ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun463

Performance Analysis of Vehicle License Plate Recognition Using Parallel and Sequential Image Processing on a Multi-Core Processor

Sajani Deshika Manamperi[1]; Aruni Nadeesha Weerasinghe[2]

[1] Department of Electrical Engineering, The STEM Centre, Technology Campus, St Helens College, United Kingdom
[2] Department of Electrical and Electronics Engineering, Advanced Technological Institute, Galle

Publication Date: 2026/06/19

Abstract: Automatic Vehicle License Plate Recognition (VLPR) has become an important application in intelligent transportation systems, traffic management, and security monitoring. However, conventional sequential image processing approaches often experience increased computational time when processing large volumes of visual data.

This research presents a performance evaluation of vehicle license plate recognition using sequential and parallel image processing approaches implemented a multi-core embedded processor. The proposed system integrates Python, OpenCV, and enhanced through parallel execution of image processing tasks using the multiprocessing module.

The system performs image acquisition, preprocessing, license plate detection, character segmentation, and character recognition using a k-Nearest Neighbour (kNN) classifier. Parallel execution is implemented to accelerate independent processing operations including loading classification datasets and flattened image training data.

Experimental results obtained from multiple vehicle license plate images demonstrate that parallel execution significantly reduces execution time compared with sequential processing while maintaining recognition performance. The study confirms that parallel processing improves computational efficiency and supports real-time image processing applications.

Keywords**:** *Image Processing, Vehicle License Plate Recognition (VLPR), Real-Time Processing, Multi-Core Processor, Parallel Processing, Python Multiprocessing, Character Recognition, Vehicle Identification, Image Segmentation, Performance Analysis.*

How to Cite: Sajani Deshika Manamperi; Aruni Nadeesha Weerasinghe (2026) Performance Analysis of Vehicle License Plate Recognition Using Parallel and Sequential Image Processing on a Multi-Core Processor. *International Journal of Innovative Science and Research Technology*, 11(6), 599-605. https://doi.org/10.38124/ijisrt/26jun463

I. INTRODUCTION

Image processing plays an essential role in modern intelligent systems where automated analysis of visual information is required. It focuses on improving pictorial information for human interpretation and processing image data for various applications [6],[9]. Image processing techniques mainly include image enhancement, image segmentation, feature extraction, image classification, and object recognition.

Today, image processing technologies are widely used in security systems, traffic monitoring, military applications, banking systems, automatic toll collection, smart transportation systems, industrial inspection, and automated recognition systems [3],[4]. Vehicle License Plate Recognition (VLPR) has emerged as an important image-processing application because of its ability to automate vehicle identification without manual intervention [1], [5], [7], [14].

Despite significant advances in recognition technologies, real-time license plate recognition remains a computationally demanding task [10], [16]. Accurate recognition depends on several image processing stages including image acquisition, preprocessing, noise reduction, license plate localization, character segmentation, and character recognition. Environmental conditions such as illumination variation, vehicle motion, camera angle, image resolution, and background complexity further increase processing requirements and reduce recognition efficiency. To overcome these challenges, high-speed image processing techniques and efficient computational platforms are required.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun463

Traditional sequential image processing approaches execute these operations one after another, resulting in increased execution time and reduced suitability for real-time applications [14]. As image datasets become larger and recognition systems become more complex, improving computational efficiency has become a major research challenge.

To address this challenge, modern multi-core processors provide an effective solution by allowing multiple tasks to execute simultaneously across several processing cores. Python multiprocessing and OpenCV libraries are utilized to improve execution performance by distributing workload among available processing cores.

This research analyses the performance improvement achieved by implementing vehicle license plate recognition using sequential and parallel image processing techniques on a multi-core embedded processor. Experimental analysis was conducted using multiple vehicle license plate images to compare processing speed and computational efficiency.

The main contributions of this research are summarized as follows:

- Development of a vehicle license plate recognition system using image processing techniques.
- Implementation of parallel execution using Python multiprocessing for performance improvement.
- Comparative analysis between sequential and parallel processing methods.
- Evaluation of processing time and system efficiency for real-time recognition applications.

The outcomes of this study highlight the potential of multi-core processing architectures for accelerating image-based recognition systems and provide a foundation for future intelligent transportation and real-time monitoring applications.

II. FUNDAMENTAL PRINCIPLES

This section presents the theoretical concepts and fundamentals supporting the development and performance evaluation of the proposed vehicle license plate recognition system. The research combines digital image processing techniques with computational performance optimization through parallel execution on a multi-core processor.

➢ *Digital Image Processing*

Digital image processing implies the acquisition, conversion, analysis, and interpretation of digital images using computational algorithms. The primary objective of digital image processing is to improve image quality and extract meaningful information for automated decision-making and recognition applications.

The basic stages of image processing include image acquisition, preprocessing, segmentation, feature extraction, and recognition. These operations enable systems to identify objects and patterns that may not be easily observable through direct visual inspection.

In this license plate recognition system, digital image processing provides the foundation for detecting vehicle plates and extracting character information from captured images. The accuracy and speed of recognition largely depend on the effectiveness of preprocessing and feature extraction techniques.

➢ *Vehicle License Recognition (VLPR)*

Vehicle License Plate Recognition (VLPR) is an automated image-processing technique used to identify vehicle license plate numbers from captured images.

A typical VLPR system consists of several sequential stages:

- Image acquisition
- Image preprocessing
- License plate localization
- Character segmentation
- Character recognition
- Post-processing and output generation

During preprocessing, image quality is improved through grayscale conversion, thresholding, and noise removal. License plate localization identifies the region containing the plate, while segmentation separates individual characters. Recognition algorithms then classify each character and generate the final license plate number. The research implements these stages using OpenCV-based image processing and character recognition methods.

➢ *Image segmentation and Character Extraction*

Image segmentation is the process of dividing an image into meaningful segments to isolate objects of significance. In license plate recognition, segmentation plays an important role because characters must be separated from the background before recognition.

Character extraction typically uses contour detection and geometric analysis to locate character boundaries and remove irrelevant regions.

The accuracy of segmentation directly affects recognition performance because incorrectly segmented characters lead to classification errors.

In this research, contour-based character grouping and extraction techniques were used to detect and isolate possible license plate characters before recognition.

➢ *Character Recognition Using k-Nearest Neighbour (kNN)*

Character recognition is the final stage of the recognition pipeline where segmented characters are converted into readable alphanumeric values.

The k-Nearest Neighbour (kNN) algorithm is a supervised machine learning method commonly used for

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun463

classification tasks. The algorithm compares an unknown input sample with previously trained samples and assigns the class that produces the smallest distance.

In this research, kNN is used to recognize segmented license plate characters by comparing them against trained character datasets.

The advantages of kNN include simplicity, ease of implementation, and suitability for small-scale classification applications.

➢ *Parallel Processing and Multi-Core Computing*

Parallel processing refers to the simultaneous execution of multiple tasks using multiple processor cores. Unlike sequential execution, where tasks are processed one at a time, parallel processing distributes computational workload across available processing resources.

Multi-core processors contain two or more independent processing cores integrated into a single unit. This architecture improves computational throughput and enables faster execution of intensive image processing operations.

For real-time image recognition systems, parallel execution reduces waiting time and improves responsiveness by utilizing processor resources more efficiently. In this research, multiprocessing was implemented to execute independent operations concurrently and evaluate performance improvements compared with sequential processing.

➢ *Python Multiprocessing Framework*

Python supports parallel execution through the multiprocessing module, which creates independent processes that can operate simultaneously across multiple CPU cores.

Unlike multithreading, multiprocessing allows separate memory spaces and avoids limitations associated with Python's Global Interpreter Lock (GIL).

The multiprocessing approach enables independent execution of tasks such as:

- Training data loading
- Image preprocessing
- Character detection
- Recognition operations

This capability makes Python suitable for computationally intensive image-processing applications.

➢ *OpenCV for Computer Vision Applications*

OpenCV (Open-Source Computer Vision Library) is a widely used computer vision framework that provides optimized algorithms for image processing and object recognition.

OpenCV supports image acquisition, transformation, filtering, segmentation, contour detection, and classification operations required for real-time vision systems.

In this research, OpenCV was used to implement preprocessing, license plate extraction, contour analysis, and visualization of recognition results.

III. SYSTEM IMPLEMENTATION AND TECHNOLOGIES

This section describes the hardware platform, software tools, and system architecture used to implement the proposed vehicle license plate recognition system. The implementation integrates image processing algorithms with parallel computing techniques to improve computational efficiency and reduce processing time.

➢ *Hardware Platform*

The proposed system was implemented on a multi-core processing platform capable of executing multiple computational tasks simultaneously. Multi-core processors provide improved performance compared with conventional single-core architectures by distributing workloads among independent processing units.

The utilization of a multi-core processing environment is particularly beneficial for image processing applications, as operations such as preprocessing, feature extraction, and segmentation require significant computational resources. By developing parallel execution, processing latency can be reduced while improving overall system responsiveness.

A digital image acquisition device was used to capture vehicle images containing license plates. The captured images served as the input dataset for the recognition process.

➢ *Software Environment*

The proposed system was developed using Python due to its simplicity, flexibility, and extensive support for computer vision applications.

Several software libraries were integrated during development:

- *Python*

Python was selected as the primary programming language because of its rapid development capability and availability of scientific computing libraries. The language also provides built-in support for parallel processing through the multiprocessing module.

- *OpenCV*

OpenCV (Open-Source Computer Vision Library) was used as the primary image processing framework. It provides optimized implementations of image manipulation, filtering, contour analysis, thresholding, and object detection algorithms required for license plate recognition.

- *NumPy*

NumPy was used for numerical computations and matrix operations. Since digital images are represented as multidimensional arrays, NumPy enables efficient manipulation of image data during processing.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun463

- *Multiprocessing Module*

The Python multiprocessing module was employed to execute independent tasks concurrently across available processor cores. This module creates separate processes that operate simultaneously and allows more effective utilization of computational resources.

➢ *System Architecture*

The proposed vehicle license plate recognition system consists of several interconnected processing stages.

Initially, an image containing a vehicle is acquired and supplied to the preprocessing stage. The image is then converted into grayscale format and enhanced to improve the visibility of license plate features.

Subsequently, character regions are detected using contour-based analysis techniques. Matching character vectors are identified and used to locate possible license plate regions within the image. Once the plate region has been extracted, individual characters are segmented and passed to the recognition stage.

Finally, a trained k-Nearest Neighbour (kNN) classifier is used to identify the alphanumeric characters and generate the recognized license plate number.

➢ *Parallel Processing Architecture*

To evaluate computational performance, both sequential and parallel execution models were implemented.

In the sequential implementation, image processing tasks were executed one after another. Each task completed before the next operation commenced.

In the parallel implementation, independent operations were executed concurrently using multiple processor cores. Specifically, the loading of classification data and flattened training images was performed simultaneously using Python multiprocessing techniques.

The parallel execution model enables better utilization of available hardware resources and forms the basis for the performance comparison presented in the subsequent sections.

➢ *Recognition Framework*

The recognition framework combines image processing and machine learning techniques to identify license plate characters.

The framework consists of the following major components:

- Image preprocessing module
- Character detection module
- License plate extraction module
- Character segmentation module
- kNN classification module
- Result visualization module

Each module contributes to the overall recognition process and collectively enables automatic extraction of vehicle registration numbers from input images.

IV. SIMULATION OF METHODOLOGY AND DESIGN

This section describes the methodology adopted for the implementation and evaluation of the proposed Vehicle License Plate Recognition (VLPR) system. The system was developed using Python and OpenCV and was designed to compare the performance of sequential and parallel image processing approaches on a multi-core processing environment.

➢ *Methodology*

The proposed recognition framework consists of a sequence of image processing and character recognition stages. Initially, a vehicle image is acquired and subjected to preprocessing operations to improve image quality and facilitate feature extraction. The processed image is then analyzed to identify potential character regions, which are subsequently grouped to determine the location of the license plate. Once the license plate region is extracted, individual characters are segmented and recognized using a trained k-Nearest Neighbour (kNN) classifier.

The overall processing workflow of the proposed system is illustrated in Figure 1.

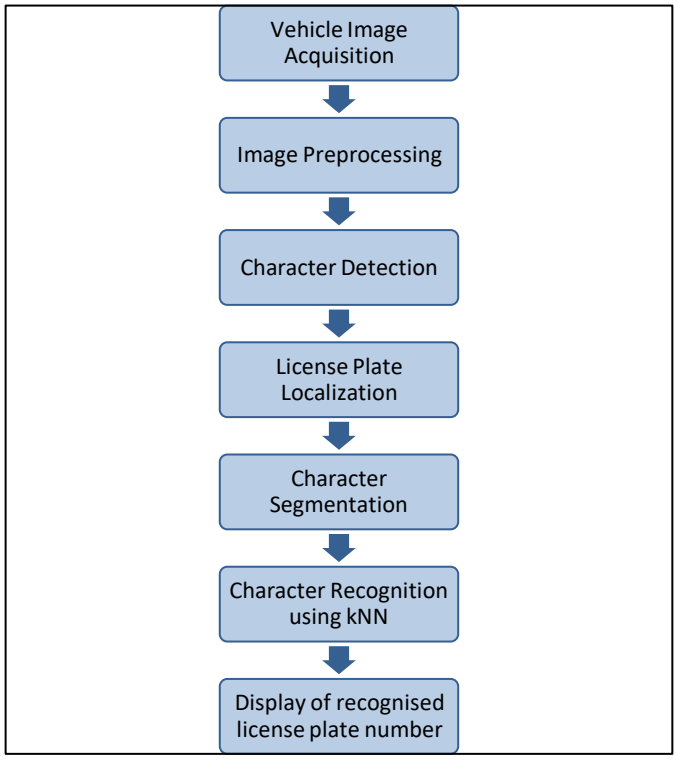

Fig 1 Overall Processing Workflow of Vehicle License Plate Recognition System

The methodology was implemented under both sequential and parallel execution environments to evaluate computational performance.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun463

➢ *Image Preprocessing*

Image preprocessing was performed to enhance image quality and improve the accuracy of subsequent processing stages. The acquired vehicle image was first converted into grayscale format to reduce computational complexity while preserving essential visual information.

Noise reduction techniques were then applied to minimize image artifacts and improve feature visibility. Adaptive thresholding was subsequently performed to generate a binary image that clearly separates foreground characters from the background. These preprocessing operations improve character detection accuracy and facilitate reliable license plate localization.

➢ *Character Detection and License Plate Localization*

The detection stage identifies potential character regions within the image using contour-based analysis techniques. Geometric properties such as area, aspect ratio, width, and height were evaluated to distinguish character regions from non-character objects.

Detected character regions were grouped according to spatial relationships and similarity measures. Groups of matching characters were then analyzed to identify possible license plate regions. The region containing the most probable character sequence was selected as the candidate license plate.

➢ *Character Segmentation*

Following plate localization, the extracted license plate image was processed to isolate individual characters. Contour detection techniques were employed to identify character boundaries within the plate region.

Unwanted contours and overlapping regions were removed based on predefined geometric constraints. The remaining segmented character images were normalized and prepared for classification.

➢ *Character Recognition Using kNN*

Prior to testing, the classifier was trained using labeled character samples stored as classification and flattened image datasets. Each segmented character was resized to a standard dimension and transformed into a feature vector. The trained kNN model then compared the input vector with stored training samples and assigned the most similar character class.

➢ *Parallel Processing Implementation*

To investigate the impact of parallel execution on system performance, multiprocessing techniques were incorporated into the implementation.

The Python multiprocessing module was utilized to execute independent computational tasks concurrently on multiple processor cores.

➢ *Sequential Processing Implementation*

In the sequential implementation, all processing tasks were executed one after another within a single execution flow. Each operation completed before the next stage commenced.

➢ *Performance Evaluation*

The effectiveness of the proposed system was evaluated using a set of vehicle license plate images under both processing modes. The primary performance metric considered in this study was total image processing time.

The evaluation focused on:

- Processing time under sequential execution
- Processing time under parallel execution
- Comparative speed improvement
- System efficiency in utilizing processor resources

Execution times were recorded for each test image, and the results were analyzed to determine the effectiveness of multiprocessing techniques in accelerating vehicle license plate recognition tasks.

V. PERFORMANCE ANALYSIS RESULTS

This section presents the experimental results obtained from the proposed Vehicle License Plate Recognition (VLPR) system and evaluates the performance of sequential and parallel image processing approaches.

➢ *Experimental Setup*

The performance evaluation was conducted using a dataset of ten vehicle license plate images. Each image was processed under both sequential and parallel execution environments using the same recognition algorithm and preprocessing procedures. Processing time was recorded for each image from the initialization stage to the completion of license plate recognition.

To ensure a fair comparison, all experiments were performed using the same hardware and software environment. The recognition methodology, preprocessing operations, and classification algorithm remained unchanged throughout the evaluation process.

➢ *Processing Time Comparison*

Table 1 Comparison of Parallel and Sequential Processing Times

Image Number	Parallel Processing (s)	Sequential Processing (s)
1	1.369	1.842
2	0.821	1.161
3	0.620	0.903
4	1.027	1.455
5	0.887	1.266
6	0.937	1.250

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun463

7	0.896	1.201
8	0.892	1.271
9	0.786	1.179
10	0.794	1.107

The results demonstrated that parallel implementation consistently required less processing time than the sequential implementation for all test images.

The average processing time for the parallel implementation was approximately 0.903 s, whereas the sequential implementation required approximately 1.264 s. This corresponds to an average reduction in execution time of approximately 28.6%.

➢ *Speedup Analysis*

Using the average execution times obtained during experimentation, the overall speedup achieved by the proposed parallel implementation was approximately 1.40. This indicates that the parallel approach executed approximately 40% faster than the sequential approach.

The observed improvement demonstrates the effectiveness of distributing independent computational tasks across multiple processor cores.

➢ *Discussion of Results*

The experimental results clearly indicate that parallel processing provides significant performance benefits for vehicle license plate recognition applications. In every test case, the multiprocessing implementation achieved lower processing times than the corresponding sequential implementation.

Furthermore, the reduction in processing time was achieved without altering the recognition algorithm or compromising recognition capability. This demonstrates that performance improvements were obtained primarily through improved computational efficiency rather than changes to the recognition methodology.

The results confirm that multiprocessing techniques can effectively accelerate computationally intensive image processing applications and support real-time license plate recognition requirements.

VI. DESIGN OPTIMIZATIONS

To improve the computational efficiency of the system, several optimization techniques were incorporated during implementation. These optimizations focused on reducing processing time, improving resource utilization, and maintaining reliable recognition performance.

➢ *Parallel Processing Optimization*

The primary optimization introduced in this research was the use of Python's multiprocessing framework to execute independent tasks concurrently across multiple processor cores. In the parallel implementation, the loading of classification data and flattened image datasets was performed simultaneously rather than sequentially. This approach improved processor utilization and reduced overall execution time, resulting in better performance compared with the conventional sequential implementation.

➢ *Image Processing Optimization*

Image preprocessing operations were optimized through grayscale conversion and adaptive thresholding to reduce image complexity and improve character visibility. In addition, contour-based filtering techniques were employed to eliminate irrelevant image regions and identify potential character candidates. These optimizations improved the efficiency of license plate localization and character segmentation while reducing computational overhead.

➢ *Recognition and Memory Optimization*

Recognition performance was enhanced by utilizing pre-trained k-Nearest Neighbour (kNN) classification data, eliminating the need for repeated training during execution. Furthermore, efficient memory management techniques and optimized OpenCV functions were employed to reduce processing overhead and accelerate image manipulation operations. These improvements contributed to faster execution while maintaining reliable recognition accuracy.

VII. CONCLUSIONS

This study investigated the performance of vehicle license plate recognition (VLPR) using sequential and parallel image processing techniques on a multi-core processing environment. A VLPR system was developed using Python, OpenCV, and the k-Nearest Neighbour (kNN) algorithm to perform image preprocessing, license plate localization, character segmentation, and character recognition.

The experimental results demonstrated that the parallel implementation consistently outperformed the sequential approach in terms of processing speed. The average processing time was reduced from 1.264 s to 0.903 s, achieving an execution time reduction of approximately 28.6% and an overall speedup factor of 1.40. These findings confirm that multiprocessing can effectively improve computational efficiency and processor utilization for image processing applications without modifying the underlying recognition algorithm.

The proposed system successfully detected and recognized vehicle license plates under various test conditions, demonstrating the effectiveness of combining OpenCV-based image processing techniques with kNN-based character recognition. The results indicate that multi-core processing architecture provides a practical solution for real-time vehicle identification and intelligent transportation applications.

However, recognition performance may be affected by factors such as poor image quality, illumination variations, motion blur, and damaged license plates. Future work may

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/26jun463

focus on deep learning-based recognition methods, GPU-accelerated processing, multilingual license plate recognition, and real-time video stream analysis to further enhance system accuracy and performance.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to all their affiliated academic institutions for providing the academic environment, facilities, and support that contributed to the successful completion of this research. The authors also acknowledge the guidance, encouragement, and assistance received from colleagues and staff members throughout the various stages of this work.

REFERENCES

[1]. D.G. Bailey, "Test Bed for Number Plate Recognition Applications," International Workshop on Electronic Design, Test and Applications, New Zealand, 2002.
[2]. T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, Introduction to Algorithms, MIT Press, 2001.
[3]. S. Davidson, "The Use of Automatic Number Plate Recognition in Managing the Boots Company Headquarters Site," IEEE International Conference on Security Technology, London, 2001.
[4]. S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1998.
[5]. H. Lee, "Real Time Automatic Vehicle Management System Using Vehicle Tracking and Car Plate Number Identification," IEEE ICME, Baltimore, USA, 2003.
[6]. J.R. Parker, Algorithms for Image Processing and Computer Vision, John Wiley and Sons, 1997.
[7]. C.A. Rahman, "A Real Time Vehicle's License Plate Recognition System," IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.
[8]. M. Sarfraz, "Saudi Arabian License Plate Recognition System," International Conference on Geometric Modeling and Graphics, London, 2003.
[9]. M. Sezgin et al., "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," Journal of Electronic Imaging, vol. 13, no. 1, pp. 317–327, 2004.
[10]. M. Shridhar, "Recognition of License Plate Images: Issues and Perspectives," Conference on Document Analysis and Recognition, 1999.
[11]. T. Sirithinaphong, "The Recognition of Car License Plate for Automatic Parking System," International Symposium on Signal Processing and its Applications, Australia, 1999.
[12]. W. Wei, "An Automatic Method of Location for Number-Plate Using Color Feature," IEEE ICIP, 2001.
[13]. Wei Xie and Y. Wu, "License Plate Automatic Recognition System Based on MATLAB-GUI," The Open Automation and Control Systems Journal, pp. 497–502, 2014.
[14]. Shan Du, "Automatic License Plate Recognition (ALPR): A State-of-the-Art Review," IEEE Transactions on Circuits and Systems for Video Technology, vol. 23, no. 2, February 2013.
[15]. H. Kaur and N. Kumar Garg, "Indian Number Plate Recognition and Segmentation Using K Mean and Neural Network Classifier," International Journal of Advances in Computer Science and Communication Engineering (IJACSCE), vol. 2, issue 2, June 2014.
[16]. P. Prabhakar and P. Anupama, "A Novel Design for Vehicle License Plate Detection and Recognition," IEEE 2nd International Conference on Current Trends in Engineering and Technology, July 2014.

BIOGRAPHIES

Eng. (Mrs.) Sajani Deshika Manamperi is a Lecturer in the Department of Electrical Engineering at Technology Campus, St. Helens College, United Kingdom. She previously served as a Lecturer in Electrical and Electronics Engineering at the Advanced Technological Institute (ATI), Galle, Sri Lanka (2013–2022), where she was actively involved in teaching, curriculum delivery, and the academic development of engineering students.

She has professional experience and applied engineering practice, with a focus on competency-based learning in electrical and electronics engineering education. Her interests include modern engineering education methods and the use of emerging technologies in engineering education and skill development.

Eng. (Mrs.) Aruni Weerasinghe is a Lecturer in the Department of Electrical Engineering under the Higher National Diploma in Engineering programme at the Advanced Technological Institute (ATI), Galle. She has been serving at the institute since 2013, contributing significantly to the academic and professional development of engineering students. Her teaching responsibilities include several electrical modules, particularly Electrical Installation Practice. Her academic interests encompass electrical engineering education, optical communication technologies, and applied engineering research.