/non-defect areas are plotted in orange color.

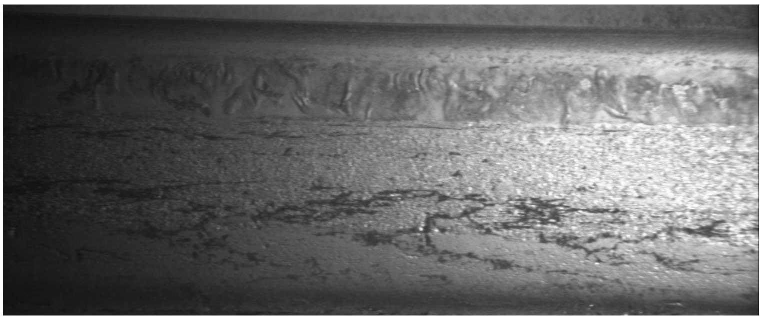
6.3.1 PCA Statistical Plots for Tiger Stripes

Principal Components (Eigen vectors with high Eigen values), their decomposition coefficients (Principal components projections) and the statistics of variance and standard deviation between them, for the tiger stripes and clear rails have been plotted against each other.

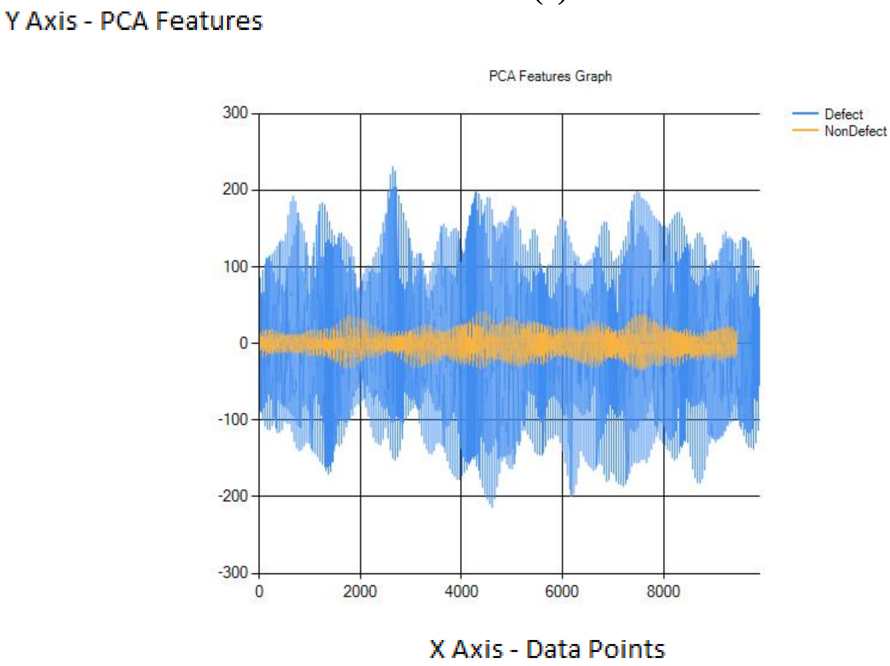
Statistics of the selected regions of rail images from the defected and clear rails, calculated as explained in last section (section 6.2.3), are saved in their respective variables. The original image with the graphs obtained by plotting the statistical points calculated on the clear and defective rail areas can be seen in Figure 6.3.

The PCA features for defects and non-defect areas are plotted as shown in Figure 6.3

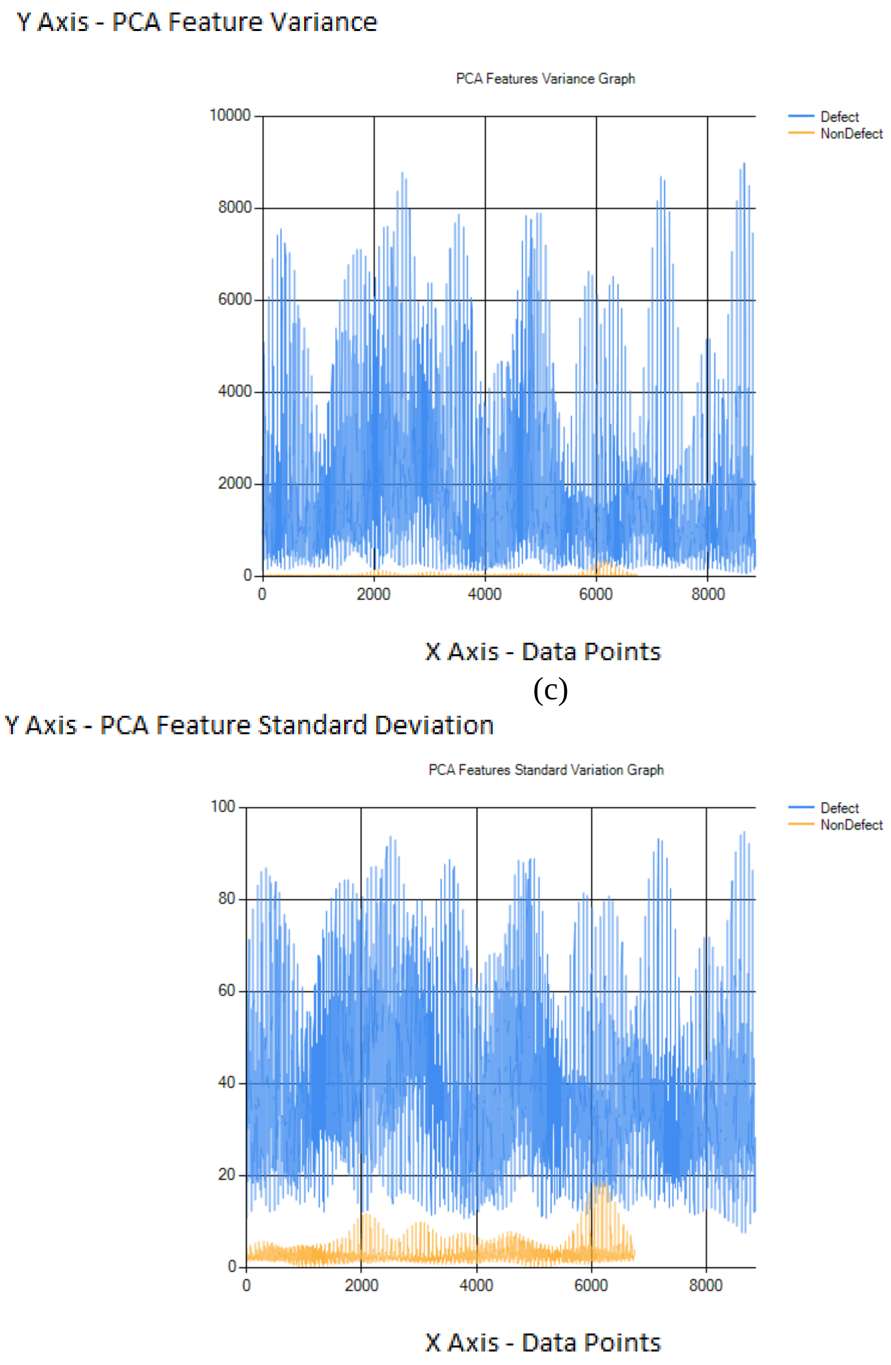
(b). It can be seen that the plots obtained are very distinct from each other. The plots of variance and standard deviation of PCA features shown in Figure 6.3 (c) and (d) are even more distinctive for defects and non-defect areas of the rail with tiger stripes.



(a)



(b)



(c)

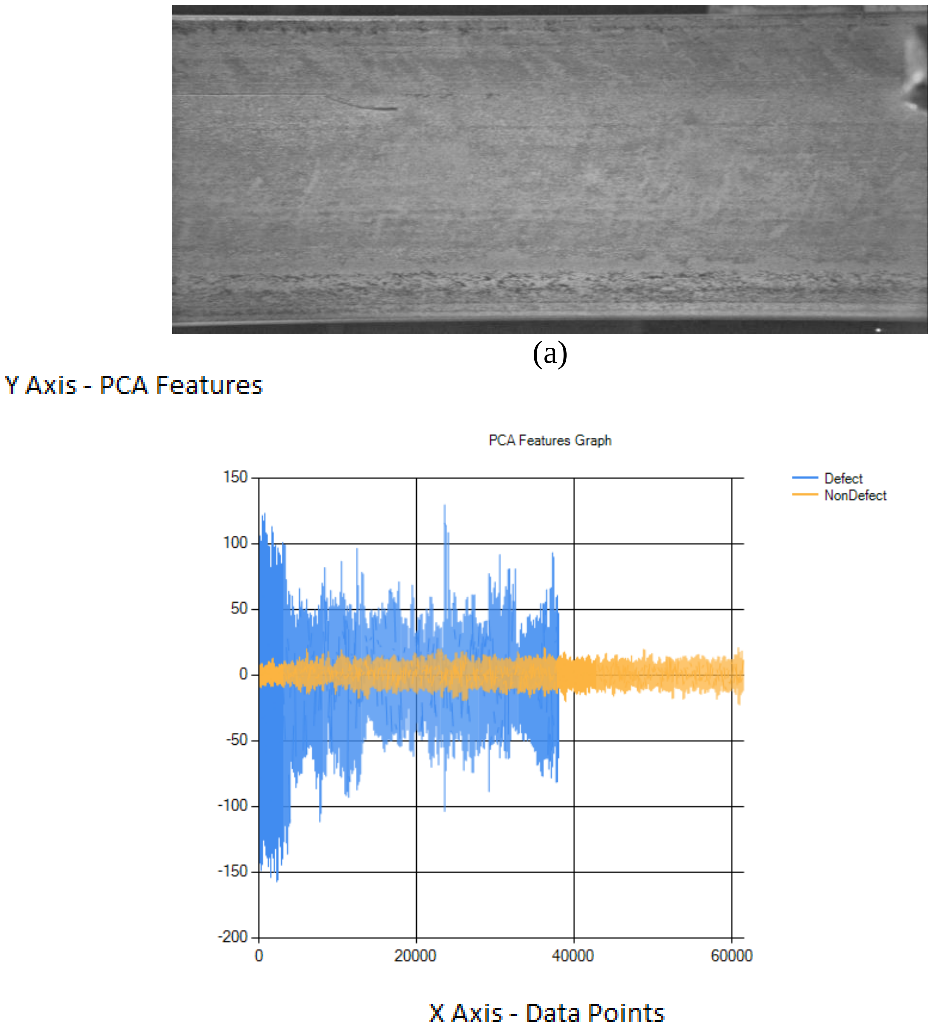
(d)

Figure 6.3: (a) Original rail image with tiger stripe type defect (b) Principal components\PCA features computed on tiger stripe rail areas plotted against the clear rail areas (c) Variance of PCA features of computed on the rail areas defected with tiger stripes and clear areas of the rail plotted together (d) Standard Deviation of PCA features of rail areas defected with tiger stripes and clear areas of the rail plotted together.

Therefore, it has been concluded that PCA features and statistics computed from them for Tiger Stripes type defect are able to differentiate the defective areas from the clear areas of the rail.

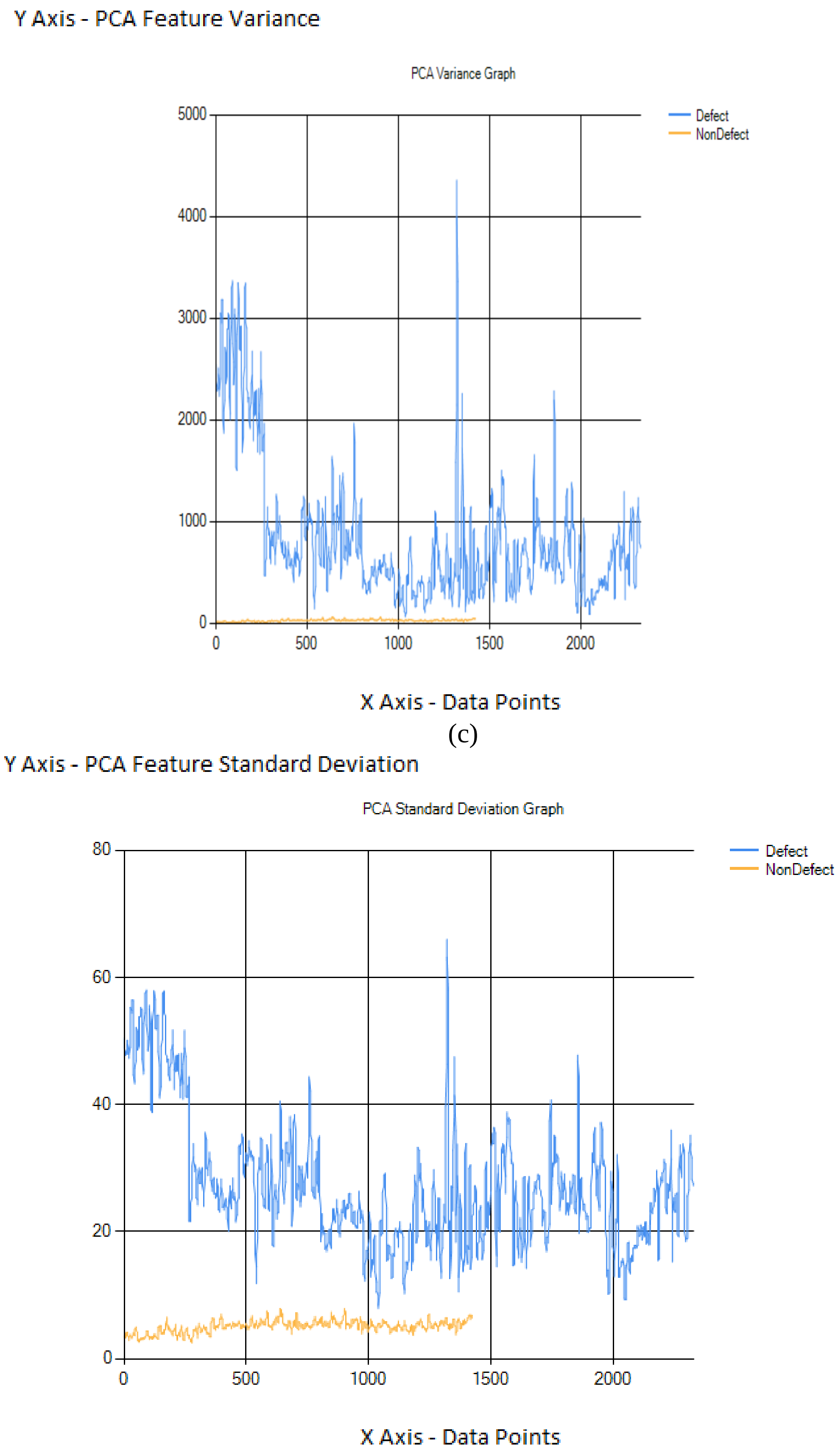
6.3.2 PCA Statistical Plots for Wire Defect

PCA features and their statistics were computed and plotted for wire defects, with the resulting plots of wire defects shown in Figure 6.4. It can be seen from the plot in Figure 6.4 (b) that features of defects and non-defects are quite distinct and are even more differentiable for the statistics plots in Figure 6.4 (c) and (d).



(a)

(b)



(c)

(d)

Figure 6.4: (a) Original rail image with wire defect (b) Principal Components\PCA features of wire defect and non-defect rail areas plotted against each other (c)

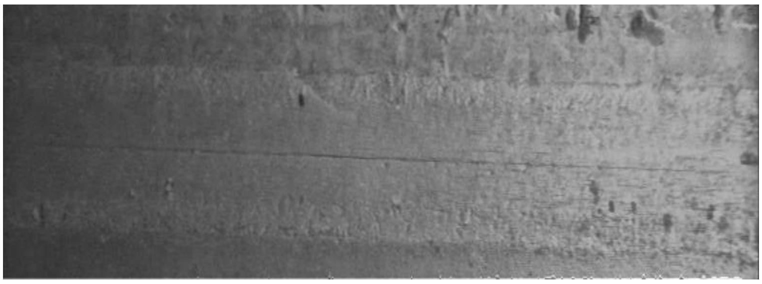
Variance of PCA features of wire defect and non-defect rail areas plotted together (d) Standard Deviation between PCA features of wire defect and non-defect rail areas plotted together.

The above plots show that PCA have a great potential to detect the defects.

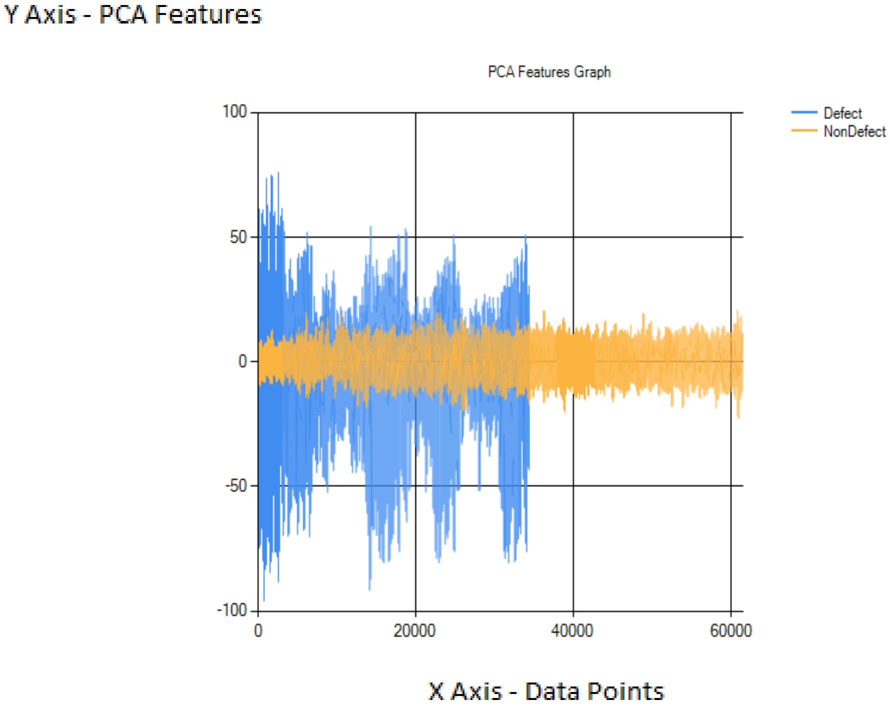
6.3.3 PCA Statistical Plots for Line on the Top of Rail

PCA features, their variance and standard deviation when plotted for line on the top of rail, comparatively similar results are obtained. The respective PCA statistics plots can be seen in Figure 6.5.

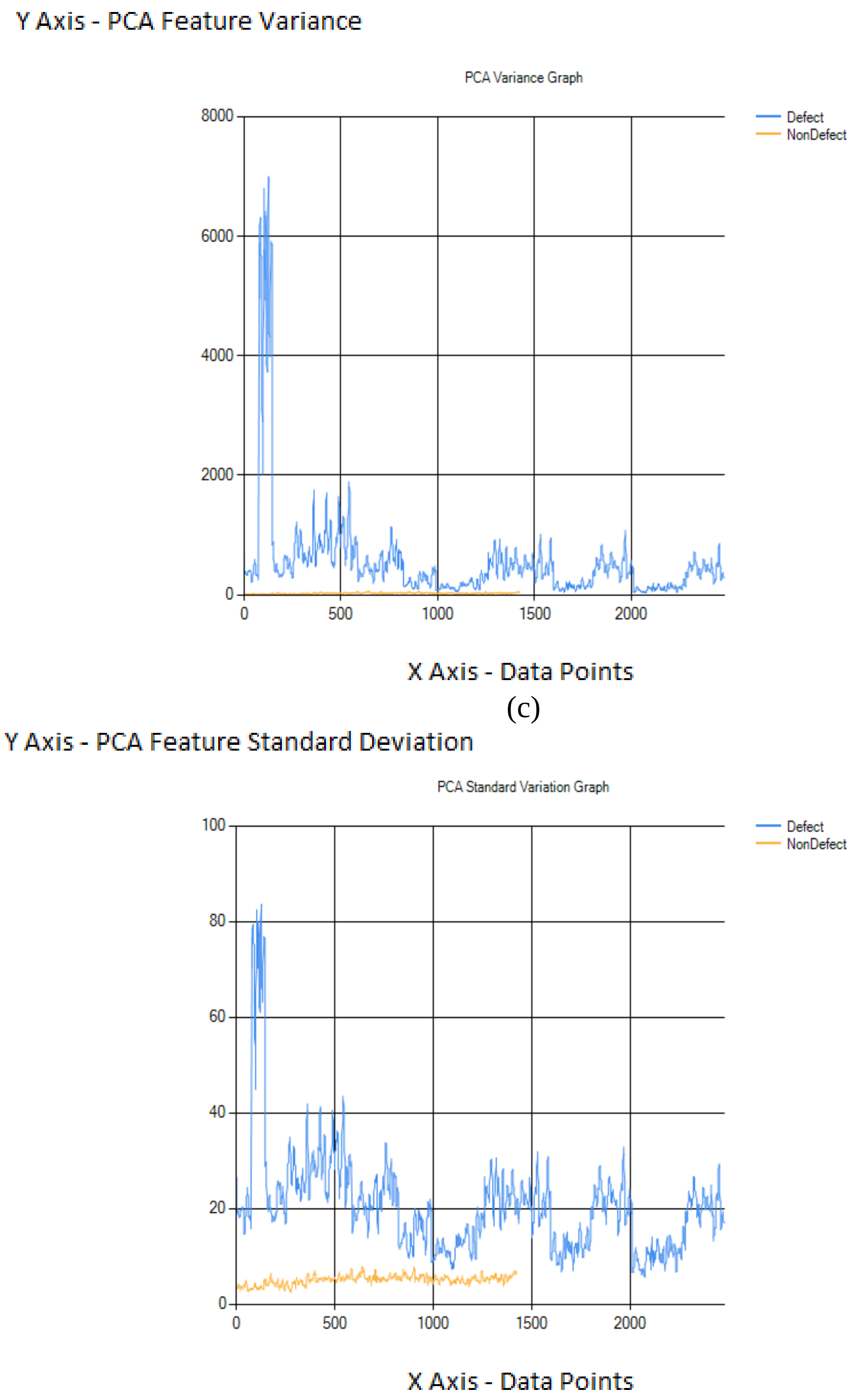
The lines for defective and clear region of the rail can be seen in different regions of the plots. Hence, PCA also appears to be effective for the line on the top type rail defects.



(a)



(b)



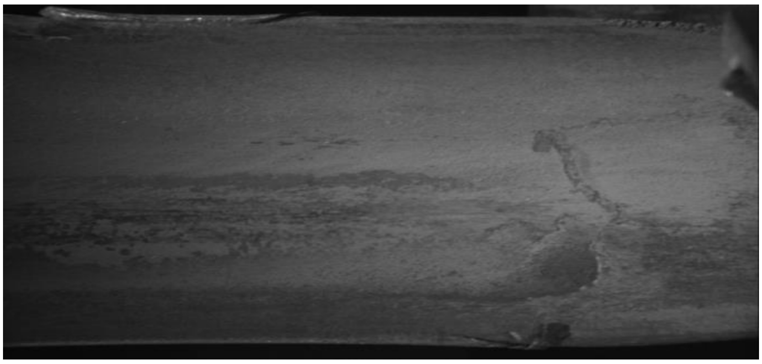
(c)

(d)

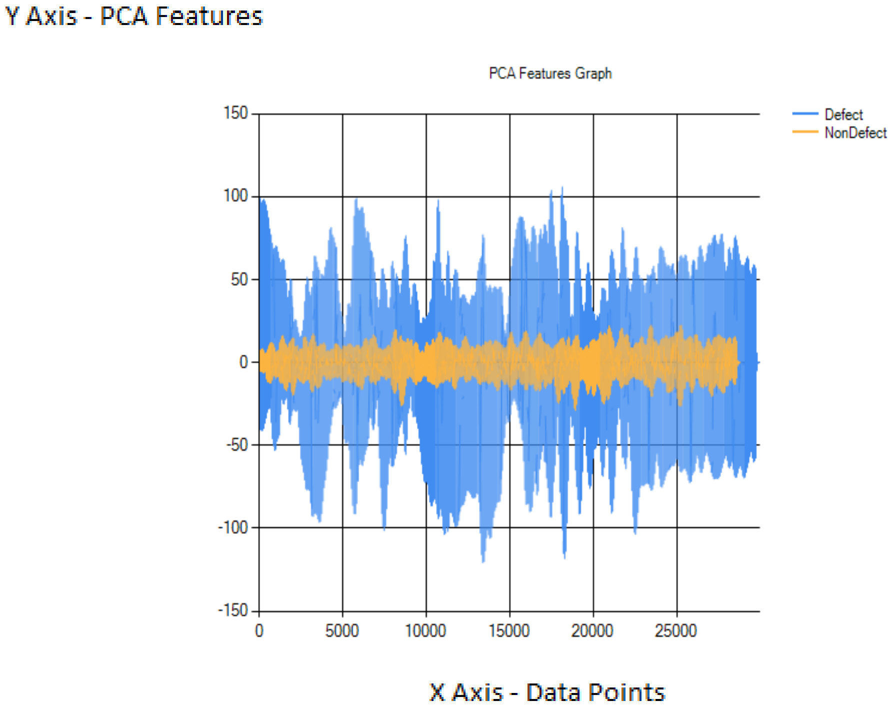
Figure 6.5: (a) Original rail image with Line on the top of rail type defect (b) Principal Components/PCA features of line defect and non-defect rail area plotted against each other (c) Variance between PCA features of the line defect and non-defect rail areas plotted together (d) Standard Deviation among PCA features of line on the top of rail and non-defect areas of the rail plotted together

6.3.4 PCA Statistical Plots for Rolled in Scrap

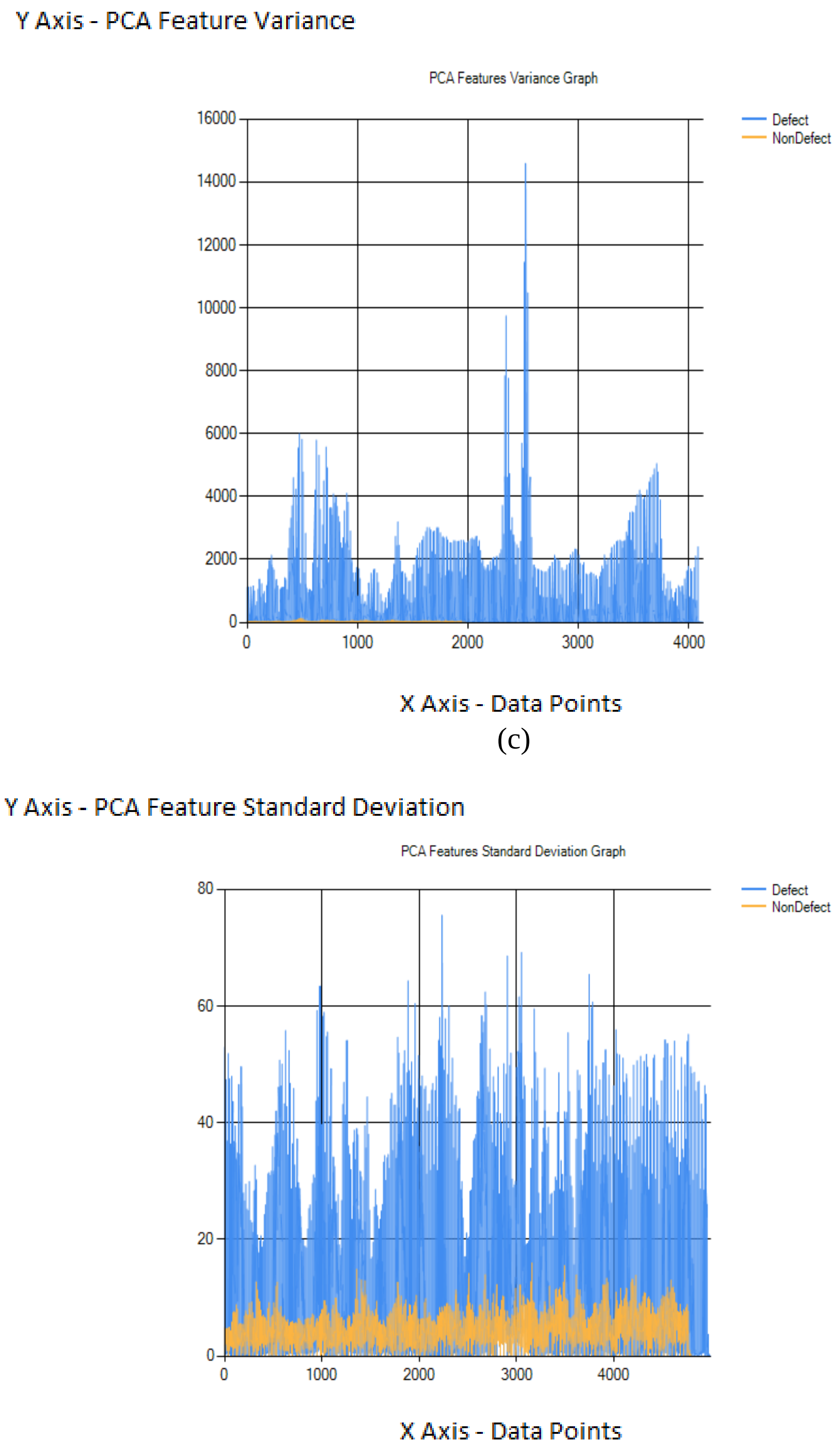
PCA statistics when plotted for Rolled in scrap type rail defects, the results were not very distinctive. The plotted lines for clear and defective rail's PCA features and statistics are overlapping each other as can be seen inFigure 6.6. Hence, PCA doesn't appear to be very effective for the rolled in scrap type rail defect.



(a)



(b)



(c)

(d)

Figure 6.6: (a) Original rail image with rolled in scrap type defect (b) Principal Components/PCA features of RIS and non-defect edge plotted together (c) Variance of PCA features of RIS and clear/smooth edge of the rail plotted together (d) Standard Deviation among PCA features of RIS and non-defect/clear edge of the rail plotted together

The range and average values for PCA features and the statistics computed for all the types of defects have been compared and analyzed in the next section.

6.3.5 Analysis of PCA Statistics

The PCA features and statistics of those components have been compared for the all the types of defects in Table 6.1 and their average values have been presented in Table 6.2. PCA features and their statistical relationship for clear and defective rails were computed for a large number of selected rail areas. More than 60,000 values from several rail images for both clear and defective regions were saved and compared to get the results shown in tables below. It is very obvious that PCA features and statistical plots differentiates the defective and clean areas of the rail exceptionally well for tiger stripes, wire and line on the top type defects. When the similar plots are done for Rolled in Scrap type defects, PCA did not work as well. Rolled in scrap, being an irregularity on the edge and being different from the three other defect types under investigation, can be detected by separating the rail from its background and measuring the irregularity of the edge. However, the existing detection method for Rolled in scrap explained in last chapter is already 100% using the developed algorithm.

Table 6.1: Comparison of PCA statistical values of clear and defective regions selected from several rail images. More than 60,000 values for Principal Components and their statistics from various rail images were computed and compared to get the results.

	Clear Rail Surface	**Wire Defect**	**Tiger Stripes**	**Line on the Top**	**Straight Edge**	**Rolled in Scrap**
PCA Features Range	-23 to 21	-158 to 130	-229 to 300	-96 to 77	-130 to 107	-163 to 212
Variance Range	6 to 63	62 to 4360	85 to 11000	32 to 7000	90 to 471	75 to 2069
STD Range	3 to 8	10 to 67	11 to 105	6 to 84	10 to 22	9 to 46

Hence, the first observation that can simply be made on the data available is; that the rail with PCA projection values within the range provided for clear/normal rail (between -23 to 21) in the Table 6.1 can be clearly marked out as a clear rail. The rail with PCA values out of the safe range needs to be investigated. Moreover, when the average values of PCA features and statistics were computed they reflected clearly different values. This suggests that the detection of the defects is possible. The average statistical values for clear rail and different types of defects are presented in Table 6.2.

Table 6.2: Average statics values for PCA clear rail surface, straight rail edge, wire, tiger stripes and line on the top of rail and RIS Type Defects

	Clear Rail Surface	**Wire Defect**	**Tiger Stripes**	**Line on the Top**	**Straight Edge Average**	**RIS Average**
PCA Projection Values Average	0.66	-1.33	1.25	0.72	0.49	2.84
Variance Average	27	854	2013	544	220.1	339.62
STD Average	5.1	27.5	37	21	14.5	17.71

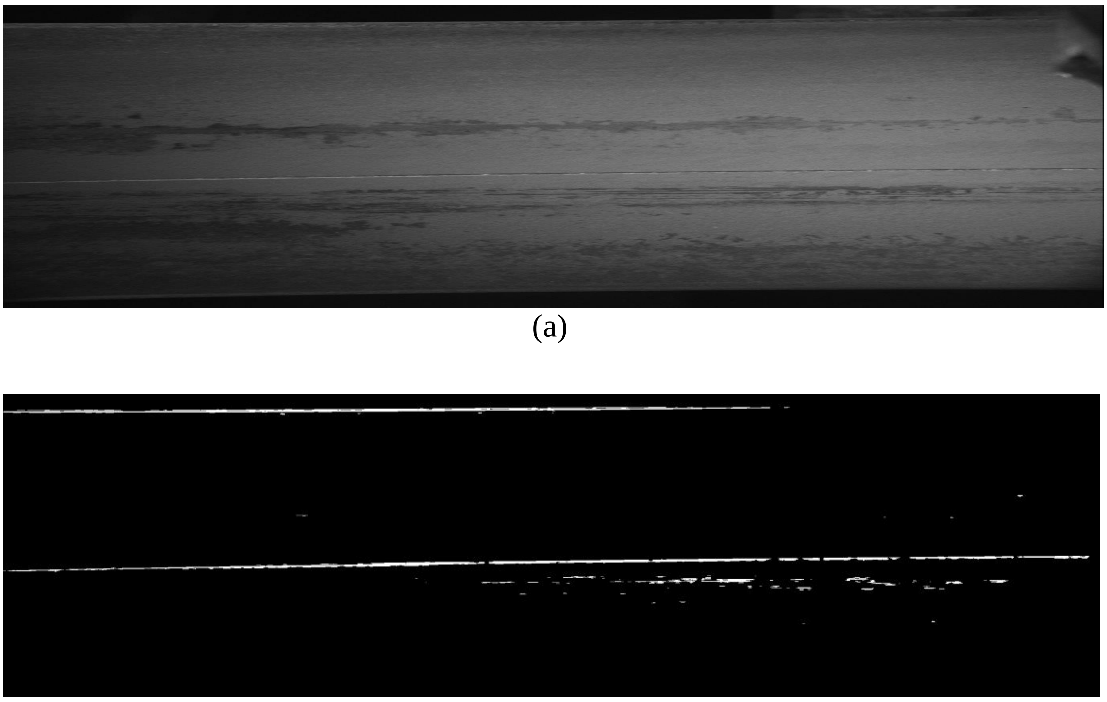
It can be concluded, therefore, from the data values presented in the table above that defect detection is possible using the PCA data and neural networks trained on the PCA data of the rail images and the statistics calculated on this data should also be able to recognize and classify them accordingly. The next section describes and analyzes the rail defect detection based on PCA features and their statistics.

6.4 Statistical Output for PCA Projection Images

Simple standard deviation computed over the whole image of PCA projections, as explained in section 6.2.3, when displayed gives exceptionally clear defects lines as can be seen in Figure 6.7 for wire defect, in Figure 6.8 for line defect, in Figure 6.9 for rolled in scrap defect and Figure 6.10 for tiger stripes.

These result images have been obtained by simply displaying the standard deviation obtained by sliding a window of variable height (mostly between 2 and 20 pixels) and width of 1 pixel, oriented vertically over complete PCA projection image. Vertical window has been used as most of the defects under investigation have horizontal spread of defect lines. Hence, statistics such as standard deviation and variation vary vertically. For each location of the window, the standard deviation is computed for all the values underneath the mask. The value computed is assigned to the top most location of the area of image currently under the window, because the window is moved with a step of 1 pixel both in x and y directions. The vertical window is moved by a step size of 1 in both vertical and horizontal directions over the image.

A standard deviation image for a wire defect PCA projection image; presented in Figure 6.7, shows the wire clearly running through the image length. However, some noise next to the defect mark can also be noticed. Though those non-defects marks are broken and faint but their PCA projection’s statistics are comparable to that of the defects. That is why they are quite visible in the image.

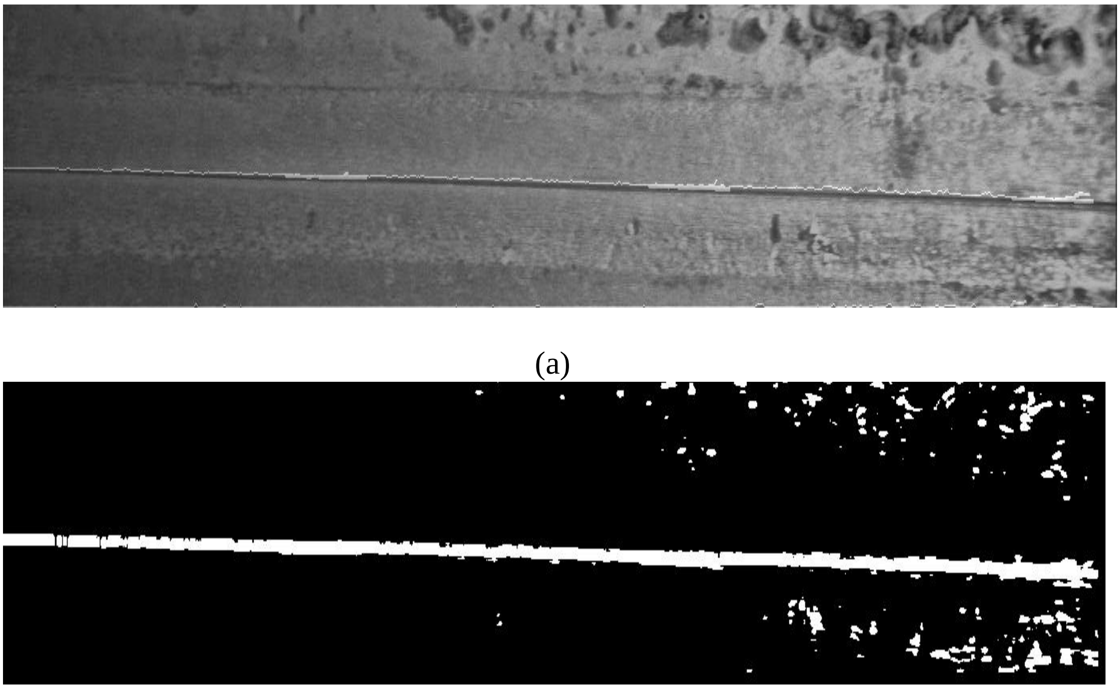


(a)

(b)

Figure 6.7: Wire defect detection using PCA Standard deviation values. (a) Original rail image with wire defect. (b) Standard deviation image computed from a PCA projection image using window size 6.

Similar plots for the line defect can be seen in Figure 6.8. The defect line is clearly visible; however, the noise marks from the background can also be seen on the right side of the image.

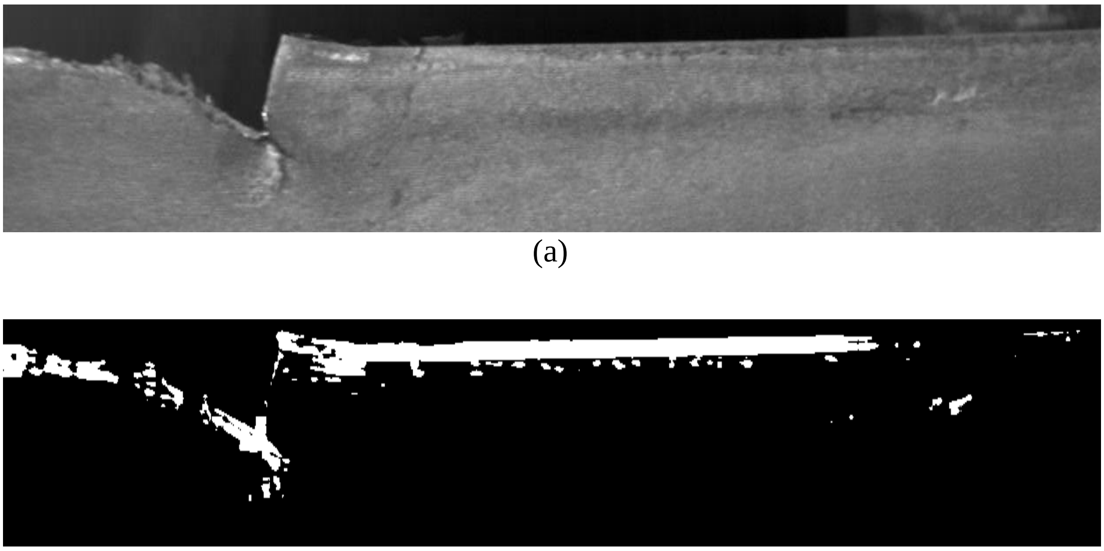


(a)

(b)

Figure 6.8: Line on the top defect detection using PCA Standard deviation values. (a) Original rail image with line on the top defect. (b)Standard Deviation image computed on PCA projection image using window size 8

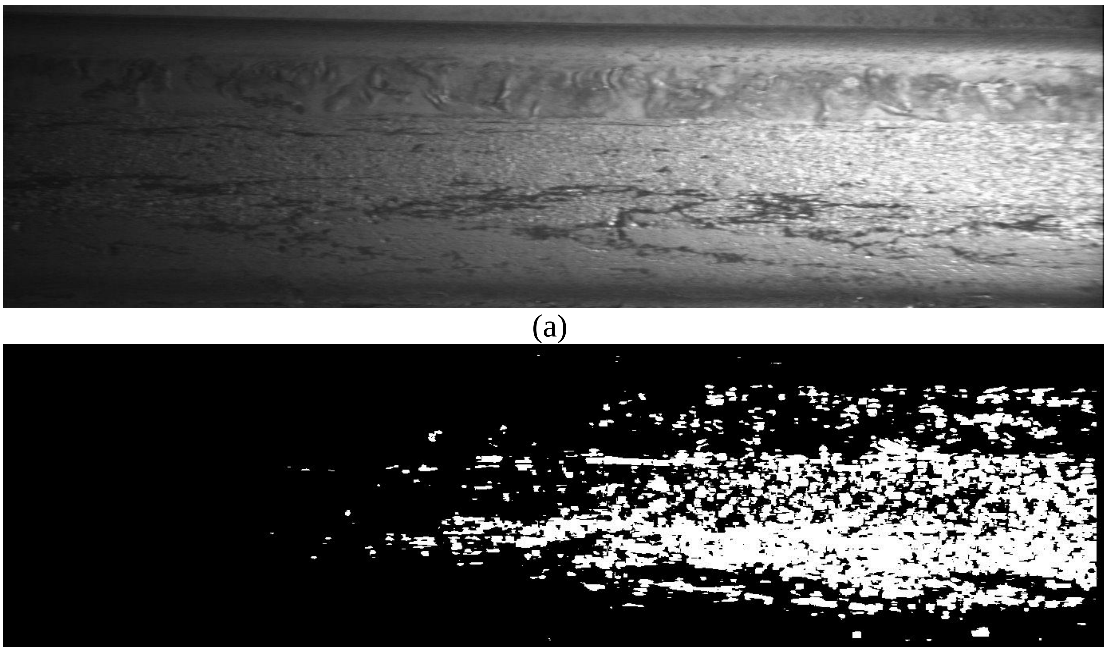
STD plots for the PCA projection image with RIS type defect has been presented in Figure 6.9. The broken edge of the rail has clearly been marked out in the image.



(a)

(b)

Figure 6.9: Edge Detection using PCA Standard deviation values. (a) Original rail edge image with RIS type defect. (b) Standard deviation image computed on PCA projection image using window size 8.



(a)

(b)

Figure 6.10: Tiger stripes detection using PCA Standard deviation values. (a) Original rail edge image with tiger stripes. (b) Standard deviation image computed on PCA projection image using window size 8.

The image result shown in Figure 6.7, Figure 6.8 and Figure 6.9 are encouraging to some extent and defects can be seen clearly on them. However, even after some post processing operations including thresholding, clearing blobs and morphological operations, some non-defect noise has been picked up and some parts of the defects have been dropped as can be seen in the results obtained for each defect and the most for the tiger stripes, as presented in Figure 6.10. Hence, classification of PCA statistics needs some further processing. Neural networks are tested for this purpose and have been discussed in the next section.

6.5 Classification of PCA Statistics using Neural Networks

The computation of the values for Eigen vectors, Eigen values, PCA projection values and statistics such as variance, means and standard deviation using PCA values have been explained and analyzed in detail in the previous sections. All the results obtained have been presented and discussed as well. In this section Neural Network has been trained on the principal components, their projections and their particular statistics to perform the classification of different types of defects. This section covers the implementation of neural network used, its training and classification and discussion of results produced for each type of defect. Neural Network somewhat succeeded in detecting tiger stripes and rolled in scrap type defect. However, wire and line type defects could not be successfully classified using neural networks so far.

6.5.1 Defect Classification using Back Propagation Neural Networks

A back propagation neural network implemented in C# has been used to classify defects and non-defects PCA features and statistics. The basic model and working of back propagation neural network has been provided in Appendix A at the end of the report. The network implemented has one input layer and one output layer. Number of input neuron depends on number of input values. Hidden layers of varying numbers have been tested, and mostly one hidden layer with 5 or 9 neurons have been used. The structure of the implemented network has been presented below in Figure 6.11.

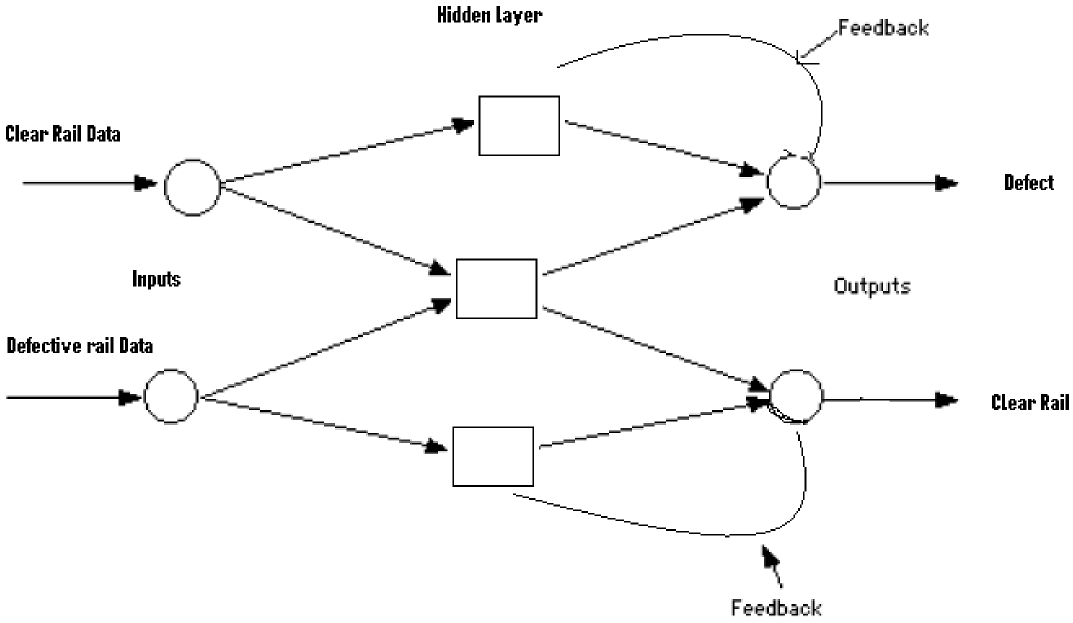


Figure 6.11: Example of a Neural Network implemented with one input layer with two neurons, one hidden layer and one output layer.

Various transfer functions such as linear, sigmoid, Gaussian and rational sigmoid transfer functions were tested for the classification purpose by neural networks. Hidden layers with sigmoid transfer function; in combination with output layer of linear function, give the better converging of the network, when trained on the statistical PCA data from the rail images. The classification results produced by neural networks and their evaluation have been presented in the next section.

6.5.2 Training Data

The network was trained on a representative set of data points from the various areas of the rail, both clear and with the defects. For example, there was an image with a tiger stripe type defect on it. The PCA image for the original image was computed and saved. Areas of defects and clear rails then were selected and saved in respective arrays. The selection was done by choosing different areas from the same image or several regions from different images. A whole block region or area around a chosen point can be selected into defect and clear rail data point arrays. The statistics of each block or area around a selected point were calculated; their respective targets are assigned and saved in individual arrays. The target values were different for different transfer functions. They can be 0 or 1, -1 and 1 for sigmoid function, 1 and 2 for Gaussian function etc. Statistics are calculated using the masking process explained in chapter 3. A small size of mask is chosen and is moved over the whole image. The function to be performed or the calculation is repeated for every position of the mask and the values are stored in input array. Most commonly tested statistics are median, mean, variation and standard deviation. Neural network is created using input; output and defined number of hidden layers with specific transfer function are trained on the input data for defects and clear rails created by the area selection performed on the rail images.

An example of the converged network plot has been presented in Figure 6.12. This has been obtained by plotting the values of error against epochs from the network training. The network was trained on statistical PCA data computed from an image with a tiger stripe type defect.

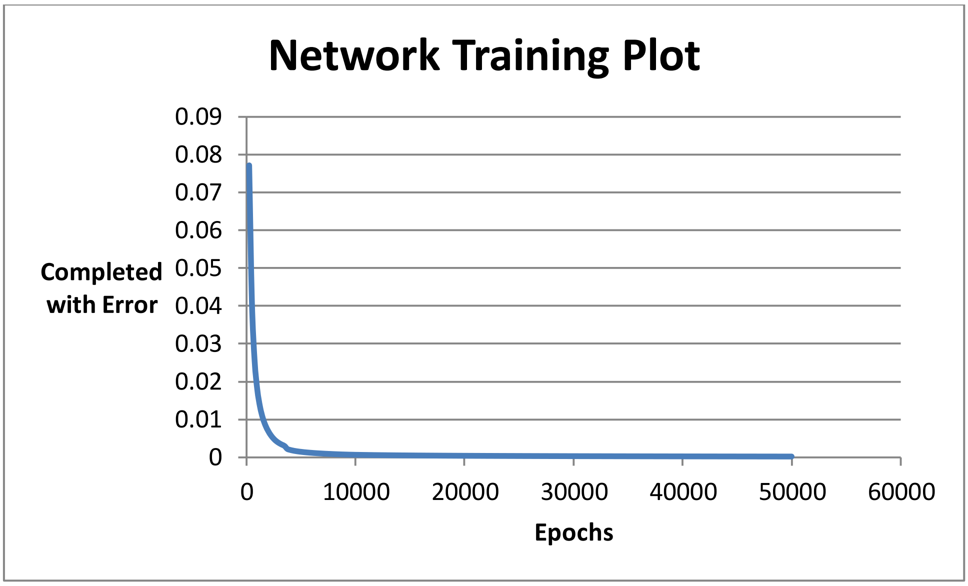
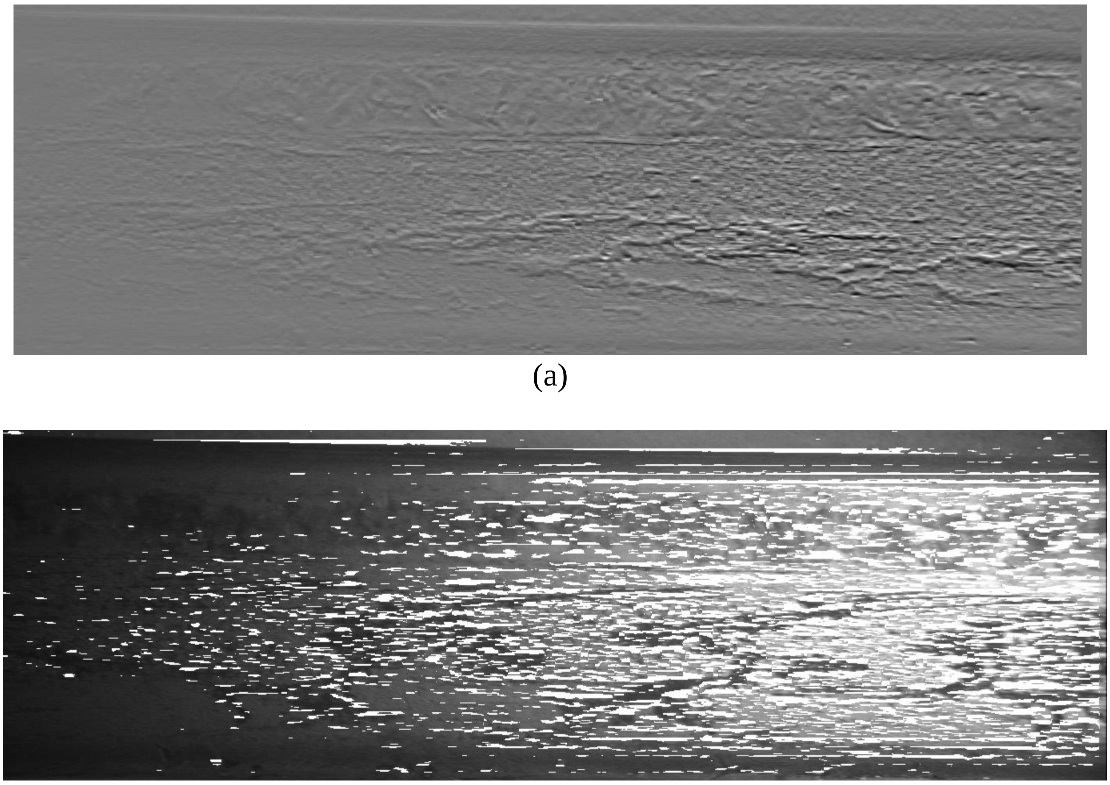


Figure 6.12: Example network training plot for a converged network trained on the statistics of Tiger stripes PCA features

6.5.3 Neural Network Results and Evaluation

The result of tiger stripes detected by neural networks trained on standard deviation of PCA features is shown in Figure 6.13. The back propagation neural network used to get these results had one input layer, 9 hidden layers with sigmoid transfer function and one linear transfer output layer. The network is converged reaching final error of only 0.000191 at 50000 epochs.



(a)

(b)

Figure 6.13: Result of Tiger stripe detection by PCA neural network method. (a) Original rail image with tiger stripes (b) Tiger stripes detected by neural networks trained on standard deviation of PCA features using a window size of 6.

Neural networks when trained on PCA statistics of rolled in scrap type defect also manages to differentiate the defective edge from the back ground. The result of rail edge with rolled in scrap can be seen in Figure 6.14. It can be seen from the figure that neural networks manages to separate the rail well from its back ground.

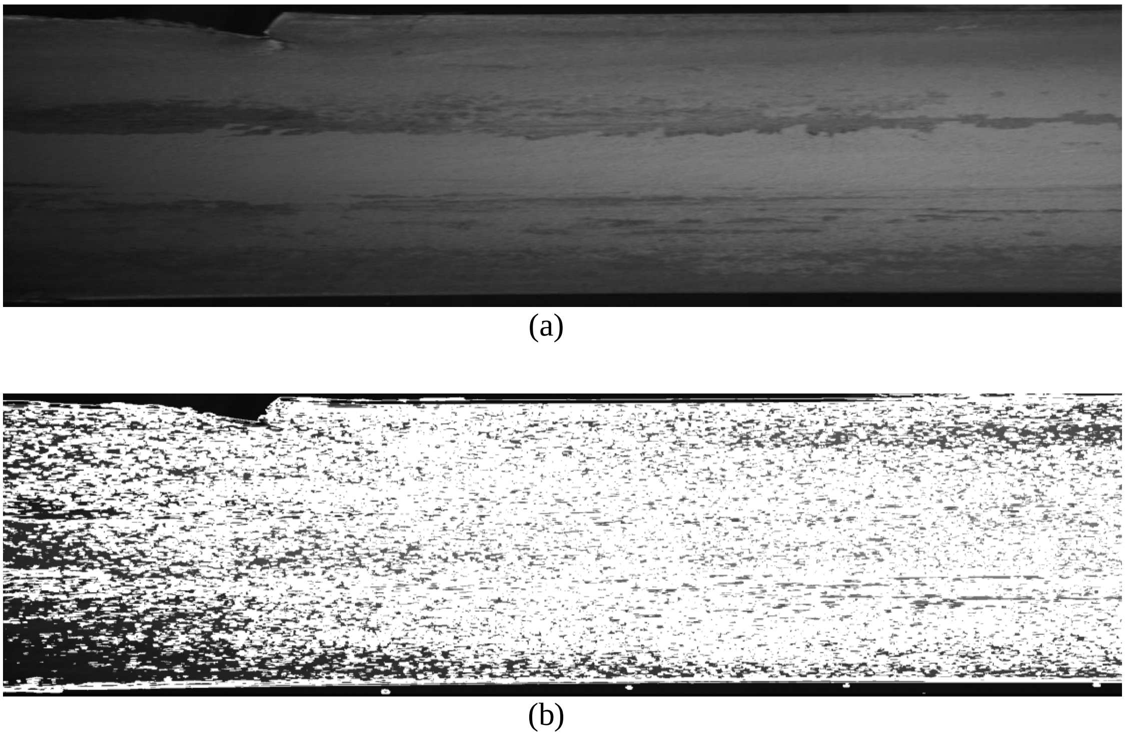


Figure 6.14: Result of Rolled in Scrap by PCA based neural network classification method. (a) Original rail image with rolled in scrap defect (b) Rolled in scrap type defect detected by neural networks trained on standard deviation of PCA features using a window size of 6.

6.6 Conclusion

Principal Component Analysis yields important information of the rail images and the defects. The analysis shows that the Eigen vectors, Eigen values, PCA projection values and statistics computed on these figures gives important information about the defects and clear areas of the rail. The PCA based detection method is found to be quite efficient in detecting clear rail areas, tiger stripes type rail defects and rail edges for rolled in scrap type defects. The method is still under initial testing stages and more work in future is expected to be done on it. The results obtained so far for all the detection methods explained in this chapter and the last one, have been analyzed and discussed in the next chapter.

7 Results

7.1 Introduction

Previous two chapters describe the detection techniques developed for the rail surface defects occurred during the manufacturing process. The developed detection algorithms for all the four types of defects under investigation have been discussed in detail in chapter 5 while PCA based alternate method have been described and discussed in chapter 6. This chapter gives a comprehensive performance evaluation of the detection algorithms developed so far for rail inspection. It discusses the defect detection techniques, results produced by those techniques, their performance evaluation and comparison; when applied on images of a complete rail. The detection algorithms have been tested on set of rail images provided by Tata Steel and the comprehensive performance and efficiency evaluation of the developed algorithms have been performed.

7.2 System & Software Requirements

The final software has been developed using the OpenCV image processing library encapsulated in EMGU for C# (Microsoft Visual Studio Professional). The software has been tested on an Intel Core I7 CPU 920 at 2.67 GHz, with 8 GB of RAM machine with Windows XP 64 Bits Professional operating system running on it. The offline images are treated by the software and a report of the detection is shown on screen for the experts to assess.

7.3 Software Processing Times

All the final implementation was performed on C# using an Intel I7 machine with 8GB of RAM. Initial testing was done on Matlab and then the codes are transferred to C#, as Matlab processing was very slow. The processing time has greatly reduced by the use of C# as can be seen in the table below.

Table 7.1: Comparison of processing time between Matlab and C#

Matlab	**C#**
• Takes approximately 10 minutes to process a rail. • Multi-threading capabilities limited	• Less than a minute for the same amount of processing • Higher processing power > less time to process a rail

The Matlab implementation of the algorithms was approximately taking 10 minutes to process all the images of one rail however C# algorithm performed the similar job in less than a minute.

The running time for a rail of approximately 110m length; with on average 440 images per camera view; is 46 sec (average time). The time between each rail while in heavy production is more than 1 min 30 sec, making the software capable of analysing rails in real-time.

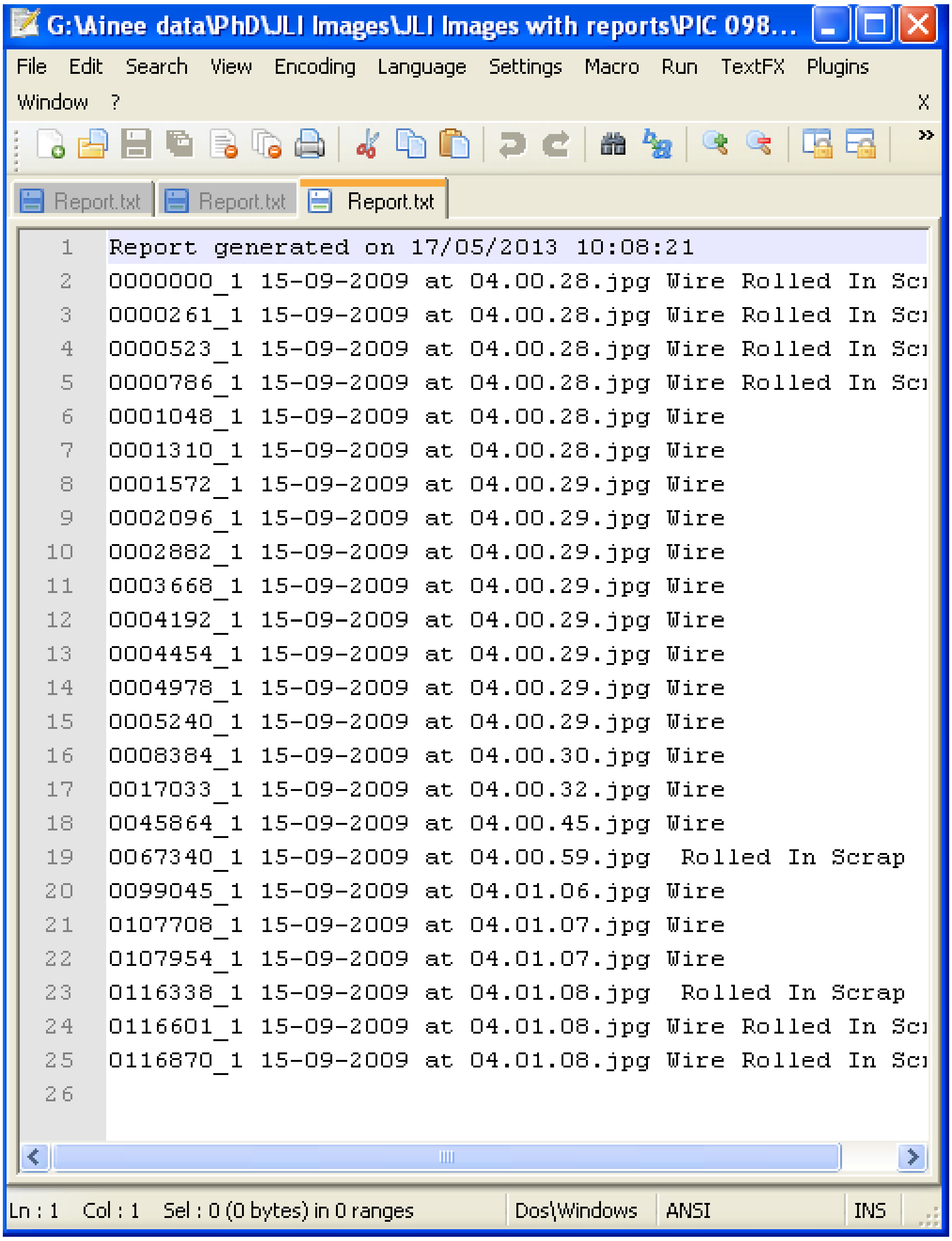
All the algorithms were firstly developed on MATLAB as it has been specially designed for engineers and have packages/libraries specifically designed for general control applications. Hence, makes the programming quick and easy. However, when the program becomes very large and requires more and more resources, Matlab does not perform very well and it becomes a necessity to switch to C\C++\C#. Hence, initial programming was done on Matlab to perform quick testing of the ideas. Once, the successful results came out, they were all coded into C#, which sometimes is a very lengthy and time consuming procedure. A trail was also made to convert the codes in to C language initially to get better performance in terms of speed. Moreover, OpenCV library is also available to be used in C environment that is a comprehensive open source computer vision library focusing on real-time image processing. However, finally the software has been developed using the OpenCV image processing library encapsulated in EMGU for C#. EMGU is a cross platform .Net wrapper that allows OpenCV functions to be called from .NET compatible languages such as C#, VB, VC++ etc.

7.4 Method 1 - Validation of the Results

Experimental results shows that the detection techniques developed for rail inspection, for this research work are successful with high efficiency and accuracy rates. They appear to work better than all the existing techniques applied as part of the investigation. They have the ability to identify and discriminate the different types of defects in these images.

7.4.1 Report Generated

A report is generated after processing each complete rail and apart from being displayed on the interface, it is also saved in the same folder that has been processed. The report contains the list of all the defects detected. The names of the defective images get written in the report file, while the algorithm processes the rail. The example of the report generated has been shown in Figure 7.1.


G:\Ainee data\PhD\JLI Images\JLI Images with reports\PIC 098...
File Edit Search View Encoding Language Settings Macro Run TextFX Plugins
Window ?
Report.txt
Report.txt
Report.txt
Ln : 1 Col : 1 Sel : 0 (0 bytes) in 0 ranges
Dos\Windows
ANSI
INS

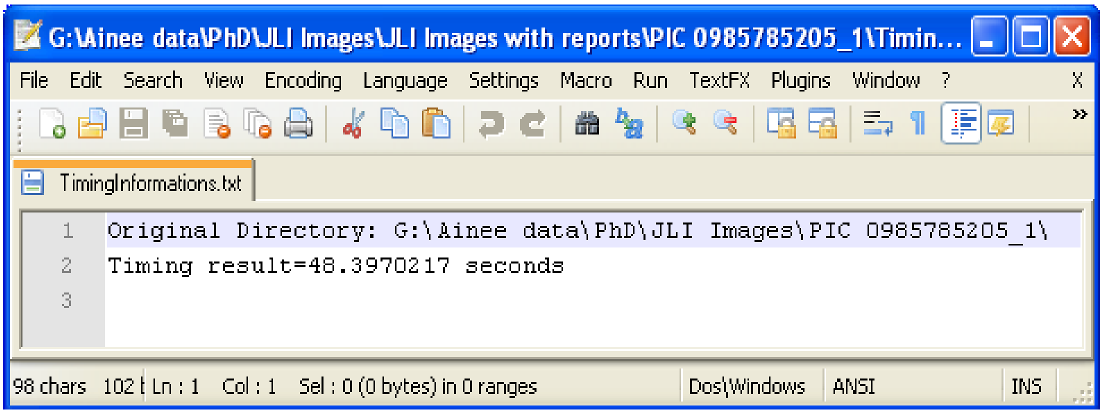
```
Report generated on 17/05/2013 10:08:21
0000000_1 15-09-2009 at 04.00.28.jpg Wire Rolled In Sc
0000261_1 15-09-2009 at 04.00.28.jpg Wire Rolled In Sc
0000523_1 15-09-2009 at 04.00.28.jpg Wire Rolled In Sc
0000786_1 15-09-2009 at 04.00.28.jpg Wire Rolled In Sc
0001048_1 15-09-2009 at 04.00.28.jpg Wire
0001310_1 15-09-2009 at 04.00.28.jpg Wire
0001572_1 15-09-2009 at 04.00.29.jpg Wire
0002096_1 15-09-2009 at 04.00.29.jpg Wire
0002882_1 15-09-2009 at 04.00.29.jpg Wire
0003668_1 15-09-2009 at 04.00.29.jpg Wire
0004192_1 15-09-2009 at 04.00.29.jpg Wire
0004454_1 15-09-2009 at 04.00.29.jpg Wire
0004978_1 15-09-2009 at 04.00.29.jpg Wire
0005240_1 15-09-2009 at 04.00.29.jpg Wire
0008384_1 15-09-2009 at 04.00.30.jpg Wire
0017033_1 15-09-2009 at 04.00.32.jpg Wire
0045864_1 15-09-2009 at 04.00.45.jpg Wire
0067340_1 15-09-2009 at 04.00.59.jpg  Rolled In Scrap
0099045_1 15-09-2009 at 04.01.06.jpg Wire
0107708_1 15-09-2009 at 04.01.07.jpg Wire
0107954_1 15-09-2009 at 04.01.07.jpg Wire
0116338_1 15-09-2009 at 04.01.08.jpg  Rolled In Scrap
0116601_1 15-09-2009 at 04.01.08.jpg Wire Rolled In Sc
0116870_1 15-09-2009 at 04.01.08.jpg Wire Rolled In Sc
```

Figure 7.1: A typical report generated by the inspection software.

It can be seen in the report shown in Figure 7.1 that complete name of the rail image is saved against the name of defect \ defects detected. The name of rail signifies the location of that image from the rail head, also the date and time at which the image was taken. Hence, with the name in the report, the defective part of the rail can be reached physically and examine easily.

7.4.2 Timing Information

The time between each rail while in heavy production is more than 1 minute and 30 seconds, making the software capable of analysing rails in real-time. Hence, one complete folder of images, which is for one complete rail, has to be examined by the software with in 1 minute and 30 seconds of time. The average software running time for a rail of on average 440 images per camera view is less than 50 seconds. Hence, making the software capable of analysing rails in real-time. Every time the rail is processed its timing information is saved in a text file, in the same folder. The screen shot of timing information file has been shown in Figure 7.2. The file stores the location of the folder that have been processed and time taken by the software to process the folder in seconds.



```
Original Directory: G:\Ainee data\PhD\JLI Images\PIC 0985785205_1\
Timing result=48.3970217 seconds
```

Figure 7.2: Screen shot of text file containing the Timing information

The timing information for the processing of several rails has been grouped together and shown in *Table 7.2*. The average processing time as calculated from the values in the table below is almost 46.23 seconds, after processing almost 40 rails, which is enough time to process a complete rail before the next rail images come in for inspection.

Table 7.2: Processing time for a complete rail in production

	Directory Name	**Processing Timing**
1	PIC 0985785101_1	38.02
2	PIC 0985785104_1	47.15
3	PIC 0985785105_1	49.50
4	PIC 0985785201_1	48.14
5	PIC 0985785204_1	46.43
6	PIC 0985785205_1	48.1
7	PIC 0985785206_1	47.14
8	PIC 0985785301_1	46.35
9	PIC 0985785302_1	10.78
10	PIC 0985785303_1	46.85
11	PIC 0985785401_1	48.05
12	PIC 0985785402_1	44.12
13	PIC 0985785403_1	50.90
14	PIC 0985785404_1	46.67
15	PIC 0985785501_1	45.55
16	PIC 0985785504_1	49.12
17	PIC 0985785601_1	46.36
18	PIC 0985785604_1	50.45
19	PIC 0985828101_1	46.48
20	PIC 0985828103_1	47.93
21	PIC 0985828104_1	49.63
22	PIC 0985828201_1	47.20
23	PIC 0985828203_1	35.74
24	PIC 0985828301_1	45.84
25	PIC 0985828401_1	47.11
26	PIC 0985828404_1	44.53
27	PIC 0985828501_1	46.66
28	PIC 0985828504_1	46.19
29	PIC 0985828601_1	45.20
30	PIC 0985910105_1	53.18
31	PIC 0985910106_1	49.55
32	PIC 0985910108_1	49.41
33	PIC 0985910204_1	49.76
34	PIC 0985910302_1	49.71
35	PIC 0985910303_1	46.64
36	PIC 0985910506_1	48.86
37	PIC 0985910603_1	49.96
	Total Images	**Average Time**
	99,500	**46.2255 Sec**

7.4.3 Success Rate of Detection Algorithms

Table 7.3 is an overview of the algorithm's defect detection efficiently when applied on large number of rail image data. As the results of the detection are to be assessed and sorted by a technician, a small amount of false positive is tolerable. Each rail to be examined has an average length of 110m, with on average 440 images per camera view. There are 6 cameras in the installed JLI system; which makes total of almost 2700 rail images to be examined per rail. JLI system stores images for each rail in separate folders using specific naming convention explained in section 3.4. For every folder of images, covering complete length of a rail, a report is generated and saved in the same folder and the list of detected defects is also displayed on the screen. The software has been tested on more than 40 complete rails; which make more than 100,000 images in number that have been tested by the software. The defect percentage is found to be 1.22% that implies that almost 89.9% of the rails processed are found to be clear and only 1.2% of the total processed rail images were marked as defective by the inspection software. Moreover, this data were validated by experts at Tata over several iterations of testing.

Table 7.3: Occurrences of defects detected by the software after processing large number of rail images.

Total Images Processed	**Occurrence of RIS**	**Occurrence of Line on Top**	**Occurrence of Tiger Stripe**	**Occurrence of Wire**	**Defect Percentage (%)**
105,600	207	42	644	399	1.22 %

The rails that are marked as defected by the software are then inspected by an inspector who is an expert on rail defects. The inspector physically examines the site of defects and approves or rejects their occurrences. The Table 7.4 represents the detection success rates of the algorithms in the form of Receiver Operation Characteristic (ROC) table. It was very difficult and time consuming to examine every image individually in the huge data of more than 100, 000 rail images available. Hence, a small data set of images for each defect has randomly been selected from the large number of rail images available. That small sub-data has been used to get the percentage values represented in the ROC table below. This has been done to get the overall performance overview of the developed rail inspection algorithm.

Table 7.4: Algorithm detection accuracy represented by Receiver Operation Characteristic (ROC) table

Type of defect	**True Positive (%)**	**False Positive (%)**	**True Negative (%)**	**False Negative (%)**
Rolled in scrap	99	1	99	1
Line on top	92	8	96	4
Tiger stripe	80	20	79	21
Wire defect	95	5	97	3

A True-positive means that the detection made by the algorithm, when examined physically was found to be a defect. Hence, it shows the correct and successful working of the algorithm.

A False positive means a false alarm, i.e. the algorithm detected and classified something as a defect which actually is not a defect. It may also be referred as over detection.

A True-negative means correct rejection that is the rail has been classified as clear (without any defect) by the algorithm, i.e. no occurrence of a defect under investigation. Hence, shows successful working of the algorithm.

A False-negative means that the rail is classified as clear by the algorithm however, when examined physically was found to have that type of defect on it. In more simple terms false negative means a defect missed by the algorithm.

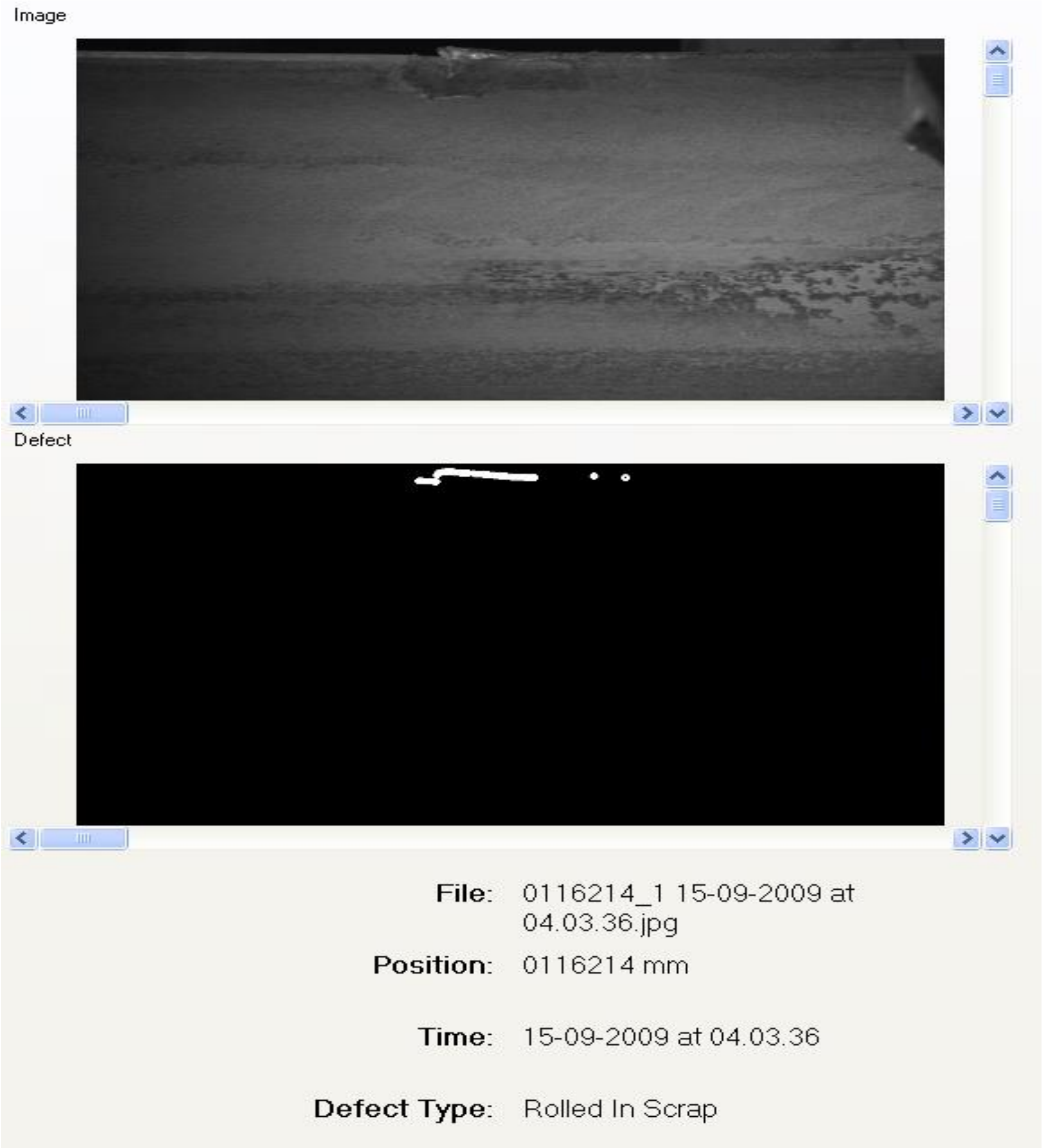
7.4.4 Rolled in Scrap Detection Algorithm

The rolled in scrap detection rate is highly accurate, almost all the defects that were present on the set of images submitted by Tata Steel were successfully detected. Rolled in scrap in the ROC Table 7.4 shows 99% of true positive, which implies that

99% of the time the alarm raised by the software for the occurrence of RIS is correct and the defect exists there when examined physically. There is however a small amount of false positive which is as low as 1% due to flying dust incorrectly detected as faults. The inspection report generated by the software after processing the rail PIC 0985785404_1 has been shown in Table 7.5. The rail image with Rolled in scrap type defect, highlighted in red in the Table 7.5, has been shown in Figure 7.3 which is a screen shot from actual software interface.

Table 7.5: Inspection report generated by the software for the rail PIC 0985785404_1.

No.	Image Name	Detected Defect Type
1	0000000_1 15-09-2009 at 04.02.56.jpg	Wire Defect
2	0000262_1 15-09-2009 at 04.02.56.jpg	Rolled In Scrap
3	0000787_1 15-09-2009 at 04.02.56.jpg	Wire Defect
4	0001049_1 15-09-2009 at 04.02.56.jpg	Wire Defect
5	0001311_1 15-09-2009 at 04.02.56.jpg	Wire Defect
6	0001574_1 15-09-2009 at 04.02.56.jpg	Rolled In Scrap
7	0001836_1 15-09-2009 at 04.02.56.jpg	Wire Defect
8	0002098_1 15-09-2009 at 04.02.56.jpg	Wire Defect
9	0003148_1 15-09-2009 at 04.02.56.jpg	Wire Defect
10	0004459_1 15-09-2009 at 04.02.57.jpg	Wire Defect
11	0018625_1 15-09-2009 at 04.03.00.jpg	Wire Defect
12	0107295_1 15-09-2009 at 04.03.34.jpg	Wire Defect
13	0107557_1 15-09-2009 at 04.03.34.jpg	Wire Defect
14	0116214_1 15-09-2009 at 04.03.36.jpg	Rolled In Scrap
15	0116477_1 15-09-2009 at 04.03.36.jpg	Rolled In Scrap
16	0116739_1 15-09-2009 at 04.03.36.jpg	Wire Defect



File: 0116214_1 15-09-2009 at 04.03.36.jpg

Position: 0116214 mm

Time: 15-09-2009 at 04.03.36

Defect Type: Rolled In Scrap

Figure 7.3: Screen shot from the software interface for Rolled in scrap detection.

The software generates report containing list of images with a defect type in the form of a table. Each defective image in the list can be navigated and original image along with the processed image with defect marked; can be viewed by the user as seen in the screen shot presented in Figure 7.3. Output also displays the image name, position of that occurrence of a particular defect from the start of the rail, time at which the image was taken and type of defect found on that image on the rail.

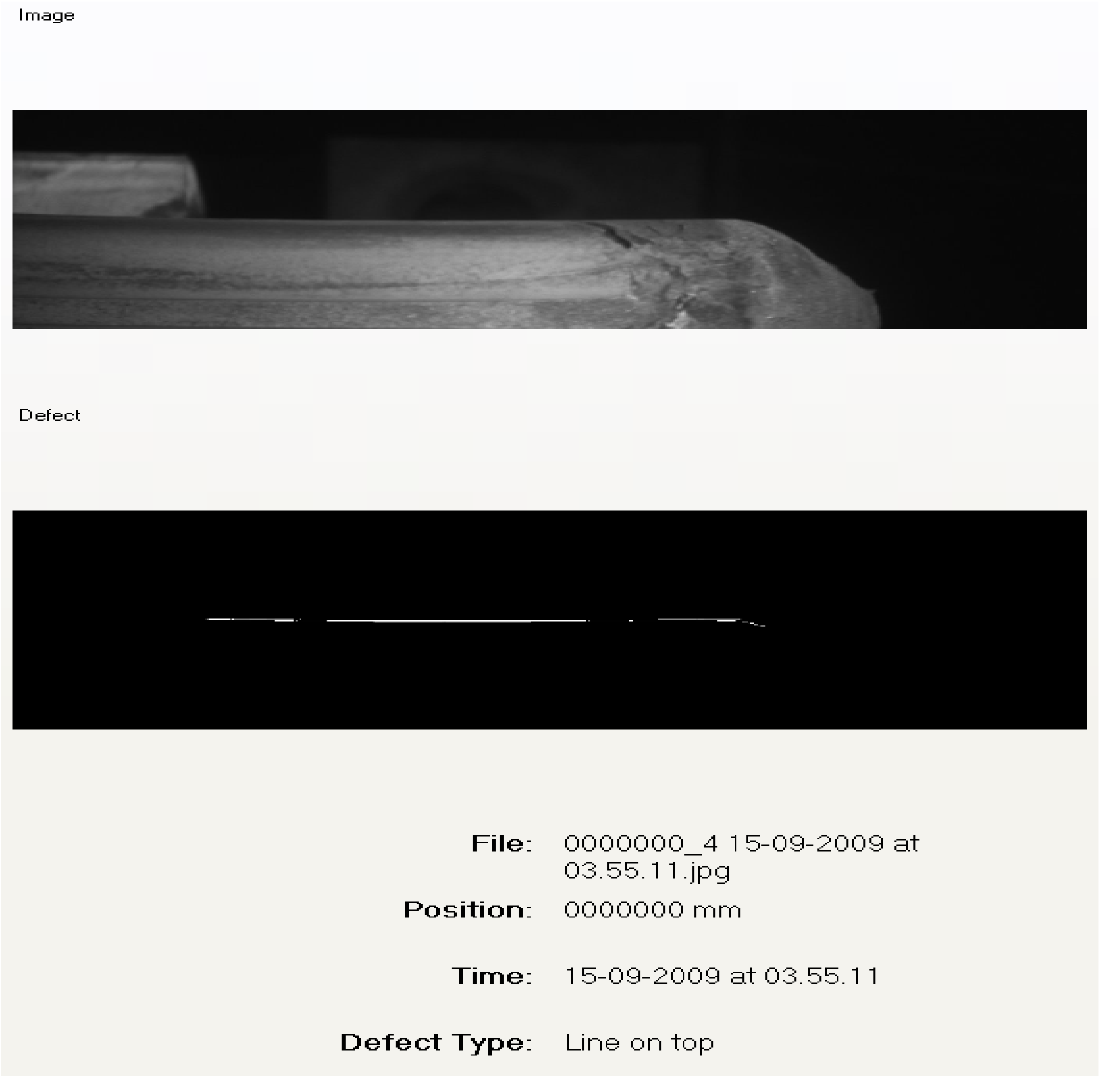
7.4.5 Line on the Top of Rail Detection Algorithm

The line on the top of rail's detection accuracy table shows 92% true positive.

However, 4% false negative detection as represented in Table 7.4, are mostly the faint lines. Those faint lines according to experts are not as critical as the deep appearing defects, successfully detected by the algorithm. The inspection report generated by the software after processing the rail PIC 0985785206_1 has been shown in Table 7.6. The rail image detected, having line on the top of rail type defect, by the software has been highlighted in red in Table 7.6 below. The representative image displayed by the software's has been shown by a screen shot presented in Figure 7.4.

Table 7.6: Inspection report generated by the software for the rail PIC 0985785206_1.

No.	Image Name	Detected Defect Type
1	0000000_1 15-09-2009 at 03.55.13.jpg	Wire Defect
2	0000262_1 15-09-2009 at 03.55.13.jpg	Rolled In Scrap
3	0000524_1 15-09-2009 at 03.55.13.jpg	Rolled In Scrap
4	0000786_1 15-09-2009 at 03.55.13.jpg	Wire Defect
5	0001048_1 15-09-2009 at 03.55.13.jpg	Rolled In Scrap
6	0001310_1 15-09-2009 at 03.55.13.jpg	Rolled In Scrap
7	0001572_1 15-09-2009 at 03.55.14.jpg	Wire Defect
8	0002620_1 15-09-2009 at 03.55.14.jpg	Wire Defect
9	0002882_1 15-09-2009 at 03.55.14.jpg	Wire Defect
10	0003144_1 15-09-2009 at 03.55.14.jpg	Wire Defect
11	0003406_1 15-09-2009 at 03.55.14.jpg	Wire Defect
12	0003942_1 15-09-2009 at 03.55.14.jpg	Wire Defect
13	0004192_1 15-09-2009 at 03.55.14.jpg	Wire Defect
14	0004454_1 15-09-2009 at 03.55.14.jpg	Wire Defect
15	0004716_1 15-09-2009 at 03.55.14.jpg	Wire Defect
16	0102713_1 15-09-2009 at 03.55.51.jpg	Wire Defect
17	0116862_1 15-09-2009 at 03.55.53.jpg	Wire Defect
18	0000000_4 15-09-2009 at 03.55.11.jpg	Line on top



File: 0000000_4 15-09-2009 at 03.55.11.jpg

Position: 0000000 mm

Time: 15-09-2009 at 03.55.11

Defect Type: Line on top

Figure 7.4: Screen shot from the software interface for Line on the top of rail type defect detected.

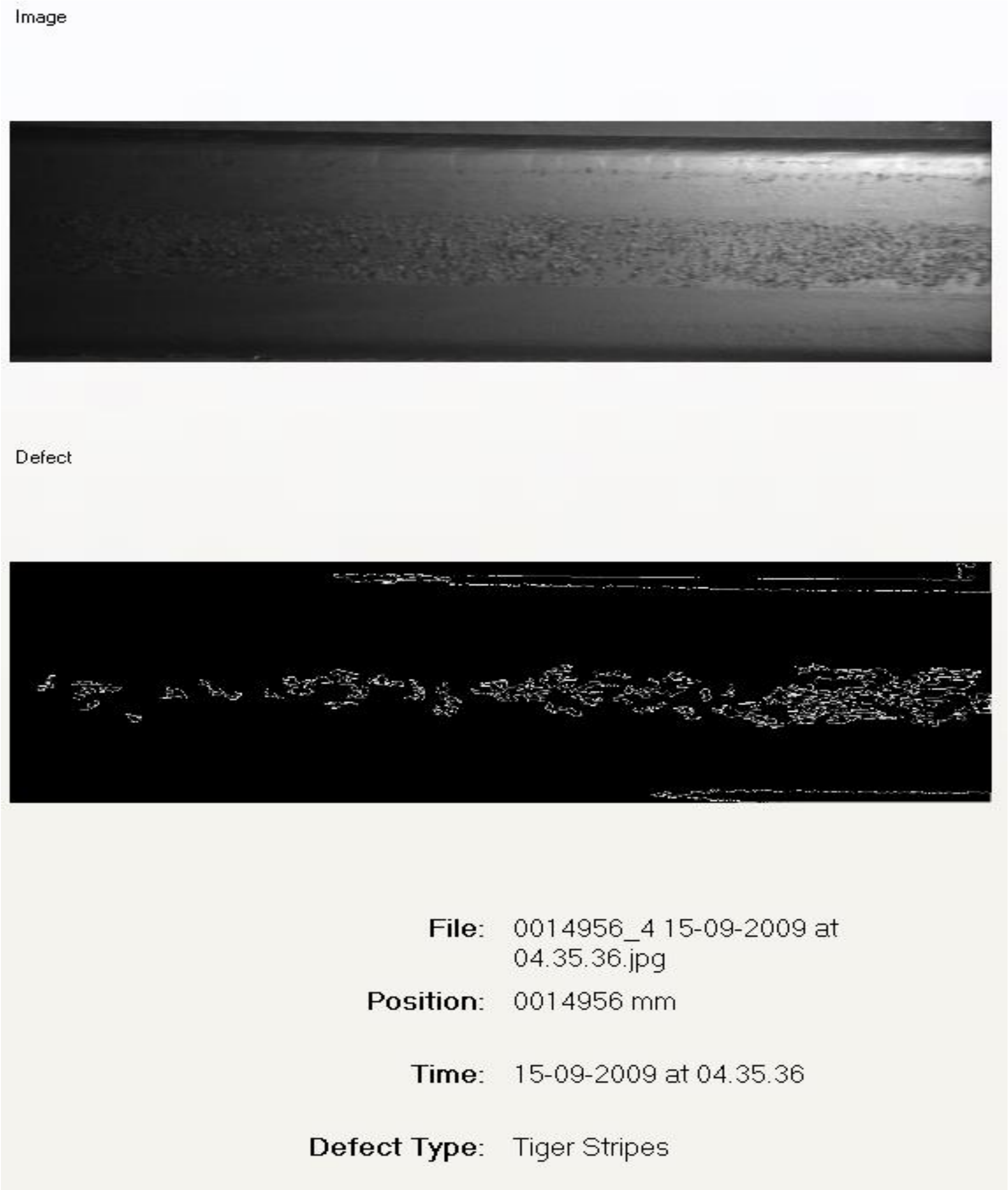
Image name, position of the rail where the defect is found in meters, time at which image was taken and type of defect detected has been displayed in the output interface to facilitate the user with maximum information.

7.4.6 Tiger Stripes Detection Algorithm

The tiger stripes detection accuracy Table 7.4 shows 85% true positive, however, there are some cases where the stripes are faint and small where the algorithm fails and are still critical to the quality of the rail itself. Therefore, false negative is comparatively high for this type of defect which is 20%. The inspection report generated by the software for PIC 0985828101_1 has been shown in Table 7.7. The image of the rail detected to be having tiger strips type defect, has been shown as highlighted in red in the said table below. The corresponding representative image displayed by the software's has been shown by a screen shot presented in Figure 7.5.

Table 7.7: Inspection report generated by the software for the rail PIC 0985828101_1

No.	**Image Name**	**Detected Defect Type**
1	0000000_1 15-09-2009 at 04.35.34.jpg	Wire Defect
2	0000261_1 15-09-2009 at 04.35.35.jpg	Rolled In Scrap
3	0000523_1 15-09-2009 at 04.35.35.jpg	Rolled In Scrap
4	0000785_1 15-09-2009 at 04.35.35.jpg	Wire Defect
5	0002881_1 15-09-2009 at 04.35.35.jpg	Wire Defect
6	0007601_1 15-09-2009 at 04.35.36.jpg	Wire Defect
7	0013902_1 15-09-2009 at 04.35.38.jpg	Wire Defect
8	0116811_1 15-09-2009 at 04.36.15.jpg	Rolled In Scrap
9	0117075_1 15-09-2009 at 04.36.15.jpg	Rolled In Scrap
10	0117336_1 15-09-2009 at 04.36.15.jpg	Rolled In Scrap
11	0000261_4 15-09-2009 at 04.35.33.jpg	Tiger Stripes
12	0000523_4 15-09-2009 at 04.35.33.jpg	Tiger Stripes
13	0007341_4 15-09-2009 at 04.35.35.jpg	Tiger Stripes
14	0008392_4 15-09-2009 at 04.35.35.jpg	Tiger Stripes
15	0011547_4 15-09-2009 at 04.35.36.jpg	Tiger Stripes
16	0013909_4 15-09-2009 at 04.35.36.jpg	Tiger Stripes
17	0014956_4 15-09-2009 at 04.35.36.jpg	Tiger Stripes
18	0016006_4 15-09-2009 at 04.35.37.jpg	Tiger Stripes
19	0017056_4 15-09-2009 at 04.35.37.jpg	Tiger Stripes
20	0018107_4 15-09-2009 at 04.35.37.jpg	Tiger Stripes
21	0019157_4 15-09-2009 at 04.35.37.jpg	Tiger Stripes
22	0019419_4 15-09-2009 at 04.35.37.jpg	Tiger Stripes
23	0038849_4 15-09-2009 at 04.35.42.jpg	Tiger Stripes



File: 0014956_4 15-09-2009 at 04.35.36.jpg

Position: 0014956 mm

Time: 15-09-2009 at 04.35.36

Defect Type: Tiger Stripes

Figure 7.5: Screen shot from the software interface for Tiger Stripe type defect detected.

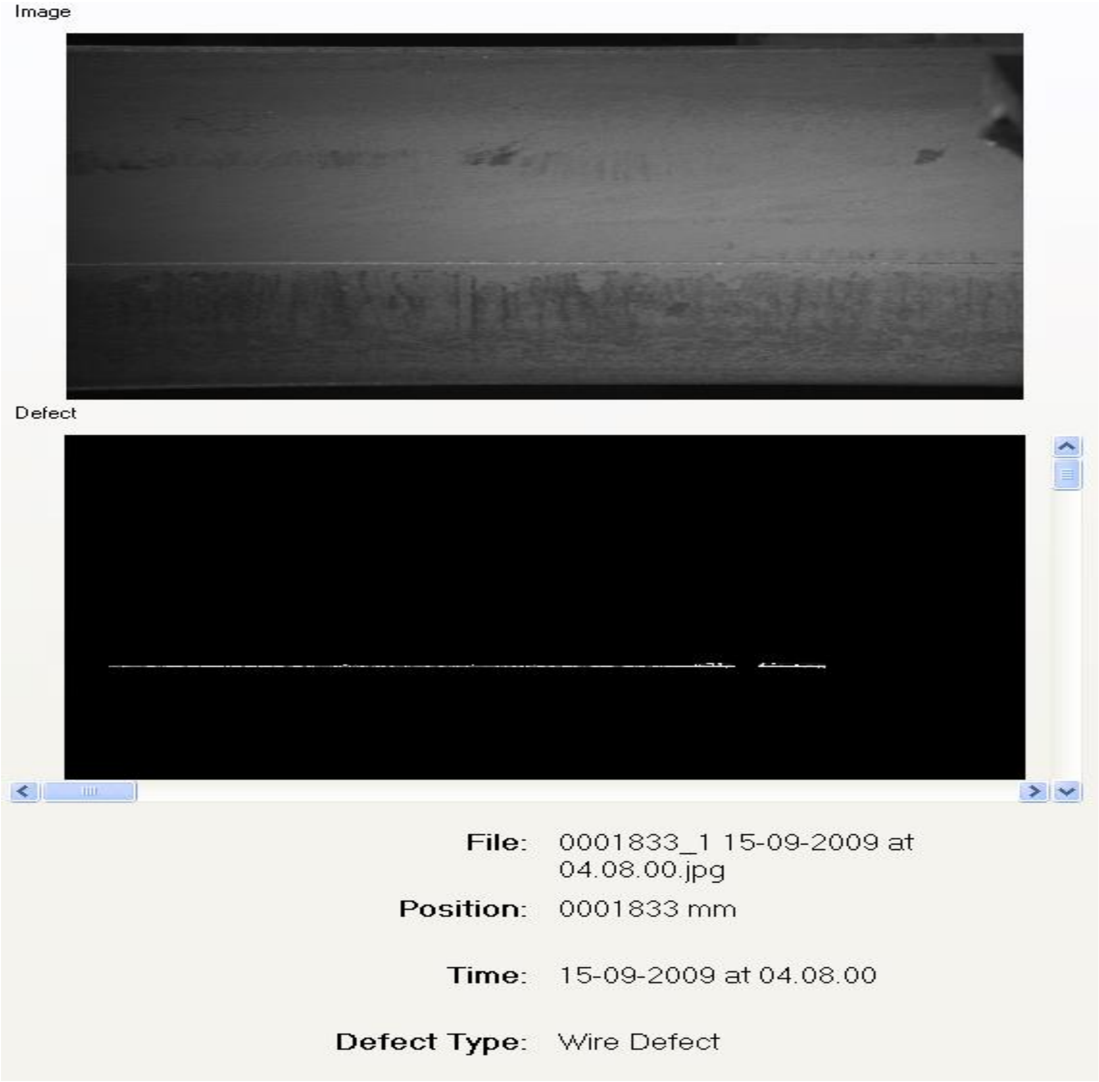
Same as other example images presented for previous defects, image name, position of the defective rail part, and time at which image was taken and type of defect detected for tiger stripes; can also be seen in the output generated by the software; which has been shown in Figure 7.5 given above.

7.4.7 Wire Defect Detection Algorithm

The algorithm for wire defect has again a high detection accuracy rate and detects almost all the occurrence of wires, with 5% of false positive due to occasional water drops that are incorrectly labelled as being faults, but that can easily be assessed by a post-viewing as being safe. These values have been given in Table 7.4 in the previous section of success rate of detection algorithm. There is also small percentage (3%) of false negative which occurs occasionally due to very poor lighting condition on that particular image. The inspection report generated by the software for PIC 0985785104_1 has been shown in Table 7.8. The image of the rail defected by wire type defect, has been shown as highlighted in red in the given Table 7.8 below. The corresponding representative image displayed by the software has been shown by a screen shot provided in Figure 7.6.

Table 7.8: Inspection report generated by the software for the rail PIC 0985785104_1

No.	Image Name	Detected Defect Type
1	0000523_1 15-09-2009 at 04.07.59.jpg	Rolled In Scrap
2	0001047_1 15-09-2009 at 04.07.59.jpg	Wire Defect
3	0001309_1 15-09-2009 at 04.07.59.jpg	Wire Defect
4	0001571_1 15-09-2009 at 04.08.00.jpg	Wire Defect
5	0001833_1 15-09-2009 at 04.08.00.jpg	Wire Defect
6	0002095_1 15-09-2009 at 04.08.00.jpg	Wire Defect
7	0002357_1 15-09-2009 at 04.08.00.jpg	Wire Defect
8	0003143_1 15-09-2009 at 04.08.00.jpg	Wire Defect
9	0003405_1 15-09-2009 at 04.08.00.jpg	Wire Defect
10	0023838_1 15-09-2009 at 04.08.05.jpg	Wire Defect
11	0108190_1 15-09-2009 at 04.08.40.jpg	Wire Defect
12	0116573_1 15-09-2009 at 04.08.42.jpg	Rolled In Scrap
13	0116836_1 15-09-2009 at 04.08.42.jpg	Rolled In Scrap
14	0117099_1 15-09-2009 at 04.08.42.jpg	Rolled In Scrap



File: 0001833_1 15-09-2009 at 04.08.00.jpg

Position: 0001833 mm

Time: 15-09-2009 at 04.08.00

Defect Type: Wire Defect

Figure 7.6: Screen shot from the software interface for Wire defect detected.

Once again for the wire defect; image name, position of the defective rail part, and time at which image was taken and type of defect detected can be seen in the output display generated by the software. The screen shot has been provided in the Figure 7.6 given above.

7.5 Method 2 - Principal Component Analysis (PCA) Method

PCA based defect detection method is so far in its earlier stages. The method has been tested for individual defects and results for few of the defects have been found quite successful and encouraging. However, the testing for the complete rail has still not been fully implemented. The results obtained from method-1 are quite efficient and accurate and currently have been installed at the Scunthorpe site and is use in very initial testing stages. However, the need for another method was to develop a single adaptable, trainable and efficient detection method for the detection of all the defects being investigated and specifically to improve the detection of tiger stripes type defect, whose detection success rate was comparatively as low as 80% as shown by the Table 7.4; which is mainly due to largely varying size of tiger stripes patterns on different rail images.

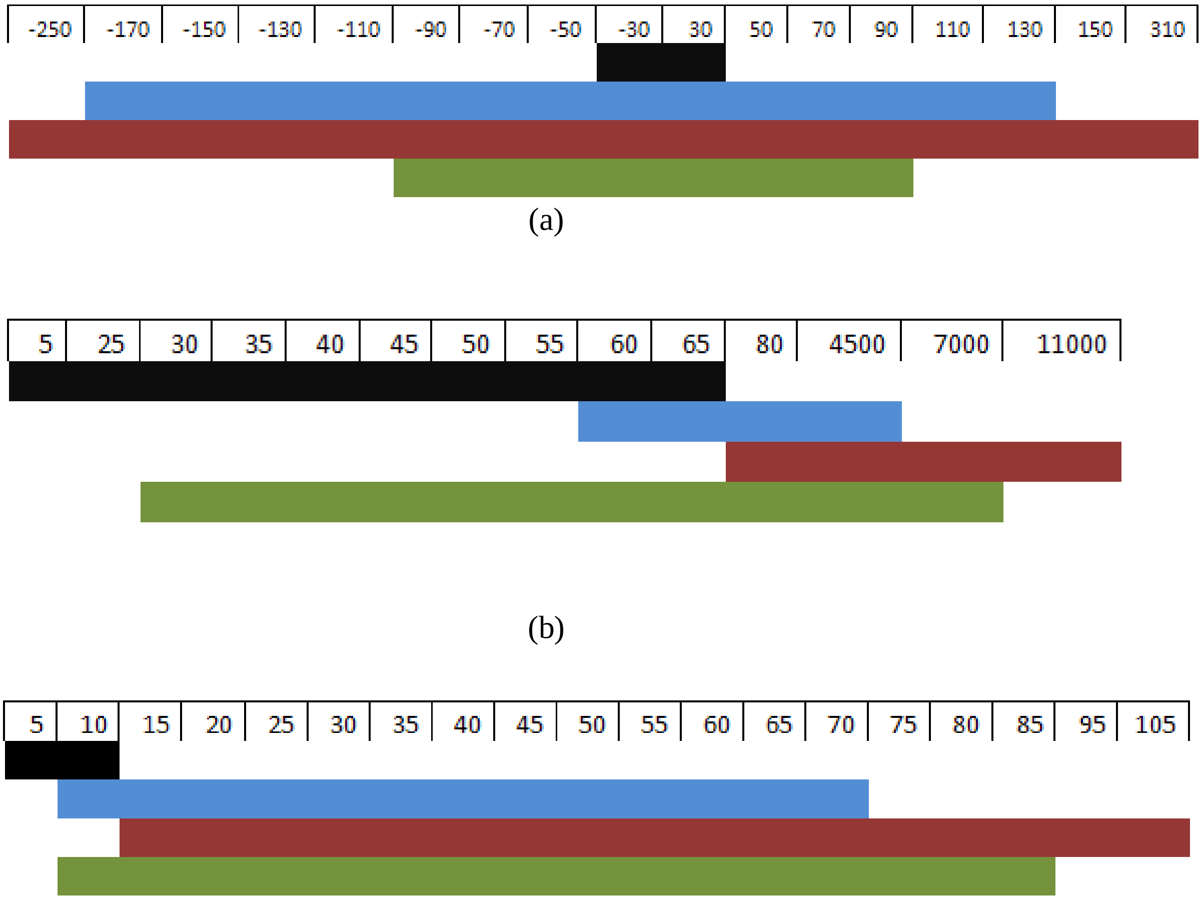
Method-2 is based on PCA features and their statistics computation followed by the classification of those features and statistics using back propagation neural networks. The method has been explained in detail in chapter 6. This section covers the discussion of the results produced by method-2, their evaluation and analysis; for each type of the rail defect under investigation.

7.5.1 Statistical Analysis of PCA Features

First phase of PCA method was to analyze PCA statistical profiles of the defects and clear areas of the rail. A small window mask of variable size slides over the whole image and PCA features of area covered by the mask at each location are calculated. Similarly, sliding the window over the image again; statistics such as standard deviation and variance are computed for each location. The step and window size can be defined by the user. The statistics can be projected to have an overview of the results and values can also be plotted to see the trend being followed. The statistics obtained for defective and clear areas of the rail have been compared, plotted and discussed in detail in section 6.5 of the report.

The respective ranges obtained from the statistics presented in Table 6.1 in the previous chapter have been drawn using colour bands in the Figure 7.7 below. Defects and clear rails have been represented by specific colours to clearly show the ranges obtained for PCA projection (PCA Features) values, standard deviation between those values and also their variance.

⬛	Normal Rail
🟦	Wire Defect
🟥	Tiger Stripes
🟩	Line on the Top



(a)

(b)

(c)

Figure 7.7: Statistics ranges of PCA features represented by colour band. (a) PCA features range for clear rail against wire, line on the top and tiger stripes type defects. (b) PCA Variance range for clear rail against wire, line on the top and tiger stripes type defects. (c) PCA Standard Deviation range for clear rail against wire, line on the top and tiger stripes type defects.

The range bands for PCA features shown in Figure 7.7 (a) shows that the upper and lower cut-offs of PCA features for defects and clear rail vary distinctively. Hence, from the bands provided above in Figure 7.7, it is clear that variance and standard deviation between the PCA features for defects and clear rails shall be the key factors for their detection.

Standard deviation values acquired for the PCA projection images represent the most of the defects distinctively. The resulting images have been presented and discussed in section 6.6 of the previous chapter. The images obtained after displaying the statistical values obtained from PCA projection images are quite successful in defect detection process, and clearly shows the defects occurrence. However, some noise has also been picked up in the images obtained as visible in the images presented in the figures 6.8, 6.9, 6.10 and 6.11 of the previous chapter. Hence, classification of PCA statistics has been tested further using neural networks to improve the results obtained up till that point.

7.5.2 Neural Network Classification of PCA Features and Statistics

Artificial Neural Network is trained on PCA features and their statistics to perform the classification of the rail defects. The classification method has so far been tested for tiger stripes and RIS type defect.

The detection success rate for tiger stripes algorithm by method 1 is comparatively low. The main issue with previous method was greatly varying size of the tiger stripes. They can be very small at some occurrences and very large at others. Hence, the requirement is to make the adaptable algorithm which is sensitive to different sizes of the defect to make the detection accurate and more efficient.

A small set of images have been processed using PCA detection method. The results obtained for tiger stripes are precise and better than the ones produced by method-1.

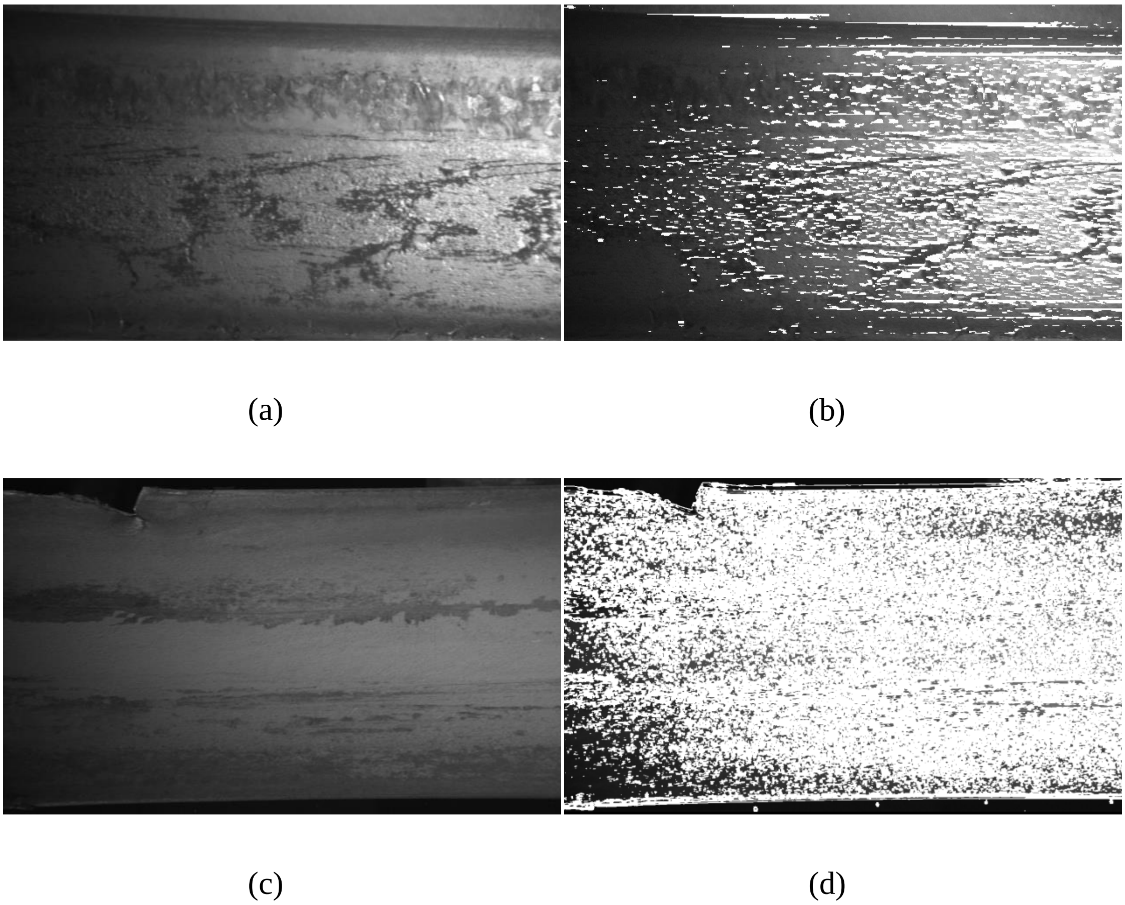
The method has also been tested for RIS type defect and results are encouraging but not as accurate as method 1. The results for RIS produced by method 1 were efficient with 99% of true positive as can be seen in Table 7.4.

These results have been produced by calculating PCA features and their statistics followed by the classification of those statistics by artificial neural networks. The method has been explained in detail in chapter 6. The success rate of method-2 for tiger stripes and RIS has been presented in Table 7.9.

Table 7.9: PCA method accuracy represented by Receiver Operation Characteristic (ROC) for tiger stripes type defect.

Type of defect	**True Positive (%)**	**False Positive (%)**	**True Negative (%)**	**False Negative (%)**
Tiger Stripes	95	5	99	1
RIS	90	10	80	20

It is clear from the table that this method is better than method-1 for tiger stripes, however RIS detection is good but not accurate as method-1.



(a) (b)

(c) (d)

Figure 7.8: Result of defect classification by PCA based neural network method. (a) Original rail image with tiger stripes (b) Tiger stripes detected by neural networks trained on standard deviation of PCA features using a window size of 6 (c) Original rail image with rolled in scrap defect (d) Rolled in scrap type defect detected by neural networks trained on standard deviation of PCA features using a window size of 6.

The resulted images produced by the classification of tiger stripes and RIS type defects; using neural networks has been presented in Figure 7.8. It can be seen from the above figure that the neural network classification for tiger stripes is quite good and for RIS, the background has been separated from the rail very efficiently. The irregularity in the edges obtained is finally classified as RIS defect.

7.6 Discussion

This chapter has described an objective performance evaluation of the detection techniques defined for the rail defects, being investigated in this research work.

Different existing imaging techniques have been tested, used and adapted at several detection stages. Those techniques have been compared in the following section to get the brief overview of the extensive research work that has been carried out.

7.6.1 Detection Algorithm Comparison

Different methods and image techniques have been exhaustively tested on the defects. Details of the extra work and testing have been given in Appendix A, B and C in the end of the report. However, generalised table has been developed to summarize the work. The summary of general imaging methods used and their contribution to the detection procedures have been summed up in Table 7.10. Table 7.10 presents the list of existing image processing techniques used at various stages of the detection procedures that have been developed for the rail inspection. The table also sums up the advantages of those imaging techniques.

Table 7.10: List of existing image techniques used and their advantages

Image Techniques & Algorithms	**Advantages**
Image Filtering	• Enhancement \ Equalization: improves contrast and hence adjust lighting in the rail images • Smoothing: remove noises from the rail images • Template matching: detects known patterns
Image Binarization	• Gives the images that have only one sample per pixel, hence makes processing easy • Help to see what areas of an image consist of pixels whose values lie within a specified range
Edge Detectors	• Helps to extract the areas of interest \ edges – defects in the current work. • Robinson and Susan Edge Detector are tested and used at various stages of detection.
Morphological Operation	• Enhances the defects by dilating \ erosion operation • Cleans up the images by reducing the back ground noise.
Principal Component Analysis (PCA)	• PCA is a method of identifying similarities or differences in a data • PCA reduces dimensionality of data, retaining the useful information.
Artificial Neural Networks (ANN)	• Advanced technique of computer programming; that is efficient at pattern recognition and solving other such problems

7.7 Conclusion

All the results produced by the detection methods developed have been presented, discussed and evaluated in detail in this chapter.

Method 1 that has further four different methods for the detection of each of the defect under investigation has been developed. Individual techniques of method 1 are a combination of customised existing image algorithms and newly developed algorithms; put together to perform the defect detection. The developed algorithms are efficient and reliable. The system has been handed over to the TATA steel group and is in initial testing phase before the proper installation is to be done in near future.

Method 2 explains alternate rail inspection method to the method 1. The alternate method is an attempt on using Principal Component Analysis (PCA) to devise an adaptable and trainable single technique for classification of all defect types being investigated. The method is based on classifying PCA features using artificial neural networks.

The performance of both the methods has been evaluated and discussed in detail in this chapter. The efficient performance of these novel techniques is evident by the resulting image, reports, table and timing information obtained by processing large number of image data.

The next chapter concludes the report with a brief summary of the research work done and also accounts the future enhancement and scope of the project developed.

8 Conclusion

8.1 Introduction

This chapter will give an overall summary of all work done for this project. It will summarise the major developments done in the research and make conclusions on the basis of the results presented in chapter 7.

Typical defects on rail in production line have been studied and characterised. The defects presented cover more than 95% of the defects that occur on the rail surface while in production. All the defects are considered critical by the customers and adequate action must be taken to avoid defective rails to be dispatched.

Human inspection has the drawbacks firstly of being subject to human error in the form of tiredness and secondly of being too slow for an exhaustive inspection of all the rails produced. Automatic inspection software has been developed to complement the human inspection by pinpointing the suspected defects in rail and to perform an exhaustive inspection.

8.2 Analytical Outcome

Standard image processing algorithms have been used as well as specially developed methods in order to design an automatic defect detection tool on images acquired with the JLI system at Tata Steel. Validation of the methods has shown that good results have been obtained and can be improved further in the case of PCA and Neural Networks techniques.

8.2.1 Rolled in Scrap

The rolled in scrap detection rate is perfect, all the defects that were present on the set of images submitted by Tata Steel were successfully detected, with a small amount of false positive due to flying dust incorrectly detected as faults. The algorithm developed detected 99% of the defects which shows the high success of it. Sometimes, due to uneven lighting and dust the background is not very clear from the rail that makes the detection difficult. The detection algorithm used the idea of separating rail edge from the background and then calculating unevenness in the rail edge, hence, better lighting conditions can clear the edge difference. A request has also been made to use the diffuse light instead of spot lights and also to improve the lighting condition to get the better view. If this happens a lot of pre-processing will be reduced and better detection will be possible.

8.2.2 Line on Top

The line on top detection rate is of 92%, the 8% not detected are faint lines that according to expert are not as critical as the deep appearing defects detected by the algorithm. It has been concluded from the results that written line algorithm is working fine. The only problem is with the edge detection, it missed the faint lines \ edges. Hence, the final detection is not 100%. Any other edge detector or different mask can be applied to improve the detection. However, rest of the steps and detection procedures are successful and producing good results.

8.2.3 Tiger Stripe

The tiger stripe detection rate is 80%, while the algorithm successfully detects all type of tiger stripes as presented in the table above, there are some cases where the stripes are faint and small where the algorithm fails and are still critical to the quality of the rail itself. A different approach has been suggested to detect this type of defect and is currently under implementation and testing. Nonetheless, these faint stripes are rare in real cases and the high non detection rate observed here is due to the faults oriented images provided by Tata Steel that contained higher than usual defects of this type. More specific and clear images have been received recently and currently under testing. The main issue with the old images was them having very large patterns of tiger stripes. However, the defect size varies greatly. They can be very small and very large. Hence, the requirement is to make the adaptable algorithm which is sensitive to different sizes of the defect. Neural Networks and PCA are currently under consideration to get the better detection rate. The idea is to train the network on different areas of different sized from the defect.

8.2.4 Wire Defect

The algorithm for wire defect has a perfect detection rate of all the wires, with 5% of false positives due to occasional water drop that are incorrectly labelled as being faults, but that can easily be assessed by a post-viewing as being safe. This was the most difficult type to detect. However, smart observation of dark and light lines next to each other made it easy to detect.

8.3 Comparison of Tested and Used Algorithms

This section presents the comparison of the algorithms tested, studied and used for the detection of each defect. There is a long list of image processing methods that have been studied, explored and tested. Moreover, few new detection algorithms have also been developed. *Table 8.1* contains the brief comparison of the algorithms that have been tested and the ones that have been used at different stages of the defect detection.

Table 8.1: Comparison of tested and used algorithms for the defect detection process and their success rate

Tested Image Techniques	**Success Rate**	**Comment**
Image Filtering	95%	It has been used at various detection stages, as it successfully reduced noise and enhanced most of the images.
Fourier Transform	60%	Some of the defects had similar frequencies as the defects. Hence, it didn't help much in removing the noise.
Binarization	100%	It has been used in all the algorithms as it segments the image making the processing simple and easy
Robinson Edge Detectors	90%	Different edge operators were tested and Robinson kernel which is a gradient type has been used, as it gave comparatively high success rate. However, needs improvement as it is picking many irrelevant marks and non-defects as well.
Susan Function	65%	Also picks up extra noise and dust of the back ground. Hence, not a best edge detector for the current problem to be used
Hough Transform	50%	The algorithm picks up the lines in all direction. However, the line on top defect has lines only in horizontal direction.
Developed Straight Line Algorithm	92%	The algorithm detected all the lines in horizontal spread apart from the faint lines missed by the edge detection process.
Morphological Operators	90%	Dilate and erode operators have helped most of the time to extend the edges and remove the non-defect parts.
Template Matching	60%	Template matching separates areas of interests out from the image using a small template or compares the parts of the image against each other
Neural Networks	65%	They are still under consideration and needs to be trained cleverly on large number of defects to increase the detection date. However, the results so far are quite encouraging.
Principal Component Analysis	70%	The process is still under inspection and being tried to be optimised to produce better detection

The imaging methods with high success rates have been used in the developed detection algorithms. While the ones with low success rates such as Hough Transform to detect lines, Susan filter to detect edges and Fourier transform to reduce noise have been replaced by using own developed line detection algorithm, Robinson edge detector and average filers for the time being. However, Neural Networks and PCA which are giving encouraging results are being considered for the future work.

8.4 Future Enhancements and Scope

Due to a large number defects that have to be detected in a short time and the very slow processing of some algorithms using neural networks, many aspects and points have been missed out. Noteworthy, the testing for PCA and Neural Networks based detection method has not been tested for large number of image data due to slow learning of ANN. Also that method 2 has not been yet tested for all the defected being investigated for the current research work. Therefore, the techniques used can be improved by applying some optimised application of artificial neural networks or the combination of both PCA and ANN. Implementation in any other software and finally on hardware can also improve the efficiency of the software developed. Their advance and practical form would definitely be a very great step forward in the field of image processing.

The next step will be to make the implementation of an adaptable and trainable single technique complete and successful for detecting all the types of faults. A system is to be achieved that can learn and adapt itself to the changing types of faults with the time. It is hoped to develop some more generic approach rather than the series of bespoke and tailored algorithms using different new and developing techniques such as neural networks and PCA.

Recent development in the field of neural networks has resulted in a large number of techniques and applications based on them. Neural networks are routinely being used for image processing, machine and computer vision. Back propagation neural networks are very efficient at prediction and classification (segmentation) but they are comparatively slow. There is no explanation about the learning of the network available i.e. a learned network does not contain any information about its learning.

The project has been developed for Tata steel’s Scunthorpe site for now. However, the developed software can be used at various similar sites around the world to replace the manual inspection to save time and money. There is also a proposal of designing a similar system to be used on another setup in Europe, known as Duris system. Apart from rail inspection the software processes rails while in production, hence can also be used to inspect metal surfaces for other industries.

9 Bibliography

Bajwa, I. S., & Hyder, S. I.(2005). PCA based image classification of single-layered cloud types. Proceedings of the IEEE Symposium on Emerging Technologies, 2005., 365–369. doi:10.1109/ICET.2005.1558909

Bernardo, V. et al. (2010). Digital Image Subtraction , Blue Filter , Enhancement (DISBE): A new approach for quantitative immunohistochemical analysis in light microscopy. , (5), pp.828–835.

Berry, A et al. (2008). High Speed Video Inspection of Joint Bars Using Advanced Image Collection and Processing Techniques. , pp.1–13.

Bezryadin, S. (2007). New generation of image editing algorithms. Electronic Imaging 2007, Digital Photography III.

Bezryadin, S., Bourov, P. (2006). Color Coordinate System for Accurate Color Image Editing Software. The International Conference Printing Technology SPb’06 proceedings, p. 145–148, St. Peterburg State University of Technology and Design.

Blackledge, J. M., & Dubovitskiy, D. A. (2008). A Surface Inspection Machine

Vision System that Includes Fractal Texture Analysis, 1–14.

Bonnot, N., Seulin, R., & Merienne, F. (2004). Machine vision system for surface inspection on brushed industrial parts ., 33.

Cannon, D. F., Edel, K.-O., Grassie, S. L., & Sawley, K. (2003). Rail defects: an overview. Fatigue Fracture of Engineering Materials and Structures, 26(10), 865–886.

Cawley, P., Lowe, M. J. S., Alleyne, D. N., Pavlakovic, B. & Wilcox, P. (2003). Practical long range guided wave testing: applications to pipes and rail," Materials Evaluation, vol. 61, no. 1, pp. 66–74.

Cheng, W., Wang, H., & Xu, D. (2008). Image Classification Based on Laplacian PCA, 1616–1619.

Choi, D., Jeon, Y., & Kim, S. W. (2011). Faulty Scarfing Slab Detection Using Machine Vision, (Icmv 2010), 434–437.

Chowdhary, C. L. (2011). Linear feature extraction techniques for object recognition: study of PCA and ICA. Journal of the Serbian Society for Computational Mechanics / Vol. 5/ No.1, 5(1), 19–26.

Clark, R. (2004). Rail flaw detection: overview and needs for future developments. NDT & E International, 37(2), pp.111–118. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0963869503001026 [Accessed November 1, 2009].

Cohen, J. (2000). Visual color and color mixture: the fundamental color space (illustrate.). University of Illinois Press, 2001.

Deutschl, E.; Gasser, C.; Niel, A.; Werschonig, J. (2004). Defect detection on rail surfaces by a vision based system. Intelligent Vehicles Symposium, 2004 IEEE , vol., no., pp. 507- 511, 14-17.

Edwards, J.R. & Hart, J.M., (2009). Advancements in Railroad Track Inspection Using Machine-Vision Technology. Proceedings of American Railway and Maintenance of Way Assocition Annual Conference. Chicago, IL.

Edwards, R. S., Dixon, S. and Jian, X.. (2006) Characterisation of defects in the railhead using ultrasonic surface waves. NDT & E INTERNATIONAL, 39 (6). pp. 468-475. ISSN 0963-8695

El-bakry, H.M., (2006). New Fast Principal Component Analysis for Face Detection. Computational Intelligence, pp.195–201.

Eleyan, A., & Demirel, H. (2005). Face Recognition System Based on PCA and Feedforward Neural Networks. 8th International Work-Conference on Artificial Neural Networks, IWANN 2005, Vilanova i la Geltrú, Barcelona, Spain, June 8-10. Proceedings (pp. 935–942).

Esther Resendiz, Luis . et al. (2010). development of a machine vision system for inspection of railway track components.12th WCTR, Lisbon, Portugal.

Frayman, Y., Zheng, H., & Nahavandi, S. (2005). Machine Vision System for Automatic Inspection of Surface Defects in Aluminum Die Casting, 281–282.

Garg, R., Mittal, B. & Garg, S. (2011). Histogram Equalization Techniques For Image Enhancement. , 7109, pp.107–111.

Gilchrist, A.O. (2006). A history of Engineering research on british Railways. working papers in Railways studies. Institute of Railways Studies and Transport History.

Grassie, S., L. (2005). Rolling contact fatigue on the British railway system: treatment. Wear, vol.258, no.7-8, pp.1310-1318.

Graves, Mark & Bruce G. B. (2003). Machine Vision for the Inspection of Natural Products. Springer. p. 5. ISBN 9781852335250.

Heckel, T., Thomas, H. & Kreutzbruck, M. (2009). High Speed Non-Destructive Rail Testing With Advanced Ultrasound and Eddy-Current Testing Techniques.

Hsieh, H.Y., Chen, N., Liao, C.L. (2007). Visual Recognition System of Elastic Rail Clips for Mass Rapid Transit Systems. JRCICE2007-40080, Proceedings of the ASME/IEEE Joint Rail Conference & Internal Combustion Engine Spring Technical Conference, ASME/IEEE, Pueblo, Colorado.

Innotrack, (2008). D4.4.1 – Rail Inspection Technologies, Available at: http://www.innotrack.net/IMG/pdf/d441.pdf. [Accessed January 15, 2010].

Jie, L., Siwei, L., Qingyong, L., Hanqing, Z. & Shengwei, R. (2009). Real-time Rail Head Surface Defect Detection: a Geometrical Approach, (ISlE), 769–774.

Karhunen, J. & Joutsensalo, J. (1995). Generalizations of Principal Component Analysis , Optimization Problems , and Neural Networks. Science, 8(4), pp.549–562.

Kenderian, S. et al. (2003). Rail track field testing using laser/air hybrid ultrasonic technique,"Materials Evaluation, vol. 61, no. 10, pp. 1129–1133.

Khan, A., & Farooq, H. (2011). Principal Component Analysis-Linear Discriminant Analysis Feature Extractor for Pattern Recognition, 8(6), 267–270.

Li, Y., Otto, C., Haas, N., Fujiki, Y., & Pankanti, S. (2011). Component-Based Track Inspection Using Machine-Vision Technology. Proceedings of the 1st ACM International Conference on Multimedia Retrieval - ICMR ’11, 1–8. D

Limited, R.T.C. (2006). Rail Defects Handbook, Some Rail Defects, their Characteristics, Causes and Control. Engineering Practices Manual, Civil Engineering, A, N. S. W. Department of Railways, ed..

Mandriota, C., Stella, E., Nitti, M., Ancona, N., Distante, A., (2004). Filter-based feature selection for rail defect detection. Machine Vision and Applications.

Mandriota, C., Stella, E., Nitti, M., Ancona, N., Distante, A., Elaborazione, I., & Amendola, V. (2001). Rail corrugation detection by Gabor filtering, (1), 626–628. Image Processing Proceedings, International Conference.

Manne, M. (2009). Machine Vision For Automatic Visual Inspection Of Wooden Railway Sleepers Using Unsupervised Neural Networks. Master’s Thesis Computer Engineering. Borlange, Sweden.

Marino, F. & Stella, E. (2007). ViSyR:a Vision System for Real-Time Infrastructure Inspection. Vision Systems Applications. pp. 111-142.

Martin, D. (2007). A Practical Guide to Machine Vision Lighting.

McNamara, J. and Lanza di Scalea, F., (2002). Air-coupled Ultrasonic Testing of Railroad Rails. Materials Evaluation, Vol. 60(12).

Molleda, J. et al. (2010). Real-time flatness inspection of rolled products based on optical laser triangulation and three-dimensional surface reconstruction. Journal of Electronic Imaging, 19(3), p.031206. Available at: http://link.aip.org/link/JEIME5/v19/i3/p031206/s1&Agg=doi [Accessed September 15, 2011].

Nejikovsky, B. et al. (2005). Automated Joint Bar Inspection Using High Speed Cameras.washington: American Railway Engineering and Maintenance-of-Way Association.

Network Rail, (2010). Britain relies on rail. Retrieved from http://www.networkrail.co.uk/Contents/AboutUs/Documents/11865_BritainReliesonRail.pdf [Accessed December 12, 2011].

Oravec, M., & Pavlovičová, J. (2007). Face Recognition Methods Based on Feedforward Neural Networks , Principal Component Analysis and Self-Organizing Map, 16(1), 51–57.

Palmer, S.B. (2005). Transverse and longitudinal crack detection in the head of rail tracks using Rayleigh wave-like wideband guided ultrasonic waves. Proceedings of SPIE, 5767, pp.70–80. Available at: http://link.aip.org/link/?PSI/5767/70/1&Agg=doi [Accessed October 10, 2010].

Panda, S. S., Deepthi, G., & Anisha, V. (2011). Face Recognition Using PCA and Feed Forward Neural Networks, 2(8), 79–82.

Papaelias, M. Ph., Lugg, M., Roberts, C. & Davis, C. L. (2009). High-Speed inspection of rails using ACFM technology. Journal of NDT&E International.

Papaelias, M. Ph., Roberts, C. & Davis, C.L. (2008). A review on nondestructive evaluation of rails: state-of-the-art and future development. Rail and rapid transit, p. 367, p. 373.

Papaelias, M., S. Kerkyras, F. Papaelias, & Graham, K. (2012). The future of rail inspection technology and the INTERAIL FP7 project (pp. 1–13). Available at: http://www.interailproject.eu/files/technical_information/NDT2012_2C3.pdf [Accessed March 10, 2013].

Raj, M., Mallik, D., Bansal, S., Saini, R. K., & Ajmeria, R. K. (2012) . Non-Destructive Testing and Inspection of Rails at JSPL – Ensuring Safety and Reliability 2 . Manufacturing process at JSPL, (April), 16–20.

Ramadge, Y. T. X. and P. J. (2009). Separable PCAfor image classification, 1805–1808.

Reinhard, E., Stark, M., Shirley, P., & Ferwerda, J. (2002). Photographic tone reproduction for digital images. ACM Transactions on Graphics, 21(3), 267–276.

Rizzo, P. et al. (2009). An unsupervised learning algorithm for fatigue crack detection in waveguides. Smart Materials and Structures, 18(2), p.025016.

Rizzo, P. et al. (2010). Ultrasonic Guided Waves-Based Monitoring of Rail Head: Laboratory and Field Tests. Advances in Civil Engineering, 2010, pp.1–13. Available at: http://www.hindawi.com/journals/ace/2010/291293/ [Accessed January 12, 2011].

Robinsztejn, Y. (2011). Automatic Detection of Objects of Interest from Rail Track Images, MSc Dissertation.

Rose, J. L., Avioli, M. J., Song, W. J. (2002). Application and potential of guided wave rail inspection. Insight, vol. 44.

Ruvo, P. De, Ruvo, G. De, Distante, a., Nitti, M., Stella, E., & Marino, F. (2008). A Visual Inspection System for Rail Detection and Tracking in Real Time Railway Maintenance. The Open Cybernetics & Systemics Journal, 2(1), 57–67.

Sandhu, P.S. et al. (2009). Face Recognition Using Eigen face Coefficients and Principal Component Analysis. Data Base, pp.498–502.

Sandhu, P.S. et al., (2009). Face Recognition Using Eigen face Coefficients and

Principal Component Analysis. Data Base, pp.498–502.

Sawadisavi, S.V., Edwards, J.R. & Hart, J.M. (2008). Machine-Vision Inspection of Railroad Track 09-1369. New York.

Shah, D. M. (2010). Automated Visual Inspection / Detection of Railroad Track. Retrieved from http://www.dot.state.fl.us/research-center/Completed_Proj/Summary_PTO/FDOT_BD550-08_rpt.pdf

Shrutisagar, C. (2004). Automatic damage detection for railroad tracks by real-time DSP based dual analysis system. in Micro-NanoMechatronics and Human Science, IEEE International Symposium.

Singh, M., Singh, S., Jaiswal, J., & Hempshall, J. (2006). Autonomous Rail Track Inspection using Vision Based System, (October), 16–17.In IEEE International Conference on Computational Intelligence for Homeland Security and Personal Safety.

Steger, Carsten, Ulrich, M., and Wiedemann, C. (2008). Machine Vision Algorithms and Applications. Weinheim: Wiley-VCH. p. 1. ISBN 9783527407347.

Stella, E., Mazzeo, P. L., Nitti, M., Cicirelli, G., Distante, A. & D'Orazio, T. (2002). Visual Recognition of Missing Fastening Elements for Railroad Maintenance, Proceedings of the IEEE 5th ITSC, Singapore.

Sudeep K C & Dr. Majumdar, J. (2011). A Novel Architecture for Real Time Implementation of Edge Detectors on FPGA. IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 1, ISSN (Online): 1694-0814, www.IJCSI.org

Tao, X. & Eades, A. (2003). Artifacts Introduced Through the Hough Transform in EBSD. , 9(Suppl 2), pp.724–725.

Thakur, S., Sing, J. K., Basu, D. K., Nasipuri, M., & Kundu, M. (2008). Face Recognition Using Principal Component Analysis and RBF Neural Networks. First International Conference on Emerging Trends in Engineering and Technology, 695–700.

Toliyat. H.A, Abbaszadeh. K, Rahimian. M.M, Olson. L.E. (2003). Rail defect diagnosis using wavelet packet decomposition. IEEE Transactions on Industry Applications . Volume 39, Issue 5, pp. 1454-1461.

Trinh, H., Haas, N., Li, Y., Otto, C., & Pankanti, S. (2012). Enhanced rail component detection and consolidation for rail track inspection. 2012 IEEE Workshop on the Applications of Computer Vision (WACV), 289–295.

Wang, X., Chen, T., Li, H., & Wu, W. (2012). Algorithms for Fast Locating & Identification on the Surface Defects of the Thermal-State Heavy Rail based on Machine Vision. 2012 IEEE 11th International Conference on Cognitive Informatics and Cognitive Computing, (2), 472–476.

Wang, Y. (2006). Image Filtering: Noise Removal, Sharpening, Deblurring. Available at: http://eeweb.poly.edu/~yao/EE3414/image_filtering.pdf.

Wilcox, P., Evans, M., Pavlakovic, B., Alleyne, D., Vine, K., Cawley, P., and Lowe, M. (2003), “Guided wave testing of rail,” Insight, vol. 45, no. 6, pp. 413–420.

Wilson T. Wamani & Villar, C. (2009). Aurora Automated Railroad Tie Condition Assessment System: The Quest For Accuracy. AREMA 2009 Annual Conference & Exposition.

Wilson, A., & Kerr, M. (2012). Rail defects handbook, (June), 1–83.

Wyszecki, G. & Stiles, W. S. (2000) . Color Science. Concepts and Methods, Quantitative Data and Formulae. Second Edition, John Wiley & Sons.

Xie, X. (2008). A Review of Recent Advances in Surface Defect Detection using Texture analysis Techniques. , 7(3), pp.1–22.

Ye, F., Shi, Z., & Shi, Z. (2009). A Comparative Study of PCA, LDA and Kernel LDA for Image Classification. 2009 International Symposium on Ubiquitous Virtual Reality, 51–54.

Yella, S. & Gupta, N.K. (2007). Automating condition monitoring of wooden railway sleepers. Engineerit, (October). ISC ’07 Proceedings of the 10th IASTED International Conference on Intelligent Systems and Control (pp.289-

295). CA, USA : ACTA Press Anaheim.

Yılmazer, P., Amini, A., & Papaelias, M. (2011). The Structural health condition monitoring of rail steel using acoustic emission techniques, in the proceedings of NDT 2012.

Ze Liu Wei Wang Xiaofei Zhang Wei Jia, (2010). Inspection of rail surface defects based on image processing. in Informatics in Control, Automation and Robotics (CAR), 2010 2nd International Asia Conference, Wuhan.

Zheng, W., & Zhang, Y. (2011). A novel improvement to PCA for image classification. 2011 International Conference on Computer Science and Service System (CSSS), 1964–1967.

Zhu, Y. & Huang, C. (2012). An Adaptive Histogram Equalization Algorithm on the Image Grey Level Mapping. Physics Procedia, 25, pp.601–608. Available at: http://linkinghub.elsevier.com/retrieve/pii/S1875389212005482 [Accessed January 7, 2013].

10 Appendix A - Back Propagation Neural Network Steps for Wire Defect Detection

Back Propagation Neural Network (BPNN) are used for implementing image segmentation, different clustering techniques, image restoration and reconstruction, nonlinear image filtering, target detection, radar imaging, medical imaging, document analysis, character, signature, face and object recognition. In high-level image processing, they are helpful for three-dimensional object recognition, motion estimation, and stereo vision. [2]

The typical BPNN model has an input layer, at least one or more hidden layer(s) and an output layer. Theoretically, there is no limit on the number of hidden layers but typically they are one or two.

The network works in two stages, Feed-Forward stage usually known as learning or training of the network, and a Back-Propagation Stage. To begin with, a network is created and the synaptic weights are set to some random values initially. Generally the network's need two different sets of data. One set is used for training and the other is used for testing by simulating the created network through it. Network can be tested on the same training data to avoid prediction or memorizing patterns.

In feed forward stage the training data set is presented to the network. The data is processed by all the layers of the network in a sequence and processed data is moved towards the output layer.

The output received after feed forward stage, contains error. These errors are the difference between the predicted output and the actual output produced. They inform the network that how far it is from the desired output. Hence, these errors are used to readjust the network. These are feed back to the network and new weights are assigned to each neuron. Back propagation method tries to minimize the sum of these errors for the training data, to make the network work in the most "desirable" way. Before training a BPNN, pre knowledge of the output attributes, number of epochs, learning rate, hidden layer nodes and test data set is necessary.

[2] **"Applications of Artificial Neural Networks in Image Processing IX (EI15)",** http://electronicimaging.org/call/04/ [online]

10.1.1 Training Data and Targets

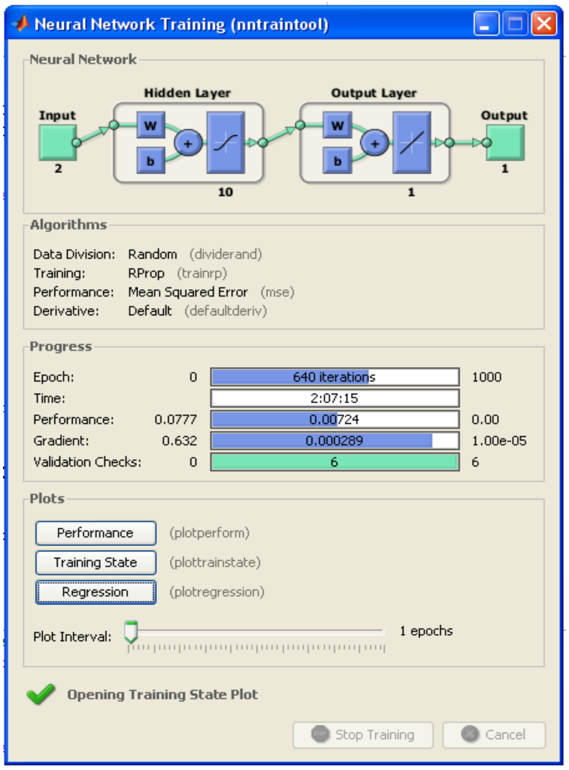
The network is trained on a small set of data points. For example, there is an image with a wire defect on it. First step is to train the network with few points selected from the defect and the clear rail. The size of the mask can also be set by the user. When he clicks any point on the image for selection, the pixels surrounding that point gets selected in an array. The statistics of each points selected by the user are calculated; their respective targets are assigned and saved in individual arrays. Statistics are calculated using the masking process explained in chapter 3. A small size of mask is chosen and is moved over the whole image. The function to be performed or the calculation is repeated for every position of the mask and the values are stored in input array as rows. Most commonly used statistics are median, mean, variation, standard deviation, local minima and maxima.

10.1.2 Training the Network

The back propagation neural network is then trained by the input. The nodes in the input and output layer needs to be defined. This is determined by number of statistics used. For example, if only standard deviation and mean is used then input layer would have 2 neurons. Output layer would have only one neuron as an input can either be a defect or not. However, neurons in the hidden layer can be defined by the use. The input vectors pass through the network again and again, errors are propagated and weights are adjusted. Network is trained till the maximum epoch is reached. Training plot and network can been seen in the figures below.

10.1.3 Simulation and Testing

Once the training of the network is finished, it can be tested on any data for verification. Simulation takes place very fast as compared to the Training, as the network that is already intelligent enough only has to classify the input to the known groups. Simulation sorts the input and divides them into the specified groups and resulting image can be viewed after that.



Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division: Random (dividerand)
Training: RProp (trainrp)
Performance: Mean Squared Error (mse)
Derivative: Default (defaultderiv)

Progress

Epoch:	0	640 iterations	1000
Time:		2:07:15	
Performance:	0.0777	0.00724	0.00
Gradient:	0.632	0.000289	1.00e-05
Validation Checks:	0	6	6

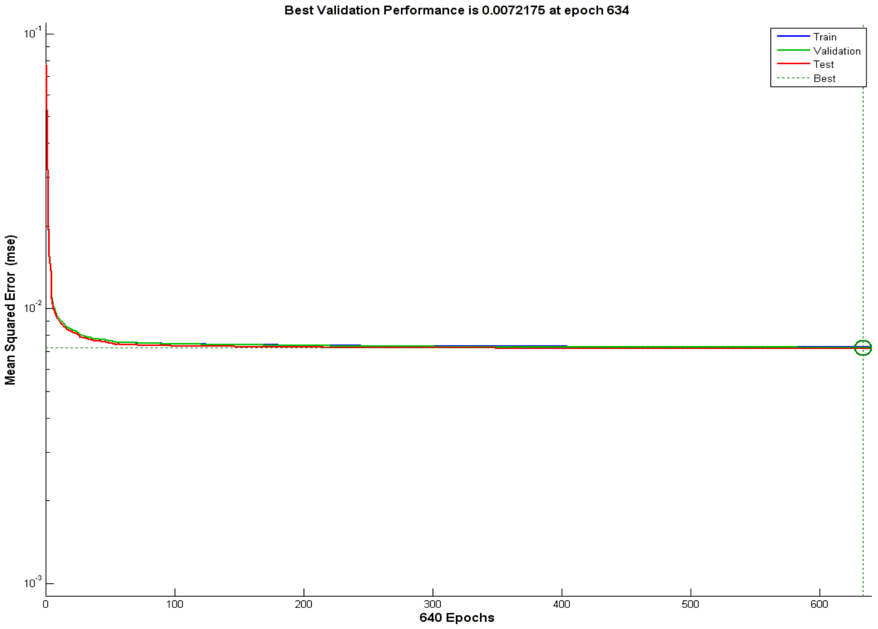
Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

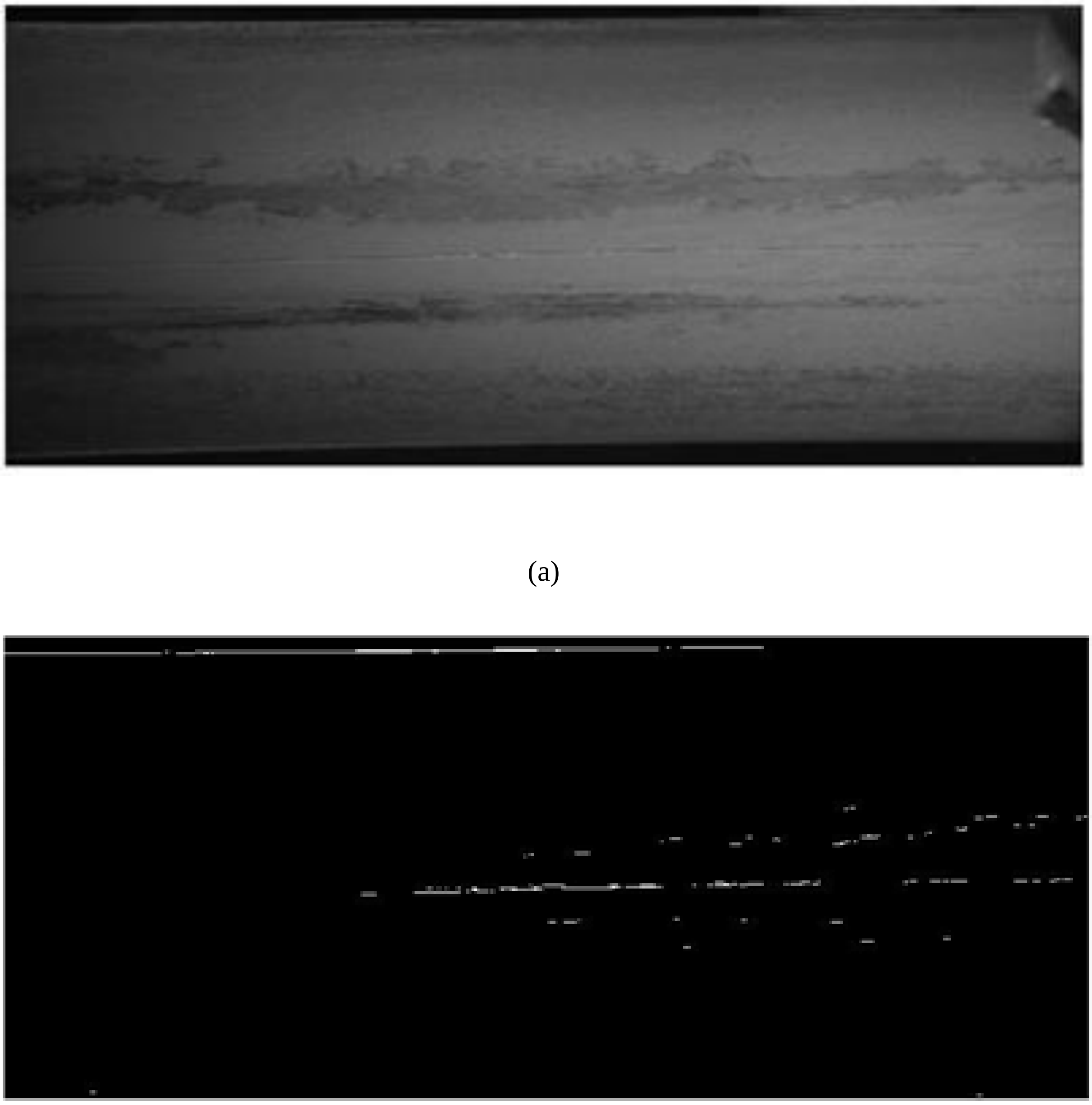
✓ Opening Training State Plot

Stop Training Cancel



10.1.4 Wire Defect Detection Results Produced by Matlab

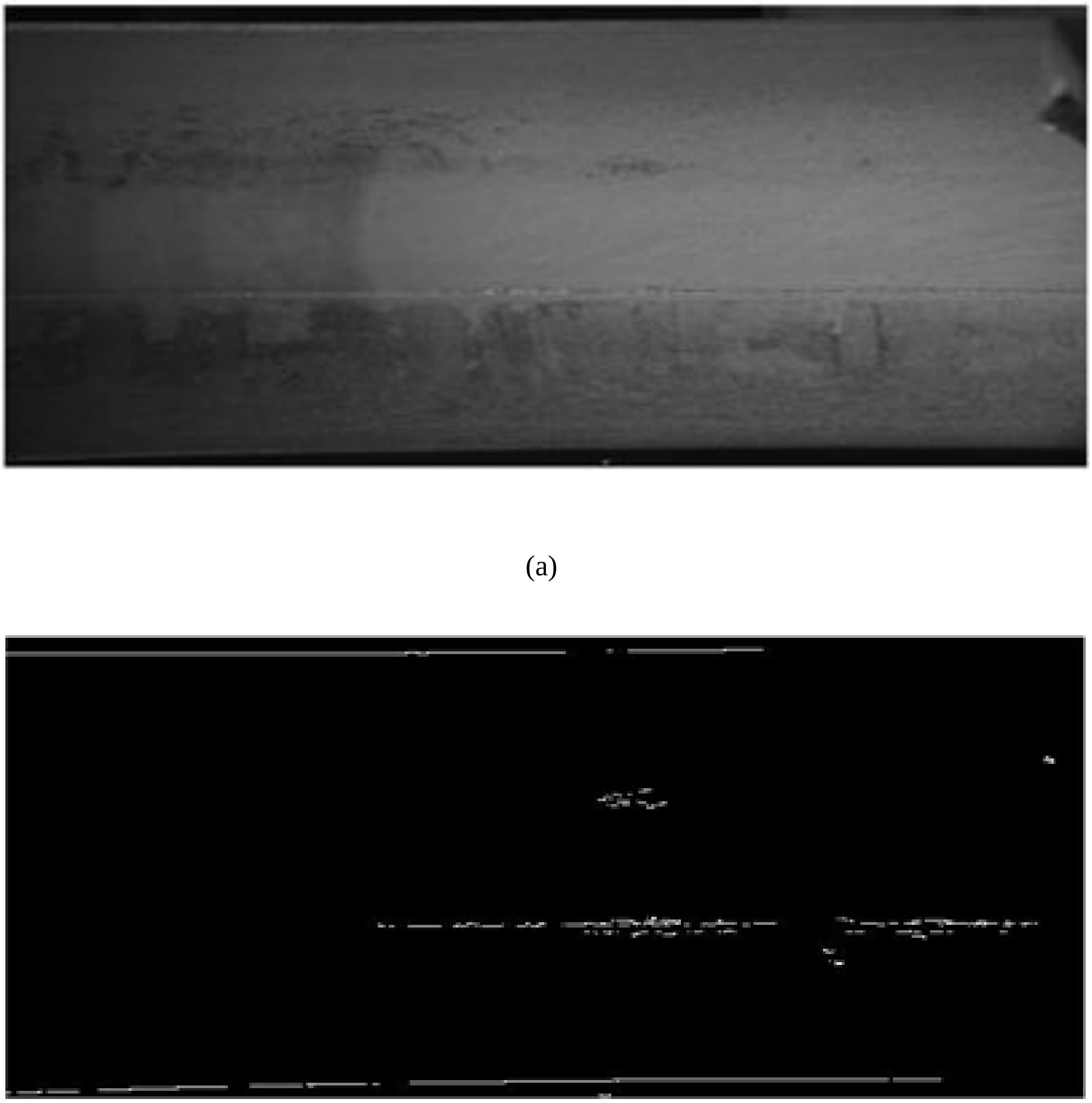
Some of the results produced by BPNN have been presented in the figures below.



(a)

(b)

Figure 10.1: Neural Network Result A. (a) Original rail image with wire defect (b) Wire defect detected by Neural Networks



(a)

(b)

Figure 10.2: Neural Network Result B. (a) Original rail image with wire defect (b) Wire defect detected by Neural Networks

10.1.5 Practical Considerations of BPNN

Bach propagation networks are very efficient at prediction and classification (segmentation) but they are comparatively slow. There is no explanation about the learning of the network available i.e. a learned network does not contain any information about its learning. Back-propagation networks have some practical consideration that needs to be considered before using them.

Speed: The neural networks are very fast on hardware but really slow when they are run on the computers. The small networks run faster it works.

The right choice of data: Anything can be fed on network and it becomes a problem, even the irrelevant data. For example noise on the images cause a problem.

Stability of a solution: Overtraining of the network usually results in error.

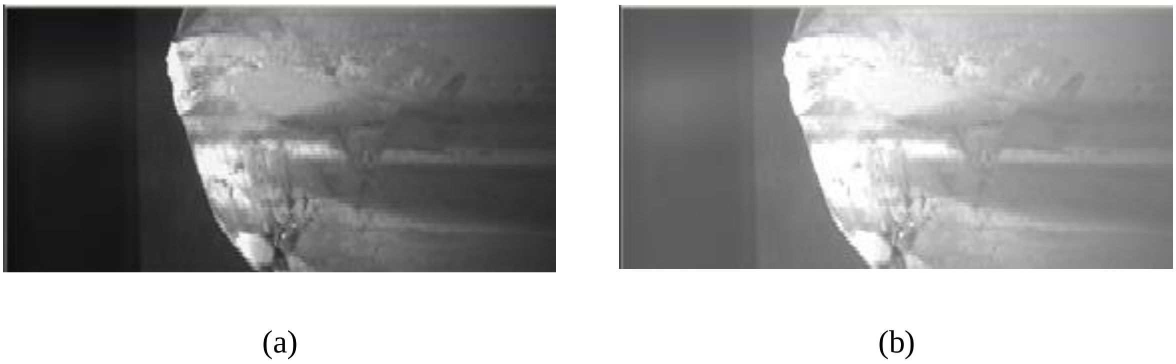
Can we use our results: Considering practical applications, we need to sort out the fact that data might change, and the network working today might not work tomorrow.

11 Appendix B – Extra Light Adjustment Techniques Tested

11.1 Hue Saturation Value (HSV) adjustment

Conversion of the images from the standard Red, Green, Blue (RGB) standard colour space to an alternative colour space for enhanced colour segmentation helps in contrast and brightness adjustment. The Hue, Saturation, Value (HSV) also known as Hue, Saturation, Brightness (HSB) or Hue Saturation lightness (HSL) spaces as known to be useful for light adjustment in an image, has also been tested for the purpose. Hue is the colour in image, saturation is the intensity and lightness is for the brightness value of the image.

The simple procedure involves the conversion of an image to HSV space, which returns an HSV colormap. An HSV colormap varies the hue component of the hue-saturation-value colour model. The colours begin with red, pass through yellow, green, cyan, blue, magenta, and return to red. The map is particularly useful for displaying periodic functions. This step is done to perform segmentation based on colour difference. As with HSV function every colour, with whatever intensity is assigned same HSV value. That helps to segment that specific colour and in the current situation can be used to adjust the brightness of rail images. The result of the test has been shown in the Figure 11.1. The image HSV space was generated and then lightness value had been increased to make look bright. However, the problem with the grey scale rail images is the use of non-diffused light, making some part of the rail very bright leaving others very dark. Hence, making whole image bright using HSV space didn't help to make the image lighting look uniform.

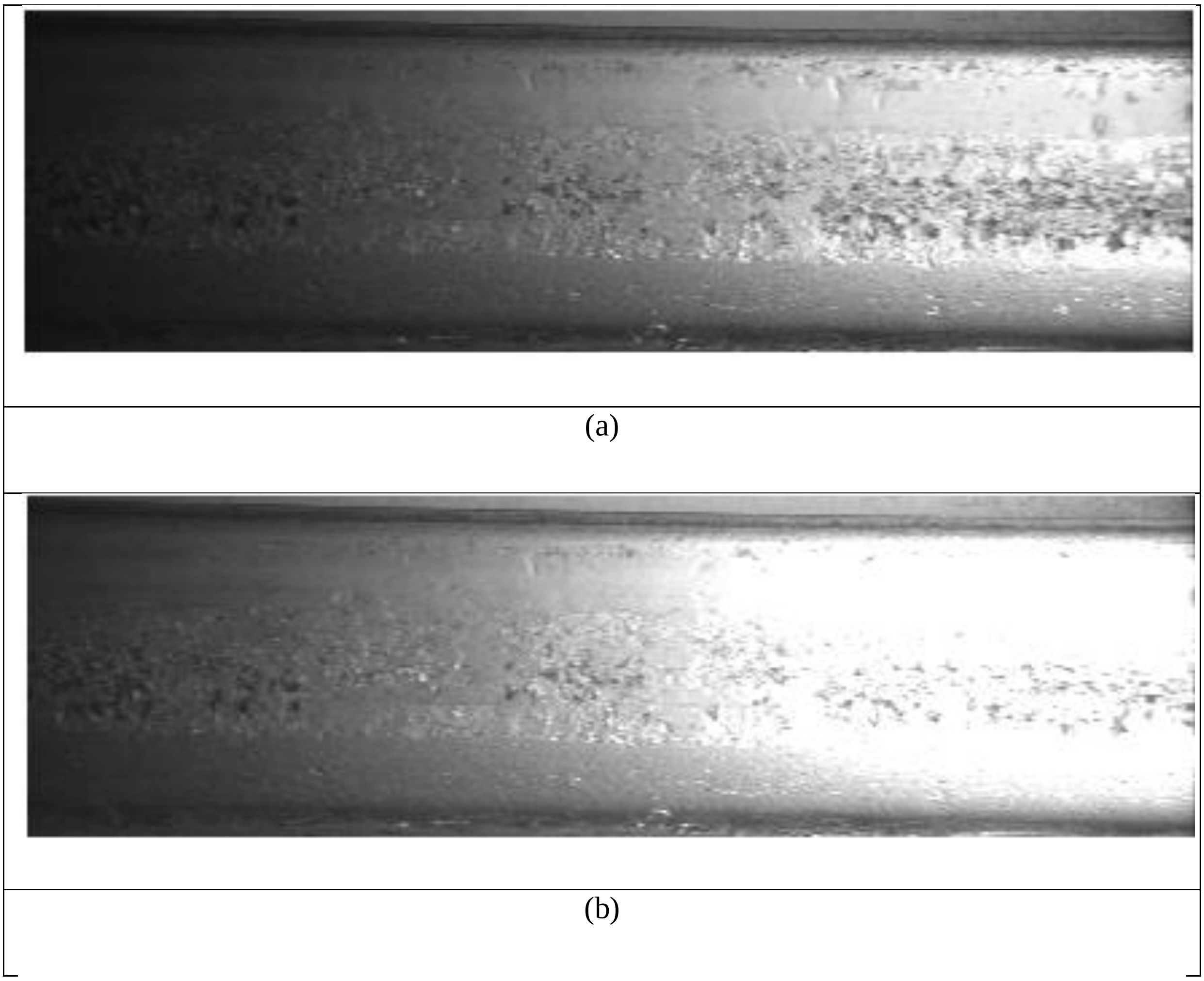


(a) (b)

Figure 11.1: Result of light adjustment using HSV color space. From left (a) Original Image (b) Lightness Value Adjusted in HSV space

11.2 Gamma Correction

Gamma correction handles the overall brightness of an image. It can be described as a relationship between and input and output image or in more technical terms the relationship between the numerical value of each of the pixel in an image and its actual luminance. Gamma correction can be applied to the images that are either too bright or too dark, to make them look proper. Gamma correction makes the features in an image more prominent without causing any washing out effect. However, it also didn't work well for type of rail images as can been seen in Figure 11.2. The results shown have been produced by self-written code. Gamma correction does make the image look brighter but fails to make it look uniform.



(a)

(b)

Figure 11.2: Result of Gamma correction. (a) Original tiger stripe Image (b) Gamma Corrected Image