Smart Areca Leaf Plate Grading System

A PROJECT PHASE-2 REPORT

Submitted to

Visvesvaraya Technological University

BELAGAVI - 590 018

by

Harsha G V Subedar	**4SU22AD016**
Laxman Sannu Gouda	**4SU22AD025**
Rahul Malatesh Sannapujar	**4SU22AD043**
Sateerth Ganapati Palkar	**4SU23AD401**

Under the guidance of

Mrs. Veena Bhat

Assistant Professor and HOD

in partial fulfilment of the requirements for the award of the degree of

Bachelor of Engineering

Department of Artificial Intelligence and Data Science

SDM INSTITUTE OF TECHNOLOGY

UJIRE - 574 240

2025-2026

SDM INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

UJIRE-574 240

Department of Artificial Intelligence & Data Science

CERTIFICATE

Certified that the Project Phase -2 Work titled '**Smart Areca Leaf Plate Grading System'** is carried out by **Mr. Harsha G V Subedar**, USN: **4SU22AD016, Mr. Laxman Sannu Gouda**, USN: **4SU22AD025**, **Mr. Rahul Malatesh Sannapujar**, USN: **4SU22AD043**, **Mr. Sateerth Ganpati Palkar**, USN:**4SU23AD401**, Bonafide students of SDM Institute of Technology, Ujire, in partial fulfilment for **VI Semester Project Phase-II** in Artificial Intelligence & Data Science of Visvesvaraya Technological University, Belagavi during the year 2025-2026. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in respect of project phase-2 work prescribed for the said degree.

-------------------------	-------------------------	---------------------------
Mrs. Veena Bhat	**Mrs. Veena Bhat**	**Dr. Ashok Kumar T**
Assistant Professor and Guide	Head of Department	Principal

Signature with Date and seal

External Viva

Name of the Examiners: Signature with Date

1.

2.

SDM INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

UJIRE – 574 240

Department of Artificial Intelligence and Data Science

DECLARATION

We declare that this project report titled '**Smart Areca Leaf Plate Grading System'** submitted in partial fulfilment of the degree of B.E in in Artificial Intelligence & Data Science is a record of original work carried out by us under the supervision of **Mrs. Veena Bhat**, and has not formed the basis for the award of any other degree, in this or any other Institution or University. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Signature of the students

Name	**Signature**
1. Harsha G V Subedar	
2. Laxman Sannu Gouda	
3. Rahul Malatesh Sannapujar	
4. Sateerth Ganapati Palkar	

Acknowledgement

It is our pleasure to express our heartfelt thanks to **Mrs. Veena Bhat**, Assistant Professor and Head of the Department, Department of Artificial Intelligence and Data Science, for her supervision and guidance which enabled us to understand and develop this mini project.

We are indebted to **Dr. Ashok Kumar T**, Principal, for his advice and suggestions at various stages of the work. We also extend our heartfelt gratitude to **Mr. Padma Prasada**, Project Coordinator for his assistance and the management of SDM Institute of Technology, Ujire, for providing us with a good learning environment, library and laboratory facilities.

Lastly, we take this opportunity to offer our regards to all of those who have supported us directly or indirectly in the successful completion of this project work.

Harsha G V Subedar

Laxman Sannu Gouda

Rahul Malatesh Sannapujar

Sateerth Ganapati Palkar

Abstract

As demand for sustainable and eco-friendly alternatives to plastic, areca leaf plates made from naturally fallen palm leaves have become a popular choice for disposable tableware. While they're great for the environment, their natural variation in colour, texture, shape, and occasional defects makes it difficult to inspect their quality by hand. Manual checks are often slow, inconsistent, and rely heavily on human judgment. With recent advances in machine learning, especially convolutional neural networks (CNNs), and the availability of affordable IoT devices, automated systems are now offering a smarter way to grade these plates quickly and accurately. This paper explores how technologies like computer vision, ML, and IoT are being used to automate the grading process, covering steps like capturing images, cleaning and processing them, identifying key features, and classifying plate quality. It also looks at the challenges these systems still face like limited datasets, difficulties with real-time processing, and trouble adapting to different types of leaves. Newer approaches such as edge computing, hybrid ML models, and adaptive learning are also discussed for their potential to make these systems faster and more efficient.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The rapid shift toward environmentally sustainable products has increased the demand for eco-friendly and biodegradable alternatives in various industries. Among these, Areca nut plates have gained widespread popularity due to their natural origin, biodegradability, and suitability for replacing single-use plastic items. As production scales up to meet growing market needs, maintaining consistent quality becomes essential for ensuring customer satisfaction and supporting sustainable business practices. However, traditional manual inspection methods used in Areca nut plate manufacturing are slow, labour-intensive, and highly prone to human error. Fatigue, subjective judgement, and inconsistent lighting conditions often lead to inaccurate grading, making manual inspection inefficient for large-scale production

To address these limitations, our project titled **“Areca Leaf Plate Grading System”** proposes an automated quality assessment solution that integrates deep learning with embedded hardware. The system leverages a Convolutional Neural Network (CNN) trained to identify and categorize Areca nut plates based on important quality parameters such as texture, colour uniformity, cracks, and shape defects. These features directly influence the usability and commercial value of the plates, making accurate grading crucial. Unlike manual inspection, a CNN can evaluate each plate with consistent precision, reducing subjectivity and improving the reliability of the grading process.

The hardware backbone of the system is a Raspberry Pi, which processes images captured by a high-resolution camera placed above a conveyor belt. As plates move along the conveyor, the camera takes real-time images and feeds them into the CNN model for instant classification. Based on the predicted grade, a gear-motor-based sorting mechanism automatically redirects the plates into their respective bins. This seamless integration of image capture, intelligent classification, and mechanical sorting creates a fully automated workflow that significantly reduces human intervention.

By improving accuracy, speed, and operational efficiency, the proposed system offers a scalable and cost-effective solution for Areca nut plate manufacturers. It not only minimizes labour costs but also ensures uniform product quality—an essential requirement in today’s competitive market.

Chapter 2

Literature Survey

The literature survey is aimed to encompass the advancements, limitations, and ongoing challenges in the field of automated quality inspection in biodegradable product manufacturing. This chapter identifies the gaps in existing manual and semi-automated grading methods for areca leaf plates, highlighting the need for a reliable, real-time, and AI-powered solution. It sets the foundation for the proposed CNN-based Smart Areca Leaf Plate Grading System to address these challenges effectively.

2.1 Existing Systems or Technologies

Several systems and technologies have been proposed and implemented for grading of different products, each with unique methodologies and capabilities. Key advancements include:

This paper presents a comprehensive overview of Convolutional Neural Networks (CNNs) for image recognition tasks. It explores the CNN structure inspired by the animal visual system, emphasizing feature extraction via convolutional layers, pooling, and classification layers. The study details the CNN recognition pipeline using the LeNet-5 architecture, applied to the MNIST dataset for handwritten digit recognition. The CNN achieved high accuracy with minimal preprocessing, leveraging hierarchical layers, weight sharing, and activation functions like ReLU. The results validate CNN's efficiency in extracting spatially relevant features and performing robust classification. The work highlights CNNs as a powerful tool for real-world image recognition applications, albeit with interpretability challenges due to their black-box nature [1].

This study proposes an advanced deep learning-based system for grading Robusta coffee beans into nine distinct quality categories. Using a dataset of high-resolution images captured under controlled conditions, the researchers trained and evaluated four CNN architectures—ResNet34, Inception v3, VGG-16, and EfficientNet-B0. Among them, EfficientNet-B0 demonstrated the highest performance, achieving 100% classification accuracy even under challenging lighting and backgrounds. The study highlights the system's potential to automate and enhance the coffee grading process, ensuring better quality control and operational efficiency for producers. This work improves upon earlier models that graded only six types, showcasing the effectiveness of deep networks in fine-grained agricultural product classification [2].

This paper introduces an automated grading system for Arecanuts using image processing techniques. Images of Arecanuts are acquired using a digital camera setup, followed by feature extraction involving colour, texture, and size measurements. Segmentation techniques like thresholding and morphological operations are applied to isolate the nut from the background. The grading algorithm classifies Arecanuts into three categories based on predefined criteria such as roundness, hue intensity, and surface quality. The study demonstrates a high correlation between the automated system and expert manual grading. This solution aims to reduce labour dependency and standardize the arecanut quality control process, with applications in co-operative processing centres and small-scale agricultural industries [3].

This study presents an automated lemon grading and sorting system using computer vision and a Raspberry Pi-based setup. Lemons are classified into three categories: ripe, semi-ripe, and a combined group of unripe and defective lemons. The process involves capturing RGB images in a controlled lighting chamber, followed by preprocessing (cropping, background removal, and noise reduction). Feature extraction includes colour metrics (RGB means), size (pixel count), and surface irregularities using global/local standard deviation and spatial filters. These features are input into a backpropagation neural network trained on 99 samples. The system achieved an overall classification accuracy of 94%, demonstrating effective and efficient performance in minimizing human errors in post-harvest lemon sorting [4].

This research presents an integrated system for automated plant disease detection and monitoring using image processing techniques and IoT-enabled Raspberry Pi and Arduino modules. The system captures plant leaf images, preprocesses them, and applies segmentation and feature extraction (colour, texture, morphology) for classification using an SVM (Support Vector Machine) model. Additionally, the setup includes fruit grading based on size and colour, using edge detection and RGB value analysis. IoT integration enables environmental monitoring through sensors for humidity, gas, and light, with real-time alerts sent to smartphones. The prototype demonstrates effective disease identification and environmental tracking, suggesting potential for scalable agricultural applications [5].

This paper proposes an embedded ARM-based system for fruit grading using image processing, integrating both visual (colour, size) and physical (weight) attributes. A camera captures fruit images on a conveyor, with MATLAB used for processing to extract RGB colour values and measure size via edge detection. Fruits are graded into categories like Red Small, Red Big, Green Small, and Green Big. A second grading option uses a load cell to classify fruits by weight. The system utilizes LPC2148 ARM processor, CMOS cameras, GSM for communication, and GUI for real-time feedback. Experiments on tomatoes and guavas achieved accurate grading with less than 1% size detection error, highlighting the system's potential for low-cost, high-accuracy agricultural automation [6].

This project demonstrates an image processing system for detecting and classifying geometric shapes using OpenCV and Arduino. The system employs edge detection (Canny algorithm), contour detection, and shape approximation (Ramer-Douglas-Peucker algorithm) to identify shapes such as triangles, rectangles, and pentagons from video input. The shape data is transmitted to an Arduino Uno, which controls a mechanical sorting arm to classify objects into designated bins. Implemented on a custom 3D-printed hardware platform with a conveyor system, this prototype highlights the potential of combining computer vision and robotics for low-cost, real-time industrial automation and object sorting applications [7].

This study presents a real-time machine vision-based grading system for dried figs, classifying them into five grades based on colon, size, and split area. The system includes a feeder, belt conveyors, CCD camera, lighting, image acquisition, and pneumatic sorting unit. A LabVIEW-based grading algorithm processes features extracted via image processing techniques colour intensity (from RGB images), equivalent diameter, and split area (via binarization and ROI segmentation). The system achieved up to 95.2% grading accuracy and a throughput of 90 kg/h. Experimental validation confirmed that using these three features enabled accurate and non-destructive grading, enhancing efficiency and consistency in postharvest fig processing [8].

This paper introduces an automated smart farming system that classifies and grades Apollo tomatoes using image processing and fuzzy logic. The system uses a machine vision pipeline—image acquisition, RGB colour extraction, segmentation, and feature extraction—followed by classification via a Fuzzy Inference System (FIS) implemented in MATLAB. Tomatoes are first classified as good or bad (damaged), and then graded into under-ripe, ripe, or overripe based on colour features. The hardware includes a conveyor integrated with Arduino for control. The system aims to assist farmers by reducing manual effort, ensuring consistency, and improving efficiency in tomato quality assessment [9].

This paper presents a system that integrates Internet of Things (IoT) and machine learning techniques to automate the grading and sorting of fruits, aiming to reduce manual labour and increase efficiency in agricultural processes. The proposed setup includes a conveyor belt, microcontroller, camera module, and motors, which work together to capture fruit images and analyse them using machine learning algorithms such as Convolutional Neural Network (CNN), Artificial Neural Network (ANN), Support Vector Machine (SVM), and

YOLO. These algorithms process features like size, shape, and colour for accurate classification. The study evaluates algorithm performance, with CNN achieving 90% accuracy, KNN 82%, and SVM 80%. Among deep learning architectures tested, VGGNet-16 demonstrated the highest accuracy at 97.34%. The paper concludes that combining machine learning with IoT offers a reliable, scalable solution for real-time fruit classification in modern agricultural systems [10].

This paper presents the design and implementation of a smart grading system for fresh milk using Internet of Things (IoT) and machine learning techniques to improve quality classification in the dairy industry. The system captures quality parameters such as temperature, pH, odour, turbidity, colour, fat, and taste to evaluate milk quality. Artificial Neural Network (ANN) is used for classification into three quality grades (low, medium, high), achieving an accuracy of 98.74%, while K-Means clustering helps group milk samples based on temperature and colour, identifying the third cluster as optimal. The system utilizes real-time monitoring through sensors like DHT22 and processes data using R Studio. It supports stakeholders like farmers and cooperative staff by automating the classification process, reducing time, and enhancing decision-making regarding milk quality and pricing [11].

This review paper provides an in-depth overview of convolutional neural networks (CNNs) in the context of computer vision, highlighting their components, architectures, and diverse applications. It explores fundamental CNN layers such as convolution, pooling, activation functions, batch normalization, dropout, and fully connected layers. The paper thoroughly examines major CNN architectures including AlexNet, VGG, GoogLeNet, ResNet, SENet, and MobileNet, and analyzes their performance in tasks like image classification, object detection, and video prediction. Challenges such as gradient vanishing, overfitting, and model complexity are discussed alongside solutions like residual connections, attention mechanisms, and lightweight networks. It concludes by outlining future directions, emphasizing the integration of CNNs with transformers, domain adaptation, and the need for more interpretable and efficient models [12].

This review paper explores the evolution of soil classification methods, focusing on modern computer vision and deep learning techniques aimed at improving accuracy and efficiency in soil type identification. It outlines two main approaches: traditional image processing-based classification, which involves steps like image acquisition, segmentation, and feature extraction (e.g., texture, colour, particle size), and deep learning-based classification using convolutional neural networks (CNNs), which automate feature learning and improve classification outcomes. The paper discusses datasets developed under diverse environmental conditions using devices like cameras and smartphones, highlights common challenges such as computational demands and data scarcity, and emphasizes dimensionality reduction techniques to boost model performance. It concludes that while both approaches are valuable, deep learning models—especially CNNs—offer higher accuracy and scalability for real-world soil classification applications [13].

This paper explores the implementation of image classification using Convolutional Neural Networks (CNNs), a powerful deep learning approach designed to improve the accuracy of automated image recognition. The study uses the MNIST dataset of grayscale handwritten digits to train and evaluate the CNN model. The model architecture involves several modules including data collection, preprocessing, architecture design, training, evaluation, and deployment. Key features include layer-based feature extraction, SoftMax loss function, and optimization through hyperparameter tuning. The proposed CNN system achieved a classification accuracy of 98%, demonstrating its effectiveness in recognizing image patterns and its potential for use in real-world applications like healthcare, autonomous vehicles, and surveillance. Future work aims to scale the model for color image classification and image segmentation tasks [14].

This research proposes an automated system for classifying arecanut (supari) grades using computer vision and machine learning, addressing the inefficiencies and inconsistencies of traditional manual grading. The system captures high-resolution images of arecanuts and extracts relevant visual features, which are then processed using a convolutional neural network (CNN) to classify the nuts into categories such as "good," "bad," or "low-quality." The automated approach enables real-time classification, enhances accuracy, reduces labor, and supports quality control and defect detection in arecanut processing. This work demonstrates the potential of AI-based solutions in agricultural automation and quality assurance [15].

This paper presents an automated tomato grading system using computer vision and image processing techniques to evaluate fruit quality based on ripeness and defects. The system comprises a hardware setup with a conveyor belt and camera for image acquisition, and a software module that preprocesses, segments, and analyzes images. Key features such as color statistics and texture (from R, G, B channels and GLCM matrices) are extracted and selected using Sequential Forward Selection. A neural network classifier is used to categorize tomatoes as defective/non-defective and ripe/unripe. Experimental results on 520 tomato images achieved 100% accuracy in defect detection and 96.47% in ripeness classification. The system enhances grading efficiency and reduces human error, though improvements in speed and performance under high specular reflection conditions are suggested for real-world deployment [16].

This review paper presents a comprehensive study of fruit grading systems used for quality inspection, highlighting both traditional and modern methods that integrate computer vision and machine learning. It covers various technologies such as imaging spectroscopy, multispectral imaging, CMOS cameras, and x-ray imaging to assess attributes like size, volume, hydration, defects, and internal structure. Classification methods including neural networks, fuzzy inference systems, and K-means clustering are explored for fruit sorting. The paper also discusses embedded grading systems, robotic manipulators, and wearable sensor platforms for real-time and automated fruit assessment. Recognition rates reported across different systems range from 80% to over 90%, demonstrating the effectiveness of integrating advanced image processing with AI for improving accuracy, efficiency, and scalability in agricultural grading applications [17].

This paper provides a detailed review of machine vision systems applied to fruit classification and grading, emphasizing non-destructive methods that utilize image processing and machine learning. It outlines the complete workflow including image acquisition, preprocessing, segmentation, feature extraction (colour, shape, size, texture), and classification using algorithms such as K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Artificial Neural Networks (ANN), and Convolutional Neural Networks (CNN). Feature descriptors like SURF, HOG, and LBP are discussed along with their relevance in improving classification accuracy. The paper highlights various studies where techniques achieved accuracies ranging from 83% to over 98%, demonstrating the efficiency of combining visual features with intelligent algorithms. It concludes by recommending future work on developing systems for local fruits, real-time grading applications, and mobile-based solutions for practical deployment in agricultural industries [18].

This paper proposes a machine learning-based classification process for rice seeds to enhance agricultural seed quality management. Using image analysis, features such as colour intensity, contour, and surface texture are extracted. Classification algorithms, including random forest and k-nearest neighbours, are evaluated for their accuracy in identifying rice seed types. The system reduces the labour required in manual sorting and provides consistent and repeatable outcomes. This research supports automation in seed processing plants, with a focus on scalability and applicability in regions with high rice production [19].

This paper presents an automatic grading system designed to evaluate handwritten mathematics homework by verifying each derivation step for correctness. The system combines Mathematical Expression Recognition (MER) and a Computer Algebra System (CAS) to convert handwritten expressions into symbolic format and check for equivalence between steps using expression trees. To address misrecognition issues caused by OCR inaccuracies, a symbol replacement module uses confusion matrix data and surrounding context to correct errors and improve accuracy. The system architecture includes modules for symbol recognition using CNNs, structural analysis for syntax parsing, and expression matching through hashing and tree comparison. Evaluation on a synthetic dataset of 6,000 derivations showed a significant increase in F1-score (from 69.41% to 95.95% for addition/subtraction and from 61.45% to 89.95% for multiplication tasks) due to the symbol replacement technique. This work demonstrates promising potential for automating formative assessment in math education while addressing challenges in handwritten content analysis [20].

2.2 Comparative analysis of related work

The table 2.1 shows the comparative analysis of the related works.

Table 2.1 Comparative analysis of related work.

Sl. No.	Focus Area	Key Technologies	Advantages	Limitations
1	Feature extraction with CNN	CNN	High classification accuracy (96.2%)	No mention of real-time grading speed or hardware deployment
2	Robusta coffee bean grading	ResNet-34, Inception v3, EfficientNet-B0, VGG-16	High accuracy using lightweight models	May perform poorly in uncontrolled environments
3	Areca nut grading	CNN, OpenCV	High accuracy (96.75%), Real-time capable	Not tested under varying lighting/motion
4	Lemon grading	Computer Vision, ANN	Good performance for small datasets	Limited by small training dataset

5	Plant monitoring	SVM, K-means	Cost-effective, IoT-integrated	Accuracy affected by lighting conditions
6	Fruit grading	Raspberry Pi, OpenCV	Real-time, low-cost, suitable for farms	Limited scalability, lighting sensitive
7	Real-time fruit grading	OpenCV, Raspberry Pi	Easy to implement, portable	Lighting-dependent, fruit-specific
8	Dried fig grading	MATLAB, LabVIEW	High precision, non-destructive	Sensitive to lighting, ambiguous splits
9	Tomato grading	Fuzzy Logic, Image Processing	Reduces labour, good multi-parameter analysis	Limited to tomatoes, controlled setup needed
10	Fruit grading & sorting	IoT, CNN, SVM, YOLO	Real-time, modern agriculture support	Needs high-quality hardware, data
11	Milk quality grading	IoT, ANN, K-Means, sensors	Real-time, transparent pricing	Sensor-heavy, data-dependent
12	CNN in vision tasks (review)	VGG, ResNet, MobileNet, SENet	End-to-end learning, strong visual task performance	High computational cost, overfitting risk
13	Soil classification	CNN, Image Processing	Adaptable to lighting changes, automation	Needs large datasets, image quality issues
14	Image classification (digits)	CNN, MNIST	High accuracy, end-to-end learning	Limited scope (grayscale digits)

15	Areca nut grading	ML, Computer Vision	Fast, consistent, labor-saving	Image quality-dependent, domain-specific
16	Tomato quality evaluation	Computer Vision	96.47% accuracy, consistent	Lighting/image quality affects performance
17	Fruit grading methods (review)	Imaging Spectroscopy, ML	Non-destructive, adaptable	Hardware complexity, lighting sensitivity
18	Fruit grading techniques	RGB/Multispectral imaging, SVM, ANN	Diverse techniques compared	Real-time deployment limited
19	Paddy seed classification	SVM, Image Features	High accuracy (94.28%) with low-cost setup	Limited to few varieties, lighting sensitive
20	Edge device image classification	Early-exit NN, NAS, Distillation	Fast inference, low power	Complex tuning, hardware-model tight coupling

2.3 Gaps in Existing Solutions

Although numerous automated grading systems have been developed for agricultural produce using machine vision and image processing techniques, the specific domain of arecanut plate grading remains significantly underexplored. Existing systems predominantly target whole Arecanuts, emphasizing parameters such as size, shape, and colour. In contrast, the grading of processed arecanut plates widely utilized in the production of eco-friendly disposable products introduces distinct challenges, including:

- Inconsistent lighting conditions and background noise, which adversely impact classification accuracy.
- Absence of standardized grading parameters tailored to the unique properties of processed arecanut plates.
- Limited adoption of real-time grading mechanisms that leverage edge computing or IoT-based frameworks for efficient deployment.

- Dependence on traditional thresholding and classical machine learning algorithms, which often struggle to address surface defects, texture irregularities, and edge distortions.
- Lack of comprehensive and annotated datasets, specifically curated for arecanut plate grading, which hinders the effective training and evaluation of deep learning models.

These limitations highlight the need for advanced, domain-specific solutions that integrate robust data collection, deep learning architectures, and real-time edge deployment to enable accurate and scalable grading of arecanut plates.

2.4 Summary

The surveyed literature highlights the rapid convergence of machine vision, deep learning, IoT, and embedded systems in addressing agricultural and industrial classification challenges. Numerous studies have explored automated grading and sorting of agricultural products such as fruits, arecanut, and plant leaves using image processing and neural networks, with objectives centred on enhancing consistency, minimizing manual intervention, and improving real-time efficiency in quality control. While many existing systems utilize classical image processing techniques in conjunction with microcontrollers or Raspberry Pi platforms, deep learning models, particularly convolutional neural networks (CNNs), have demonstrated superior performance in handling complex, noisy, and non-linear data. Additionally, IoT-based solutions for monitoring plant health and environmental parameters are increasingly adopted for their scalability and integration capabilities. However, recurring gaps across these works include the absence of standardized datasets, limited focus on real-time performance optimization, and challenges in scaling solutions to industrial-grade applications. Moreover, most approaches are crop-specific and lack generalizability. In this context, the proposed Arecanut Plate Grading System aims to fill a critical niche by delivering robust, real-time classification tailored to the unique surface textures and structural variations of arecanut plates an area that remains largely unexplored in current research.

Chapter 3

Problem Formulation

Problem formulation is the process of clearly defining a problem by identifying its objectives, constraints, inputs, and expected outputs. It provides a focused and structured understanding of what needs to be solved, ensuring that the research or project stays aligned with the actual requirement and leads to meaningful, measurable outcomes.

3.1 Problem Statement

The grading of areca leaf plates plays a crucial role in ensuring product quality, consumer satisfaction, and sustainable production. At present, the process is predominantly manual, where workers visually inspect plates based on texture, shape, and colour variations. This method is highly subjective, labour-intensive, and inconsistent, leading to errors in classification.

The challenge is to design and implement an automated grading system that:

- Captures high-quality images of areca leaf plates in real time.
- Classifies them accurately into categories (Grade A, Grade B, Reject).
- Operates efficiently on affordable embedded hardware such as Raspberry Pi, ensuring cost-effectiveness and scalability.

Hence, the problem can be formulated as:

"How can an AI-driven, vision-based system be developed to automate areca leaf plate grading with high accuracy, consistency, and real-time performance under practical industrial conditions?"

3.2 Objectives of the Project

The specific objectives of the project are:

i. **Develop an automated grading system**: Use Convolutional Neural Networks (CNNs) to classify Areca nut plates based on quality factors like shape, color, texture and defects.

ii. **Enhance grading precision**: Ensure consistent quality control, reduce human error and support reliable sorting into categories such as Grade A, Grade B, or reject.

iii. **Automate the inspection process**: Reduce manual labor, improve efficiency, cut costs, and remove subjectivity in quality checks.

iv. **Enable scalable and sustainable production**: Streamline grading to speed up production and meet the growing demand for high-quality biodegradable plates in an eco-friendly market

3.3 Summary

This chapter presented the problem formulation of the project, outlining the limitations of existing manual grading methods and emphasizing the need for automation using AI and computer vision. The problem statement highlighted the challenges in ensuring accuracy, speed, and consistency, while the objectives clearly define the roadmap for developing a CNN-based automated grading system. These objectives form the foundation for the subsequent chapters, which detail the implementation, experimental results, and performance evaluation of the proposed solution.

Chapter 4

Requirements and Methodology

Software and hardware requirements define the essential specifications that must be fulfilled to install, operate, and maintain a software system effectively. These requirements ensure optimal compatibility, performance, and reliability by aligning the software's needs with the capabilities of the underlying hardware and operating environment.

4.1 Software Requirements

The software components required for the AI-Based Smart Areca Leaf Plate Grading system are listed in Table 4.1

Table 4.1 Software requirements

Sl. No.	Category	Description
1	Operating System	Raspberry Pi OS (Linux-based)
2	Programming Environment	Python is widely used for implementing machine learning models, image processing algorithms, and hardware interfacing.
3	Libraries and Frameworks	**OpenCV:** For image acquisition, preprocessing, and defect detection. **TensorFlow / Karas:** For building and deploying CNN models like MobileNet. **NumPy:** For matrix operations and image data handling. **Flask:** For creating APIs or lightweight dashboards (if needed)
4	Development tools	**Visual Studio Code:** for on-device development and debugging. **Google Colab:** for cloud-based training and testing of ML models with GPU support.

4.2 Hardware Requirements

The hardware setup is critical to ensure seamless functionality of the AI-Based Smart Areca Leaf Plate Grading system are referred in Table 4.2.

Table 4.2 Hardware Requirements

SL. No.	Component	Description
1	Raspberry Pi 5	Acts as the processing unit for image analysis and control tasks, supporting GPIO interfacing with motors and sensors.
2	Camera Module	High-resolution camera (e.g., 1080p) captures detailed images of areca plates for defect detection and grading.
3	Conveyor belt System	A motor-driven conveyor belt transports areca plates under the camera for consistent and automated image capture. Speed can be adjusted based on processing time.
4	Lighting System	LED lights provide uniform illumination to ensure high-quality image capture, reducing the effects of shadows or ambient lighting.
5	Cooling System	Heat sinks and fans attached to the Raspberry Pi prevent thermal throttling during continuous grading operations.
6	Power Supply	A reliable 5V/3A power adapter supports the Raspberry Pi, motors, and other peripherals.
7	Display Unit	Optional monitor or LCD for displaying real-time grading output and system status.
8	Sorting Mechanism	Gear or stepper motors can be used to sort plates into categories based on grading results, enhancing automation.
9	Motor Driver	L298N is a dual H-bridge motor driver module used to control the speed and direction of DC motors and stepper motors.

4.3 Methodology

The methodology of the proposed Smart Areca Leaf Plate Grading System follows a structured pipeline that integrates computer vision, deep learning, and embedded systems to achieve automated, real-time grading and sorting of areca leaf plates. The complete methodology is divided into sequential stages as described below.

i. **Data Collection and Image Acquisition:** High-resolution images of areca leaf plates are captured using a USB camera or Raspberry Pi camera module mounted above a motor-driven conveyor belt. Plates are placed individually on the conveyor to ensure proper alignment and minimal overlap. Uniform LED lighting is used to reduce shadows and illumination variations, enabling consistent image capture under controlled indoor conditions.

ii. **Image Preprocessing:** The captured images are preprocessed using OpenCV to enhance visual quality and standardize inputs for the deep learning model. Preprocessing steps include resizing images to 224 × 224 pixels, median filtering to remove noise, background normalization, and pixel value scaling using ImageNet normalization. These operations reduce the effect of lighting variations and highlight critical surface features such as texture, cracks, and discoloration.

iii. **Dataset Preparation and Augmentation:** The preprocessed dataset is divided into training and validation sets using an 80:20 split. To improve model robustness and prevent overfitting, data augmentation techniques such as rotation, flipping, zooming, and shifting are applied. This allows the model to generalize well across variations in plate orientation and surface patterns.

iv. **CNN Model Training and Fine-Tuning:** A ResNet50 convolutional neural network pre-trained on ImageNet is employed using a transfer learning approach. Initially, the base layers are frozen and used as a feature extractor, while a custom classification head is added to classify plates into Grade A, Grade B, and Reject categories.

v. Subsequently, selected deeper layers are unfrozen and fine-tuned with a low learning rate to learn domain-specific features related to areca leaf plate texture and defects. The model is trained using the Adam optimizer and categorical cross-entropy loss until optimal accuracy is achieved.

vi. **Edge Deployment on Raspberry Pi:** The trained CNN model is saved in .h5 format and deployed directly on a Raspberry Pi 5, enabling on-device inference without cloud dependency. The Raspberry Pi handles real-time image capture, preprocessing, and classification, ensuring low latency and continuous operation suitable for small-scale industrial environments.

vii. **Real-Time Classification:** As each plate reaches the inspection point, the conveyor momentarily stabilizes and the camera captures an image. The image is preprocessed and passed to the CNN model, which predicts the quality category.

Inference is completed within 300–400 ms per plate, enabling near real-time grading.

viii. **Automated Sorting Mechanism:** Based on the predicted class label, the Raspberry Pi triggers appropriate GPIO signals to control gear motors via an L298N motor driver. Grade A plates are diverted to Bin A, Grade B plates are diverted to Bin B, Rejected plates continue to the reject tray. This synchronized mechanical sorting ensures accurate and consistent segregation with minimal human intervention.

ix. **Monitoring, Logging, and Validation:** Classification results, timestamps, and system status are logged locally for performance analysis. The system's output is validated by comparing automated grading results with manual expert grading. Metrics such as accuracy, sorting reliability, and real-time performance are evaluated to verify system effectiveness and robustness.

4.4 Summary

This chapter presents the software and hardware requirements along with the methodology of the AI-Based Smart Areca Leaf Plate Grading System. The system integrates computer vision, deep learning, and embedded hardware to enable automated, real-time grading and sorting of areca leaf plates. Image acquisition and preprocessing ensure consistent and high-quality inputs for CNN-based classification. A transfer learning approach using a pre-trained deep learning model improves grading accuracy and robustness. The trained model is deployed on a Raspberry Pi for edge-based inference with low latency. Automated motor-driven sorting and continuous monitoring ensure efficient, reliable, and scalable operation with minimal human intervention.

Chapter 5

System Design

System design outlines the overall architecture and workflow of the project, detailing how various hardware and software components interact. It defines data flow, module functionalities, and technology choices to ensure efficient performance. This section serves as a blueprint for implementing and integrating all parts of the system cohesively.

5.1 Architecture diagram of Proposed System

The figure 5.1 shows the architectural design of the Smart Areca Plate Grading System that integrates a Raspberry Pi 5 with a camera module and conveyor belt to automate plate inspection. Captured images are processed using a CNN model for quality classification. Based on the results, a gear motor sorts the plates, and a display shows the classification report.

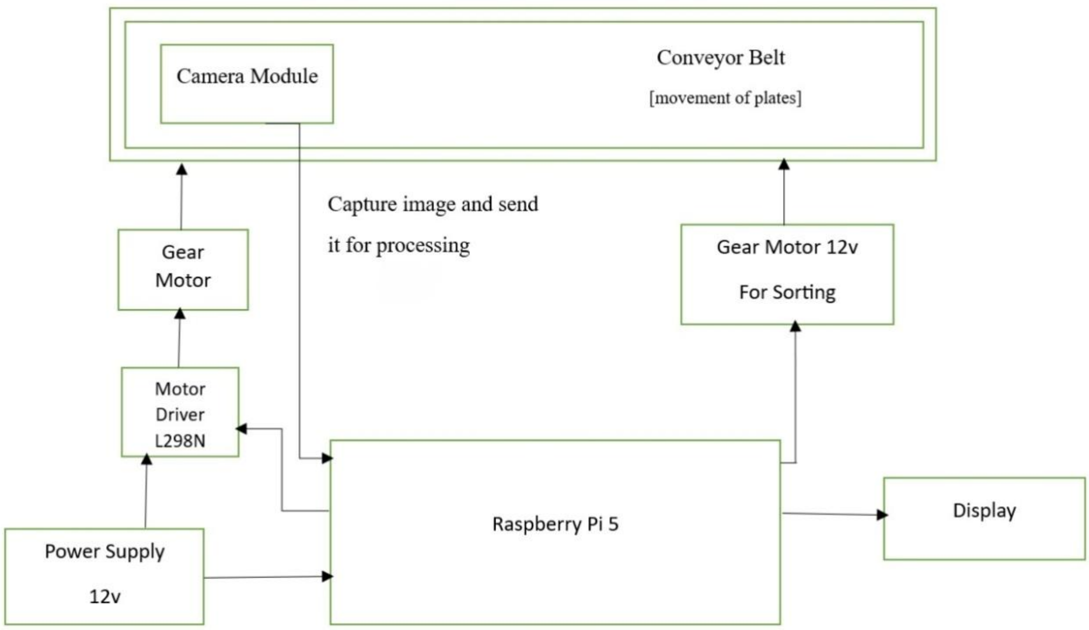


Figure 5.1 Architectural diagram of proposed system

i. **Raspberry Pi 5:** The Raspberry Pi 5 acts as the central processing unit of the system, coordinating all major functions involved in plate classification. It receives images captured by the camera module and performs preprocessing operations such as resizing and normalization to prepare the data for analysis. Once pre-processed, the images are passed through a Convolutional Neural Network (CNN) model running on the device to determine the grade of each plate. After classification, the Raspberry Pi sends the results simultaneously to the display unit for visualization and to the gear motor for appropriate sorting actions.

ii. **Camera Module: The** camera module is positioned above or beside the conveyor belt to ensure a clear and consistent view of the passing plates. As each plate moves along the conveyor, the module captures snapshots at the correct moment and transmits these images to the Raspberry Pi 5. Its operation is crucial for obtaining accurate visual data required for image processing and classification.

iii. **Conveyor Belt:** The conveyor belt is responsible for mechanically transporting the plates through various stages of the system. It moves the plates steadily into the camera's field of view, ensuring proper alignment for image capture. The belt is driven by a DC motor, enabling smooth and continuous plate movement required for automated inspection.

iv. **DC Motor:** The DC motor powers the movement of the conveyor belt, enabling plates to pass through the system in a controlled manner. It receives power from the system's power supply and may also be controlled by the Raspberry Pi to perform timed or trigger-based movements. This synchronization helps ensure that image capture and classification occur at the right moment.

v. **Gear Motor:** The gear motor operates based on the classification output generated by the Raspberry Pi. Its primary function is to divert plates into their appropriate bins according to their assigned grades. When the CNN classifies a plate as Grade A or Grade B, the Raspberry Pi activates the gear motor to push the plate into the corresponding bin. If the plate is rejected, no motor action is taken, and the plate continues to the reject tray.

vi. **Display Unit:** The display unit provides a clear and immediate visual representation of the system's output. It shows the classification result for each plate along with status messages that help the operator monitor system performance. The display receives real-time data directly from the Raspberry Pi, ensuring accurate and up-to-date information.

vii. **Power Supply:** The power supply serves as the main energy source for the system, delivering stable electrical power to all components including the Raspberry Pi 5, DC motor, and gear motor. Its reliability is essential for maintaining consistent operation, preventing interruptions, and ensuring that all mechanical and electronic parts function properly throughout the classification process.

5.2 Interface Diagram and Description

The figure 5.2 illustrates the interfacing between the Raspberry Pi (e.g., Raspberry Pi 4 Model B), a camera module, and the system's output peripherals. The Raspberry Pi serves as the core single-board computer (SBC), responsible for running the full operating system, executing the AI model, and managing all peripherals. A camera (such as a Raspberry Pi Camera Module connected via the CSI interface or a USB webcam connected to a USB port) is used for image acquisition.

Power is supplied via a dedicated USB-C power adapter to meet the Pi's processing demands. The Raspberry Pi's GPIO pins are used to interface with motor drivers, gears, or actuators that control the physical grading and sorting mechanism. For development, programming, and debugging, the Pi is typically accessed remotely via SSH or VNC (over WiFi or Ethernet) or by directly connecting a monitor (via HDMI) and keyboard/mouse (via USB).

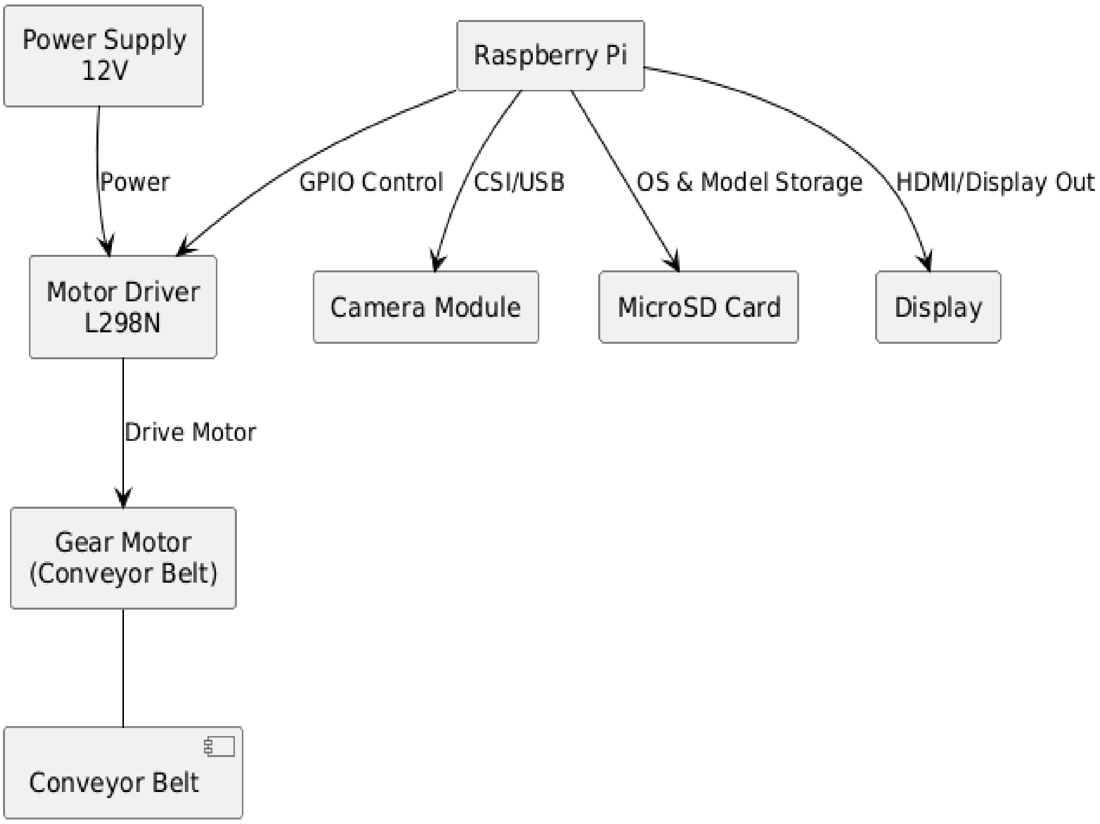


Figure 5.2 Raspberry Pi System Interface Diagram

i. Interfaces: The Raspberry Pi connects to multiple components through dedicated interfaces to enable smooth system operation. The camera module is interfaced either through the high-speed CSI port for the Raspberry Pi Camera Module or through a USB port when using a standard webcam. For controlling the mechanical sorting system, the

Raspberry Pi communicates with output peripherals such as motor drivers and gear mechanisms using GPIO pins that deliver PWM or digital control signals, with all components sharing a common ground for stable operation. Power is supplied to the Raspberry Pi through a USB-C adapter, ensuring reliable energy delivery to the SBC and its connected modules. For development, debugging, and network connectivity, the Raspberry Pi supports WiFi or Ethernet access to routers for SSH or remote management, while HDMI ports allow monitor connection and USB ports support keyboard and mouse inputs for direct programming.

ii. Processes: The system begins with the Raspberry Pi booting from the MicroSD card, loading the operating system and automatically initializing the grading application. Development of the software can be performed directly on the Pi or remotely via SSH or VS Code Remote tools, enabling efficient code updates. During operation, the Python script activates the connected camera—using either Picamera2 or OpenCV—to capture images of incoming arecanut plates. These images are then preprocessed and passed into the trained CNN model, which predicts the plate's grade such as "Grade A" or "Reject." Based on the model's inference, the Raspberry Pi sends precise signals through its GPIO pins to the motor driver or gear mechanisms to route the plate to the correct sorting bin, after which the system waits for the next plate to arrive.

5.3 Module Description

i. Image Capture Module: The Image Capture Module is responsible for acquiring high-resolution images of Areca nut plates as they pass through the conveyor or designated inspection area. It employs a camera—typically a USB web camera—positioned to ensure optimal viewing and consistent lighting, which is essential for producing uniform and clear images. The module supports both automated and manual triggering mechanisms: images can be captured at fixed intervals, through sensors that detect plate movement, or through user-initiated commands. This flexibility ensures that the system can adapt to varying operational requirements while maintaining high-quality visual data for further processing.

ii. Preprocessing Module: The Preprocessing Module prepares the raw images for accurate classification by enhancing their quality and making them suitable for input to the deep learning model. Using OpenCV, this module performs essential image processing tasks such as filtering, resizing, segmentation, and contrast adjustment to remove background noise and highlight important features of the plate. These steps ensure that variations in lighting, background, or orientation do not affect the system's performance. Once processed, the images are converted into a standardized format compatible with

Convolutional Neural Network (CNN) input, enabling reliable and consistent analysis during the classification stage.

iii. Classification Module: The Classification Module serves as the core intelligence of the system, using a trained Convolutional Neural Network (CNN) to categorize Areca nut plates into predefined quality grades. Models such as MobileNet or custom CNN architectures implemented using TensorFlow or PyTorch are used to analyse key attributes like surface texture, colour uniformity, defects, shape irregularities, and edge quality. Based on these visual characteristics, the module assigns grade labels such as Grade A, Grade B, or Defective. By automating this decision-making process, the module ensures consistent, fast, and highly accurate grading without human intervention.

iv. Edge Development Module: The Edge Deployment Module integrates the entire classification pipeline onto a compact embedded device such as a Raspberry Pi or Jetson Nano. This module enables real-time grading with minimal latency by performing all computations locally rather than relying on remote servers or cloud-based processing. Its offline capability ensures uninterrupted operation even in remote or low-connectivity environments. Additionally, the module interfaces directly with local displays, indicators, or actuators to provide instant feedback and trigger mechanical actions based on the classification output, making the system efficient and self-contained.

v. User Interface Module: The User Interface Module provides operators with a clear, interactive display of real-time grading results, system status, and performance metrics. It may feature a local LCD display or a web-based dashboard accessible through the network. The interface shows the live camera feed, classification outputs, and operational logs, offering full visibility into ongoing processes. It also allows users to perform manual overrides, adjust configuration settings, or initiate system commands, giving operators convenient control and ensuring smooth monitoring and management of the overall system.

vi. Sorting Module: The Sorting Module automates the physical separation of Areca nut plates into their respective quality categories based on the decisions made by the Classification Module. Using gear motors or other suitable actuators, this module redirects each plate into its appropriate bin—such as Grade A, Grade B, or Defective—as it moves along the conveyor. The sorting operation is synchronized with both the conveyor and the classification results, ensuring timely and accurate segregation. This automated mechanism minimizes human involvement and ensures a smooth and efficient sorting process with high precision.

5.4 Data Flow Diagram of proposed system

A Data Flow Diagram (DFD) visually represents the flow of data within a system, showing how inputs are processed into outputs through various modules. It helps in understanding the system's functionality, data interactions, and process dependencies in a structured manner. The below figure 5.3 describes the dataflow of proposed system.

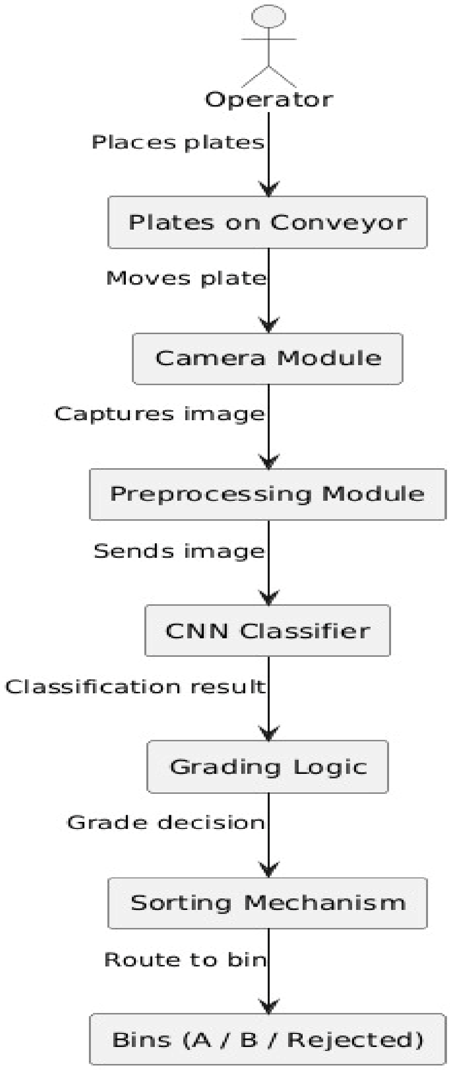


Figure 5.3 Data flow diagram of proposed system

i. **Operator Input:** The process begins with the operator placing areca plates onto the conveyor system.
ii. **Plate Movement:** The conveyor belt moves the plates through the system, initiating the grading workflow.
iii. **Image Acquisition:** As plates move, the camera module captures real-time images of each plate for analysis.

iv. Image Preprocessing: Captured images are sent to the Preprocessing Module, where operations like resizing, filtering, and normalization are performed to prepare the image for classification.

v. CNN-Based Classification: The processed image is passed to the CNN Classification module, which uses a trained convolutional neural network to classify plates into categories (e.g., Grade A, Grade B, or Rejected).

vi. Decision Logic: Based on the classification result, the Grading Decision Logic determines the appropriate action or bin category for each plate.

vii. Sorting Mechanism: The system triggers a sorting mechanism (likely a gear motor) to physically redirect the plate toward the appropriate bin.

viii. Final Output: Plates are sorted into designated bins labelled A, B, or Rejected, completing the automated grading process.

5.5 Sequence Diagram

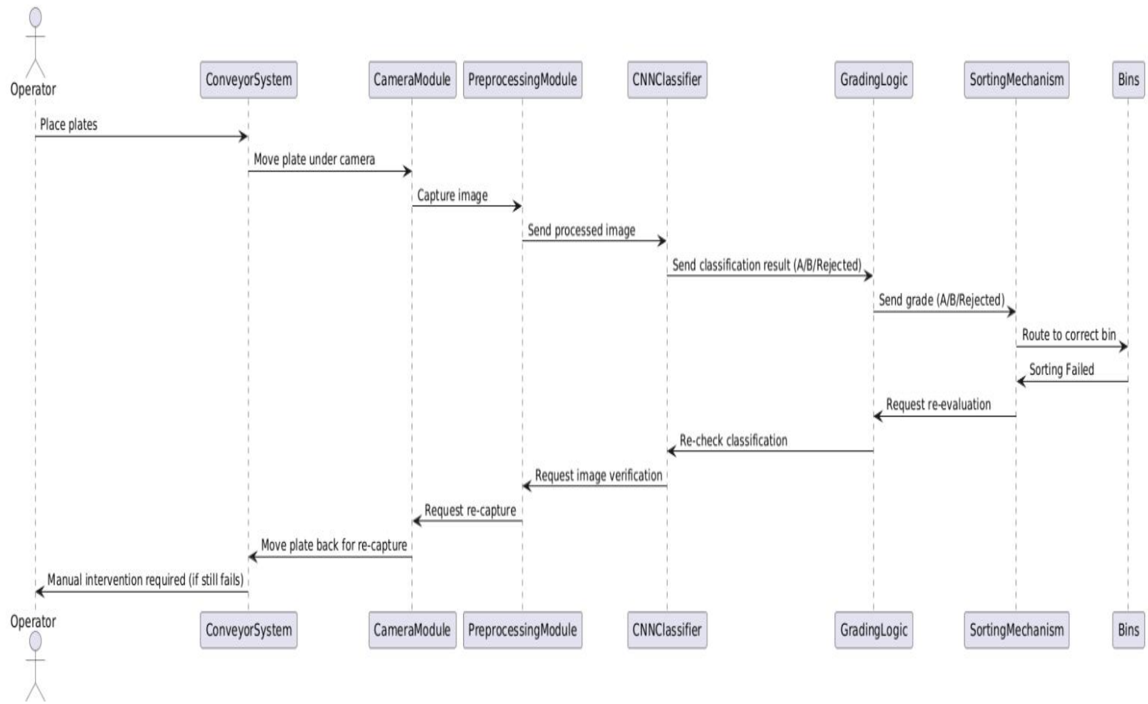


Figure 5.4 Sequence Diagram of proposed system

The above figure 5.5 illustrates the sequence diagram of the Areca Leaf Plate Grading System, detailing the step-by-step interaction among various components during the grading process. The process begins when the operator places areca leaf plates onto the conveyor system, which then moves the plates under the camera module. The camera captures an image of each plate, which is sent to the preprocessing module where it is resized and filtered for clarity. The pre-processed image is then forwarded to the CNN classifier, which analyses it and predicts the quality grade as A, B, or Rejected. This classification result is sent to the grading logic module, which determines the appropriate sorting action. Subsequently, the sorting mechanism receives the grading instruction and directs the plate into the correct bin based on its quality. Figure 4.4 effectively represents the data flow and interaction between human and machine actors to achieve an automated and accurate plate grading process.

5.6 Hardware Description

The below figure 5.5 shows the hardware connection diagram of proposed system.

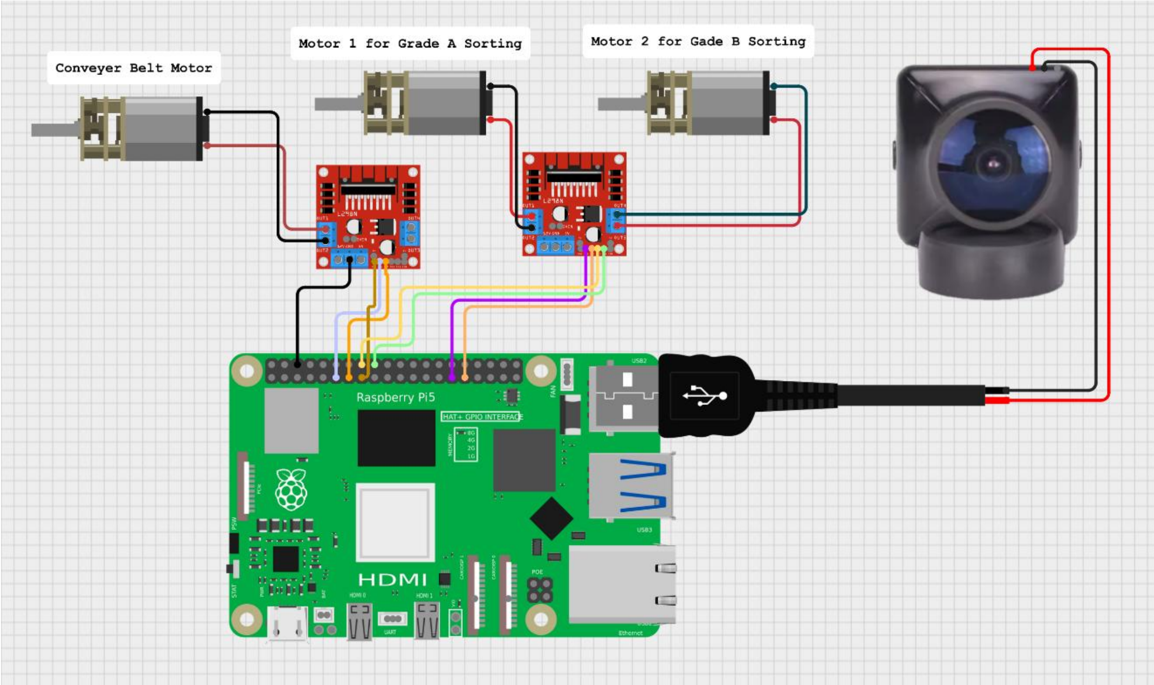


Figure 5.5 Hardware connection of proposed system

i. **Sensing Unit (Inputs):** The sensing unit consists of a Pi Camera Module (V2/V3) connected through the mini-CSI interface, which serves as the primary input device for the system. It continuously captures high-resolution images of the areca plates as they move along the conveyor for inspection. The captured visual data is transmitted to the Raspberry Pi through the CSI interface in the form of high-speed camera data. This sensing mechanism ensures accurate image acquisition, enabling reliable preprocessing, feature extraction, and grading by the AI model.

ii. **Control & Processing Unit (Software Logic):** The control and processing unit serves as the computational core of the system, where all image analysis and decision-making operations take place. Once an image is captured, it is processed using OpenCV to perform denoising, resizing, and other essential pre-processing tasks required for accurate inference. The refined image is then passed into the deep learning model, which evaluates the visual features and classifies the plate based on the trained grading criteria. The control logic interprets the model's predicted class label and prepares the appropriate actuation response. Using the gpiozero library, these control signals are translated into GPIO outputs, enabling precise manipulation of motors, servos, or gates to physically sort the plates in real time.

iii. **Power Supply Unit:** The Power Supply Unit provides stable and isolated power to all major components in the system. A 12V DC adapter powers the conveyor motor and other high-current actuators that require higher voltage and torque. In parallel, a 5V USB-C PD supply (minimum 27W recommended) powers the Raspberry Pi and its associated control circuits, ensuring consistent performance during image processing and deep-learning inference. This dual-power setup ensures that the computational components and mechanical components operate reliably without voltage drops or interference.

iv. **Actuation Unit (Outputs):** The Actuation Unit is responsible for performing all mechanical operations based on commands received from the control logic. The L298N Motor Driver acts as an interface between the Raspberry Pi and the DC conveyor motor, receiving PWM signals to regulate speed and direction while handling the required high current. In addition, a gear motor is used to control the sorting or reject gate mechanism. This motor is activated using GPIO signals and PWM control to precisely remove defective plates from the conveyor. Together, these actuators enable smooth, automated sorting and continuous operation of the grading system.

v. **Monitoring Unit (Interface):** The Monitoring Unit provides real-time visibility and interaction for the operator. A web-based dashboard is used to display system status, plate classification results, motor operation state, and other diagnostics in real time. Through this interface, the operator can monitor overall system performance, initiate manual controls if required, and view logs or alerts generated by the processing unit. This interactive monitoring layer ensures ease of use, improves operational efficiency, and enhances the reliability of the automated grading system.

5.7 Conveyor Details

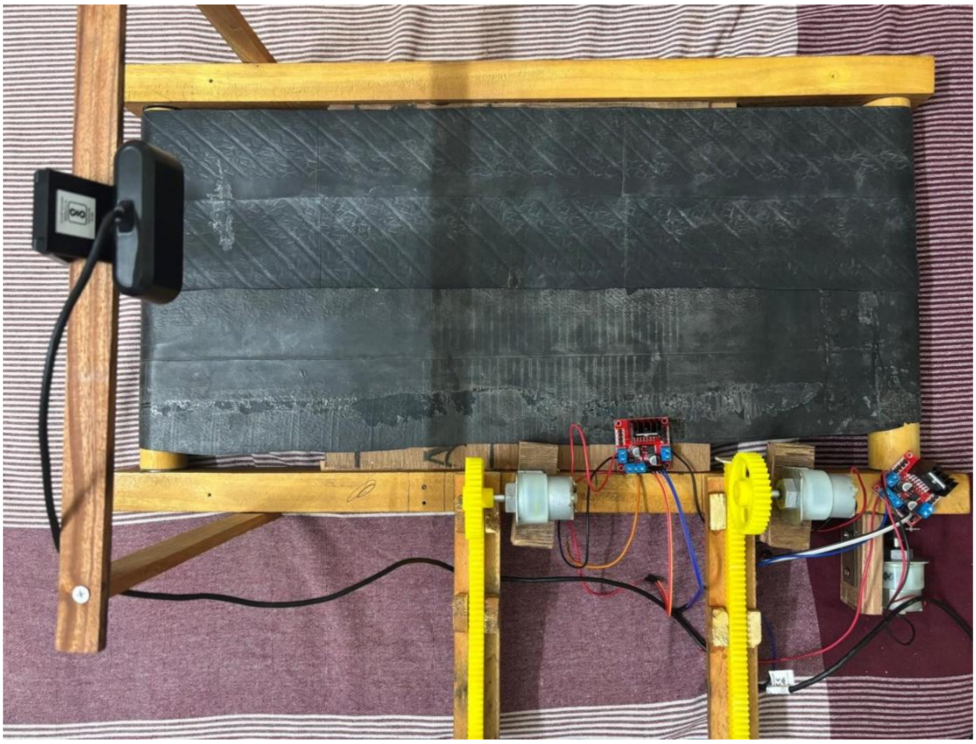


Figure 5.6 Conveyer System

Figure 5.8 shows that the conveyor system is responsible for transporting areca leaf plates in a controlled and linear motion beneath the camera module for image capture. It ensures consistent positioning and spacing of plates, enabling accurate classification by the AI model. The conveyor belt is driven by a DC gear motor through a roller mechanism, providing stable and uniform movement. The conveyor system measures 2 ft × 1 ft, with a rubber tire-tube belt. The camera is positioned 0.5 ft above the belt for effective image acquisition.

5.8 Motor Details

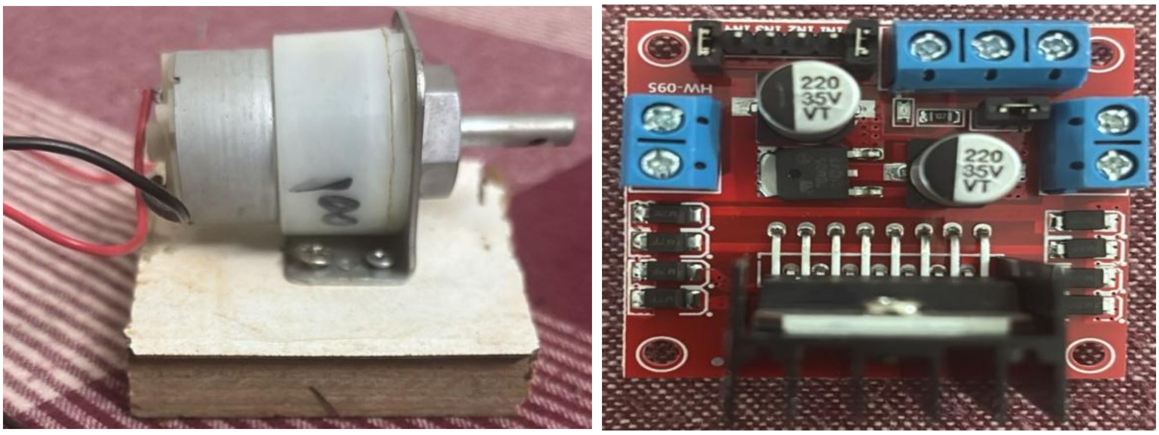


Figure 5.7 Motor and L298N Driver

Figure 5.9 shows the 12V DC gear motor and L298N motor driver used to drive the conveyor belt. The gear motor enables smooth and controlled plate movement, while the L298N motor driver interfaces with the Raspberry Pi GPIO pins to receive PWM and directional control signals. The gear motor provides high torque at low RPM, ensuring steady belt motion and preventing slippage under load. The motor driver facilitates safe power handling, speed regulation, and bidirectional control. This motor–driver combination ensures reliable mechanical actuation suitable for continuous and real-time grading operations.

5.9 Microcontroller Details

Figure 5.8 Microcontroller (Raspberry Pi 5)

Figure 5.10 shows the microcontroller (Raspberry Pi 5) used as the central processing and control unit of the system. It runs a Linux-based operating system and executes the trained CNN model for image classification using Python, TensorFlow, and OpenCV. Through its GPIO pins, the Raspberry Pi controls the motor driver, gear motors, and sorting mechanism based on the classification output. It interfaces with the camera module via CSI or USB for real-time image acquisition. Additionally, built-in Wi-Fi, USB ports, and HDMI support enable easy program deployment, remote monitoring, and future system expansion, making it a compact and powerful controller for the complete grading setup.

Chapter 6

Implementation

The system is implemented using a Raspberry Pi 5, a camera module, and a fine-tuned ResNet50 model for real-time plate grading. Captured images are pre-processed and classified directly on the device without cloud dependence. Based on the prediction, gear motors sort the plates into their respective categories. The integrated setup ensures fast, consistent, and automated grading suitable for practical deployment.

6.1 Pseudocode

```
BEGIN
  INITIALIZE Raspberry Pi 5 system
  INITIALIZE Raspberry Pi Camera Module for capturing plate images
  INITIALIZE GPIO pins for:
    - gear motors
    - Conveyor belt control
  LOAD the fine-tuned ResNet50 model (.h5 format) using TensorFlow/Keras
  WHILE system is active DO
    CAPTURE image of the areca leaf plate using the Raspberry Pi Camera
    PREPROCESS the captured image:
      - Resize image to 224 × 224 pixels
      - Remove noise using median filtering
      - Normalize pixel values using ImageNet mean and standard deviation
      - Expand image dimensions (1, 224, 224, 3) for model input
    PERFORM prediction using the loaded Keras .h5 model
    OBTAIN predicted class:
      - Grade A
      - Grade B
      - Reject
    IF predicted class = "Grade A" THEN
      ACTIVATE Gear Motor 1 → direct the plate to Bin A
    ELSE IF predicted class = "Grade B" THEN
      ACTIVATE Gear Motor 2 → direct the plate to Bin B
    ELSE
      NO gear activation → plate moves to the reject end of conveyor
```

END IF

LOG the following information:

- Captured image
- Predicted class label
- Timestamp

SAVE log entry to Raspberry Pi local storage or a simple database file

MOVE conveyor belt to bring the next plate under the camera

END WHILE

END

6.2 System Features

The system performs **on-device inference** by running the original Keras `.h5` model directly on the Raspberry Pi 5, eliminating the need for model conversion while enabling fully local and efficient predictions. A **USB-C webcam** is integrated with the Raspberry Pi to capture high-quality images of the areca leaf plates as they move along the conveyor. To ensure accurate image capture, the Raspberry Pi precisely **synchronizes the conveyor movement** using GPIO-controlled motors, positioning each plate perfectly before triggering the camera. After the image is processed, the system performs **automated sorting** based on the model's prediction—Grade A, Grade B, or Reject. For Grade A and Grade B plates, the Raspberry Pi activates dedicated gear motors to divert the plates into their respective bins. In the case of a rejection, no motor is triggered, allowing the plate to naturally pass through to the reject tray at the end of the conveyor. This end-to-end automation ensures efficient grading, minimal manual intervention, and consistent product quality.

6.3 Algorithms and Techniques

Algorithm 1: Image Classification (ResNet50 Fine-Tuned)

The image classification algorithm uses a two-stage transfer learning strategy to train an accurate model for grading Areca nut plates. In the training phase, the process begins by loading a pre-trained ResNet50 backbone with ImageNet weights and include_top=False, using it solely as a deep feature extractor. All layers of this backbone are frozen to preserve its general visual feature representations. Each training and validation image is preprocessed through a pipeline that converts it to uint8, applies median filtering with a kernel size of 3 to remove dust and salt-and-pepper noise, and normalizes it using ImageNet's preprocess_input scaling. Real-time data augmentation is performed using ImageDataGenerator with rotations up to 90°, width and height shifts, shear and zoom distortions, horizontal and vertical flips, and a 20% validation split to enhance model robustness. A custom classification head is added to the frozen base consisting of a Global Average Pooling layer, a Dense layer with 512 ReLU units, a 0.4 dropout layer, and a Softmax output layer with three units corresponding to the grading classes. The model is compiled using the Adam optimizer (learning rate 0.001), categorical crossentropy loss, and accuracy as the evaluation metric, and trained for an initial 75 epochs.

An accuracy-threshold-based early stopping mechanism ensures efficient learning: training continues uninterrupted until 90% accuracy is reached, after which validation loss governs early stopping, automatically restoring the best weights. This completes Phase 1, and the resulting model is saved as **resnet_custom_preprocessed.h5**. In Phase 2, fine-tuning is performed by unfreezing higher layers of the ResNet50 backbone above index 140, allowing the model to learn deeper, task-specific features relevant to leaf plate texture and defects. The model is recompiled with a very low learning rate of 1e-5 to avoid catastrophic forgetting and is trained for an additional 25 epochs using the same augmentation and callbacks.

Algorithm 2: Automated Sorting Mechanism Control (Raspberry Pi 5)

The automated sorting algorithm on the Raspberry Pi 5 is implemented using a decoupled dual-process architecture to optimize both hardware control and computational inference. A master shell script initiates and manages two concurrent Python processes: a "Motor Service" running with administrative privileges (sudo) to manage accurate GPIO timing, and a "Grading Logic" script running within a specialized virtual environment to handle the deep learning tasks. These two distinct processes synchronize their operations via a local socket connection (localhost), allowing the AI system to send precise actuation commands to the hardware controller with minimal latency.

The Raspberry Pi loads the fine-tuned .h5 Keras model directly into the grading logic. As the conveyor transports plates towards the inspection point, the camera captures top-view images in real-time. Each captured image undergoes specific preprocessing procedures—resizing to 224 X 224 pixels, applying median filtering for noise removal, performing ImageNet normalization, and expanding dimensions to match the model's input shape of (1, 224, 224, 3).

The processed image is passed to the ResNet50 model, which predicts one of three categories: Grade A, Grade B, or Reject. Based on this prediction, the grading logic transmits a specific command string (e.g., "SORT_A", "SORT_B") over the local network socket to the motor service. The motor service interprets this signal and instantly activates the appropriate actuator: Gear Motor 1 runs a forward-reverse sequence to direct Grade A plates into Bin A, while Gear Motor 2 activates similarly for Grade B plates. Plates categorized as Reject proceed forward without motor activation and fall into the reject tray. This socket-based design ensures that the heavy computational load of the neural network does not interfere with the stability of the motor control signals, enabling a robust and efficient sorting loop.

6.4 Data Structures Used

- **Image Arrays (NumPy)**: Captured frames from the USB camera are stored as NumPy arrays. These arrays are used during preprocessing operations such as resizing to 224×224, applying median filtering for noise removal, and performing ImageNet-based normalization before feeding the images into the classification model.
- **Keras Model Object:** The fine-tuned ResNet50 model is loaded on the Raspberry Pi 5 as a Keras model object. This structure stores the model architecture, weights, and inference graph, enabling direct prediction using model.predict() without requiring TensorFlow Lite conversion.
- **Prediction Mappings:** A simple dictionary structure is used to map numeric model outputs to human-readable class labels. Example:

  ```
  { 0: "Grade A",
    1: "Grade B",
    2: "Reject"}
  ```

  This mapping allows the Raspberry Pi to convert raw prediction indices into meaningful plate categories for sorting decisions.
- **Gear Control Table:** A gear control table is used to define a logical mapping between the predicted quality class and the corresponding gear motor action. When the model classifies a plate as **Grade A**, **Gear Motor 1** is activated to divert the plate to the appropriate bin. If the plate is classified as **Grade B**, **Gear Motor 2** is triggered for sorting. In the case of a **Reject**, no gear motor is activated, allowing the plate to move directly to the reject tray. This structured mapping ensures deterministic and reliable control of the sorting mechanism based on the model's classification output.

6.5 Integration Workflow

i. Training: The ResNet50 model is fine-tuned using the custom areca leaf plate dataset in a workstation or cloud environment such as Google Colab. The training pipeline includes preprocessing (median filtering, resizing, normalization), data augmentation, transfer learning, and final fine-tuning of selected layers to achieve high classification accuracy.

ii. Model Saving: After successful training and fine-tuning, the model is exported in .h5 format, which preserves both the architecture and the learned weights. This format is chosen because it can be loaded directly on the Raspberry Pi 5 without requiring TensorFlow Lite conversion.

iii. Deployment: The .h5 model is transferred to the Raspberry Pi 5, where it is loaded using TensorFlow/Keras. The Raspberry Pi interfaces with the USB camera for real-time image capture and controls the conveyor motor and gear motors through GPIO pins. This enables the entire inference and sorting process to be executed locally on the device.

iv. Testing: During system testing, areca plates are placed on the motorized conveyor belt. The camera captures images as each plate reaches the inspection point. The model processes the image, predicts the plate category (Grade A, Grade B, or Reject), and the appropriate gear mechanism directs the plate into the correct sorting bin.

v. Validation: Model predictions are compared against manually graded results to assess system performance. Metrics such as accuracy, precision, recall, and reliability are calculated to validate the effectiveness of the automated grading and sorting system in real-world conditions.

Chapter 7

Testing, Validation and Results

This section presents the systematic evaluation of the system's performance through functional testing, model validation, and real-time execution analysis. It highlights accuracy metrics, error rates, and observed outcomes under different conditions. The results demonstrate the effectiveness, reliability, and consistency of the proposed grading system.

7.1 System Testing and Validation

The Smart Areca Leaf Plate Grading System was tested through both software simulation (model training and validation on a workstation) and hardware deployment (Raspberry Pi 5 + camera module + conveyor + gear motors). The testing process aimed to evaluate the following key performance parameters:

- Classification Accuracy: To assess how accurately the fine-tuned ResNet50 model classifies areca plates into Grade A, Grade B, or Reject.
- Sorting Reliability: To verify that the correct bin is selected by the gear motor mechanism based on the CNN output.
- Real-Time Performance: To measure the processing speed and overall sorting throughput per plate during live operation.
- All tests were carried out under controlled indoor lighting to ensure consistent visual input for the camera module.

7.2 Test Cases and Results

Test Case 1: Image Classification Accuracy

This test case focuses on evaluating the accuracy of the CNN model in correctly classifying areca leaf plates from captured images. A dataset containing approximately *N* labeled plate images was used, with an 80–20 split between training and validation subsets. Before training, all images were resized to 224×224 pixels and normalized, then processed through a fine-tuned ResNet50 model built using TensorFlow/Keras. The final .h5 model was deployed on the Raspberry Pi 5 to perform on-device inference for real-time grading. The expected output for each image is a predicted class label—Grade A, Grade B, or Reject. The trained model achieved a training accuracy of approximately 95% and a validation accuracy of around 88%. Confusion matrix analysis revealed strong classification performance between Grade A and Reject categories, while a small degree of overlap between Grade A and Grade B was observed due to similarities in surface texture.

Test Case 2: Sorting Accuracy on Conveyor System

This test case evaluates the accuracy of the automated sorting mechanism in real-time operation. Plates were placed sequentially on the conveyor belt, which was controlled through Raspberry Pi GPIO pins to ensure synchronized movement. As each plate passed beneath the camera, the Raspberry Pi Camera Module captured an image that was then processed by the CNN model for classification. Based on the predicted label, the gear motors activated to divert each plate into the appropriate bin—Bin A for Grade A, Bin B for Grade B, and the Reject Bin for rejected plates. The expected behavior was that every plate would be sorted accurately according to its classification. During testing with 100 plates, the system achieved a sorting accuracy of approximately 92%. Observed sources of error included minor lighting variations affecting plate edge visibility, slight motion blur at higher conveyor speeds, and occasional gear misalignment during rapid, successive sorting actions.

7.3 Validation and Verification

Validation: The automated grading results produced by the Raspberry Pi system were cross-checked with manual expert grading. The system achieved an agreement rate of over 90%, validating the reliability of the deep learning approach for areca plate quality classification.

Verification: The deployed Raspberry Pi 5 system was verified for:

Correct execution of the .h5 model using TensorFlow/Keras. Stable gear motor control through GPIO without incorrect actuation. Continuous operation for several hours without overheating or performance degradation. Consistent real-time inference within 300–400 ms per plate, suitable for small-scale industrial automation.

7.4 Results

The fine-tuned ResNet50 CNN model achieved high classification accuracy of over 90%, demonstrating its effectiveness for multi-class areca leaf plate grading in real-time applications. Deployment on the Raspberry Pi 5 enabled efficient on-device inference with acceptable latency and reliable performance during continuous operation. The seamless integration of image-based classification with gear-driven sorting significantly reduced manual labour while improving grading speed and maintaining consistent quality standards. Compared to traditional manual inspection, the proposed system delivered substantial time savings and enhanced consistency, making it well suited for semi-automated areca leaf plate manufacturing environments. The below table 7.1 shows the test case and results.

Table 7.1: Test cases and results

Test case no.	Input	Expected grade	Classification Result	Result status P – Pass F – Fail
1		Class A	As expected	P
2		Class A	As expected	P
3		Class A	Not expected	F
4		Class B	As expected	P
5		Class B	As expected	P
6		Class B	Not expected	F

7		Reject	As expected	P
8		Reject	As expected	P
9		Reject	Not expected	F

7.5 Result Analysis

The proposed Raspberry Pi–based system achieves higher accuracy and speed compared to manual and existing vision-based grading methods. It reduces labour and overall cost while offering better scalability through a low-cost modular design. The system also maintains good energy efficiency, making it suitable for continuous operation. The below table 7.2 shows the Comparisons between Existing and Proposed System.

Table 7.2: Comparisons between Existing and Proposed System

Aspect	Manual Grading	Existing Vision-Based Systems	Proposed Raspberry pi-Based System
Accuracy	~70–80% (human error)	85–90% (Raspberry Pi with Computer Vision)	90–92% (ResNet50 fine-tuned)
Speed (plates/hour)	~80–100	~150	~180–200
Cost	High (labor-intensive)	Medium (requires Pi/PC hardware)	Low (ESP32-CAM, gear, belt)
Scalability	Low	Medium	High (low-cost, modular)
Energy Efficiency	N/A	Moderate	Moderate-High (Rapberry pi 5: ~3W-9W)

Chapter 8

Conclusion and Future Scope

This section summarizes the overall achievements of the proposed system and highlights how effectively it meets the project objectives. It reflects on the system's performance, limitations, and practical impact. The future scope outlines potential enhancements and advanced features that can further improve accuracy, scalability, and real-world deployment.

8.1 Conclusion

Overall, this work highlights how machine learning, computer vision, and IoT technologies are transforming the quality inspection of areca leaf plates by replacing slow and inconsistent manual checks with smarter automated solutions. By capturing, processing, and classifying images in real time, these systems ensure reliable grading even with natural variations in texture, shape, and defects. Although challenges such as limited datasets and difficulty in adapting to diverse leaf types persist, advancements in edge computing and hybrid models present promising directions for improvement. With continued development, automated grading systems can significantly support sustainable production and wider adoption of eco-friendly tableware in the manufacturing industry. The intended objectives of the project that have been successfully met are listed below:

- Developed an automated grading system using computer vision and Convolutional Neural Networks (CNNs) to classify areca leaf plates based on quality parameters such as shape, colour, texture, and surface defects.
- Achieved enhanced grading precision by ensuring consistent and reliable quality assessment while significantly reducing human error and subjectivity.
- Successfully automated the inspection and sorting process, thereby minimizing manual labour, improving operational efficiency, and reducing overall production costs.
- Enabled a scalable and sustainable grading solution capable of supporting increased production demands for high-quality, biodegradable areca leaf plates in an eco-friendly manufacturing environment.

8.2 Future Enhancements

The proposed system can be further improved and extended in the future to enhance accuracy, robustness, and scalability for large-scale industrial deployment. Some of the potential future enhancements are listed below:

- **Enhanced Environmental Robustness:** Develop an enclosed imaging chamber with controlled internal lighting to eliminate misclassifications caused by varying ambient light conditions.
- **Throughput Optimization:** Integrate higher-speed cameras and explore edge accelerators (like Google Coral TPUs) to faster process images, allowing the conveyor to run at higher speeds for large-scale production.
- **IoT-Enabled Remote Monitoring:** Add Internet of Things (IoT) capabilities to stream real-time production data—such as counts of Grade A, Grade B, and rejected plates—to a remote dashboard for factory managers to monitor from anywhere.

References

[1] Liu, Y. H. (2018). Feature Extraction and Image Recognition with Convolutional Neural Networks. Journal of Physics: Conference Series, 1087(6), 062032.

[2] Bipin Nair B. J., Abrav Nanda K. M., Shalwin A. S., Pranav V., & Venkatesh R. (2024). Coffee Bean Graded Based on Deep Net Models. International Journal of Electrical and Computer Engineering (IJECE), 14(3), 3084-3093.

[3] Pushparani, M. K., Vinod Kumar, D., & Gubbi, A. (2019). Arecanut Grade Analysis using Image Processing Techniques. International Journal of Engineering Research & Technology (IJERT), 7(10).

[4] Mukhtar, T., Jamil, S., Arif, U., Razzaq, W., & Wasif, M. (2021). Lemon Grading and Sorting Using Computer Vision. Engineering Proceedings, 12, 55.

[5] Patil, B., Panchal, H., Yadav, S., Singh, A., & Patil, D. (2017). Plant Monitoring Using Image Processing, Raspberry Pi & IoT. International Research Journal of Engineering and Technology (IRJET), 4(10), 1337-1340.

[6] ARM-Based Fruit Grading and Management SystemM. B. Avhad and S. M. Turkane, "ARM Based Fruit Grading and Management System Using Image Processing," International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 2, no. 1, pp. 243–248, Jan. 2013.

[7] Shape Detection and Classification Using OpenCV and ArduinoX. Poda and O. Qirici, "Shape Detection and Classification Using OpenCV and Arduino Uno," University of Tirana, 2020.

[8] Machine Vision System for Grading of Dried FigsM. Baigvand, A. Banakar, S. Minaei, J. Khodaei, and N. Behroozi-Khazaei, "Machine Vision System for Grading of Dried Figs," Computers and Electronics in Agriculture, vol. 119, pp. 158–165, 2015.

[9] Automated Classifying and Grading System of Tomatoes Using Fuzzy Logic L. C. Dorado, J. I. C. Aguila, and R. B. Caldo, "Smart Farm: Automated Classifying and Grading System of Tomatoes Using Fuzzy Logic," LPU-Laguna Journal of Engineering and Computer Studies, vol. 3, no. 3, pp. 65–72, Oct. 2016.

[10] IoT and Machine Learning-Based Grading and Sorting System P. Nirale and M. Madankar, "Analytical Study on IoT and Machine Learning Based Grading and Sorting System for Fruits," IEEE, 2021.

[11] Fresh Milk Smart Grading System Based on IoT W. Habsari, F. Udin, and Y. Arkeman, "An Analysis and Design of Fresh Milk Smart Grading System Based on Internet of Things," IOP Conf. Series: Earth and Environmental Science, vol. 1063, no. 1, p. 012059, 2022.

[12] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A review of convolutional neural networks in computer vision," Artificial Intelligence Review, vol. 57, no. 99, pp. 1–43, 2024.

[13] P. Srivastava, A. Shukla, and A. Bansal, "A comprehensive review on soil classification using deep learning and computer vision techniques," Multimedia Tools and Applications, vol. 80, pp. 14887–14914, Jan. 2021.

[14] M. A. Hasan, T. Bhargav, V. Sandeep, V. S. Reddy, and R. Ajay, "Image classification using convolutional neural networks," Int. J. Mech. Eng. Res. & Tech., vol. 16, no. 2, pp. 155–162, Apr. 2024.

[15] A. Bhat, S. Sowmya, J. Lobo, P. G. K., and S. Devadiga, "Automatic arceanut grade analysis and classification using computer vision and machine learning," Global J. Res. Eng. Comput. Sci., vol. 4, no. 2, pp. 149–151, Mar.-Apr. 2024.

[16] M. P. Arakeri and Lakshmana, "Computer vision-based fruit grading system for quality evaluation of tomato in agriculture industry," Procedia Comput. Sci., vol. 79, pp. 426-433, 2016.

[17] R. SwarnaLakshmi and B. Kanchanadevi, "A review on fruit grading systems for quality inspection," Int. J. Comput. Sci. Mobile Comput., vol. 3, no. 7, pp. 615–621, Jul. 2014.

[18] S. Naik and B. Patel, "Machine Vision based Fruit Classification and Grading – A Review," 12 International Journal of Computer Applications, vol. 170, no. 9, pp. 22–33, Jul. 2017.

[19] K. Kiratiratanapruk, P. Temiranrat, W. Sinthupinyo, P. Prempree, K. Chaitavon, S. Porntheeraphat, and A. Prasertsak, "Development of Paddy Rice Seed Classification Process using Machine Learning Techniques for Automatic Grading Machine," Journal of Sensors, vol. 2020, Art. ID 7041310, pp. 1–14, Jul. 2020.

[20] E. Chaowicharat and N. Dejdumrong, "A Step Toward an Automatic Handwritten Homework Grading System for Mathematics," Information Technology and Control, vol. 52, no. 1, pp. 169–184, 2023.

Appendix

The appendix provides supporting materials referenced throughout the report, including diagrams, tables, raw data, and supplementary explanations. It helps readers validate the methodology and understand additional technical details without interrupting the flow of the main content.

i. Model Description

The Smart Areca Leaf Plate Grading System follows a modular architecture that integrates machine learning, image processing, and embedded hardware to perform fully automated plate grading. The system is built around a Raspberry Pi 5, which functions as the central processing unit. Images of plates are captured using a camera module mounted above the conveyor belt. These images are preprocessed and passed through a fine-tuned deep learning model (ResNet50) deployed directly on the Raspberry Pi for real-time classification into Grade A, Grade B, or Reject.

To enable physical sorting, the Raspberry Pi controls the conveyor and sorting mechanism through GPIO pins connected to a motor driver, gear motor, and gear actuators. The display unit provides real-time feedback of classification status and system operation. Power is supplied using a regulated 12V source for motors and an independent 5V supply for the Raspberry Pi, ensuring stable operation.

This architecture ensures smooth coordination between hardware and software components, enabling efficient image acquisition, accurate grading, and automated sorting. The modular nature of the design supports easy enhancements, scalability, and integration into small and medium-scale areca plate manufacturing environments.

ii. Model Code

```
import tensorflow as tf

import numpy as np

import cv2

from tensorflow.keras.preprocessing.image import ImageDataGenerator

from tensorflow.keras.applications import ResNet50

from tensorflow.keras.applications.resnet50 import preprocess_input

from tensorflow.keras.models import Model
```

```
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Dropout
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
DATA_ROOT = r"D:\FinalPlates"
IMAGE_SIZE = (224, 224)
BATCH_SIZE = 32
NUM_CLASSES = 3
LEARNING_RATE = 0.001
EPOCHS = 75
FINE_TUNE_EPOCHS = 25
def user_custom_logic(img_array):
    if img_array.dtype != np.uint8:
        img_array = img_array.astype(np.uint8)
    return cv2.medianBlur(img_array, ksize=3)
def combined_preprocessing(image):
    image = user_custom_logic(image)
    return preprocess_input(image.astype(np.float32))
train_datagen = ImageDataGenerator(
    preprocessing_function=combined_preprocessing,
    rotation_range=90,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
```

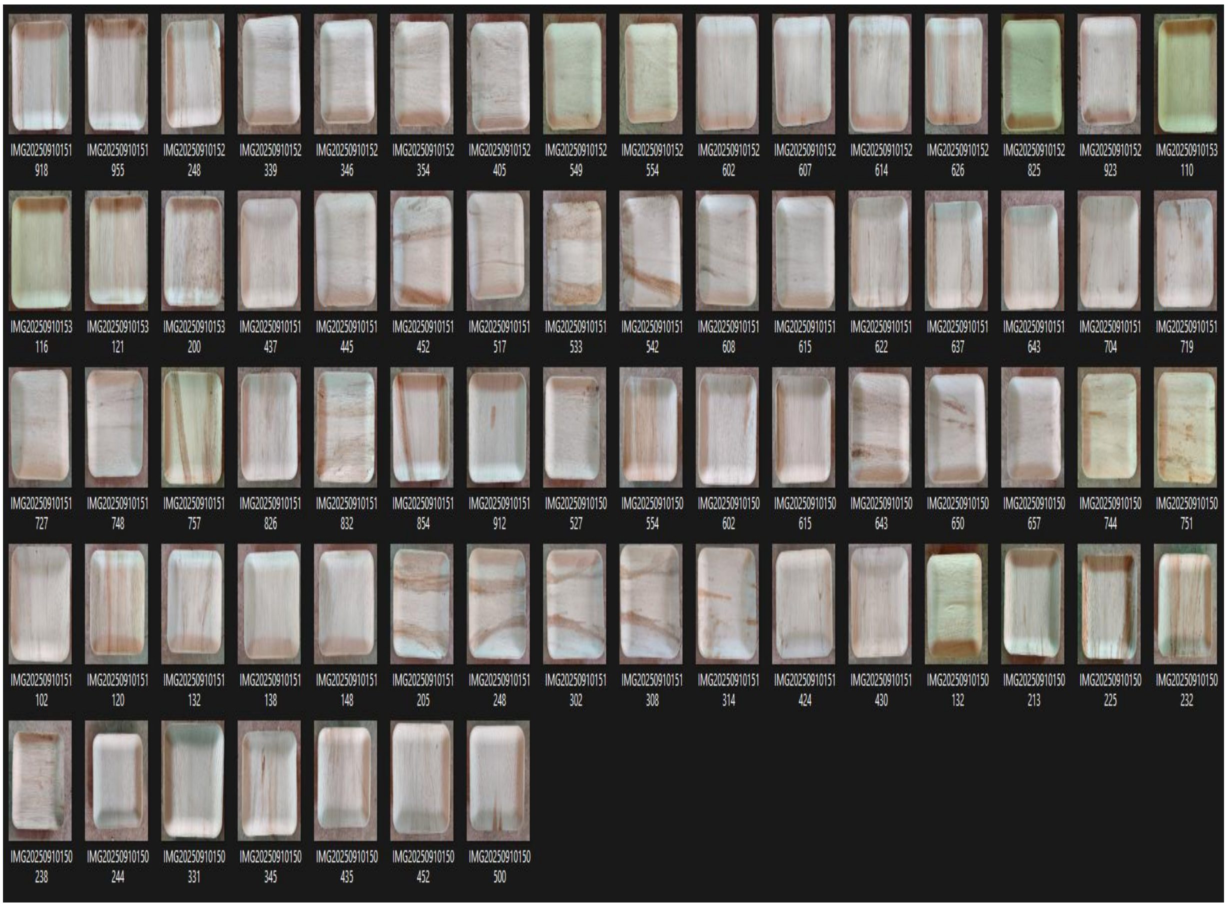
```
    validation_split=0.2)
validation_datagen = ImageDataGenerator(
    preprocessing_function=combined_preprocessing,
    validation_split=0.2)
train_generator = train_datagen.flow_from_directory(
    DATA_ROOT, target_size=IMAGE_SIZE, batch_size=BATCH_SIZE,
    class_mode='categorical', subset='training', seed=42)
validation_generator = validation_datagen.flow_from_directory(
    DATA_ROOT, target_size=IMAGE_SIZE, batch_size=BATCH_SIZE,
    class_mode='categorical', subset='validation', seed=42)
input_shape = (IMAGE_SIZE[0], IMAGE_SIZE[1], 3)
base_model=ResNet50(weights='imagenet',include_top=False, input_shape=input_shape)
base_model.trainable = False
x = GlobalAveragePooling2D()(base_model.output)
x = Dense(512, activation='relu')(x)
x = Dropout(0.4)(x)
output_layer = Dense(NUM_CLASSES, activation='softmax')(x)
model = Model(inputs=base_model.input, outputs=output_layer)
class AccuracyThresholdEarlyStopping(EarlyStopping):
    def __init__(self, monitor='val_loss', accuracy_monitor='accuracy',
                 threshold=0.90, patience=5, restore_best_weights=True):
        super().__init__(monitor=monitor, patience=patience,
                         restore_best_weights=restore_best_weights)
        self.accuracy_monitor = accuracy_monitor
        self.threshold = threshold
    def on_epoch_end(self, epoch, logs=None):
```

```
        current_acc = logs.get(self.accuracy_monitor)
        if current_acc is not None and current_acc < self.threshold:
            self.wait = 0
            return
        super().on_epoch_end(epoch, logs)
model.compile(optimizer=Adam(learning_rate=LEARNING_RATE),
              loss='categorical_crossentropy', metrics=['accuracy'])
callbacks = [
    AccuracyThresholdEarlyStopping(
        monitor='val_loss', accuracy_monitor='accuracy',
        threshold=0.90, patience=5, restore_best_weights=True),
    ModelCheckpoint('resnet_custom_preprocessed.h5',
                    save_best_only=True, monitor='val_loss')]
history = model.fit(train_generator, epochs=EPOCHS,
                    validation_data=validation_generator,
                    callbacks=callbacks, verbose=1)
base_model.trainable = True
fine_tune_at = 140
for layer in base_model.layers[:fine_tune_at]:
    layer.trainable = False
model.compile(optimizer=Adam(learning_rate=1e-5),
              loss='categorical_crossentropy', metrics=['accuracy'])
initial_epoch = history.epoch[-1] + 1
total_epochs = initial_epoch + FINE_TUNE_EPOCHS
history_fine = model.fit(train_generator, epochs=total_epochs,
                         initial_epoch=initial_epoch,
```

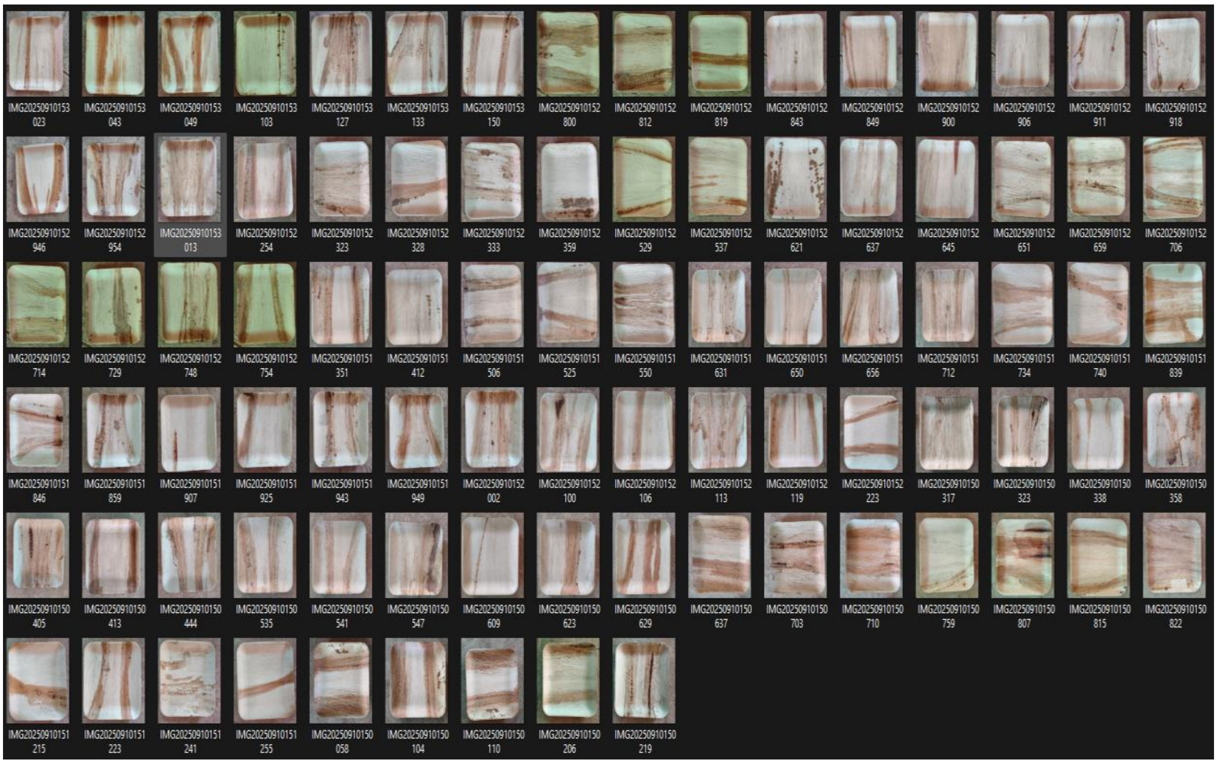
```
                  validation_data=validation_generator,
                  callbacks=callbacks, verbose=1)
model.save('resnet_custom_preprocessed_finetuned.h5')
```

iii. Datasets

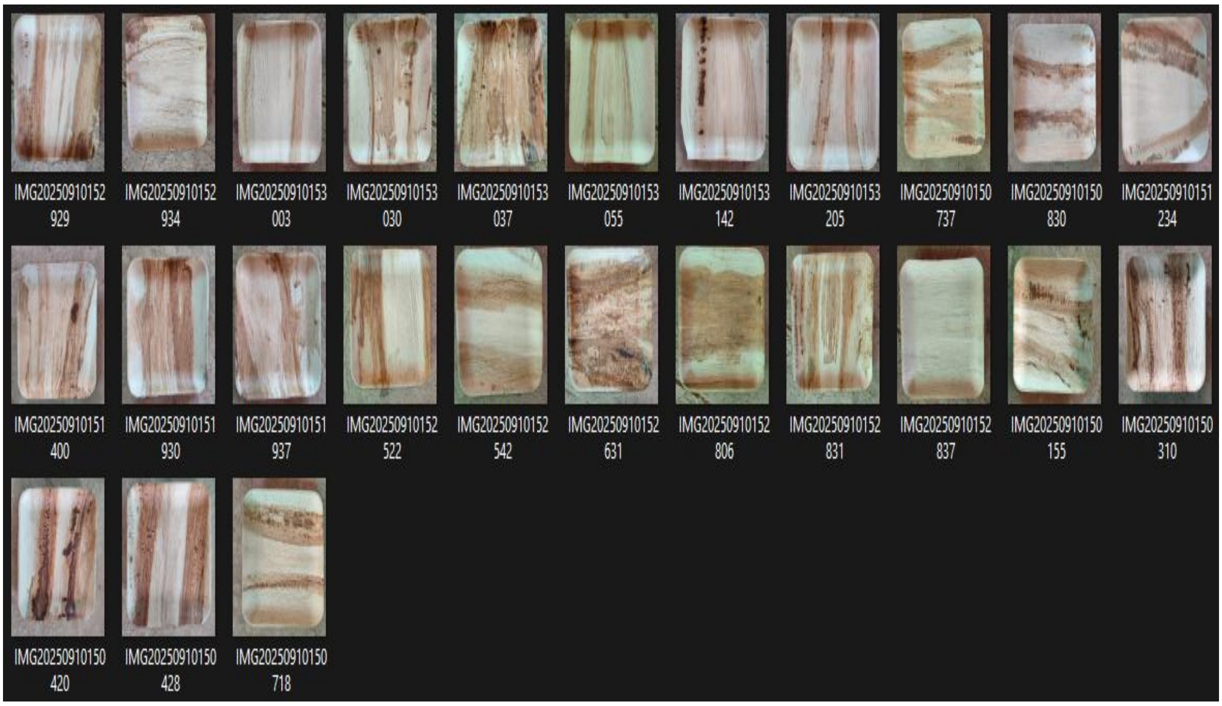
Class A:



Class B:



Reject:



Personal Profile

Mrs. Veena Bhat
Asst. Professor & HOD,
Project Guide

Mrs. Veena Bhat completed her M.Tech in Computer Science and Engineering from NMAM Institute of Technology in 2014.

Her core areas of expertise include Embedded Systems, Wireless Sensor Networks, Java Programming, Software Engineering, Database Management Systems, and Python Programming.

She was awarded the **First Prize for Best Paper** for her research work titled *"A Review: On EDM-Based Energy Aware Routing Protocol for Wireless Sensor Networks"*, presented at an International Conference held at NMAM Institute of Technology, Nitte, in May 2014.

Name: Harsha G V Subedar

USN: 4SU22AD016

Address: Chandanakere, Yedehalli, Bhadravathi, Shivamogga 577227

E-mail ID: 4su22ad016@sdmit.in

Contact Phone No: 9148405947

Name: Laxman Sannu Gouda

USN:4SU22AD025

Address: Gokarna, Kumta, Uttara kannada -581326

E-mail ID: 4su22ad025@sdmit.in

Contact Phone No: 7795288116

Name: Rahul Malatesh Sannapujar

USN: 4SU22AD043

Address: S/o Malatesh Sannapujar, Arer Keri Shikaripur, Shivamogga -577427

E-mail ID: 4su22ad043@sdmit.in

Contact Phone No: 9886873463

Name: Sateerth Ganapati Palkar

USN: 4SU23AD401

Address: #2656A, Hubli, Karwar Road, Opp Forest Naka, Kalaghatgi, Dharwad-581204

E-mail ID: 4su23ad401@sdmit.in

Contact Phone No: 7483852240
