DETECTION SYSTEM FOR COVER TAPE OFFSET IN TAP AND REEL PROCESS WITH PREDICTIVE ANALYSIS

MUHAMMAD IRFAN BIN ROSLI

اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

DETECTION SYSTEM FOR COVER TAPE OFFSET IN TAP AND REEL PROCESS WITH PREDICTIVE ANALYSIS

MUHAMMAD IRFAN BIN ROSLI

This report is submitted in partial fulfilment of the requirements for the degree of Bachelor of Electronic Engineering with Honours

Faculty of Electronic and Computer Technology and Engineering
Universiti Teknikal Malaysia Melaka

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2024

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
FAKULTI TEKNOLOGI DAN KEJUTERAAN ELEKTRONIK DAN KOMPUTER

BORANG PENGESAHAN STATUS LAPORAN
PROJEK SARJANA MUDA II

Tajuk Projek : Detection System for Cover Tape Offset in Tap and Reel Process with Predictive Analysis
Sesi Pengajian : 2023/2024

Saya MUHAMMAD IRFAN BIN ROSLI mengaku membenarkan laporan Projek Sarjana Muda ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Laporan adalah hakmilik Universiti Teknikal Malaysia Melaka.
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan laporan ini sebagai bahan pertukaran antara institusi pengajian tinggi.
4. Sila tandakan (✓):

☐ **SULIT*** (Mengandungi maklumat yang berdarjah keselamatan atau kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972)

☑ **TERHAD*** (Mengandungi maklumat terhad yang telah ditentukan oleh organisasi/badan di mana penyelidikan dijalankan.

☐ **TIDAK TERHAD**

Disahkan oleh:

(TANDATANGAN PENULIS)

(COP DAN TANDATANGAN PENYELIA)

KHAIRUN NISA BINTI KHAMIL (Ph.D)
PENSYARAH KANAN
FAKULTI TEKNOLOGI & KEJ. ELEKTRONIK & KOMPUTER (FTKEK)
UNIVERSITI TEKNIKAL MALAYSIA MELAKA
HANG TUAH JAYA
76100, DURIAN TUNGGAL
MELAKA
HP No: +60107667004 (OFFICE HOUR)

Alamat Tetap: No. 26, Jalan
Teratai 2/1,
Saujana Utama 3,
Sungai Buloh
47000, Selangor

Tarikh : 12 Jan 2024

Tarikh : 12 Jan 2024

*CATATAN: Jika laporan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh laporan ini perlu dikelaskan sebagai SULIT atau TERHAD.

DECLARATION

I declare that this report entitled "Detection System for Cover Tape Offset in Tap and Reel Process with Predictive Analysis" is the result of my own work except for quotes as cited in the references.

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Signature : ……………………………………
Author : Muhammad Irfan Bin Rosli
Date : 12 Januari 2024

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Bachelor of Electronic Engineering with Honours.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Signature : ……………………………………

Supervisor Name : Khairun Nisa binti Khamil

Date : 23 Januari 2024

DEDICATION

I would like to express my sincere gratitude to Allah, the most gracious and merciful, for blessing me with the strength and patience to successfully accomplish my final year project. Furthermore, I am extremely thankful to my parents for their constant and unwavering support in my educational endeavours. I would also like to extend my heartfelt appreciation to Dr. Khairun Nisa Binti Khamil for his invaluable guidance throughout this project. Finally, I am grateful to all my friends for their assistance and companionship during times of need.

اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRACT

This thesis presents a comprehensive study on the detection of cover tape offset or misalignment during the tape and reel process, crucial for packaging electronic components into individual pockets of carrier tape. The research aims to develop an efficient system utilizing the Raspberry Pi Camera Module for detecting and analysing cover tape misalignment. The methodology involves integrating the Raspberry Pi Camera Module with a microcontroller to capture and process images of the carrier tape, employing image processing techniques for misalignment detection. The resulting data is displayed in a user-friendly dashboard format using Node-RED. Additionally, the data is analysed in MATLAB for predictive analysis. The findings of this research, including the analysis of training results, demonstrate the successful implementation of a reliable cover tape misalignment detection system. Notably, Bayesian Regularization (BR) training algorithm outperformed Scaled Conjugate Gradient (SCG) training algorithm for cover tape offset's predictive analysis, exhibiting lower Mean Squared Error (MSE) with 0.0015874 for BR compared to 0.0017839 for SCG, consistently lower Mean Absolute Error values, stronger linear correlations, and superior overall performance, emphasizing its effectiveness for accurate predictions.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

ABSTRAK

Tesis ini menerangkan tentang satu kajian menyeluruh mengenai pengesanan ketidakseimbangan pada penutup pita semasa proses “Tap and Reel”. Matlamat kajian ini adalah untuk membangunkan satu sistem yang efisien menggunakan Modul Kamera Raspberry Pi untuk mengesan dan menganalisis ketidakseimbangan penutup pita. Rasional bagi kajian ini adalah untuk meningkatkan kualiti proses ini dengan mengautomatikkan pengesanan ketidakseimbangan. Metodologi yang digunakan dalam projek ini melibatkan penggunaan Modul Kamera Raspberry Pi untuk merakam imej pita pembawa itu. Imej yang dirakam dianalisis menggunakan teknik pemprosesan imej untuk mengesan dan mengukur ketidakseimbangan penutup pita. Data yang diperoleh kemudiannya dipaparkan dalam format papan pemuka yang dikenali sebagai Node-RED. Selain itu, data tersebut dianalisis di MATLAB untuk analisis prediktif. Hasil kajian ini, termasuk analisis keputusan latihan, menunjukkan pelaksanaan yang berjaya bagi sistem pengesanan yang boleh dipercayai. Secara signifikan, algoritma latihan Bayesian Regularization (BR) telah mengatasi algoritma latihan Scaled Conjugate Gradient (SCG) untuk analisis prediktif ketidaksempurnaan penutup, menunjukkan Nilai Purata Kuasa Dua (MSE) yang lebih rendah iaitu 0.0015874 untuk BR yang dapat dibandingkan dengan 0.0017839 untuk SCG.

ACKNOWLEDGEMENTS

This is an acknowledgement for everyone who helps me accomplish this project. First of all, I am grateful to the faculty, the Faculty of Electronic and Computer Technology and Engineering for giving me the opportunity to do this project titled 'Detection System for Cover Tape Offset in Tap and Reel Process with Predictive Analysis'.

Then, I would like to thank my supervisor, Dr. Khairun Nisa Binti Khamil for her expert advice and encouragement throughout this project. Furthermore, to my parents, who always supported me in doing this project. It was really touching when they have provided a huge amount of their precious time and effort.

Finally, I would like to express gratitude to all my friends for their support and willingness to spend their time answering all questions about this project.

اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

TABLE OF CONTENTS

Declaration

Approval

Dedication

Abstract **i**

Abstrak **ii**

Acknowledgements **iii**

Table of Contents **iv**

List of Figures **viii**

List of Tables **xi**

List of Symbols and Abbreviations **xii**

List of Appendices **xiii**

CHAPTER 1 INTRODUCTION **1**

1.1 Background of Project 1

1.2 Problem Statement 3

1.3 Objective 4

1.4 Scope of Project 5

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

1.5 Thesis Outline 6

CHAPTER 2 BACKGROUND STUDY 7

2.1 Canny Edge Detection 8

2.2 Hough Line Transform 10

2.3 Cover Tape Offset in Tap and Reel Process 11

2.4 Node-RED 15

2.5 Training Algorithms 17

2.6 Summary 22

CHAPTER 3 METHODOLOGY 24

3.1 Measurement from Industry 25

3.2 Initialize Raspberry Pi Camera Module 28

3.3 Computer Vision Processing 30

3.4 Canny Edge Detection 30

3.5 Hough Line Transform 31

3.6 Display Real-Time Data on Node-Red Dashboard 31

3.7 Generate CSV File 33

3.8 Neural Net Time Series Apps for Predictive Analysis 34

3.9 Summary 35

CHAPTER 4 RESULTS AND DISCUSSION 36

4.1 Node-RED 36

UTeM
اونيۈرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

4.2 Bayesian Regularization (BR) 37

4.2.1 Training results 37

4.2.2 Correlation coefficient (R) 42

4.2.3 Mean Squared Error (MSE) 46

4.3 Scaled Conjugate Gradient (SCG) 49

4.3.1 Training results 49

4.3.2 Correlation coefficient (R) 54

4.3.3 Mean Squared Error (MSE) 59

4.4 Comparison for Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG) 63

4.4.1 Training Results 63

4.4.2 Correlation coefficient (R) 65

4.4.3 Mean Squared Error (MSE) 67

4.4.4 Prediction results 69

4.5 Summary 73

CHAPTER 5 CONCLUSION AND FUTURE WORKS 74

5.1 Conclusion 74

5.2 Sustainable Development Goals (SDG) 76

5.3 Future Work 77

5.3.1 Real-time Monitoring and Feedback 77

5.3.2 Data Logging and Analytics 77

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

REFERENCES **79**

UTeM
اونيۏرسيتي تيکنيکل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF FIGURES

Figure 2.1: Detected Edges in an Image [5]..9

Figure 2.2: Original Picture and Line Detection Results [9]. 11

Figure 2.3: Tape and Reel Structure [2]..13

Figure 2.4: Node-RED Dashboard [14]. 16

Figure 2.5: Regression Plot for BR Algorithm [16]. 20

Figure 2.6: Regression Plot for SCG Algorithm [16]. 20

Figure 2.7: The Performance of BR Algorithm [16]. 21

Figure 2.8: The Performance of SCG Algorithm [16]. 21

Figure 3.1: Flowchart for the whole process. 25

Figure 3.2: The offset's measurement visualization...26

Figure 3.3: Example visualization of offsets in the industry. 26

Figure 3.4: Cover tape position away from sprocket hole. 27

Figure 3.5: Conveyer belt on FMS 200. 27

Figure 3.6: Camera’s design setup. 28

Figure 3.7: Connection from the camera to the Raspberry Pi....................................28

Figure 3.8: Raspberry Pi Camera Module 3 installed on the FMS 200. 29

Figure 3.9: Camera connection to the Raspberry Pi. 29

Figure 3.10: Python script on Terminal. 30

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.11: Hough Line detection on the selected Region of Interest......................31

Figure 3.12: Node-RED workflow. 32

Figure 3.13: Dashboard generated from Node-RED. 33

Figure 3.14: Real-time monitoring view..33

Figure 3.15: Data logged into CSV file. 34

Figure 3.16: Implementation of Training Algorithm using MATLAB. 34

Figure 4.1: Offset graph in Node-RED dashboard. 37

Figure 4.2: Training results in Day 1 (BR). 38

Figure 4.3: Training results in Day 2 (BR). 39

Figure 4.4: Training results in Day 3 (BR). 40

Figure 4.5: Training results in Day 4 (BR). 41

Figure 4.6: Scatter plot for Day 1 (BR). 43

Figure 4.7: Scatter plot for Day 2 (BR). 43

Figure 4.8: Scatter plot for Day 3 (BR). 44

Figure 4.9: Scatter plot for Day 4 (BR). 45

Figure 4.10: Graph for Best Training Performance (Day 1). 46

Figure 4.11: Graph for Best Training Performance (Day 2). 47

Figure 4.12: Graph for Best Training Performance (Day 3). 48

Figure 4.13: Graph for Best Training Performance (Day 4). 48

Figure 4.14: Training results in Day 1 (SCG)...50

Figure 4.15: Training results in Day 2 (SCG)...51

Figure 4.16: Training results in Day 3 (SCG)...52

Figure 4.17: Training results in Day 4 (SCG)...53

Figure 4.18: Scatter plot for Day 1 (SCG). 55

Figure 4.19: Scatter plot for Day 2 (SCG). 56

Figure 4.20: Scatter plot for Day 3 (SCG). 57

Figure 4.21: Scatter plot for Day 4 (SCG). 58

Figure 4.22: Graph for Best Training Performance (Day 1). 59

Figure 4.23: Graph for Best Training Performance (Day 2). 60

Figure 4.24: Graph for Best Training Performance (Day 3). 61

Figure 4.25: Graph for Best Training Performance (Day 4). 62

Figure 4.26: Training results for combined data (BR). 64

Figure 4.27: Training results for combined data (SCG). 64

Figure 4.28: Scatter plot for combined data (BR). 66

Figure 4.29: Scatter plot for combined data (SCG). 66

Figure 4.30: Graph for Best Training Performance (Combined data). 68

Figure 4.31: Graph for Best Training Performance (Combined data). 68

Figure 4.32: Comparison between actual and predicted value using BR training. 70

Figure 4.33: Comparison between actual and predicted value using SCG training. . 70

Figure 4.34: Plot for MAE between actual and predicted value (BR). 71

Figure 4.35: Plot for MAE between actual and predicted value (SCG). 72

Figure 4.36: Comparison plot between MAE for BR and SCG. 73

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF TABLES

Table 2.1: Method summary 22

Table 4.1: Performance for 4 days of data using BR 42

Table 4.2: Correlation coefficient comparison for BR training 45

Table 4.3: Best Training Performance for 4 days of data using BR. 49

Table 4.4: Performance for 4 days of data using SCG. 54

Table 4.5: Correlation coefficient comparison for SCG training 58

Table 4.6: Best Training Performance for 4 days of data using SCG. 63

Table 4.7: Overall performance for combined data. 65

Table 4.8: Correlation coefficient comparison between BR and SCG. 67

Table 4.9: Best Training Performance for combined data. 69

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF SYMBOLS AND ABBREVIATIONS

SMD : Surface Mount Device

HT : Hough Transform

FPGA : Field Programmable Gate Array

WLCSP : Wafer-Level Chip-Scale Package

SSPF : Single Side Peel Force

AOI : Automated Optical Inspection

LED : Light-Emitting Diode

MSE : Mean Square Error

MAE : Mean Absolute Error

CSV : Comma-Separated Values

BR : Bayesian Regularization

SCG : Scaled Conjugate Gradient

mm : millimeter

اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

LIST OF APPENDICES

Appendix A: Coding for The System 84

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 1

INTRODUCTION

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The tape-and-reel process is a widely used method for packaging electronic components, which involves placing them in individual pockets of carrier tape for protection and ease of handling during manufacturing and shipping. However, during the tape and reel process of packaging electronic components, the cover tape offset, or misalignment occurs, leading to product defects and waste. This problem can be difficult to detect and correct in a timely manner, leading to increased costs and decreased productivity.

1.1 Background of Project

The packaging of electronic components is a critical process in the electronics industry, ensuring the protection and efficient handling of these delicate devices. One commonly used method is the tape and reel process, which involves packaging individual components into pockets of carrier tape for ease of handling and automated assembly. However, during this process, misalignment or offset of the cover tape can occur, leading to various issues such as component damage, improper feeding, and increased production defects.

The current state of knowledge in this area highlights the importance of accurate and reliable detection of cover tape misalignment to ensure the quality and efficiency of the packaging process. Manual inspection methods are commonly employed, which are time-consuming, labor-intensive, and prone to human error. There is a need for an automated system that can detect cover tape misalignment in real-time, providing prompt feedback for corrective actions.

Despite the significance of this problem, there is a gap in research regarding the development of an integrated and efficient system for cover tape misalignment detection. Existing studies have focused on individual components of the system, such as image processing techniques or machine vision algorithms. However, there is a lack of comprehensive solutions that integrate various technologies to provide a complete and user-friendly system for real-time monitoring and analysis.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The aim of this research project is to address this gap by developing a robust and automated system for detecting cover tape misalignment during the tape and reel process. The system will utilize the Raspberry Pi Camera Module as a sensor to capture images of the carrier tape, which will then be processed and analyzed to detect any misalignment. The data from the sensor will be merged into a Raspberry Pi microcontroller, which will serve as the central processing unit for the system.

To provide a user-friendly interface for data visualization and analysis, the system will utilize Node-RED, a powerful dashboard tool. This will enable operators to monitor the real-time data and quickly identify any instances of cover tape misalignment. Additionally, the data will be transferred to MATLAB to apply predictive analysis training algorithms to identify patterns and potential future problems, thereby enabling proactive maintenance and minimizing production disruptions.

The scope of this work will encompass the design, development, and integration of the complete system for cover tape misalignment detection. The research will focus on the technical aspects of image processing, data integration, and system implementation.

1.2 Problem Statement

The packaging of electronic components is a critical process that ensures the safe transport and use of electronic devices. However, this process can be affected by various factors that may result in defects and waste, such as misalignment or improper handling. C. F. Jeffrey Kuo et al. have addressed the limitations of existing manual inspection methods for surface mount devices (SMDs) that are prone to human errors and time-consuming [1]. In addition, S. Qiao and L. Q. Tao addressed the issue of inconsistent peel force during the tape and reel packaging process for electronic components [2]. The authors explain that the peel force, which is the force required to remove the carrier tape from the adhesive tape during assembly, must be consistent to ensure that the electronic components are properly positioned and aligned. Inconsistent peel force can result in misalignment or damage to the components, which can lead to defects and waste.

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Existing solutions for detecting packaging issues in electronic components are often expensive and complex, requiring specialized equipment and expertise. Additionally, many of these solutions only detect issues after they have already occurred, which may be too late to prevent product defects or waste.

Therefore, to address this problem, our proposed project aims to develop a cost-effective and user-friendly system for detecting packaging issues in electronic components. Our system will have a Raspberry Pi Camera Module, capture images of the carrier tape, which will then be processed and analyzed to detect any misalignment. The system will be designed to be easy to use and integrate into existing manufacturing processes, making it a valuable tool for manufacturers looking to improve efficiency and reduce waste. The project outcome is expected to improve productivity, reduce waste and lower costs, resulting in significant benefits for the electronics industry.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

1.3 Objective

These objectives collectively focus on developing an efficient cover tape offset detection system, integrating it with Node-RED for real-time data visualization, and enhancing the system's functionality through predictive analysis using MATLAB.

The objectives of this project are:

1. To design a detection system that detects the cover tape offset in tape and reel process using a Raspberry Pi Camera Module (R01).
2. To formulate a Node-RED dashboard for live data monitoring (R02).
3. To analyze the cover tape offset behavior using predictive analysis (R03).

1.4 Scope of Project

The scope of this project encompasses the development of a system to detect cover tape offset or misalignment during the tape and reel process used in packaging electronic components into individual pockets of carrier tape. The system will utilize the Raspberry Pi Camera Module, employing Canny Edge Detection and Hough Line Transform techniques from computer vision to analyze and process images captured by the camera module. The data from the sensor will be merged into a Raspberry Pi microcontroller for further processing and analysis.

The project operates within certain constraints. It will be implemented using the Raspberry Pi platform and rely on the Raspberry Pi Camera Module for image capture. The system design will integrate computer vision techniques, specifically grayscale conversion, edge detection, and contour detection, to enable accurate detection of cover tape misalignment. Furthermore, the system will be using Node-RED as the dashboard for real-time data display.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The components to be packaged are assumed to have standardized dimensions and packaging requirements, facilitating consistent detection procedures. Additionally, it is assumed that the Raspberry Pi Camera Module will provide sufficient image quality and resolution for effective cover tape misalignment detection.

The prototype scope of the project focuses primarily on functionality rather than mass production considerations. The specific size and weight of the products being packaged are not explicitly defined within the scope, as the project's emphasis is on developing the cover tape misalignment detection system rather than the physical components being packaged.

The final deliverables of this project will include a working prototype of the cover tape misalignment detection system, implemented on the Raspberry Pi platform. The system will provide real-time data display through the Node-RED dashboard, facilitating cover tape misalignment monitoring.

Acceptance criteria for the project include the accurate detection of cover tape misalignment within a specified tolerance level and the provision of clear, intuitive, and responsive real-time data on the Node-RED dashboard. Excluded from the project scope are the design of the tape and reel equipment itself.

1.5 Thesis Outline

This thesis is organized systematically into the following chapters. Chapter 2 shows an overview of the Canny Edge Detection Method, Hough Line Transform, the past project that relate with the detection of cover tape offset in Tap and Reel Process, and the overview of Node-Red. Chapter 3 highlights the overall process throughout this project, focusing on the design and development of the cover tape offset detection system using Raspberry Pi camera module and microcontroller. The chapter starts by discussing the system requirements and specifications, including the resolution and field of view of the camera module, the processing capabilities of the microcontroller, and the necessary connectivity interfaces. Chapter 4 highlights the comparative analysis between two distinct training algorithms, namely Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG), to analyze their impact on the performance of the predictive model. Lastly, Chapter 5 highlights the conclusion for this whole project, future recommendation, and the relationship between this project to the Sustainable Development Goals (SDG).

UTeM
اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 2

BACKGROUND STUDY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The utilization of image processing techniques, specifically Canny edge detection, holds significance in offset detection within the tap and reel process. The Canny edge detection algorithm offers an effective approach for identifying edges and accurately measuring offsets. However, challenges such as image noise and threshold selection need to be addressed for robust offset detection. The Hough Line Transform is another relevant algorithm that can extract and analyze lines in images, aiding in offset detection. Variations of the Hough Transform, such as the random and probabilistic versions, optimize computational efficiency. Several studies highlight the importance of cover tape offset detection in the tap and reel process and propose various approaches to address this issue. Additionally, Node-RED, a versatile data integration and visualization platform, is applied in both industrial and healthcare domains, showcasing its adaptability and effectiveness. In summary, the combined literature underscores the relevance of Canny edge detection, the Hough Line Transform, cover tape offset detection, and Node-RED in advancing offset detection processes and improving outcomes in manufacturing and healthcare. In this chapter, the combined literature underscores the relevance of Canny edge detection, the Hough Line Transform, cover tape offset detection, and Node-RED in advancing offset detection processes and improving outcomes in manufacturing and healthcare.

2.1 Canny Edge Detection

The use of Canny edge detection in offset detection within the tap and reel process utilizing image processing techniques holds significance in the field of manufacturing and quality control. In this literature review, we will examine the relevance of Canny edge detection in offset detection, evaluate its achievements and limitations, and justify the need for further investigation in this area.

Image segmentation plays a vital role in preparing images for the detection step by extracting specific features from the source image [3]. In offset detection, edge segmentation is a crucial aspect as it helps locate relevant characteristics of objects in the image by detecting their edges. Edge-based segmentation reduces image size and facilitates analysis by removing extraneous data. Canny edge detection is a widely known and effective approach in edge-based segmentation.

The Canny edge detection algorithm, introduced by John F. Canny in 1986, is considered one of the optimal methods for edge detection. It employs a multi-stage approach, including Gaussian filters and intensity gradient changes, to identify edges in images. The algorithm aims to strike a balance between reducing noise interference and maintaining accurate edge detection [4]. Its ability to identify the best edge detection method makes it particularly relevant in offset detection tasks.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Offset detection in the tap and reel process involves identifying deviations or misalignments in the positions of components. By detecting edges using the Canny algorithm, it becomes possible to locate the boundaries of the components and accurately measure the offsets. This information is crucial for ensuring the quality and reliability of the manufacturing process.

However, despite its effectiveness, the application of the Canny edge detection algorithm, which the result can be seen on Figure 2.1, is not without limitations. One significant limitation is the sensitivity of edge detection outcomes to image noise [5]. The mathematical operations involved in edge detection, based on derivatives, can be affected by noise, leading to inaccurate edge detection results. To address this, preprocessing techniques such as Gaussian filters are commonly used to reduce noise. Implementing Gaussian blur to smooth the image helps eliminate noise and enhance the accuracy of offset detection. In addition, the algorithm's performance relies on the careful selection of threshold values, which are often set experimentally [6]. Various approaches have been proposed to enhance the algorithm, such as adaptive thresholding methods and incorporating fuzzy reasoning and genetic algorithms [7].

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

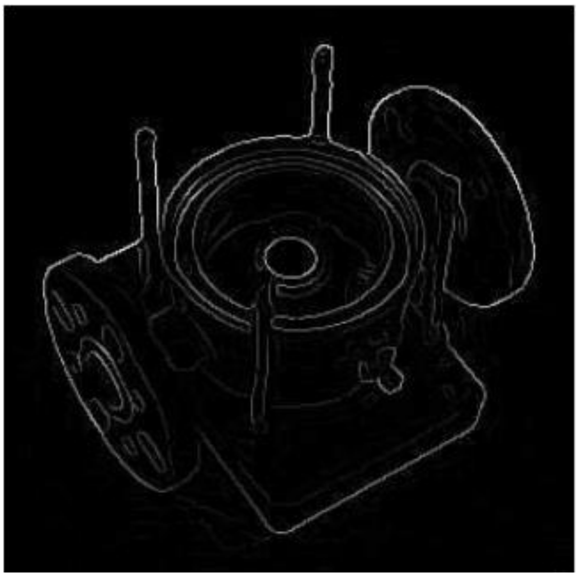


Figure 2.1: Detected Edges in an Image [5].

In conclusion, the utilization of the Canny edge detection algorithm in offset detection within the tap and reel process using image processing techniques is highly relevant. It offers a reliable method for identifying and measuring offsets in manufacturing. However, challenges related to image noise and threshold selection need to be addressed to ensure accurate and robust offset detection. Further investigation in these areas will contribute to the advancement of edge detection techniques and enhance the performance of the Canny algorithm in offset detection applications.

2.2 Hough Line Transform

The Hough Transform (HT) is a widely used algorithm in computer vision and pattern recognition, and it holds significant relevance to your project on offset detection in the tap and reel process using a Raspberry Pi camera module. By applying the HT technique, you can effectively extract and analyze lines in the images captured by the Raspberry Pi camera module, aiding in the detection and measurement of offsets in the tap and reel process.

اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The HT algorithm, originally introduced by Paul Hough in 1962, transforms spatially extended patterns into compact features in a parameter space, simplifying the line detection problem [8]. In your project, the HT can be leveraged to identify and quantify offsets by detecting lines associated with the target objects in the tap and reel process.

The variations of the HT, such as the random Hough transform (RHT) and the probabilistic Hough transform (PHT), can be particularly useful in optimizing the computational efficiency for real-time offset detection on the Raspberry Pi platform. The result for the real-time offset detection can be seen in Figure 2.2 which the method can be applied in this project. The RHT, through its reduction of unnecessary computations, can enhance the processing speed, while the PHT's utilization of random sampling significantly reduces the computation load [9]. These variations can assist in achieving real-time performance, a necessary requirement for applications like robot navigation and object tracking.

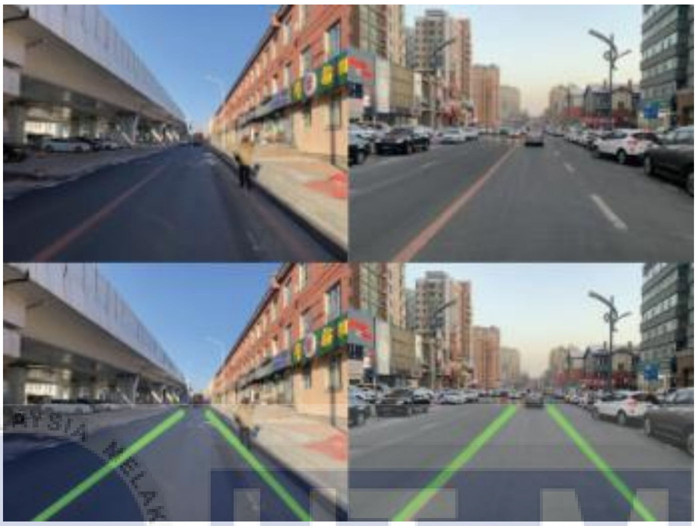


Figure 2.2: Original Picture and Line Detection Results [9].

Implementing the HT on the Raspberry Pi hardware platform, possibly using a Field Programmable Gate Array (FPGA), can further enhance the real-time line detection capabilities in your project. By leveraging the hardware capabilities of the Raspberry Pi, you can efficiently detect and measure offsets, addressing the challenges of memory capacity and time-consuming computations highlighted in the literature [10]. Additionally, by simplifying the hardware architecture and avoiding computationally intensive trigonometric calculations, the implementation can optimize the performance and resource requirements.

2.3 Cover Tape Offset in Tap and Reel Process

Chung-Feng Jeffrey Kuo et al. presented an automated optical inspection system for surface mount device light emitting diodes. The paper highlighted a machine vision system that utilizes various image processing techniques, such as binarization,

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

morphology, and feature extraction, to detect defects in SMD LEDs which consists of a high-resolution camera, lighting sources, and image processing software [1]. The research has a success rate of 97.5% in detecting LED defects, demonstrating the effectiveness of the proposed approach. In addition, the predictive maintenance aspect of the proposed system also aligns with the concept of automation, as it helps prevent machine breakdowns before they occur, reducing downtime and maintenance costs. However, the paper only focusses on the development of an automated optical inspection system and demonstrates the effectiveness of using vision systems for only defect detection. Therefore, this study will focus on the concepts and techniques to develop a low-cost monitoring system for cover tape offset detection and predictive maintenance in the tape and reel process.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Moreover, Lynn Khine and Joel C. Alimagno discuss the die sticking quality issue of tape-and-reel packaging for Wafer-Level Chip-Scale Package (WLCSP) [11]. The WLCSP package is widely used in microelectronics industries due to its small form factor, light weight, and low cost. The paper highlighted a method to investigate the die sticking issue in WLCSP by using a high-speed camera to capture the real-time images of the tape-and-reel process. However, the paper only focusses on identifying the critical stage where the die sticking occurred, which was found to be caused by the excessive deformation of the cover tape during the peeling process. Therefore, this study will focus on monitoring the cover tape offset, which the system can help to prevent die sticking during the tape-and-reel process. The system can reduce yield loss and improve production efficiency.

Shuai Qiao et al. presented a tape and reel single side peel force test verification that discusses the importance of ensuring the quality of the tape and reel packaging process in electronic manufacturing [2]. In this process, cover tape is used to seal the components in a carrier tape for protection and transportation. The cover tape is sealed by an adhesive, and the single side peel force (SSPF) test is used to measure the strength of this adhesive. The paper highlights the significance of the SSPF test in verifying the tape and reel packaging quality. The test can identify whether the cover tape is properly sealed and can detect any variations in peel force that may indicate the presence of defects or issues such as cover tape offset.

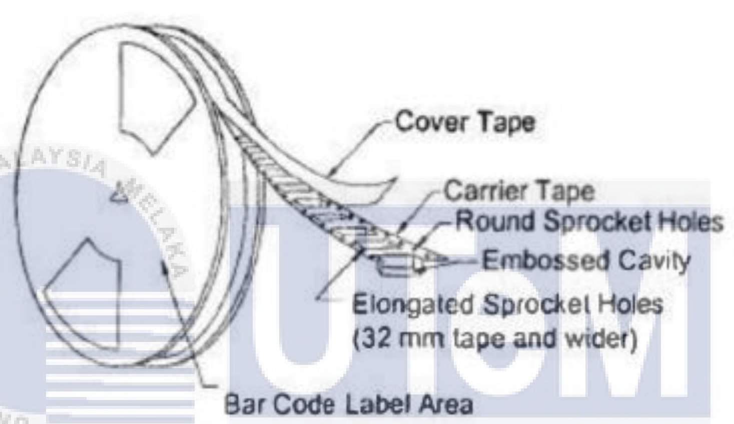


Figure 2.3: Tape and Reel Structure [2].

اونيۆرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

However, the paper only focusses on analyzing the results of the SSPF test and manufacturers can take corrective actions to improve the tape and reel packaging process. The proposed project of developing a low-cost monitoring system using Raspberry Pi Camera Module to detect cover tape offset in tape and reel process is highly relevant to this research paper. The SSPF test measures the strength of the adhesive used to seal the cover tape, and the proposed monitoring system can detect variations in the position of the cover tape that may affect the strength of the adhesive. By detecting and analyzing cover tape offset, the monitoring system can provide early warning of potential defects or issues in the tape and reel packaging process, allowing for timely corrective actions to be taken.

Alejandro Gallegos-Hernandez and Francisco J. Ruiz-Sanchez present a 2D automated visual inspection system for the remote quality control of surface mount device (SMD) assembly [12]. The system is designed to detect defects such as missing components, component misplacement, and polarity errors, among others. In the proposed project, diffuse-reflective sensor is used to detect cover tape offset in tape and reel process, whereas in the Gallegos-Hernandez and Ruiz-Sanchez paper, a camera is used to capture images of SMD board for defect detection. Moreover, the proposed project aims to develop a low-cost monitoring system for detecting cover tape offset, while the Gallegos-Hernandez and Ruiz-Sanchez paper presents a remote quality control system for SMD assembly. Both the proposed project and the Gallegos-Hernandez and Ruiz-Sanchez paper aim to improve the quality of electronic components, which is crucial for the reliability and durability of electronic devices.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Yuqiao Cen et al. presented a Defect patterns study of pick-and-place machine using automated optical inspection data assembly that discusses the use of automated optical inspection (AOI) to detect defects in the pick-and-place process of surface mount devices (SMDs) [13]. The paper highlighted that AOI systems are widely used in SMD assembly lines to improve quality control by detecting defects such as missing components, misplaced components, and soldering defects. However, AOI systems can be limited in their ability to detect defects that occur during the pick-and-place process, such as defects caused by the offset of cover tape in tape and reel packaging. Additionally, the development of a predictive maintenance system based on the situation can improve the overall efficiency of the SMD assembly process by reducing downtime and minimizing the need for manual inspection and maintenance.

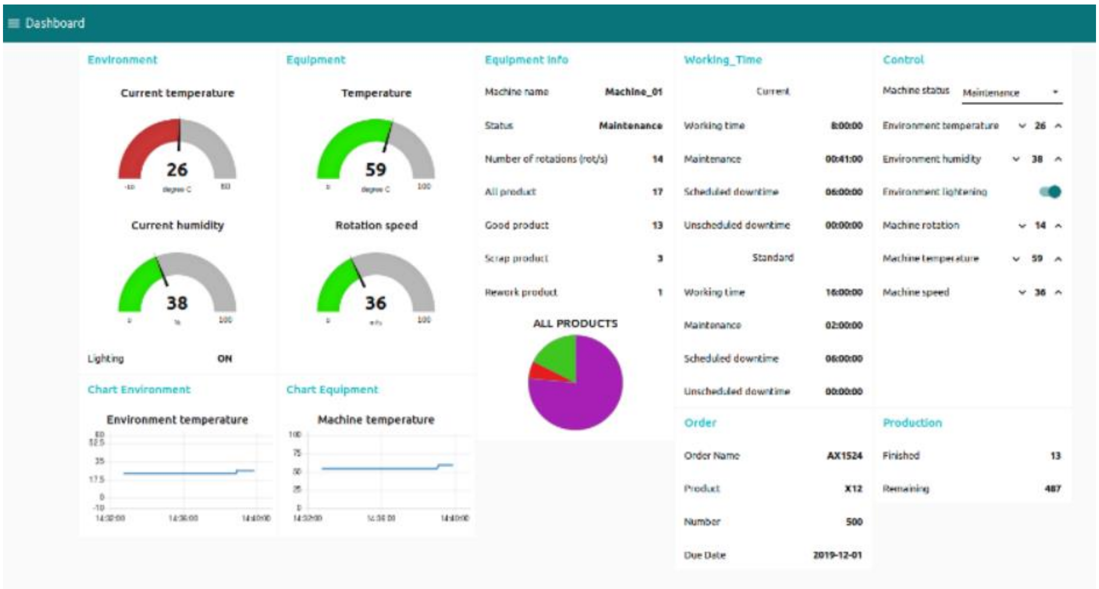
2.4 Node-RED

This literature review aims to compare the applications of Node-RED in two distinct domains: industrial environments for cover tape offset detection and predictive maintenance, as discussed in the research paper by Katalin Ferencz and József Domokos, and healthcare systems for connecting patients, staff, and medical equipment, as explored in the research paper by Junior Asante and Joel Olsson. By examining the similarities and differences between these two domains, this review seeks to provide insights into the versatility and effectiveness of Node-RED in diverse application areas.

Both research papers highlight the benefits of using Node-RED as a data integration and visualization platform. Node-RED's visual programming interface simplifies the development process, enabling the creation of event-driven applications and workflows. Its ability to connect disparate data sources, devices, and services allows for seamless data integration and interoperability. The customizability and scalability of Node-RED make it adaptable to the evolving needs of both industrial and healthcare environments. Additionally, Node-RED's real-time visualization capabilities facilitate efficient decision-making and enhance overall system performance.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

In the industrial context, the research paper by Ferencz and Domokos focuses on detecting cover tape offset and performing predictive maintenance using Node-RED. The research emphasizes the benefits of Node-RED in creating a user-friendly dashboard for monitoring and analyzing cover tape offset data [14].



Lighting ON

Machine name Machine_01
Status Maintenance
Number of rotations (rot/s) 14
All product 17
Good product 13
Scrap product 3
Rework product 1

Current
Working time 8:00:00
Maintenance 00:41:00
Scheduled downtime 06:00:00
Unscheduled downtime 00:00:00
Standard
Working time 16:00:00
Maintenance 02:00:00
Scheduled downtime 06:00:00
Unscheduled downtime 00:00:00

Machine status Maintenance
Environment temperature 26
Environment humidity 38
Environment lightening
Machine rotation 14
Machine temperature 59
Machine speed 36

Order Name AX1524
Product X12
Number 500
Due Date 2019-12-01

Finished 13
Remaining 487

Figure 2.4: Node-RED Dashboard [14].

The research paper by Asante and Olsson explores the application of Node-RED in healthcare systems to connect patients, staff, and medical equipment. Node-RED is utilized to integrate patient monitoring devices, enabling real-time data collection and transmission for enhanced patient care. The platform also streamlines staff communication by integrating various communication channels, facilitating effective collaboration and prompt response to emergencies [15]. Moreover, Node-RED enables the integration of medical equipment, allowing for real-time monitoring, predictive maintenance, and data-driven decision-making in healthcare settings.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

While both applications share common benefits and features of Node-RED, there are notable domain-specific differences. In the industrial domain, the focus is on detecting and preventing issues related to cover tape offset, contributing to product quality and reliability. On the other hand, in the healthcare domain, Node-RED plays a vital role in connecting patients, staff, and medical equipment, enhancing patient care, and improving operational efficiency. The integration of medical devices and analytics platforms in healthcare applications enables data-driven decision support and personalized treatment [15].

In conclusion, Node-RED demonstrates its versatility and effectiveness in diverse application areas, as highlighted by the research papers reviewed. In industrial environments, Node-RED aids in cover tape offset detection and predictive maintenance, while in healthcare systems, it connects patients, staff, and medical equipment. Both applications leverage Node-RED's capabilities for seamless data integration, real-time visualization, and interoperability. The comparison of these applications emphasizes the adaptability of Node-RED to different domains and underscores its potential for enhancing processes and outcomes in various industries.

2.5 Training Algorithms

This section explores key studies that utilize various training techniques, including Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG), in diverse applications such as load forecasting, crosstalk analysis, and neural network training. In the context of machine learning, training algorithms are computational procedures that enable a model to learn patterns, relationships, and representations from data. These algorithms iteratively adjust the model's parameters based on input data and their corresponding output labels, with the goal of minimizing the difference between predicted and actual outcomes [16]. These studies provide important insights into the importance and applications of the training algorithms, providing foundations for additional research in the context of cover tape offset detection and predictive analysis in this project.

A. Abbasi presented Bayesian Regularization Based Electrical Load Forecasting that discusses the use of Bayesian Regularization in short-term load forecasting. The paper highlights that accurate short-term load forecasting is crucial for optimized load generation planning, decreasing the generation-demand gap, and reducing power losses [17]. The paper also discusses the performance of different training techniques, including Bayesian Regularization, Scaled Conjugate Gradient, and Levenberg–Marquardt [17].

Y. Wang presented Application of Neural Network Analysis Based on Bayesian Regularization in Crosstalk of Cable that discusses the use of an artificial neural network based on Bayesian regularization training function for the internal twisted pair electromagnetic crosstalk system of aircraft. The prediction results show that the prediction value of the neural network based on Bayesian regularization training function is close to the input value, and the regression analysis shows that the prediction reliability of the network is high [18]. In relation to the project, this article can provide valuable insights into the application of Bayesian Regularization in predictive models. Specifically, the methodologies used in the paper which to apply Bayesian Regularization in the Neural Net Time Series Apps for predictive maintenance. The comparison of different training techniques in the paper can be referred to for the analysis of Bayesian Regularization and Scaled Conjugate Gradient (SCG) in your project.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Furthermore, Murat Kayri concludes that the Bayesian regularization training algorithm shows better performance than the Levenberg-Marquardt algorithm. The objective of this study is to compare the predictive ability of Bayesian regularization with Levenberg-Marquardt Artificial Neural Networks. The study examines the best architecture of neural networks by testing one-, two-, three-, four-, and five-neuron architectures. MATLAB was used for analyzing the Bayesian regularization and

Levenberg-Marquardt learning algorithms. Bayesian regularization artificial neural networks have the advantage of revealing potentially complex relationships, making them suitable for quantitative studies and providing a robust model [19].

The paper by Thibaut Le Magueresse discusses the application of instantaneous Bayesian regularization to RT-NAH for the reconstruction of non-stationary sound sources using a planar microphone array. It introduces Bayesian estimation of the regularization parameter based on prior knowledge of the problem, allowing for the consideration of fluctuating properties of the sound field [20]. The superiority of Bayesian regularization over state-of-the-art methods is observed numerically and experimentally for the reconstruction of non-stationary sources. This could be relevant to the project as it involves the use of Bayesian regularization that could potentially improve the accuracy of the predictive maintenance system.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Bayesian regularization improves accuracy by providing a more robust and adaptive approach to regularization in the reconstruction of non-stationary sources [20]. It introduces the estimation of the regularization parameter based on prior knowledge of the problem, allowing for the consideration of fluctuating properties of the sound field. Compared to state-of-the-art methods, Bayesian regularization has been observed to offer superior results in the reconstruction of non-stationary sources.

Furthermore, the paper by Al Bataineh compares the performance of three algorithms, Levenberg-Marquardt, Bayesian Regularization, and Scaled Conjugate Gradient, in training artificial neural networks using the housing dataset. The algorithms are evaluated based on their ability to fit curves to the data, measured using Mean Square Error (MSE). The mean squared error is computed for each algorithm, with Levenberg-Marquardt having an MSE of 7.0902, Scaled Conjugate Gradient at

15.2932, and Bayesian Regularization at 5.3480 [16]. It was found that Bayesian Regularization gave the best accuracy at 96.78%, followed by Levenberg-Marquardt at 94.53% and Scaled Conjugate Gradient at 90.51%.

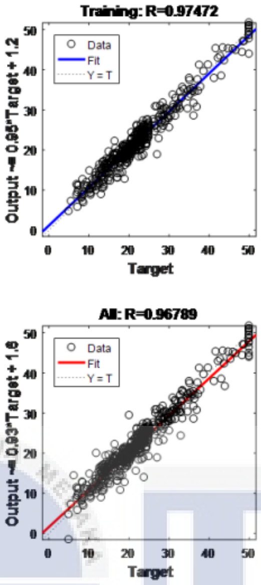


Figure 2.5: Regression Plot for BR Algorithm [16].

اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

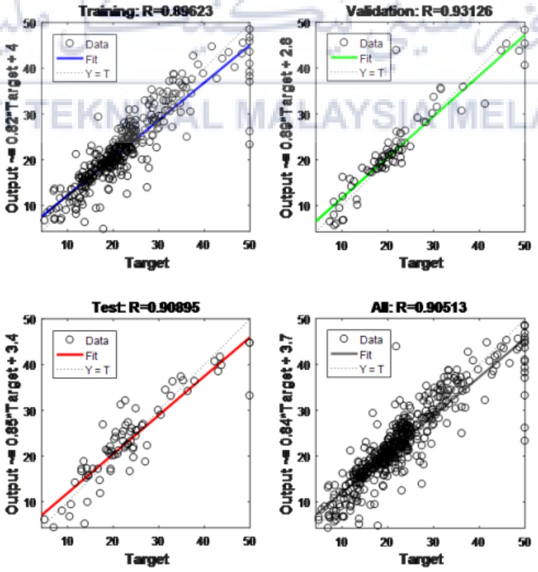


Figure 2.6: Regression Plot for SCG Algorithm [16].

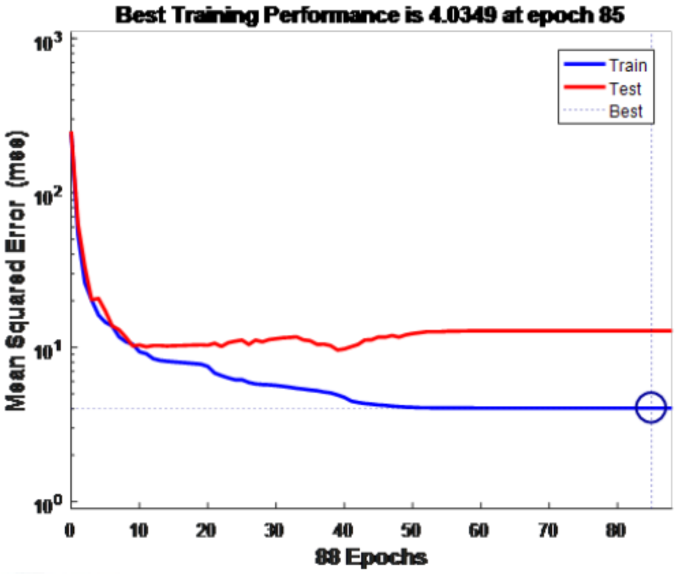
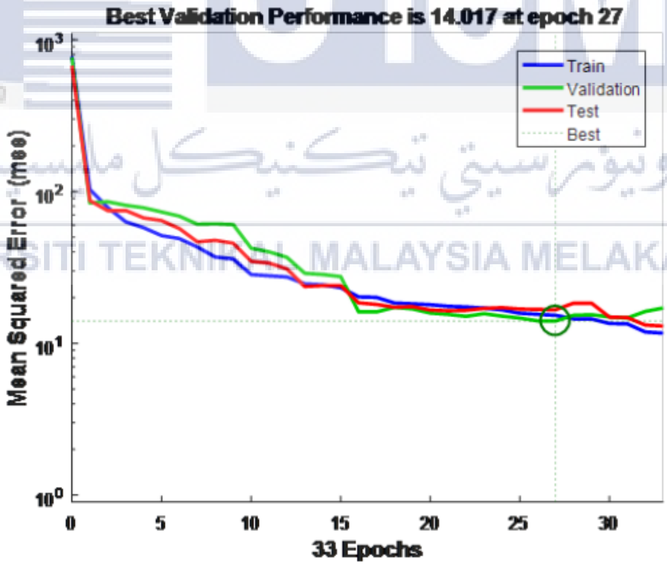


Figure 2.7: The Performance of BR Algorithm [16].



اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 2.8: The Performance of SCG Algorithm [16].

Based on the visualized plots, Bayesian Regularization is the optimal solution for achieving higher prediction accuracy. However, it requires more iterations for training and is more time-consuming compared to other algorithms. The Scaled Conjugate

Gradient algorithm is less accurate than the other techniques mentioned, however it has a faster training time than Bayesian Regularization. The Scaled Conjugate Gradient algorithm is advantageous because of its lower memory consumption.

2.6 Summary

Table 2.1: Method Summary

Reference	Year	Title	Summary Method
[1]	2016	Automated optical inspection system for surface mount device light emitting diodes	Chung-Feng Jeffrey Kuo et al. introduced an automated optical inspection system for detecting defects in surface mount device light-emitting diodes (SMD LEDs).
[11]	2019	Die sticking quality issue of tape-and-reel packaging for WLCSP	Lynn Khine and Joel C. Alimagno address the die-sticking quality issue in tape-and-reel packaging for Wafer-Level Chip-Scale Packages (WLCSP).
[2]	2016	Tape & Reel single side peel force test verification	Shuai Qiao and collaborators introduced a tape and reel single-side peel force (SSPF) test verification method to ensure the quality of the tape and reel packaging process
[12]	2002	2D automated visual inspection system for the remote quality control of SMD assembly	Alejandro Gallegos-Hernandez and Francisco J. Ruiz-Sanchez introduced a 2D automated visual inspection system tailored for remote quality control of surface mount device (SMD) assembly.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[13]	2022	Defect patterns study of pick-and-place machine using automated optical inspection data	Yuqiao Cen and team explore defect patterns in pick-and-place machines during surface mount device (SMD) assembly, utilising automated optical inspection (AOI) data.

Table 2.1 provides an extensive background study for the project, focusing on the significance of studies on cover tape offset detection, contributing unique insights, and highlighting the need for further research to overcome challenges and improve the performance of these techniques.

Furthermore, the chapter also includes studies on image processing techniques, specifically Canny edge detection and the Hough Line Transform in offset detection within the tap and reel process. It provides the relevance of these methods in manufacturing and quality control, highlighting their crucial role in detecting cover tape misalignments. Node-RED's versatility in data integration and visualisation for industrial applications, such as cover tape offset detection and healthcare systems, is explored.

Furthermore, the chapter also includes a brief section comparing Scaled Conjugate Gradient with Bayesian Regularization, emphasising their roles in training neural networks.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 3

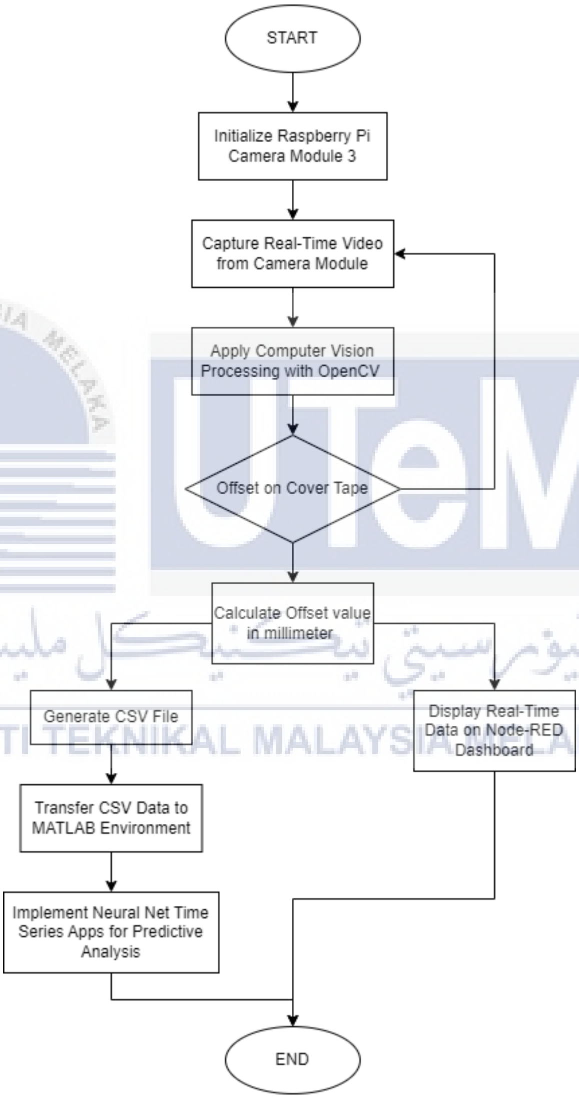
METHODOLOGY

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The methodology section is important in detecting cover tape offset or misalignment from the carrier tape during the tape and reel process using computer vision and neural network techniques. It outlines the systematic approach taken to answer the research question. This part is an organised roadmap outlining the sequential actions and techniques to achieve the study's objectives.

The flowchart in Figure 3.1 outlines the project process, which aims to detect cover tape offset or misalignment from the carrier tape during the tape and reel process. The process begins with the initialisation of the Raspberry Pi Camera Module 3, which captures real-time video of the ongoing process. This video feed is then subjected to computer vision processing using the OpenCV library. The system checks for any offset on the cover tape. The system calculates the offset value in millimetres if an offset is detected.

This information is then used in two ways, which is, it is recorded in a CSV file, and it is displayed in real-time on a Node-RED dashboard. The CSV data is subsequently transferred to a MATLAB environment. The data is implemented in MATLAB on Neural Net Time Series Apps for predictive analysis. This marks the completion of the process.


START
Initialize Raspberry Pi Camera Module 3
Capture Real-Time Video from Camera Module
Apply Computer Vision Processing with OpenCV
Offset on Cover Tape
Calculate Offset value in millimeter
Generate CSV File
Display Real-Time Data on Node-RED Dashboard
Transfer CSV Data to MATLAB Environment
Implement Neural Net Time Series Apps for Predictive Analysis
END

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.1: Flowchart for the whole process.

3.1 Measurement from Industry

Analysing the offset's measurement from the industry is an important step in the project. It helps to understand the industry standards and requirements related to detecting cover tape offset or misalignment during the tape-and-reel process. In the industry, detecting cover tape offset or misalignment during the tape-and-reel process is critical to ensure the quality of the packaged components. Any offset greater than 8mm would be considered a defect and could result in the rejection of the packaged components, as shown in Figure 3.2.

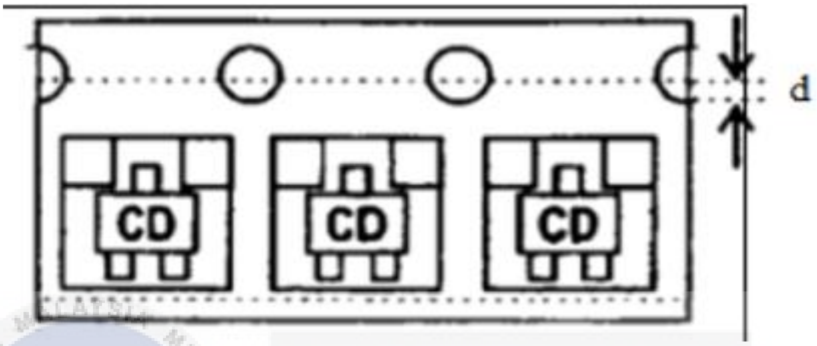


Figure 3.2: The offset's measurement visualisation.

The cover tape also can be in the offset position with more than 20mm towards the sprocket hole as shown in Figure 3.3.

UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

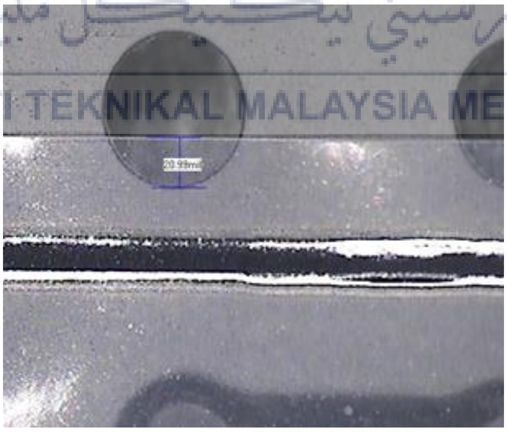


Figure 3.3: Example visualisation of offsets in the industry.

There is also a case where the cover tape offset is away from the sprocket hole as shown in Figure 3.4.

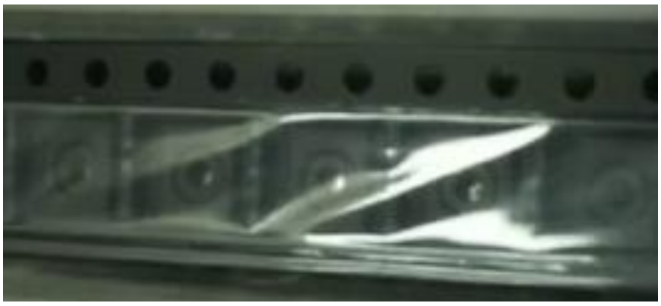


Figure 3.4: Cover tape position away from sprocket hole.

In the context of the project, understanding the industry standards for cover tape offset measurement is important to design a system that meets these requirements. Mainly, the cover tape should be precisely at the very bottom area of the sprocket hole. The Raspberry Pi Camera Module 3 used in the project should be capable of providing accurate measurements of the cover tape position within the industry-standard tolerance levels. This will ensure that the components meet the industry standards.

For the testing phase of the system, the camera was not evaluated on an actual tape and reel process. Instead, the testing was conducted on the conveyor belt of the FMS 200 as shown in Figure 3.5, which has some misalignments that can be evaluated and analyzed. This serves as a good initial test before implementing it in the actual tape and reel process.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

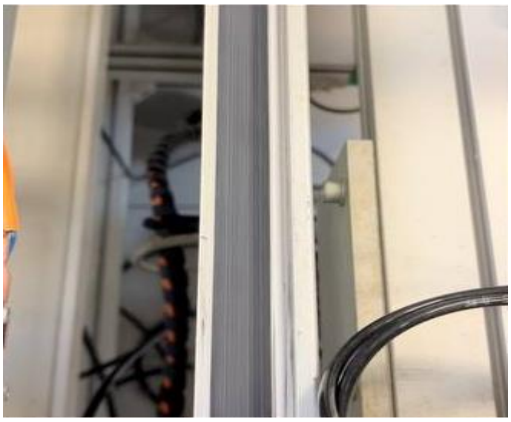


Figure 3.5: Conveyer belt on FMS 200.

3.2 Initialize Raspberry Pi Camera Module

In this initial phase, the Raspberry Pi Camera Module is set up to capture real-time data during the tape and reel process. Figure 3.6 shows the setup for the camera with the support of camera case and adjustable arm to protect the camera and to easily make the camera mounted in the industry.

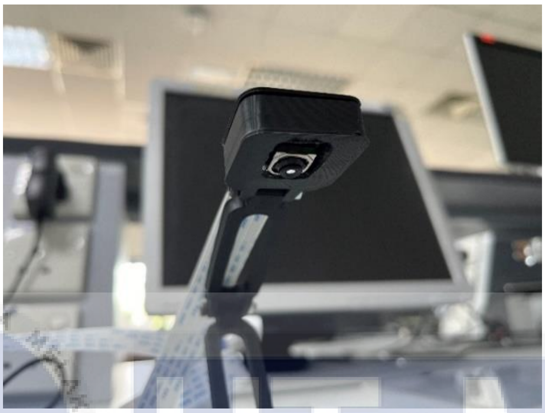


Figure 3.6: Camera's design setup.

UTeM UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The configuration involves ensuring proper connectivity, initializing the camera module, and configuring settings such as resolution and frame rate. This step establishes the foundation for subsequent image acquisition and processing. Figure 3.7 shows the connection from the camera to the Raspberry Pi via CSI port.

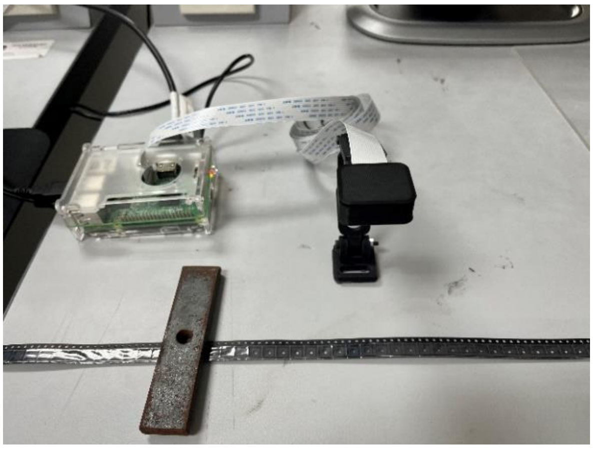


Figure 3.7: Connection from the camera to the Raspberry Pi.

Figure 3.8 shows the Raspberry Pi Camera Module 3 installed on the FMS 200, precisely positioned to observe the conveyor belt. The camera’s lens focuses on the ongoing processes.

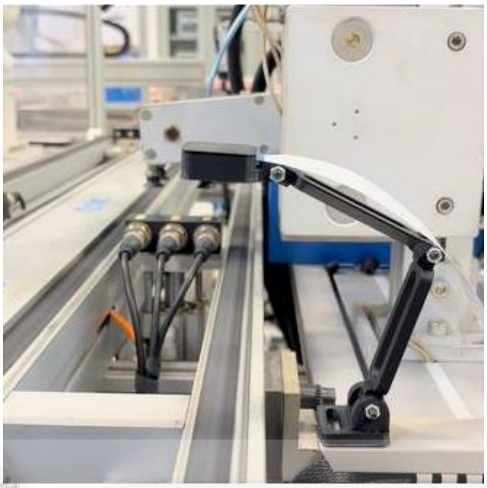


Figure 3.8: Raspberry Pi Camera Module 3 installed on the FMS 200.

Figure 3.9 shows the Raspberry Pi Camera Module 3 was connected to the Raspberry Pi via the CSI port on the FMS 200 machine.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

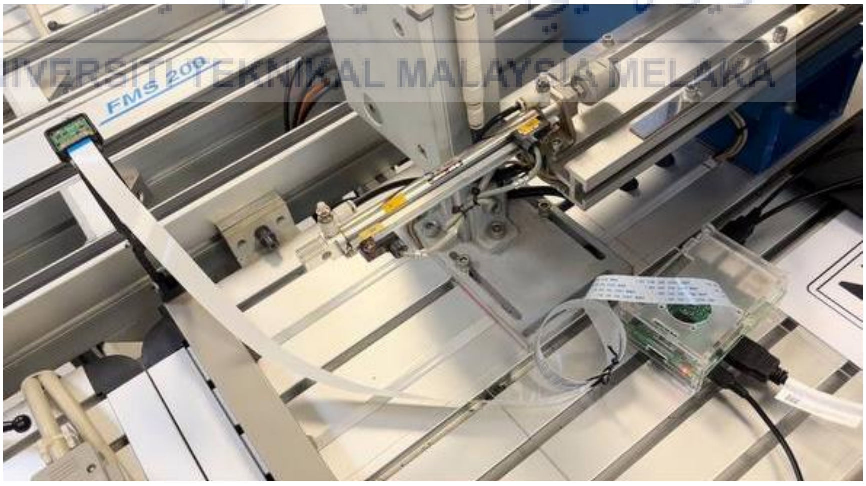
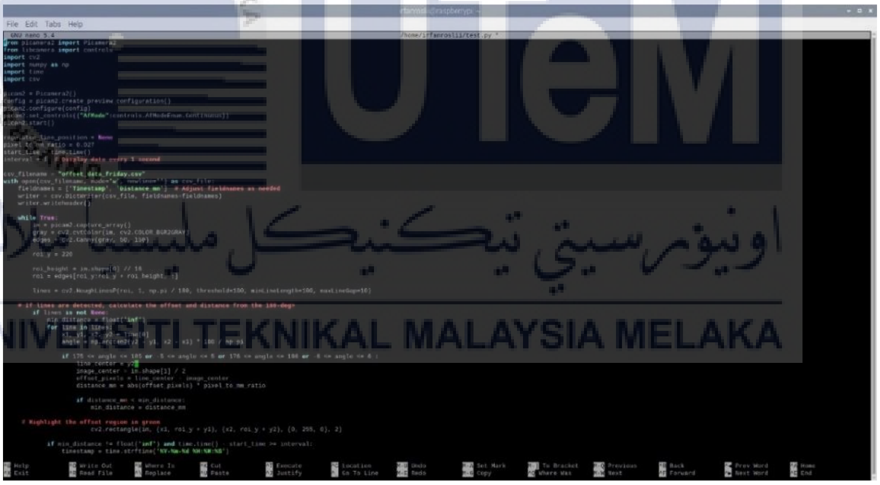


Figure 3.9: Camera connection to the Raspberry Pi.

3.3 Computer Vision Processing

Utilizing the OpenCV library, this section focuses on image processing techniques to extract meaningful information from the captured frames. This step aims to enhance the quality of the captured images and prepare them for subsequent analysis.

OpenCV, a powerful computer vision library, offers the Canny edge detection method as a robust technique for identifying edges within images [21]. Moreover, it provides the Hough Line Transform method, which plays a pivotal role in identifying the lines that represent the boundaries of the tape [9]. Figure 3.10 shows the Python script on Terminal where the code started with the declaration of OpenCV that used to implement the computer vision processing.



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 3.10: Python script on Terminal.

3.4 Canny Edge Detection

Within the broader context of computer vision processing, Canny Edge Detection is specifically employed to identify edges within the images. This technique enhances the delineation of objects in the images, providing a foundation for subsequent steps in the analysis [22].

In python script, the lower threshold for the edges was set to 50. Any gradient value below this threshold is considered not to be an edge. Meanwhile, the higher threshold for the edges was set to 150. Any gradient value above this threshold is considered to be a strong edge, and values between the lower and higher thresholds are considered to be weak edges unless they are connected to strong edges.

3.5 Hough Line Transform

Building on the results of Canny Edge Detection, the Hough Line Transform is applied to detect lines within the images. This step is crucial for identifying features such as the alignment and orientation of the cover tape during the tape and reel process [23].

Figure 3.11 shows the detection of straight lines on the selected region of interest. The most prominent feature in this image is the green line, which represents the straight line detected by the Hough line detection algorithm.

MALAYSIA MELAKA
UTeM
اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

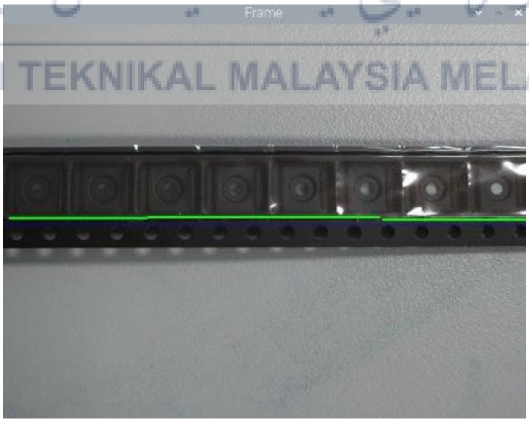


Figure 3.11: Hough Line detection on the selected Region of Interest.

3.6 Display Real-Time Data on Node-Red Dashboard

To facilitate real-time monitoring, the captured and processed data is seamlessly integrated into Node-Red. Figure 3.12 shows the Node-RED workflow. The “Camera” node is used to capture images or data from the Raspberry Pi Camera Module 3. This data is then passed to the next node in the flow. Then, the “debug 2” node used to display messages during development or testing phases. It might be showing real-time data or logging information about the cover tape alignment process. Finally, all data will be passed to the dashboard node to generate a dashboard in the graph style.

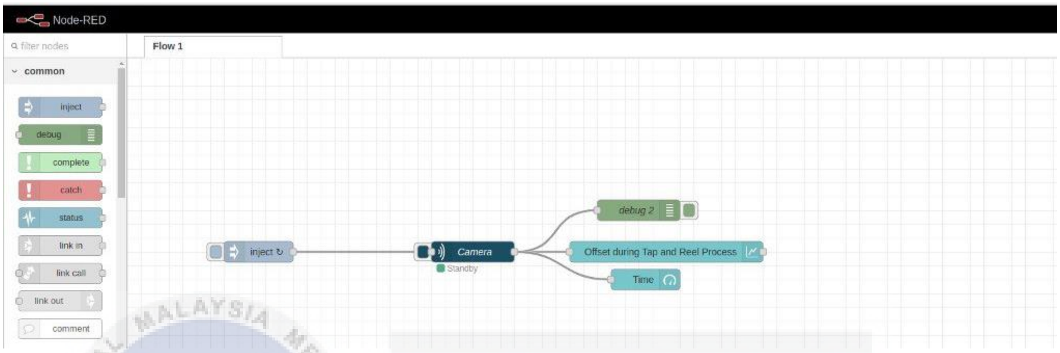


Figure 3.12: Node-RED workflow.

Figure 3.13 shows the dashboard generated after the data was passed to the Node-RED. A custom dashboard was created to visually represent the real-time information, allowing for easy observation and analysis of the cover tape alignment during the process. The graph on the dashboard, titled “Offset during Tap and Reel Process”, appears to be showing fluctuations in offset over time. The offset values range approximately from 8.2 mm to 8.7 mm. This could represent the variation in the alignment of the cover tape during the tape and reel process. If the offset value is high, it means the cover tape is not properly aligned with the carrier tape.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

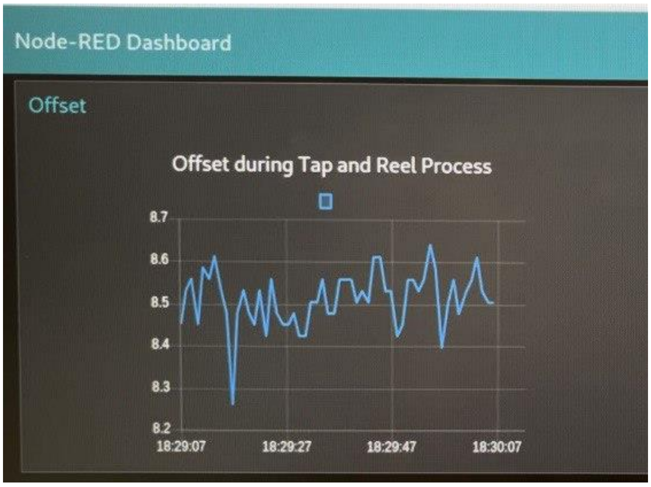


Figure 3.13: Dashboard generated from Node-RED.

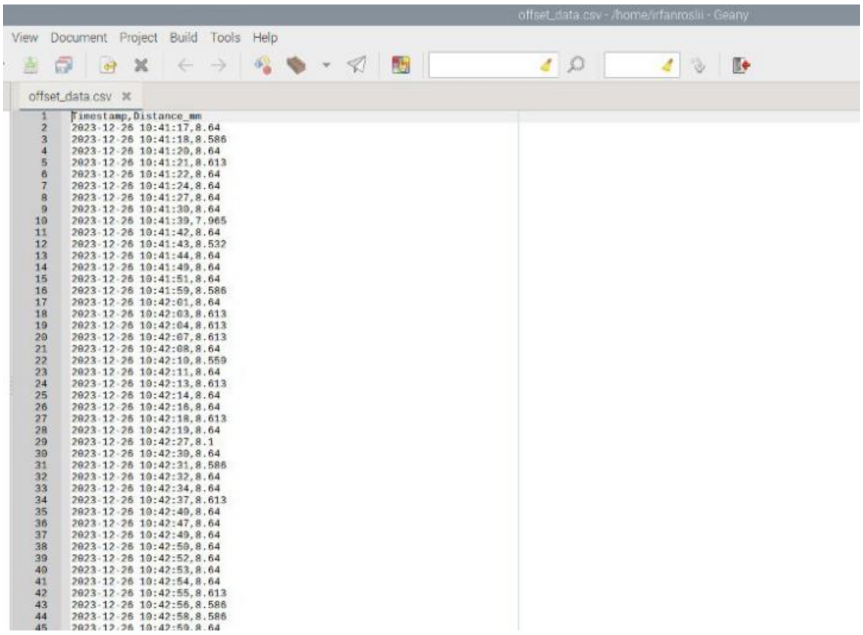
Figure 3.14 shows a real-time monitoring view of the captured images from the conveyor belt and the display of the offset values on the Node-RED dashboard in a graph form.

Figure 3.14: Real-time monitoring view.

3.7 Generate CSV File

Simultaneously with real-time data display, the processed data is logged into a CSV file as shown in Figure 3.15. This file serves as a structured repository of the captured information, enabling further analysis and providing a basis for data transfer to external platforms, such as MATLAB, for advanced analysis.

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA



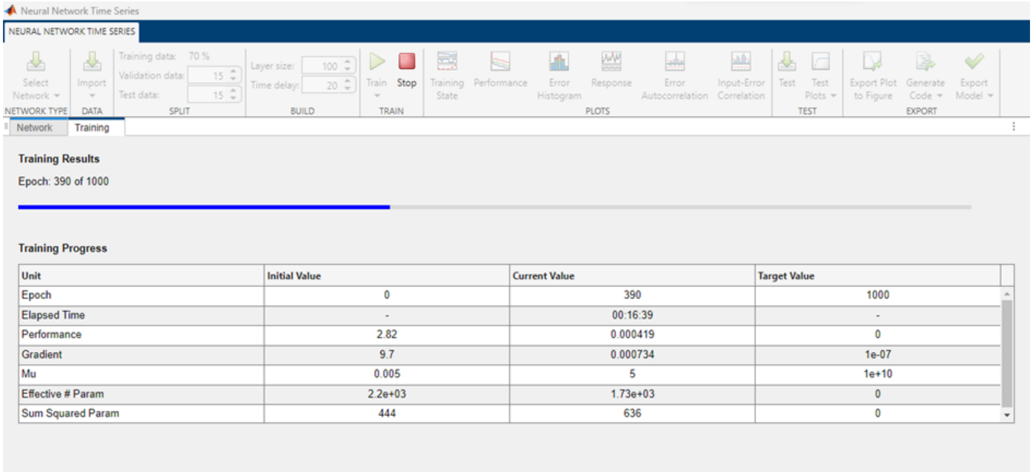
```
Timestamp,Distance_mm
2023-12-26 10:41:17,8.64
2023-12-26 10:41:18,8.586
2023-12-26 10:41:20,8.64
2023-12-26 10:41:21,8.613
2023-12-26 10:41:22,8.64
2023-12-26 10:41:24,8.64
2023-12-26 10:41:27,8.64
2023-12-26 10:41:30,8.64
2023-12-26 10:41:39,7.965
2023-12-26 10:41:42,8.64
2023-12-26 10:41:43,8.532
2023-12-26 10:41:44,8.64
2023-12-26 10:41:49,8.64
2023-12-26 10:41:51,8.64
2023-12-26 10:41:59,8.586
2023-12-26 10:42:01,8.64
2023-12-26 10:42:03,8.613
2023-12-26 10:42:04,8.613
2023-12-26 10:42:07,8.613
2023-12-26 10:42:08,8.64
2023-12-26 10:42:10,8.559
2023-12-26 10:42:11,8.64
2023-12-26 10:42:13,8.613
2023-12-26 10:42:14,8.64
2023-12-26 10:42:16,8.64
2023-12-26 10:42:18,8.613
2023-12-26 10:42:19,8.64
2023-12-26 10:42:27,8.1
2023-12-26 10:42:30,8.64
2023-12-26 10:42:31,8.586
2023-12-26 10:42:32,8.64
2023-12-26 10:42:34,8.64
2023-12-26 10:42:37,8.613
2023-12-26 10:42:40,8.64
2023-12-26 10:42:47,8.64
2023-12-26 10:42:49,8.64
2023-12-26 10:42:50,8.64
2023-12-26 10:42:52,8.64
2023-12-26 10:42:53,8.64
2023-12-26 10:42:54,8.64
2023-12-26 10:42:55,8.613
2023-12-26 10:42:56,8.586
2023-12-26 10:42:58,8.586
2023-12-26 10:42:59,8.64
```

Figure 3.15: Data logged into CSV file.

3.8 Neural Net Time Series Apps for Predictive Analysis

This section involves integrating the CSV data into MATLAB and implementing Neural Net Time Series Apps for predictive maintenance analysis. A key focus is placed on the comparison analysis between two distinct training algorithms, Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG). Figure 3.16 shows the process of implementing a training algorithm using MATLAB.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



Training data: 70 %
Validation data: 15
Test data: 15
Layer size: 100
Time delay: 20

Training Results

Epoch: 390 of 1000

Training Progress

Unit	Initial Value	Current Value	Target Value
Epoch	0	390	1000
Elapsed Time	-	00:16:39	-
Performance	2.82	0.000419	0
Gradient	9.7	0.000734	1e-07
Mu	0.005	5	1e+10
Effective # Param	2.2e+03	1.73e+03	0
Sum Squared Param	444	636	0

Figure 3.16: Implementation of Training Algorithm using MATLAB.

3.9 Summary

In summary, the methodology outlined in this section provides a comprehensive and systematic approach for detecting cover tape offset or misalignment during the tape and reel process using computer vision and neural network techniques. The use of Canny Edge Detection and Hough Line Transform enhances the analysis of captured frames, and real-time data is displayed on a Node-RED dashboard while being logged into a CSV file for further analysis. The final step involves implementing Neural Net Time Series Apps in MATLAB for predictive maintenance analysis, comparing the efficiency of Bayesian Regularization and Scaled Conjugate Gradient algorithms.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

CHAPTER 4

RESULTS AND DISCUSSION

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

In this analysis, a comparison is performed between two different training techniques, Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG), to determine their impact on the prediction model's performance. BR is well-known for its ability to reduce overfitting by using a probabilistic approach to weight regularization [24]. SCG, a quasi-Newton optimization approach, is well-known for its ability to quickly reach minima on the error surface [25]. Both techniques are implemented in the Neural Network Time Series applications in MATLAB, and their impacts on training dynamics and overall forecast accuracy are examined. Furthermore, this analysis also discuss about the results in Node-RED's graph along with the behavior of it.

4.1 Node-RED

The graph on the Node-RED dashboard as shown in Figure 4.1 displays fluctuations in offset values over a specific time period, allowing for monitoring and analysis of variations that occur during this process.

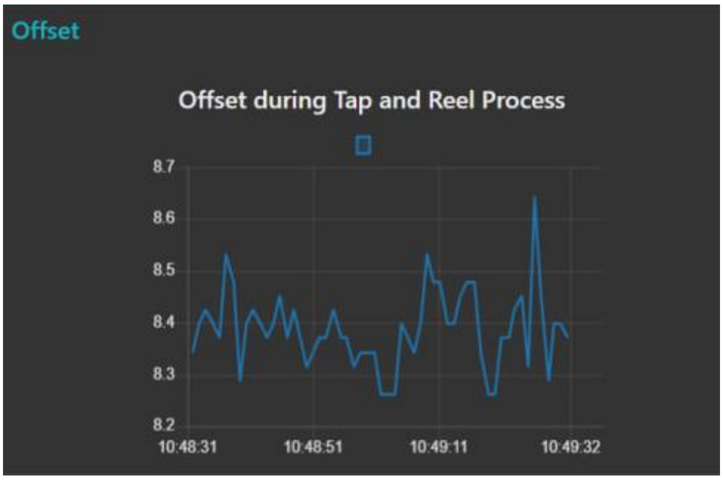


Figure 4.1: Offset graph in Node-RED dashboard.

The offset values fluctuate between 8.2 mm and 8.7 mm. This range could be indicating the variability in the alignment of the cover tape during the tape and reel process. This also suggests that the alignment of the cover tape is not constant and varies throughout the process. Apart from that, there is a noticeable spike in the offset value.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

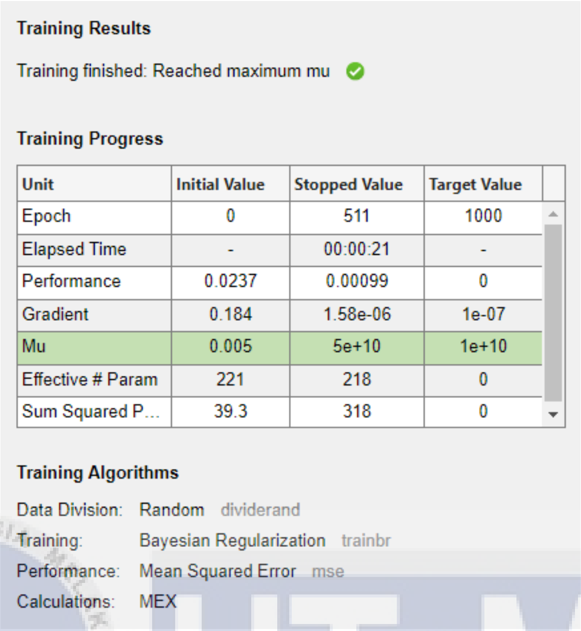
The Fluctuations and the noticeable spike in the offset values could be due to wear and tear of the conveyor belt on the FMS 200 machine. As the conveyor belt wears out, it can cause misalignments or variations in the alignment. By detecting changes in the offset, prediction can be made for potential issues with the conveyor belt and corrective action can be taken before failure occurs [26].

4.2 Bayesian Regularization (BR)

4.2.1 Training results

For this section, the training results were shown to evaluate the performance of the model. By looking at metrics such as accuracy, loss, or mean squared error, the model can understand how well our model is learning and predicting. The training algorithm used was Bayesian Regularization (trainbr), and the performance metric used was

Mean Squared Error (MSE). Figure 4.2 shows the training results from the data for day 1.



Training Results

Training finished: Reached maximum mu ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	511	1000
Elapsed Time	-	00:00:21	-
Performance	0.0237	0.00099	0
Gradient	0.184	1.58e-06	1e-07
Mu	0.005	5e+10	1e+10
Effective # Param	221	218	0
Sum Squared P...	39.3	318	0

Training Algorithms

Data Division: Random dividerand
Training: Bayesian Regularization trainbr
Performance: Mean Squared Error mse
Calculations: MEX

Figure 4.2: Training results in Day 1 (BR).

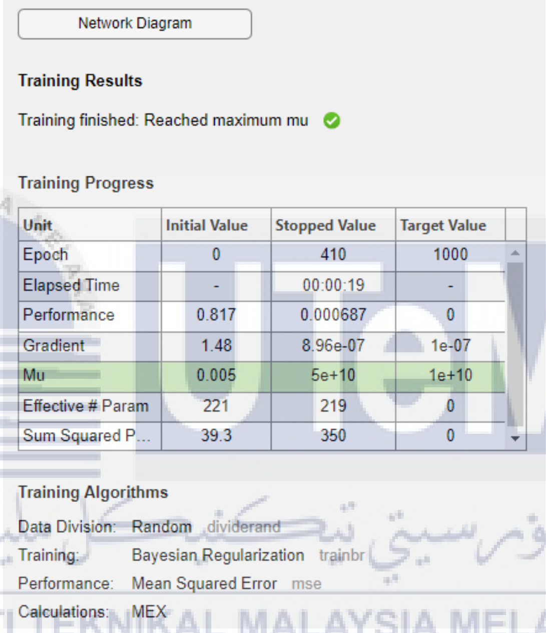
The training stopped at epoch 511, before reaching the target value of 1000 epochs. This could be due to reaching the maximum μ (mu), which is a parameter used in the BR algorithm. The elapsed time for the training was 21 seconds. The performance of the model improved significantly during the training process, from an initial mean squared error (MSE) of 0.0237 to a final MSE of 0.00099. This indicates that the model was able to learn and improve its predictions over time.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The value of μ increased from an initial value of 0.005 to its maximum at 5e+10. This is a parameter in the BR algorithm that controls the balance between fitting the data closely (which risks overfitting) and keeping the weights small (which improves generalization). The fact that μ reached its maximum value suggests that the algorithm was prioritizing generalization over fitting the training data closely [27].

Figure 4.3 shows the training results from the data for day 2. The training stopped at epoch 410, which is earlier than the previous run that stopped at epoch 511. This indicates a faster convergence of the model.

The elapsed time for the training was 19 seconds, which is slightly less than the 21 seconds from the previous run. This could suggest an improvement in computational efficiency.



Training Results

Training finished: Reached maximum mu ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	410	1000
Elapsed Time	-	00:00:19	-
Performance	0.817	0.000687	0
Gradient	1.48	8.96e-07	1e-07
Mu	0.005	5e+10	1e+10
Effective # Param	221	219	0
Sum Squared P...	39.3	350	0

Training Algorithms

Data Division: Random dividerand
Training: Bayesian Regularization trainbr
Performance: Mean Squared Error mse
Calculations: MEX

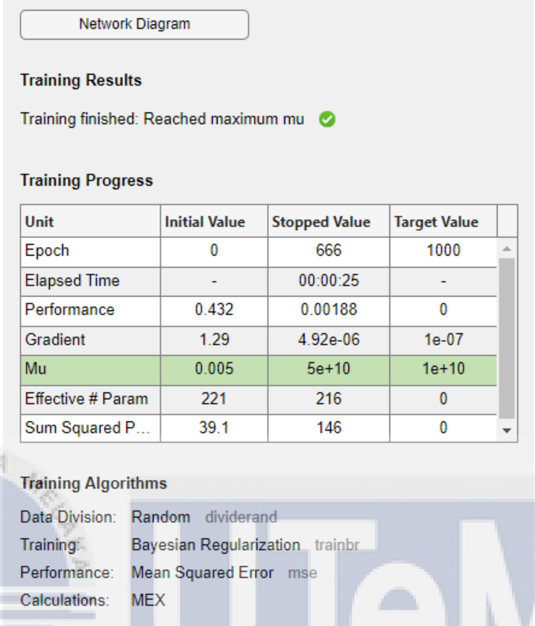
اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.3: Training results in Day 2 (BR).

The performance of the model, as measured by the mean squared error (MSE), improved from an initial MSE of 0.817 to a final MSE of 0.000687. The performance (as measured by MSE) improved significantly on both days, but the starting and ending MSE were different. This could be due to differences in the data or initial conditions.

The value of µ increased from an initial value of 0.005 to its maximum at 5e+10, similar to the previous run. This suggests that the algorithm was prioritizing generalization over fitting the training data closely in both runs.

Figure 4.4 shows the training results from the data for day 3. The training stopped at epoch 666, which is later than the previous run that stopped at epoch 410. This could indicate a slower convergence of the model.



Training Results

Training finished: Reached maximum mu ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	666	1000
Elapsed Time	-	00:00:25	-
Performance	0.432	0.00188	0
Gradient	1.29	4.92e-06	1e-07
Mu	0.005	5e+10	1e+10
Effective # Param	221	216	0
Sum Squared P...	39.1	146	0

Training Algorithms

Data Division: Random dividerand
Training: Bayesian Regularization trainbr
Performance: Mean Squared Error mse
Calculations: MEX

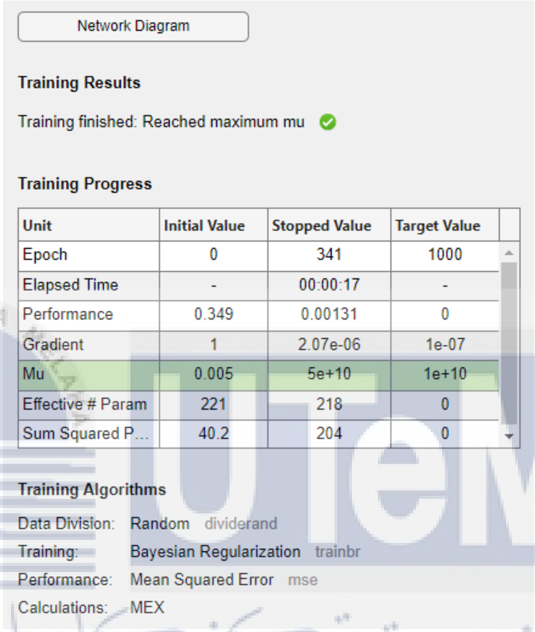
Figure 4.4: Training results in Day 3 (BR).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The elapsed time for the training on Day 3 was 25 seconds, which is longer than the 19 seconds from the previous run. This could suggest a decrease in computational efficiency. The performance of the model, as measured by the mean squared error (MSE), improved from an initial MSE of 0.432 to a final MSE of 0.00188. Moreover, the value of μ increased from an initial value of 0.005 to its maximum at 5e+10, similar to the previous run.

Figure 4.5 shows the training results from the data for day 4. The training stopped at epoch 341, which is earlier than the previous run that stopped at epoch 666. This could indicate a faster convergence of the model.

The elapsed time for the training on Day 4 was 17 seconds, which is shorter than the 25 seconds from the previous run. This could suggest an improvement in computational efficiency. Moreover, the performance of the model, as measured by the mean squared error (MSE), improved from an initial MSE of 0.349 to a final MSE of 0.00131.



Training Results

Training finished: Reached maximum mu ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	341	1000
Elapsed Time	-	00:00:17	-
Performance	0.349	0.00131	0
Gradient	1	2.07e-06	1e-07
Mu	0.005	5e+10	1e+10
Effective # Param	221	218	0
Sum Squared P...	40.2	204	0

Training Algorithms

Data Division: Random dividerand
Training: Bayesian Regularization trainbr
Performance: Mean Squared Error mse
Calculations: MEX

Figure 4.5: Training results in Day 4 (BR).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Based on the training results over the four days as shown in Table 4.1, we can observe that the number of epochs at which training stopped varied each day, indicating differences in the speed of convergence. Furthermore, faster convergence could suggest that the model is learning the patterns in the data more quickly. For computational efficiency, the elapsed time for training decreased over the four days, suggesting improvements in computational efficiency. Moreover, the performance of the model, as measured by the mean squared error (MSE), improved each day. This indicates that the model's predictions are becoming more accurate.

Table 4.1: Performance for 4 days of data using BR

Day	Mean Squared Error (MSE)		Number of Epochs
1	0.0237	0.00099	511
2	0.817	0.000687	410
3	0.432	0.00188	666
4	0.349	0.00131	341

4.2.2 Correlation coefficient (R)

For this section, scatter plot was generated to represent the results of a Bayesian Regularization (BR) training neural network from data for Day 1,2,3 and 4. The scatter plot is a graphical representation of the relationship between the target values and the output values. The scatter plot provides a visual representation of the model's performance. The closer the points are to the diagonal line (y=x), the better the model's predictions are to the actual values [28]. The identity line (Y = T) represents perfect predictions. If all data points were on this line, it would mean that the model's predictions were exactly equal to the actual values.

اونيۏرسيتي تيكنيکل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Based on Figure 4.6, the correlation coefficient is 0.90385, which indicates a strong positive linear relationship between the target and output values. This suggests that the model has a high degree of accuracy in its predictions.

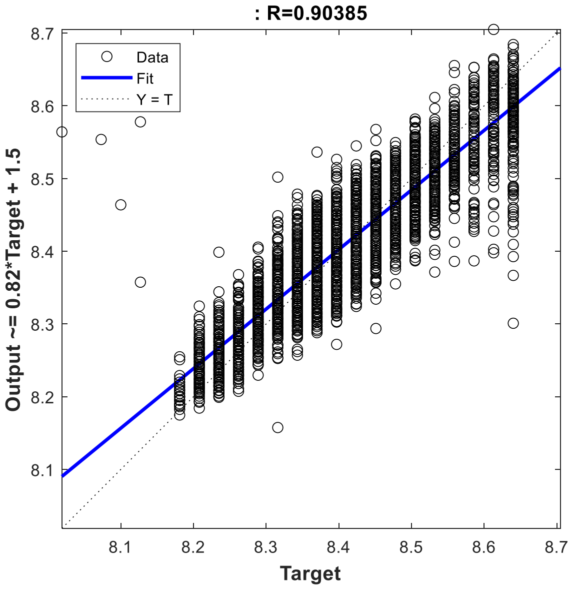


Figure 4.6: Scatter plot for Day 1 (BR).

Figure 4.7 shows scatter plot of a neural network trained using Bayesian Regularization (BR) on Day 2. The correlation coefficient for Day 2 is 0.92842, which is higher than the R value from Day 1 (0.90385).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

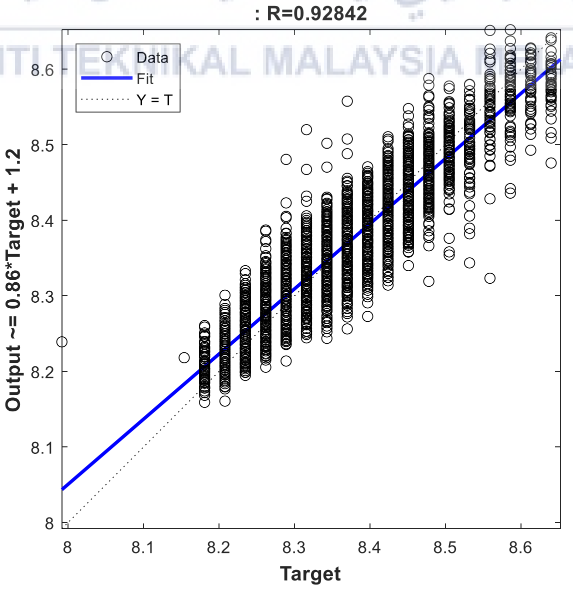


Figure 4.7: Scatter plot for Day 2 (BR).

Figure 4.8 shows scatter plot of a neural network trained using Bayesian Regularization (BR) on Day 3. The correlation coefficient for Day 3 is 0.87387, which is lower than the R value from Day 2 (0.92842). This suggests that the model’s predictions on Day 3 have a slightly weaker linear relationship with the actual values compared to Day 2.

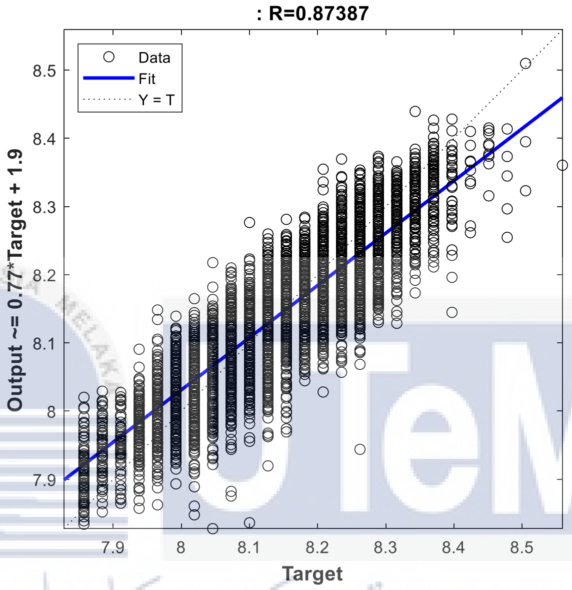


Figure 4.8: Scatter plot for Day 3 (BR).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.9 shows scatter plot of a neural network trained using Bayesian Regularization (BR) on Day 4. The correlation coefficient for Day 4 is 0.9511, which is higher than the R value from Day 3 (0.87387). This suggests that the model’s predictions on Day 4 have a stronger linear relationship with the actual values compared to Day 3.

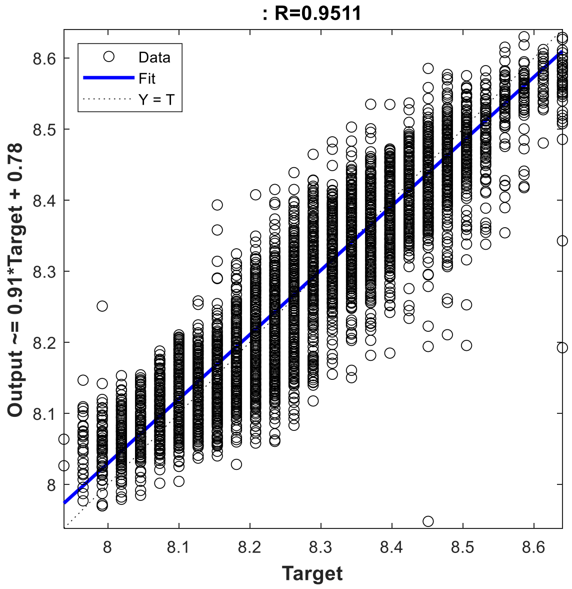


Figure 4.9: Scatter plot for Day 4 (BR).

Over the four days, the model's performance seems to be improving, as indicated by the increasing R value and the data points becoming closer to the fit line as shown in Table 4.2. This suggests that the model is learning effectively, and its predictions are becoming more accurate over time.

Table 4.2: Correlation coefficient comparison for BR training

Day	Correlation Coefficient (R)
1	0.90385
2	0.92842
3	0.87387
4	0.9511

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

4.2.3 Mean Squared Error (MSE)

For this section, a graph representing the Mean Squared Error (MSE) was generated to represent the results of a Bayesian Regularization (BR) training neural network from data for day 1,2,3 and 4.

The graph in Figure 4.10 shows the MSE for both the training and testing datasets over epochs from the data for day 1. The graph indicates that the best training performance, an MSE of approximately 0.00098985, was achieved at epoch 267. The MSE for both the training and testing datasets appears to decrease rapidly during the initial epochs, then continues to decrease more gradually. This suggests that the model is learning effectively from the data, as it can reduce the errors over time [29]. Moreover, both training and testing errors are decreasing. This suggests that the model is not overfitting, as it is performing well on both the training data and the unseen testing data.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

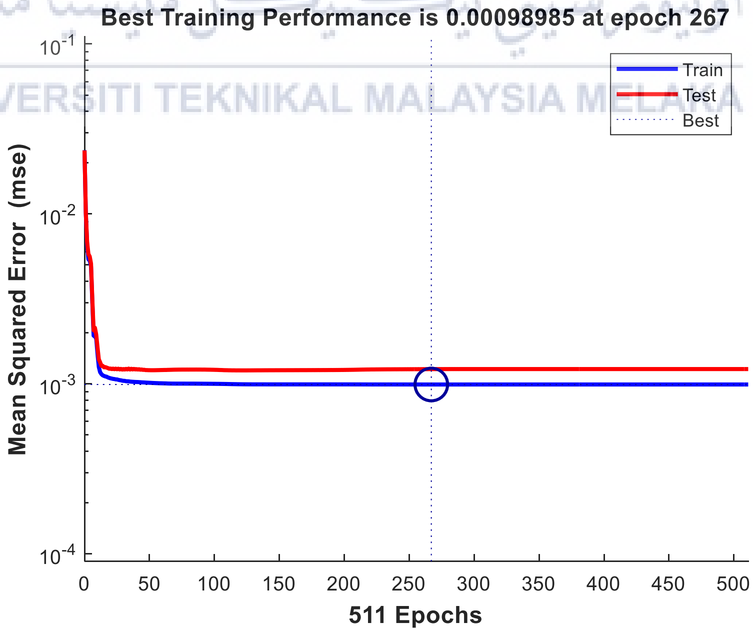
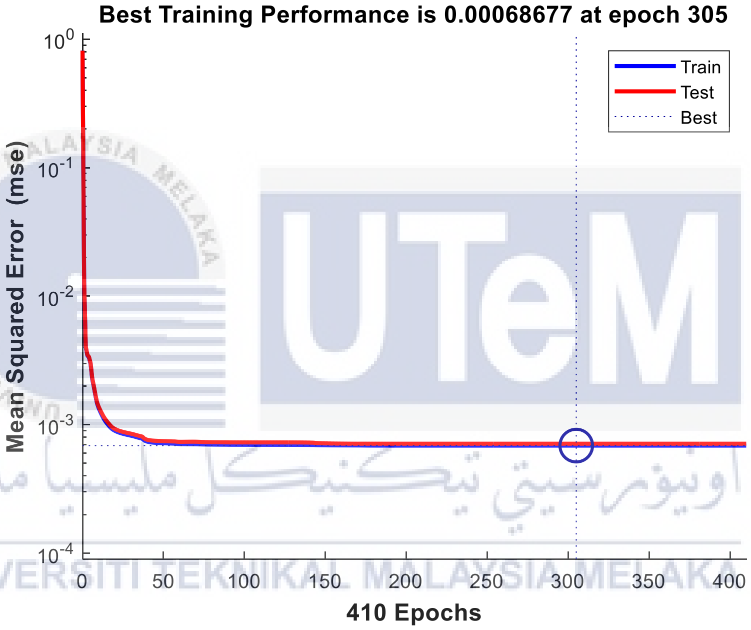


Figure 4.10: Graph for Best Training Performance (Day 1).

The graph in Figure 4.11 shows the MSE for both the training and testing datasets over epochs from the data for day 2. The graph indicates that the best training performance, an MSE of approximately 0.00068677, was achieved at epoch 305. It appears that the model's performance has improved on Day 2, as indicated by the lower MSE at epoch 305 compared to the MSE at epoch 267 on Day 1. The model seems to be learning effectively and is not overfitting on both days, as evidenced by the decreasing errors for both training and testing datasets.



UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.11: Graph for Best Training Performance (Day 2).

The graph in Figure 4.12 shows the MSE for both the training and testing datasets over epochs from the data for day 3. The graph indicates that the best training performance, an MSE of approximately 0.0018774, was achieved at epoch 488. It appears that the model's performance has slightly decreased on Day 3, as indicated by the higher MSE at epoch 488 compared to the MSE at epoch 305 on Day 2.

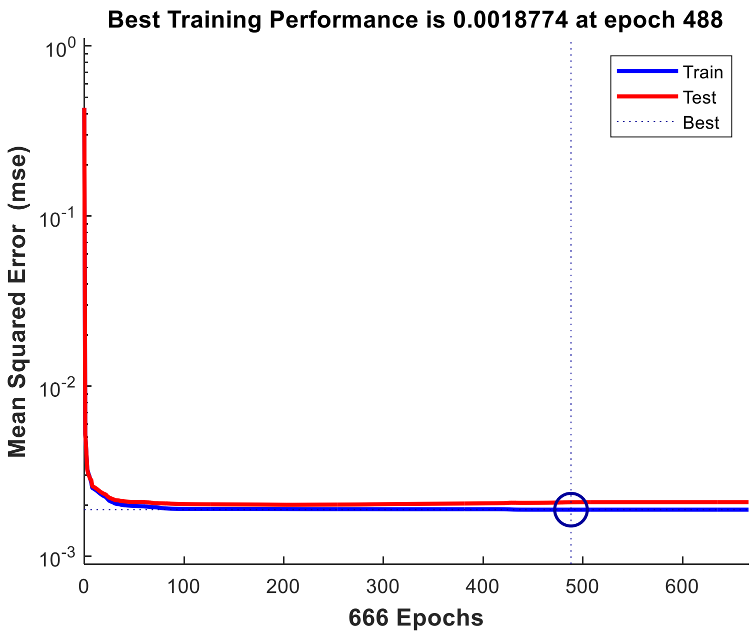


Figure 4.12: Graph for Best Training Performance (Day 3).

The graph in 3 shows the MSE for both the training and testing datasets over epochs from the data for day 4. The graph indicates that the best training performance, an MSE of approximately 0.001307, was achieved at epoch 340.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

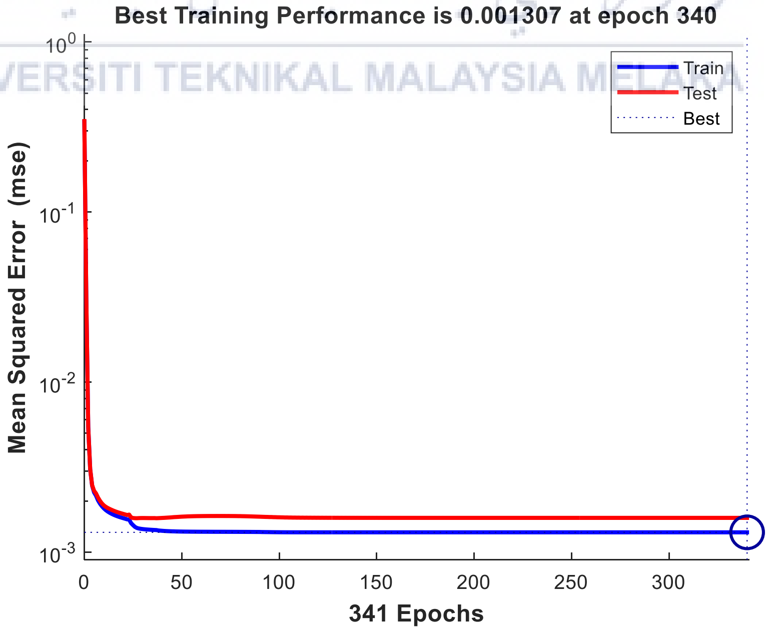


Figure 4.13: Graph for Best Training Performance (Day 4).

Over the four days, the model's performance varied, with the best performance achieved on Day 2 as shown in Table 4.3. The model seemed to be learning effectively on all days, as evidenced by the decreasing MSE. However, the speed of convergence varied, with the model converging fastest on Day 1 and slowest on Day 3.

Table 4.3: Best Training Performance for 4 days of data using BR.

Day	Epoch	Best Training Performance (MSE)
1	267	0.00098985
2	305	0.00068677
3	488	0.0018774
4	340	0.001307

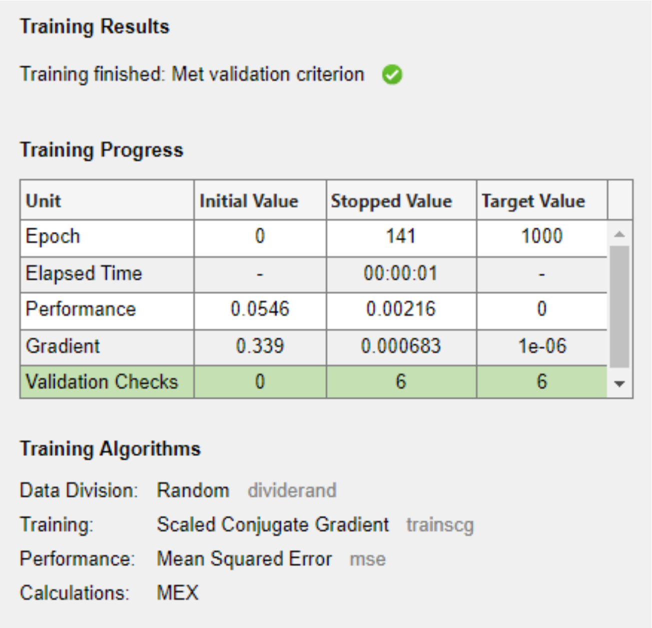
4.3 Scaled Conjugate Gradient (SCG)

4.3.1 Training results

For this section, the training results were shown to evaluate the performance of the model. By looking at metrics such as accuracy, loss, or mean squared error, the model can understand how well our model is learning and predicting. The training algorithm used was Scaled Conjugate Gradient (SCG). Figure 4.14 shows the training results from the data for day 1.

The training stopped at epoch 141, which is significantly less than the target value of 1000. This indicates that the model achieved convergence early. Meanwhile, the performance improved from an initial value of 0.0546 to a stopped value of 0.00216. This shows that the model's accuracy increased during the training. For the gradient optimization, the gradient decreased to a very small number (0.000683), suggesting that the optimization was successful and reached a minimum error.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA



Training Results

Training finished: Met validation criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	141	1000
Elapsed Time	-	00:00:01	-
Performance	0.0546	0.00216	0
Gradient	0.339	0.000683	1e-06
Validation Checks	0	6	6

Training Algorithms

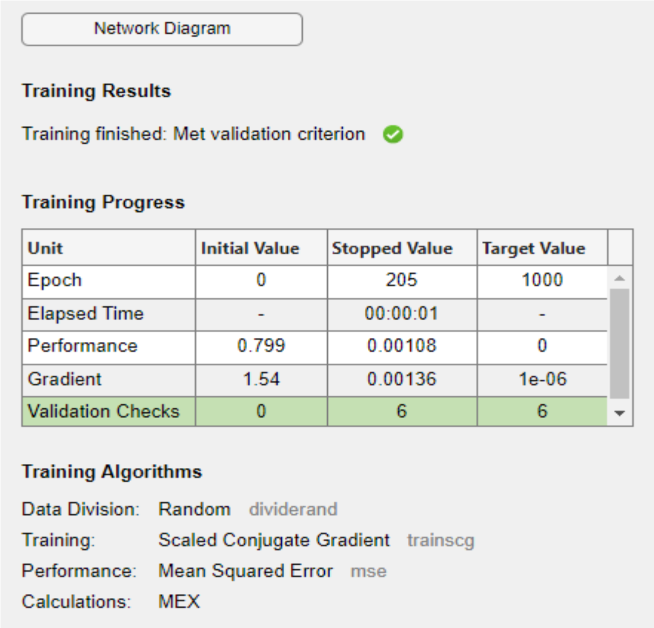
Data Division: Random dividerand
Training: Scaled Conjugate Gradient trainscg
Performance: Mean Squared Error mse
Calculations: MEX

Figure 4.14: Training results in Day 1 (SCG).

Figure 4.15 shows the training results from the data for day 2. The training on Day 2 stopped at a later epoch (205) compared to Day 1 (141). This might indicate that the model required more iterations to converge on Day 2. Meanwhile, the initial performance on Day 2 (0.799) was significantly higher than on Day 1 (0.0546), indicating that the initial error was much larger on Day 2. However, the stopped value of performance on Day 2 (0.00108) is comparable to Day 1 (0.00216), suggesting that despite starting from a higher error, the model was able to reduce the error to a similar level.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The stopped value of the gradient on Day 2 (0.00136) is slightly higher than on Day 1 (0.000683), indicating that the optimization process on Day 2 might not have been as effective in reducing the error rate as on Day 1.



Training Results

Training finished: Met validation criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	205	1000
Elapsed Time	-	00:00:01	-
Performance	0.799	0.00108	0
Gradient	1.54	0.00136	1e-06
Validation Checks	0	6	6

Training Algorithms

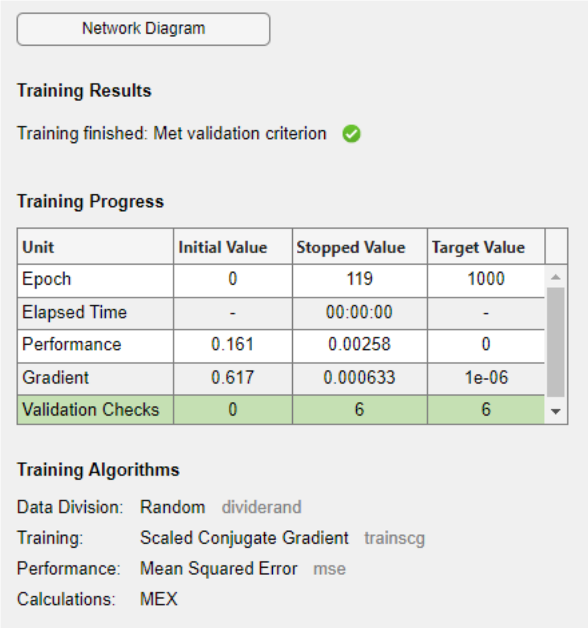
Data Division: Random dividerand
Training: Scaled Conjugate Gradient trainscg
Performance: Mean Squared Error mse
Calculations: MEX

Figure 4.15: Training results in Day 2 (SCG).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.16 shows the training results from the data for day 3. The training on Day 3 stopped at an earlier epoch (119) compared to Day 2 (205). This might indicate that the model required fewer iterations to converge on Day 3. Meanwhile, the initial performance on Day 3 (0.161) was significantly lower than on Day 2 (0.799), indicating that the initial error was much smaller on Day 3. However, the stopped value of performance on Day 3 (0.00258) is slightly higher than Day 2 (0.00108), suggesting that despite starting from a lower error, the model was not able to reduce the error to the same level as on Day 2.

The stopped value of the gradient on Day 3 (0.00633) is higher than on Day 2 (0.00136), indicating that the optimization process on Day 3 might not have been as effective in reducing the error rate as on Day 2.



Training Results

Training finished: Met validation criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	119	1000
Elapsed Time	-	00:00:00	-
Performance	0.161	0.00258	0
Gradient	0.617	0.000633	1e-06
Validation Checks	0	6	6

Training Algorithms

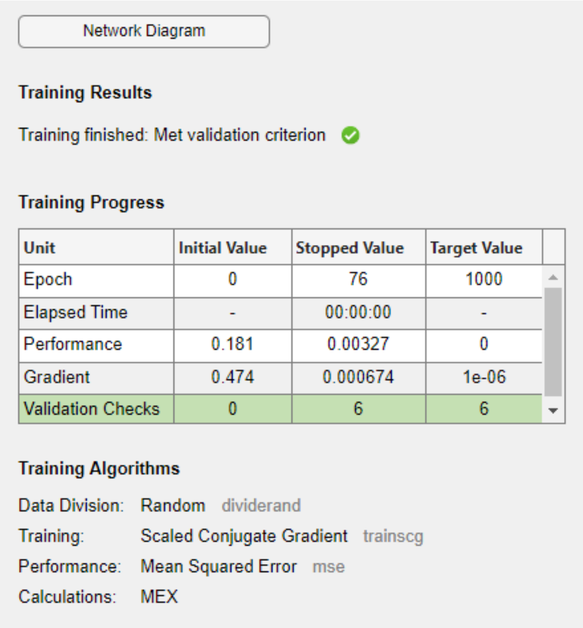
Data Division: Random dividerand
Training: Scaled Conjugate Gradient trainscg
Performance: Mean Squared Error mse
Calculations: MEX

Figure 4.16: Training results in Day 3 (SCG).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.17 shows the training results from the data for day 4. The training on Day 4 stopped at an earlier epoch (76) compared to Day 3 (119). This might indicate that the model required fewer iterations to converge on Day 4. Meanwhile, the initial performance on Day 4 (0.181) was slightly higher than on Day 3 (0.161), indicating that the initial error was a bit larger on Day 4. However, the stopped value of performance on Day 4 (0.00327) is slightly higher than Day 3 (0.00258), suggesting that despite starting from a slightly higher error, the model was not able to reduce the error to the same level as on Day 3.

The stopped value of the gradient on Day 4 (0.00674) is slightly higher than on Day 3 (0.00633), indicating that the optimization process on Day 4 might not have been as effective in reducing the error rate as on Day 3.



Training Results

Training finished: Met validation criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	76	1000
Elapsed Time	-	00:00:00	-
Performance	0.181	0.00327	0
Gradient	0.474	0.000674	1e-06
Validation Checks	0	6	6

Training Algorithms

Data Division: Random dividerand
Training: Scaled Conjugate Gradient trainscg
Performance: Mean Squared Error mse
Calculations: MEX

Figure 4.17: Training results in Day 4 (SCG).

Based on the training results from day 1 to day 4, The model converged at different epochs each day, with the earliest convergence on day 4 (epoch 76) and the latest on day 2 (epoch 205) as shown in Table 4.4. This suggests that the model's learning rate or initial weights might have varied across the days.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Moreover, the initial performance varied each day, with the highest initial error on day 2 (0.799) and the lowest on day 1 (0.0546). However, by the end of training each day, the model was able to reduce the error significantly. The lowest final error was achieved on day 2 (0.00108), suggesting the best performance on that day.

Furthermore, the gradient at the end of training also varied each day, with the smallest value on day 1 (0.000683) and the largest on day 4 (0.00674). This indicates that the optimization process was most effective on day 1.

Table 4.4: Performance for 4 days of data using SCG.

Day	Mean Squared Error (MSE)		Number of Epochs
1	0.0546	0.00216	141
2	0.799	0.00108	205
3	0.161	0.00258	119
4	0.181	0.00327	76

4.3.2 Correlation coefficient (R)

For this section, scatter plot was generated to represent the results of a Scaled Conjugate Gradient (SCG) training neural network from data for Day 1,2,3 and 4.

Based on Figure 4.18, the correlation coefficient (R) is 0.78294, which indicates a strong positive correlation between the Target and Output. This means that as the Target value increases, the Output value also tends to increase.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The line labeled “Output = 0.61*Target+3.3” is the line of best fit for the data. It represents the relationship between Target and Output. The slope of the line (0.61) suggests that for each unit increase in the Target, the Output increases by approximately 0.61 units. Apart from that, the spread of data points around the line of best fit gives an indication of the variability or noise in the data.

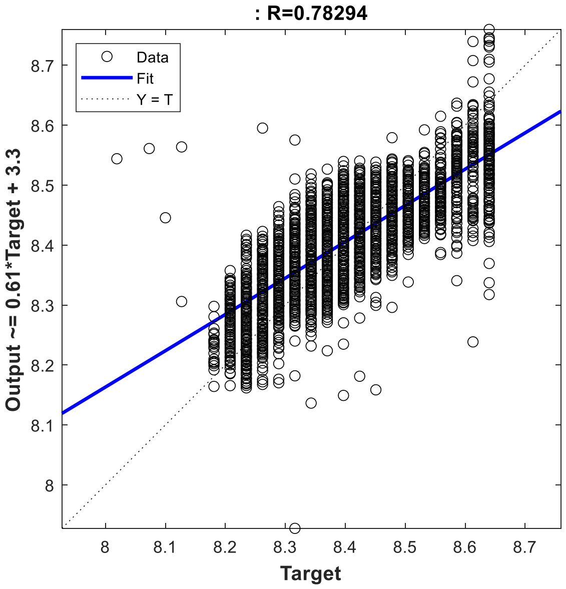


Figure 4.18: Scatter plot for Day 1 (SCG).

Based on Figure 4.19, the R value is higher on day 2 (0.8832) compared to day 1 (0.78294), indicating a stronger positive correlation between the Target and Output on day 2. Moreover, the slope of the line of best fit is higher on day 2 (0.7) compared to day 1 (0.61), suggesting that for each unit increase in the Target, the Output increases by a larger amount on day 2.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

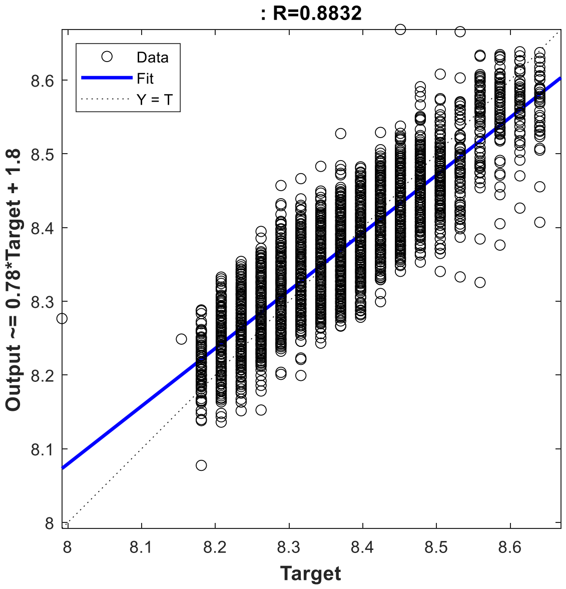


Figure 4.19: Scatter plot for Day 2 (SCG).

Based on Figure 4.20, the R value is lower on day 3 (0.82129) compared to day 2 (0.8832), indicating a slightly weaker positive correlation between the Target and Output on day 3. Furthermore, the slope of the line of best fit is lower on day 3 (0.67) compared to day 2 (0.7), suggesting that for each unit increase in the Target, the Output increases by a slightly smaller amount on day 3.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

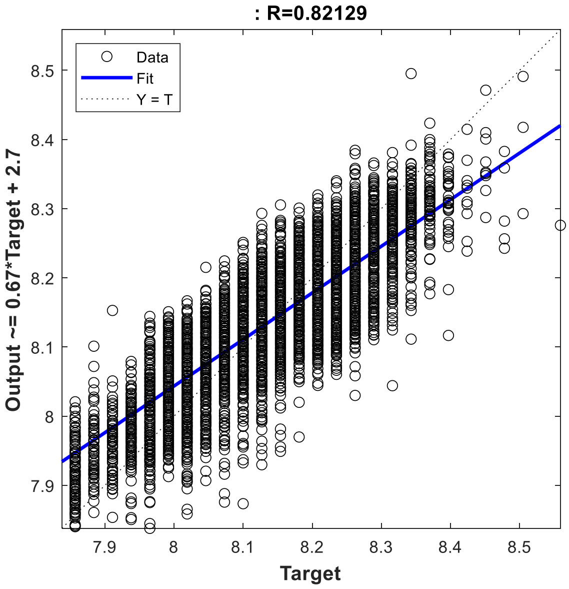


Figure 4.20: Scatter plot for Day 3 (SCG).

Based on Figure 4.21, the R value is higher on day 4 (0.87764) compared to day 3 (0.82129), indicating a stronger positive correlation between the Target and Output on day 4. Moreover, the slope of the line of best fit is slightly higher on day 4 (0.7) compared to day 3 (0.67), suggesting that for each unit increase in the Target, the Output increases by a slightly larger amount on day 4.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

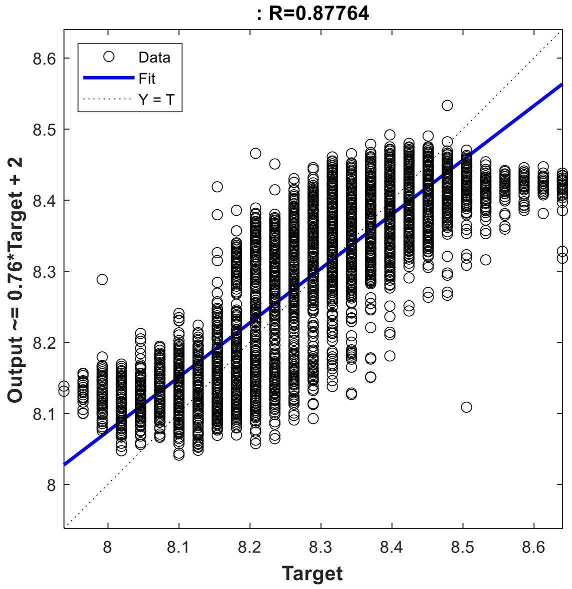


Figure 4.21: Scatter plot for Day 4 (SCG).

Over the four days, these results suggest that the model's predictions were most accurate on day 2, as indicated by the highest R value which is (0.8832) as shown in Table 4.5. The highest slope was on day 2 and day 4 (0.7), indicating a larger increase in Output for each unit increase in the Target on those days.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Table 4.5: Correlation coefficient comparison for SCG training

Day	Correlation Coefficient (R)
1	0.78294
2	0.8832
3	0.82129
4	0.87764

4.3.3 Mean Squared Error (MSE)

For this section, a graph representing the Mean Squared Error (MSE) was generated to represent the results of a Scaled Conjugate Gradient (SCG) training neural network from data for day 1,2,3 and 4.

The graph in Figure 4.22 represents the Mean Squared Error (MSE) during the training of a neural network using the Scaled Conjugate Gradient (SCG) method from the data for Day 1. The errors for all datasets rapidly decrease during the initial epochs, indicating that the model is learning quickly at the start of the training process.

The best validation performance is achieved at epoch 135 with an MSE of 0.0021512. This is the point where the model has the lowest error on the validation set, indicating the most effective learning up to that point.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

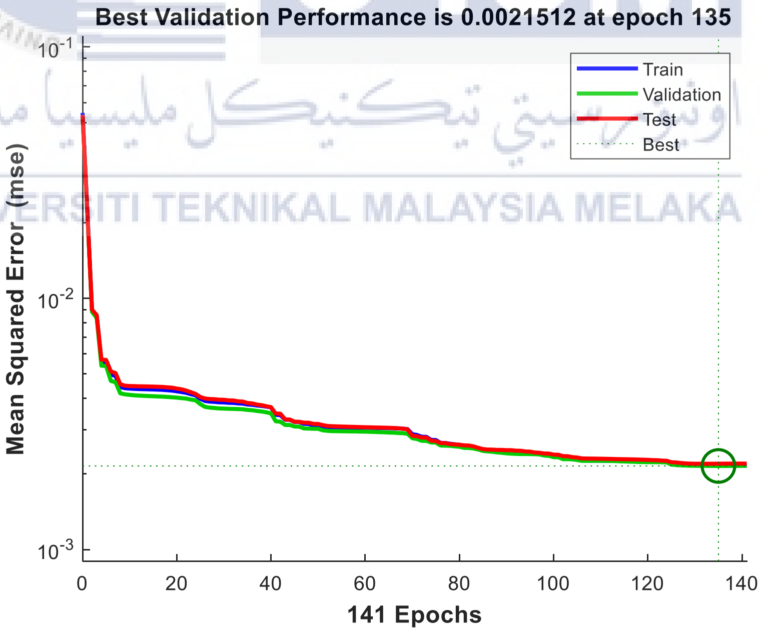


Figure 4.22: Graph for Best Training Performance (Day 1).

The graph in Figure 4.23 represents the Mean Squared Error (MSE) from the data for Day 2. The best validation performance improved from 0.0021512 at epoch 135 on day 1 to 0.0011513 at epoch 199 on day 2. This indicates that the model performed better on the validation set on day 2. However, the number of epochs increased from 141 on day 1 to 205 on day 2. This could suggest that the model needed more time to converge on the second day.

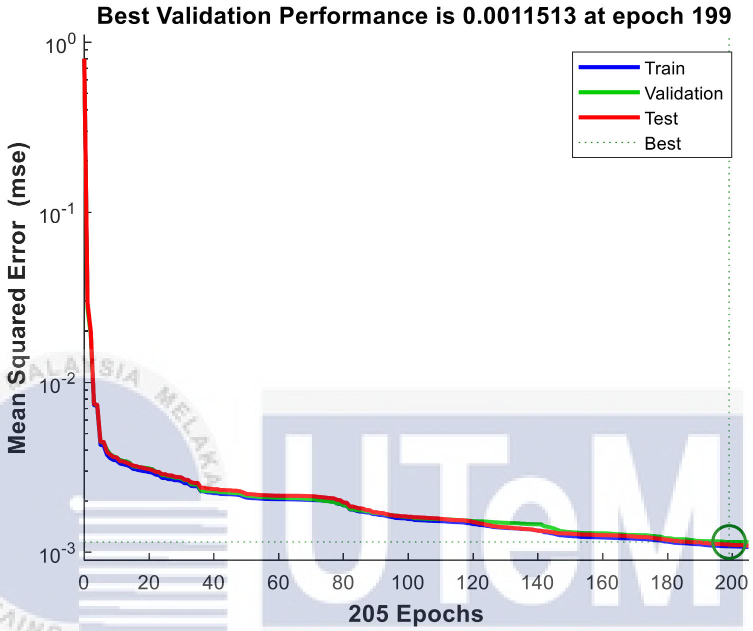


Figure 4.23: Graph for Best Training Performance (Day 2).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The graph in Figure 4.24 represents the Mean Squared Error (MSE) from the data for Day 3. The best validation performance changed from 0.0011513 at epoch 199 on day 2 to 0.002633 at epoch 113 on day 3. This indicates that the model performed slightly worse on the validation set on day 3 compared to day 2. However, the number of epochs at which the best validation performance was achieved decreased from 199 on day 2 to 113 on day 3. This could suggest that the model converged faster on the third day.

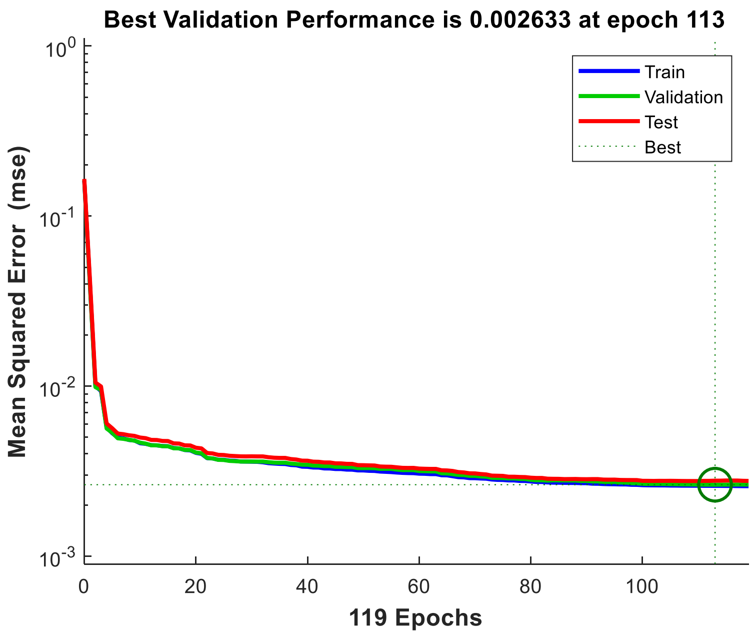


Figure 4.24: Graph for Best Training Performance (Day 3).

The graph in Figure 4.25 represents the Mean Squared Error (MSE) from the data for Day 4. The best validation performance changed from 0.002633 at epoch 113 on day 3 to 0.0032003 at epoch 70 on day 4. This indicates that the model performed slightly worse on the validation set on day 4 compared to day 3. However, the number of epochs at which the best validation performance was achieved decreased from 113 on day 3 to 70 on day 4. This could suggest that the model converged faster on the fourth day.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

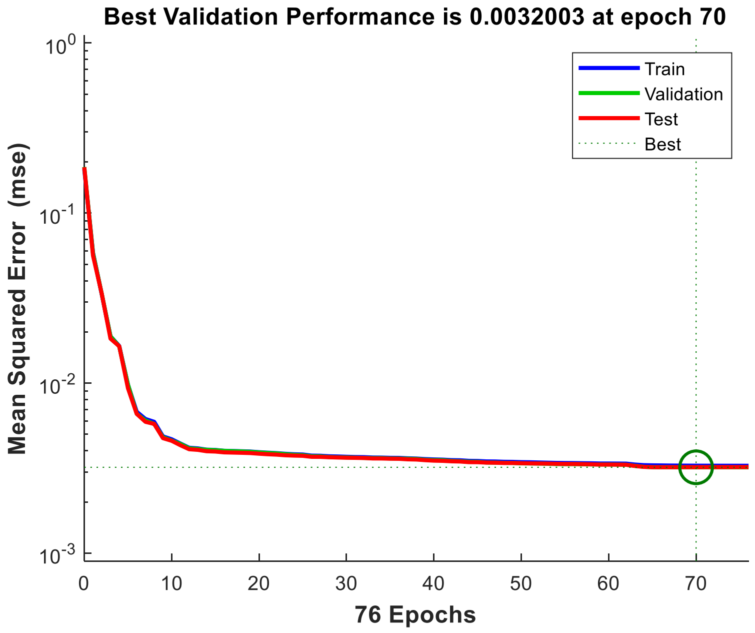


Figure 4.25: Graph for Best Training Performance (Day 4).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

The MSE for all four days shows a rapid decrease in the initial epochs, indicating that the model quickly learned to minimize errors as shown in Table 4.6. After approximately 20 epochs, the MSE stabilizes, suggesting that the model has essentially learned the pattern in the data and is now refining its parameters for optimal performance. However, the number of epochs at which the best validation performance was achieved varied each day. This could suggest that the model's speed of convergence was inconsistent across the four days.

The model's performance, as indicated by the best validation performance, improved from day 1 to day 2 but then worsened on days 3 and 4. This could be due to various factors such as changes in the data or model parameters.

Table 4.6: Best Training Performance for 4 days of data using SCG.

Day	Epoch	Best Training Performance (MSE)
1	135	0.0021512
2	199	0.0011513
3	113	0.0026330
4	70	0.0032003

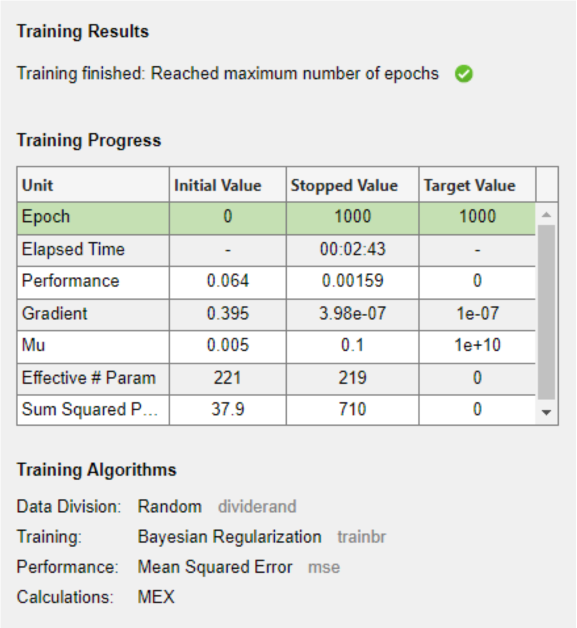
4.4 Comparison for Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG)

4.4.1 Training Results

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.26 shows the training results for combined data (4 days). The training algorithm used was Bayesian Regularization. The training reached the maximum number of epochs (1000). This could indicate that the training was stopped because it reached this limit. Moreover, the performance improved from an initial value of 0.064 to a stopped value of 0.00159. This shows a significant enhancement in model performance.

Furthermore, the gradient decreased significantly from an initial value of 0.395 to a very small number close to zero. This suggests that the model may have reached a minimum in the loss function, indicating convergence. When the gradient is close to zero, it means that the model's parameters are not changing much with each update, suggesting that the model has found a minimum in the loss function [30]. This is usually a good sign as it indicates that the model has likely learned to predict the target variable accurately.



Training Results

Training finished: Reached maximum number of epochs ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	1000	1000
Elapsed Time	-	00:02:43	-
Performance	0.064	0.00159	0
Gradient	0.395	3.98e-07	1e-07
Mu	0.005	0.1	1e+10
Effective # Param	221	219	0
Sum Squared P...	37.9	710	0

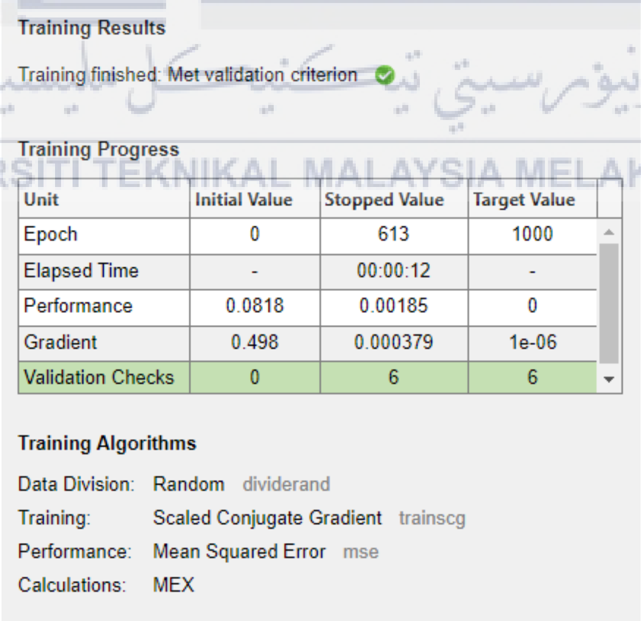
Training Algorithms

Data Division: Random dividerand
Training: Bayesian Regularization trainbr
Performance: Mean Squared Error mse
Calculations: MEX

Figure 4.26: Training results for combined data (BR).

Figure 4.27 shows the training results for combined data (4 days). The training algorithm used was Scaled Conjugate Gradient (SCG).

UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA



Training Results

Training finished: Met validation criterion ✓

Training Progress

Unit	Initial Value	Stopped Value	Target Value
Epoch	0	613	1000
Elapsed Time	-	00:00:12	-
Performance	0.0818	0.00185	0
Gradient	0.498	0.000379	1e-06
Validation Checks	0	6	6

Training Algorithms

Data Division: Random dividerand
Training: Scaled Conjugate Gradient trainscg
Performance: Mean Squared Error mse
Calculations: MEX

Figure 4.27: Training results for combined data (SCG).

The training met the validation criterion at 613 epochs. This is fewer than the 1000 epochs it took with Bayesian Regularization (BR), suggesting that SCG may have converged faster. However, the performance metric stopped at 0.00185. This is slightly higher than the 0.00159 achieved with BR, indicating that BR might have resulted in a slightly better fit to the training data.

Furthermore, the gradient is very low at 0.000379, indicating convergence. This is similar to the result with BR, suggesting that both methods were able to find a minimum in the loss function. Table 4.7 shows the comparison for the overall performance for combined data between BR and SCG.

Table 4.7: Overall performance for combined data.

Training Algorithm	Mean Squared Error (MSE)		Number of Epochs
BR	0.0640	0.00159	1000
SCG	0.0818	0.00185	613

4.4.2 Correlation coefficient (R)

For this section, scatter plot was generated to represent the results of a Bayesian Regularization (BR) training neural network for the combined data (4 days). Based on Figure 4.28, the correlation coefficient (R) is 0.9546, indicating a strong positive linear relationship between the Target and Output variables. This suggests that as the Target value increases, the Output value also increases. Moreover, the data points are closely aligned with the fit line, suggesting that the model has effectively captured the relationship between these two variables.

اونيۏرسيتي تيکنيکل مليسيا ملاک
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

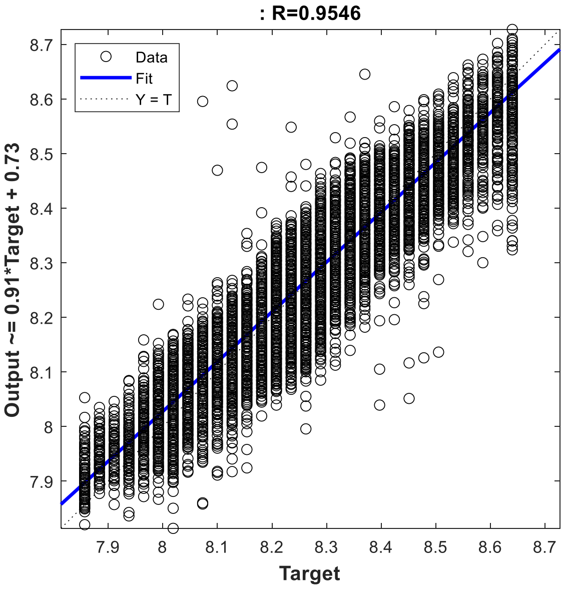


Figure 4.28: Scatter plot for combined data (BR).

Figure 4.29 shows the scatter plot that was generated to represent the results of a Scaled Conjugate Gradient (SCG) training neural network for the combined data (4 days).

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

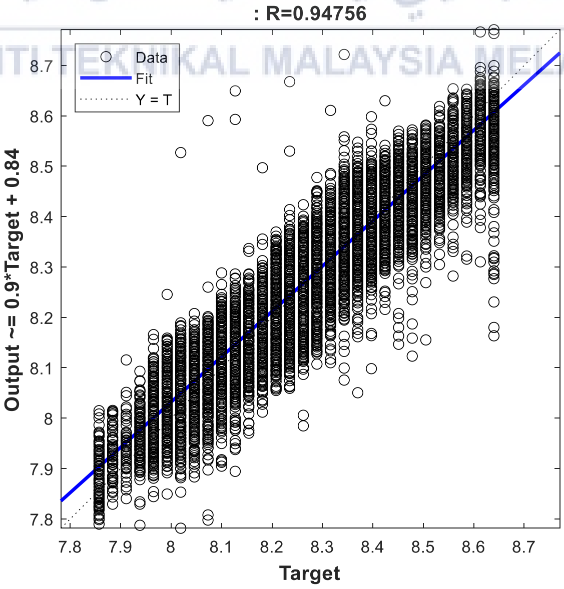


Figure 4.29: Scatter plot for combined data (SCG).

The correlation coefficient (R) is 0.94756, indicating a strong positive linear relationship between the Target and Output variables. This is slightly lower than the 0.9546 achieved with Bayesian Regularization (BR), suggesting that BR might have resulted in a slightly stronger linear relationship. Meanwhile, the data points are closely aligned with the fit line, similar to the results with BR. This suggests that both methods were able to effectively capture the relationship between these two variables. Table 4.8 shows the comparison between BR and SCG in terms of correlation coefficient.

Table 4.8: Correlation coefficient comparison between BR and SCG.

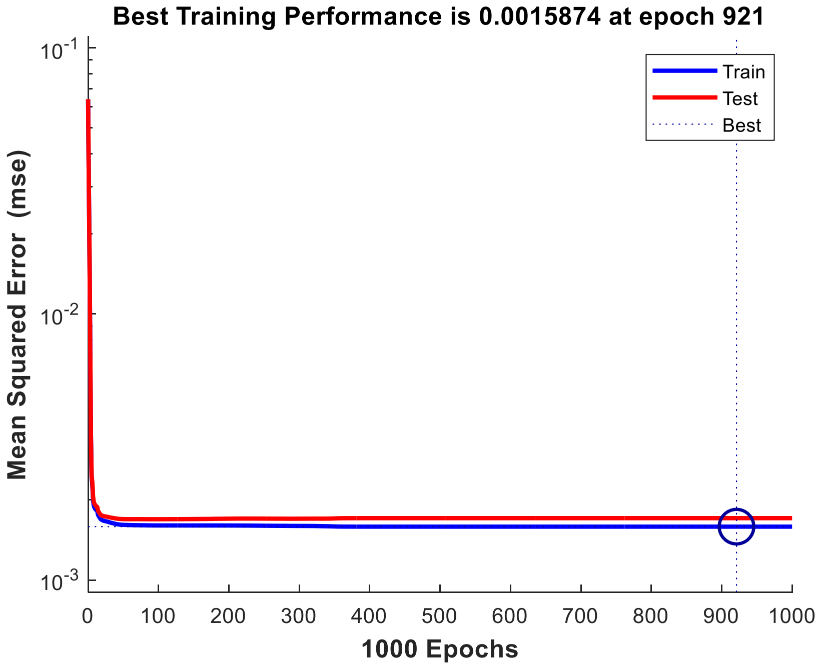
Training Algorithm	Correlation Coefficient (R)
BR	0.95460
SCG	0.94756

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

4.4.3 Mean Squared Error (MSE)

For this section, a graph representing the Mean Squared Error (MSE) was generated to represent the results of Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG) training neural network for the combined data (4 days).

The graph in Figure 4.30 represents the Mean Squared Error (MSE) during the training of a neural network using the Bayesian Regularization (BR) method. The best training performance is achieved at epoch 921 with an MSE of 0.0015874. This suggests that the model was able to minimize the error significantly during the training process. The plot also shows both training and test errors. If these two lines are close to each other and both decreasing, it is a good sign that the model is not overfitting.



UTeM

Figure 4.30: Graph for Best Training Performance (Combined data).

The graph in Figure 4.31 represents the Mean Squared Error (MSE) during the training of a neural network using the Scaled Conjugate Gradient (SCG) method.

اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

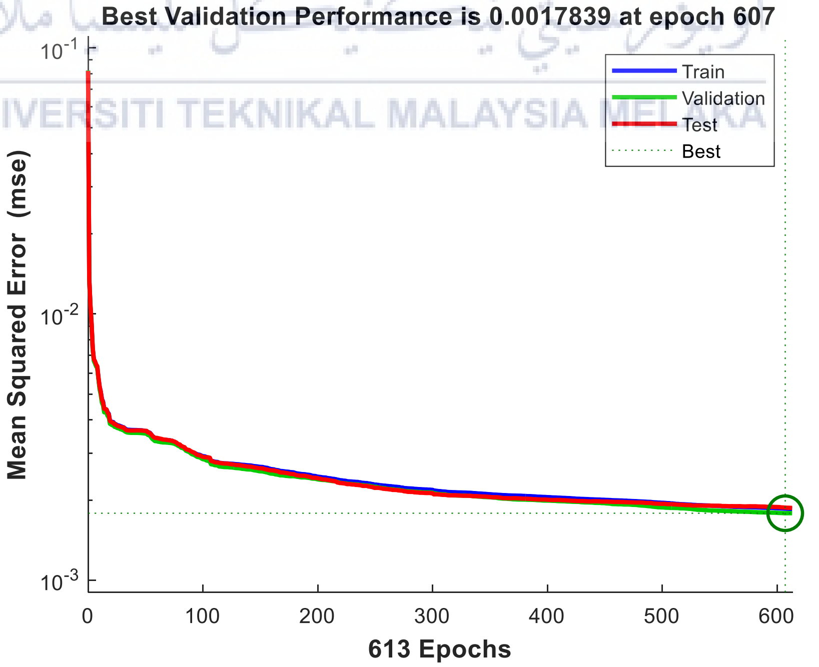


Figure 4.31: Graph for Best Training Performance (Combined data).

The best validation performance is achieved at epoch 607 with an MSE of 0.0017839. This is slightly higher than the 0.0015874 achieved with Bayesian Regularization (BR), suggesting that BR might have resulted in a slightly better fit to the training data. However, the model took 607 epochs to achieve the best performance. This is fewer than the 921 epochs it took with BR, suggesting that SCG may have converged faster. Table 4.9 shows the comparison between BR and SCG in terms of performance and convergence time.

Table 4.9: Best Training Performance for combined data.

Day	Epoch	Best Training Performance (MSE)
BR	921	0.0015874
SCG	607	0.0017839

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

4.4.4 Prediction results

For this section, a graph representing the actual value was compared with the predicted value. This graph is generated to represent the results of Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG) training neural network for the combined data (4 days).

The red line represents the predicted offset, and the blue line represents the actual offset as shown in Figure 4.32 and 4.33. If the two lines are closely aligned, it indicates that the predictions are accurate. In this case, the red and blue lines do not align closely, suggesting that the predictions may not be very accurate [30]. If the two lines are closely aligned, it indicates that the predictions are accurate. In this case, the red and blue lines do not align closely, suggesting that the predictions may not be very accurate.

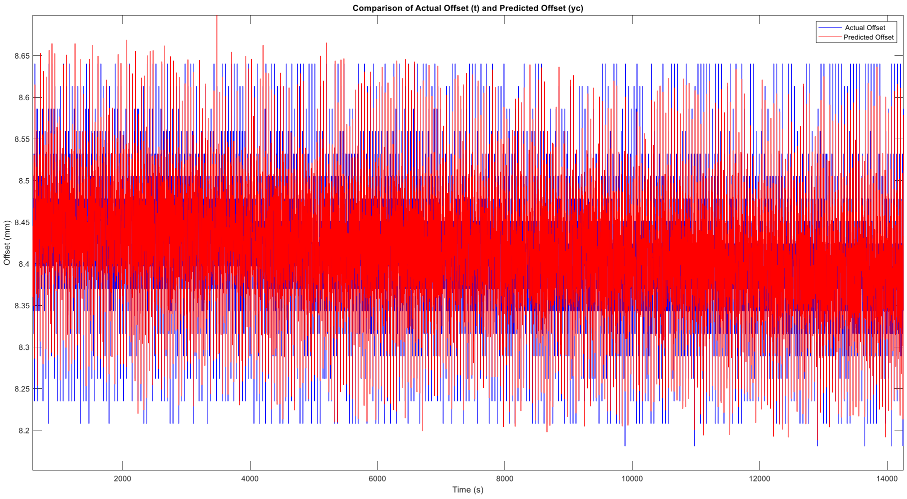


Figure 4.32: Comparison between actual and predicted value using BR training.

However, without quantitative measures such as Mean Absolute Error (MAE), it is difficult to definitively say which method is performing better [31].

UTeM
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

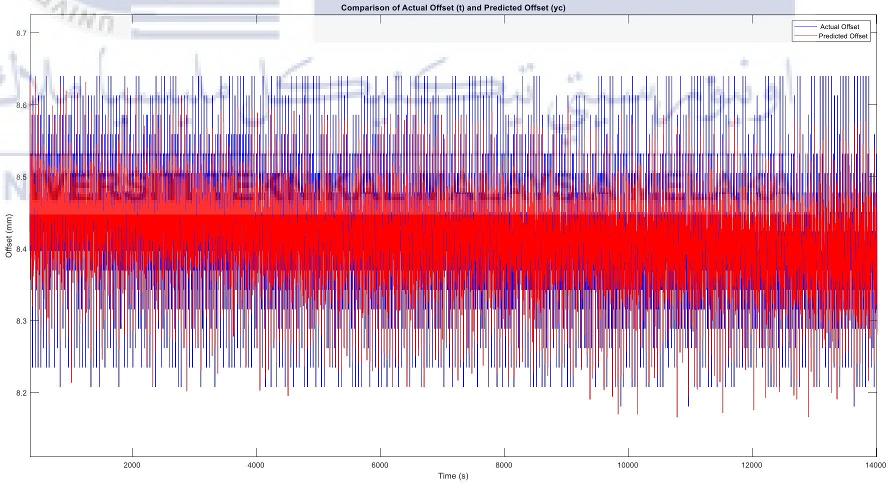
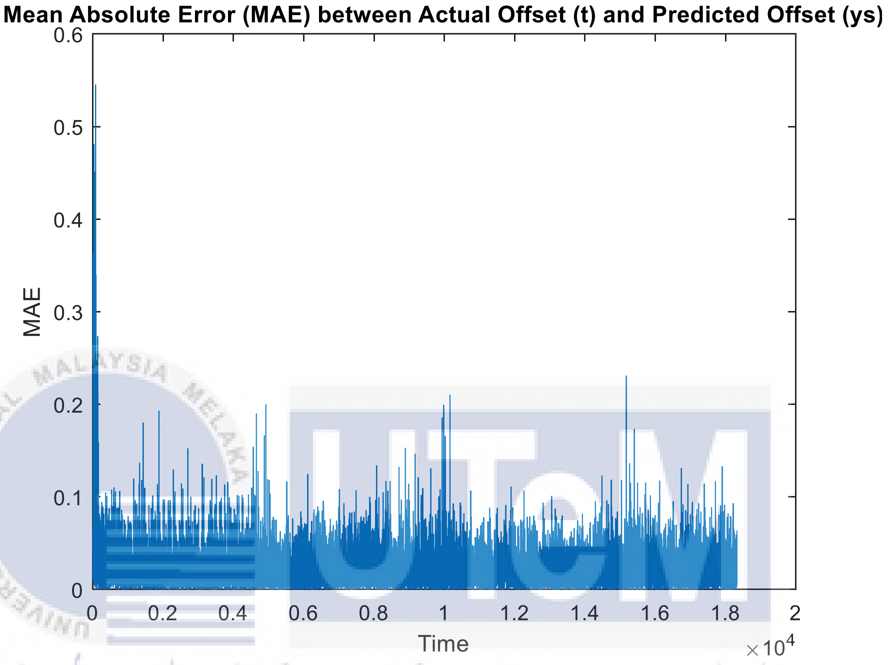


Figure 4.33: Comparison between actual and predicted value using SCG training.

Figure 4.34 and 4.35 shows the Mean Absolute Error (MAE) graph to compare the performance of Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG) training neural network in prediction.

Most of the MAE values are concentrated below 0.2, indicating that the predictions are generally close to the actual values as shown in Figure 4.33. However, there are noticeable peaks reaching up to approximately 0.5, suggesting moments of significant error between predicted and actual offsets.



اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Figure 4.34: Plot for MAE between actual and predicted value (BR).

The numerous spikes in MAE throughout the graph could suggest that the model struggles with certain patterns in the data. These could be due to outliers, noise, or complex patterns in the data that the model fails to capture [32]. The graph is also similar to the graph shown in Figure 4.35 which is quite hard to compare just by looking at the graph.

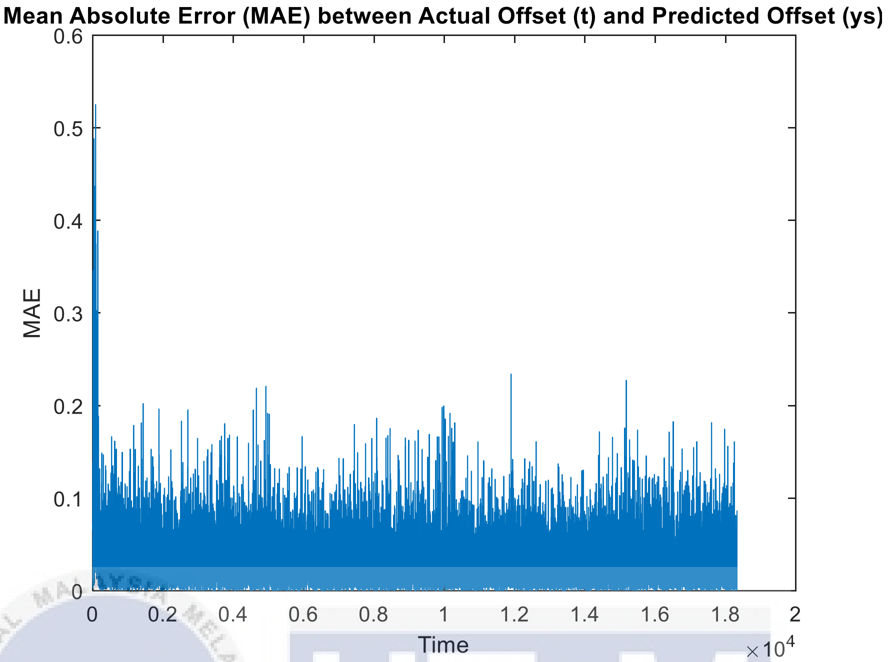


Figure 4.35: Plot for MAE between actual and predicted value (SCG).

The plot in Figure 4.36 shows the comparison plot between Mean Absolute Error for BR (blue) and SCG (orange). The MAE for the SCG method is consistently higher than the MAE for BR method. This suggests that the BR method has lower average prediction errors and thus performs better than the SCG method.

Lower MAE values indicate better model performance as they represent smaller average errors between the predicted and actual values. Therefore, the BR method, which has lower MAE values, is likely to provide more accurate predictions.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

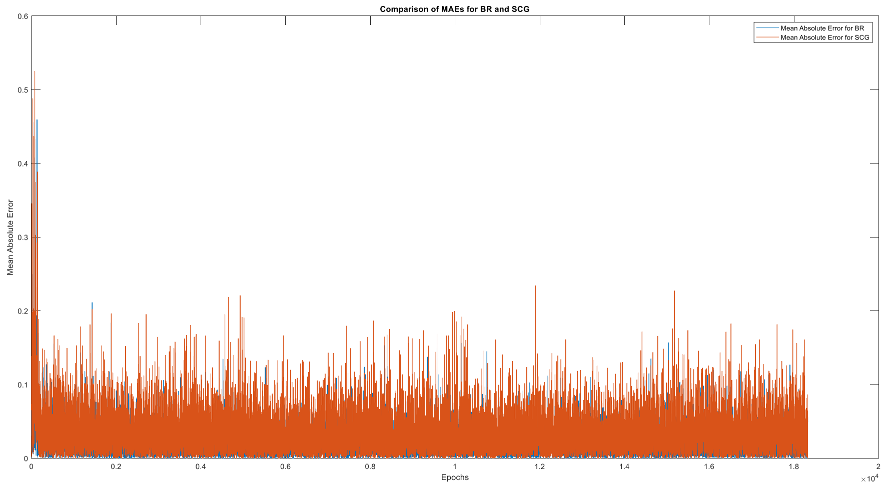


Figure 4.36: Comparison plot between MAE for BR and SCG.

4.5 Summary

UNIVERSITI TEKNIKAL MALAYSIA MELAKA
UTeM
اونيورسيتي تيكنيكل مليسيا ملاك

The analysis section of this project provided a thorough examination of the performance of Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG) training algorithms in the context of cover tape offset detection.

The exploration of training results indicated that BR achieved a lower Mean Squared Error (MSE) and demonstrated a faster convergence compared to SCG. Scatter plots illustrated strong positive linear relationships between predicted and actual values for both algorithms, with a slightly better correlation observed in BR. However, the analysis cautioned against solely relying on visual inspection and emphasized the need for quantitative measures like Mean Absolute Error (MAE).

The MAE analysis uncovered nuances in prediction accuracy, revealing both the overall closeness of predictions to actual values and specific instances of significant errors. Comparison plots highlighted consistently lower MAE values for BR, suggesting superior performance in terms of average prediction errors.

CHAPTER 5

CONCLUSION AND FUTURE WORKS

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

5.1 Conclusion

The completion of this comprehensive project focused on the detection of cover tape offset or misalignment during the tape and reel process has yielded valuable insights and outcomes. Leveraging the capabilities of the Raspberry Pi Camera Module 3 and employing computer vision techniques through Python's OpenCV library, real-time monitoring and analysis of the process were achieved. The integration with Node-RED facilitated a user-friendly dashboard for real-time data visualization, enhancing the project's accessibility. The utilization of MATLAB, coupled with Neural Net Time Series Apps, allowed for the implementation of predictive maintenance using Bayesian Regularization (BR) and Scaled Conjugate Gradient (SCG) training algorithms.

The analysis of training results indicated that both BR and SCG exhibited successful convergence, with BR showcasing slightly better performance in terms of model fit to the training data. The examination of Mean Squared Error (MSE) graphs provided insights into the training dynamics, highlighting the efficiency of both methods in minimizing errors. Subsequently, the comparison between actual and predicted values revealed challenges in accurately predicting offsets, emphasizing the importance of further quantitative measures.

The Mean Absolute Error (MAE) analysis offered a detailed understanding of prediction accuracy, showcasing occasional spikes that indicated the model's struggle with specific patterns, potentially influenced by outliers or complex data structures. Significantly, the comparison between BR and SCG consistently favored BR, affirming its superiority in providing lower average prediction errors.

Overall, this project has successfully integrated hardware, software, and advanced analytical techniques to meet its primary objectives. The design and implementation of a detection system utilizing the Raspberry Pi Camera Module (R01) effectively addressed the critical aspect of detecting cover tape offset in the tape and reel process. The creation of a Node-Red dashboard for live data monitoring (R02) not only facilitated real-time visualization but also enhanced the project's accessibility.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

Furthermore, the utilization of predictive analysis (R03) through computer vision, machine learning, and the implementation of Bayesian Regularization and Scaled Conjugate Gradient training algorithms opened avenues for enhanced efficiency and reduced downtime in industrial settings. The successful achievement of these objectives underscores the project's ability to contribute meaningfully to the manufacturing process, offering valuable insights and paving the way for advanced predictive maintenance strategies.

5.2 Sustainable Development Goals (SDG)

This thesis strongly corresponds to Sustainable Development Goals (SDGs) 9 and 12, encompassing the core principles of encouraging innovation and advocating for responsible consumption and production. SDG 9, often known as "Industry, Innovation, and Infrastructure," highlights the significance of building durable and sustainable infrastructure, promoting innovation, and improving scientific research. This project demonstrates the innovative nature of electronic engineering by combining the Raspberry Pi camera module, OpenCV library, and predictive maintenance using MATLAB. The initiative aims to improve industrial processes and drive technological improvements in manufacturing by accurately identifying cover tape misalignment during the tape and reel operation.

Furthermore, SDG 12, also known as "Responsible Consumption and Production," promotes sustainable methods that result in a successful use of resources and decreased ecological footprint. This thesis aims to optimize the manufacturing process by introducing predictive maintenance. Detecting cover tape misalignment in real-time enhances the efficiency of the tape and reel process while also minimizing downtime and resource loss. This approach is in line with the overall objective of promoting sustainable consumption and production practices in the field of electronic manufacturing. The project's focus was to demonstrate the development of a CSV file and the real-time visualization of data in Node-RED. This highlights the dedication to optimizing resource usage and promoting sustainable practices in industrial environments. This project demonstrates how developments in electronic engineering can effectively contribute to the achievement of global sustainability goals defined in SDGs 9 and 12.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

5.3 Future Work

5.3.1 Real-time Monitoring and Feedback

Looking ahead to future improvements for the cover tape offset detection system, a key focus is on adding real-time feedback features to enhance its functionality [33]. The main goal is to create a reliable alert system that quickly notifies operators when cover tape misalignments occur in the tape and reel process. By using various technologies like email notifications, text messages, or on-screen messages, we can ensure operators receive timely notifications in the way that suits them best. Additionally, upgrading the Node-Red dashboard with visual indicators or notifications will be a great improvement. The operators can easily understand and respond to misalignments. This interactive feature will let operators acknowledge or dismiss alerts right from the dashboard.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

5.3.2 Data Logging and Analytics

In the context of future work, an imperative focus revolves around the augmentation of data logging capabilities within the cover tape offset detection system. In addition to the current reliance on CSV file structures, a proposition involves the incorporation of a more resilient database system to facilitate the storage and analytical processing of historical data [34]. This proposition serves the dual purpose of enhancing the organization and accessibility of extensive datasets while providing a foundation for discerning overarching trends, intricate patterns, and optimizing maintenance schedules.

Furthermore, a prospective avenue for refinement involves reevaluating the reliance on MATLAB exclusively for the implementation of Neural Net Time Series Apps. An exploration into the feasibility of directly integrating machine learning functionalities into the Python codebase is warranted. This strategic shift aims to streamline the computational workflow and foster a unified coding environment. Through this approach, the system can be fortified to autonomously train models, leveraging historical data for predictive analysis of potential cover tape misalignments. The adoption of such methodologies aligns with contemporary research trends, enhancing the system's predictive capabilities and fortifying its technological underpinnings.

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

REFERENCES

[1] C.-F. J. Kuo, T. Fang, C.-L. Lee, and H.-C. Wu, “Automated optical inspection system for surface mount device light emitting diodes,” *J. Intell. Manuf.*, vol. 30, no. 2, pp. 641–655, Oct. 2016, doi: 10.1007/s10845-016-1270-6.

[2] S. Qiao, L. Q. Tao, T. L. Ren, and Z. L. Liu, “Tape & Reel single side peel force test verification,” *2016 17th Int. Conf. Electron. Packag. Technol. ICEPT 2016*, pp. 1483–1486, 2016, doi: 10.1109/ICEPT.2016.7583404.

[3] Tanviruzzama and S. Mehfuz, “Review: Lane Detection for Autonomous Vehicles Using Image Processing Techniques,” *2023 Int. Conf. Power, Instrumentation, Energy Control. PIECON 2023*, pp. 1–6, 2023, doi: 10.1109/PIECON56912.2023.10085756.

[4] E. A. Sekehravani, E. Babulak, and M. Masoodi, “Implementing canny edge detection algorithm for noisy image,” *Bull. Electr. Eng. Informatics*, vol. 9, no. 4, pp. 1404–1410, 2020, doi: 10.11591/eei.v9i4.1837.

[5] R. R, N. Saklani, and V. Verma, “A Review on Edge detection Technique ‘Canny Edge Detection,’” *Int. J. Comput. Appl.*, vol. 178, no. 10, pp. 28–30,

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2019, doi: 10.5120/ijca2019918828.

[6] R. Biswas and J. Sil, “An Improved Canny Edge Detection Algorithm Based on Type-2 Fuzzy Sets,” *Procedia Technol.*, vol. 4, pp. 820–824, 2012, doi: 10.1016/j.protcy.2012.05.134.

[7] R. Liu and J. Mao, “Research on Improved Canny Edge Detection Algorithm,” *MATEC Web Conf.*, vol. 232, pp. 3–6, 2018, doi: 10.1051/matecconf/201823203053.

[8] A. S. Hassanein, S. Mohammad, M. Sameer, and M. E. Ragab, “A Survey on Hough Transform, Theory, Techniques and Applications,” 2015, [Online]. Available: http://arxiv.org/abs/1502.02160.

[9] D. Duan, M. Xie, Q. Mo, Z. Han, and Y. Wan, “An improved Hough transform for line detection,” *ICCASM 2010 - 2010 Int. Conf. Comput. Appl. Syst. Model. Proc.*, vol. 2, no. Iccasm, pp. V2-354-V2-357, 2010, doi: 10.1109/ICCASM.2010.5620827.

[10] X. Wang, H. Tan, F. Zhou, and Y. Zhao, “Real-time Classified Hough Transform Line Detection Based on FPGA,” *Proc. 2015 5th Int. Conf. Comput. Sci. Autom. Eng.*, vol. 42, no. Iccsae 2015, pp. 548–554, 2016, doi: 10.2991/iccsae-15.2016.101.

[11] L. Khine and J. C. Alimagno, “Die sticking quality issue of tape-and-reel packaging for WLCSP,” *2019 IEEE 21st Electron. Packag. Technol. Conf. EPTC 2019*, pp. 676–678, 2019, doi: 10.1109/EPTC47984.2019.9026568.

[12] A. Gallegos-Hernández, F. J. Ruiz-Sánchez, and J. R. Villalobos-Cano, “2D

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

automated visual inspection system for the remote quality control of SMD assembly," *IECON Proc. (Industrial Electron. Conf.*, vol. 3, pp. 2219–2224, 2002, doi: 10.1109/iecon.2002.1185317.

[13] Y. Cen, J. He, and D. Won, "Defect patterns study of pick-and-place machine using automated optical inspection data," *Solder. Surf. Mt. Technol.*, vol. 34, no. 2, pp. 69–78, 2022, doi: 10.1108/SSMT-03-2021-0007.

[14] K. Ferencz and J. Domokos, "Using Node-RED platform in an industrial environment," *Jubil. Kandó Konf.*, no. February, p. 13, 2020, [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://www.researchgate.net/profile/Katalin-Ferencz/publication/339596157_Using_Node-RED_platform_in_an_industrial_environment/links/5e5ab48c4585152ce8fc6a6c/Using-Node-RED-platform-in-an-industrial-env.

[15] J. Asante and J. Olsson, "Using Node-Red to Connect Patient, Staff and Medical Equipment," pp. 1–77, 2016, [Online]. Available: https://www.diva-portal.org/smash/get/diva2:949264/FULLTEXT01.pdf.

[16] A. Al Bataineh, "A Comparative Study of Different Curve Fitting Algorithms in Artificial Neural Network using Housing Dataset," *NAECON 2018 - IEEE Natl. Aerosp. Electron. Conf.*, pp. 174–178, 2018.

[17] A. Abbasi and M. Jamil, "Bayesian Regularization Based Electrical Load Forecasting," *2023 Int. Conf. Recent Adv. Electr. Electron. Digit. Healthc. Technol.*, pp. 415–419, 2023, doi: 10.1109/REEDCON57544.2023.10150744.

اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

[18] X. Qi, Y. Wang, and G. Zhen, “Application of Neural Network Analysis Based on Bayesian Regularization in Crosstalk of Cable,” *2020 6th Glob. Electromagn. Compat. Conf.*, pp. 1–3, doi: 10.1109/GEMCCON50979.2020.9456685.

[19] M. Kayri, “Predictive Abilities of Bayesian Regularization and Levenberg – Marquardt Algorithms in Artificial Neural Networks : A Comparative Empirical Study on Social Data,” 2016, doi: 10.3390/mca21020020.

[20] T. Le Magueresse *et al.*, “Instantaneous Bayesian regularization applied to real-time near-field acoustic holography To cite this version : HAL Id : hal-01714326,” 2021, doi: 10.1121/1.4998571.

[21] S. Srivastava, “Path Detection for Self-Driving Carts by using Canny Edge Detection Algorithm,” *2021 9th Int. Conf. Reliab. Infocom Technol. Optim. (Trends Futur. Dir.*, pp. 1–5, 2021, doi: 10.1109/ICRITO51393.2021.9596109.

[22] C. Wu, “An Improved Canny Edge Detection Algorithm with Iteration Gradient Filter,” *2022 6th Int. Conf. Imaging, Signal Process. Commun.*, pp. 16–21, 2022, doi: 10.1109/ICISPC57208.2022.00011.

[23] L. Chandrasekar and G. Durga, “Implementation of Hough Transform for Image Processing Applications,” *2014 Int. Conf. Commun. Signal Process.*, pp. 843–847, 2014, doi: 10.1109/ICCSP.2014.6949962.

[24] J. Kolluri, V. K. Kotte, M. S. B. Phridviraj, and S. Razia, “Using Novel L1 / 4 Regularization Method,” no. Icoei, pp. 934–938, 2020.

[25] M. Ma, Y. Zhang, Z. Yang, and Y. Shang, “A class of Nonmonotone Spectral

UTeM
اونيورسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

conjugate gradient methods with the generalized quasi-Newton equation," *2011 Int. Conf. Mechatron. Sci. Electr. Eng. Comput.*, pp. 1880–1883, 2011, doi: 10.1109/MEC.2011.6025852.

[26] C. Webb, "Developing and evaluating predictive conveyor belt wear models," 2020, doi: 10.1017/dce.2020.1.

[27] H. Li, J. Li, X. Guan, B. Liang, Y. Lai, and X. Luo, "Research on overfitting of deep learning," *2019 15th Int. Conf. Comput. Intell. Secur.*, pp. 78–81, 2019, doi: 10.1109/CIS.2019.00025.

[28] J. Yuan, "Research on Employee Performance Prediction Based on Machine Learning," *2022 IEEE 5th Int. Conf. Electron. Technol.*, pp. 1296–1302, 2022, doi: 10.1109/ICET55676.2022.9824477.

[29] G. J. Dolecek, "Using OFB and LDPC Coding to Decrease MSE Produced by Gaussian and Impulse Noise," *2018 IEEE 61st Int. Midwest Symp. Circuits Syst.*, pp. 639–642, 2018, doi: 10.1109/MWSCAS.2018.8624047.

[30] R. B. Roy, "A Comparative Performance Analysis of ANN Algorithms for MPPT Energy Harvesting in Solar PV System," *IEEE Access*, vol. 9, pp. 102137–102152, 2021, doi: 10.1109/ACCESS.2021.3096864.

[31] N. Ahmad, Y. Ghadi, and S. Member, "Load Forecasting Techniques for Power System : Research Challenges and Survey," *IEEE Access*, vol. 10, no. June, pp. 71054–71090, 2022, doi: 10.1109/ACCESS.2022.3187839.

[32] J. Qi *et al.*, "On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression," vol. 27, pp. 1485–1489, 2020, doi:

UTeM
اونيۈرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

10.1109/LSP.2020.3016837.

[33] D. Torres, P. Dias, I. Tec, H. S. Ferreira, and I. Tec, “Real-time Feedback in Node-RED for IoT Development : An Empirical Study.”

[34] M. M. Maran and N. A. Paniavin, “Alternative Approaches to Data Storing and Processing,” pp. 6–9, 2020.

.

UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

APPENDIX A

CODING FOR THE SYSTEM

```
from picamera2 import Picamera2
from libcamera import controls
import cv2
import numpy as np
import time
import csv

picam2 = Picamera2()
config = picam2.create_preview_configuration()
picam2.configure(config)
picam2.set_controls({"AfMode":controls.AfModeEnum.Continuous})
picam2.start()

regulated_line_position = None
pixel_to_mm_ratio = 0.027
start_time = time.time()
interval = 1

csv_filename = "offset_data.csv"
with open(csv_filename, mode='w', newline='') as csv_file:
    fieldnames = ['Timestamp', 'Distance_mm']
    writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
    writer.writeheader()

    while True:
        im = picam2.capture_array()
        gray = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
        edges = cv2.Canny(gray, 50, 150)

        roi_y = 220
        roi_height = im.shape[0] // 16
        roi = edges[roi_y:roi_y + roi_height, :]
        lines = cv2.HoughLinesP(roi, 1, np.pi / 180, threshold=100, minLineLength =100,
maxLineGap=10)
```

UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA

```
    if lines is not None:
        min_distance = float('inf')
        for line in lines:
            x1, y1, x2, y2 = line[0]
            angle = np.arctan2(y2 - y1, x2 - x1) * 180 / np.pi

            if 175 <= angle <= 185 or -5 <= angle <= 5 or 176 <= angle <= 186 or -6 <= angle <= 6:
                line_center = y2
                image_center = im.shape[1] / 2
                offset_pixels = line_center - image_center
                distance_mm = abs(offset_pixels) * pixel_to_mm_ratio

                if distance_mm < min_distance:
                    min_distance = distance_mm

                    cv2.rectangle(im, (x1, roi_y + y1), (x2, roi_y + y2), (0, 255, 0), 2)

        if min_distance != float('inf') and time.time() - start_time >= interval:
            timestamp = time.strftime('%Y-%m-%d %H:%M:%S')
            writer.writerow({'Timestamp': timestamp, 'Distance_mm': min_distance})
            print("Distance to regulated line (mm):", min_distance)
            start_time = time.time()

    cv2.line(im, (0, im.shape[0] // 2), (im.shape[1], im.shape[0] // 2), (255, 0, 0), 1)

    cv2.imshow("Frame", im)

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

UTeM
اونيۏرسيتي تيكنيكل مليسيا ملاك
UNIVERSITI TEKNIKAL MALAYSIA MELAKA