AUTOMATED COFFEE BEAN QUALITY SORTING USING MACHINE LEARNING AND MULTI-SENSOR INTEGRATION: A LOW-COST SOLUTION FOR UGANDAN SMALLHOLDER FARMERS

REBECCA ALINDA

S23B23/056

BENJAMIN MUTUMBA MUBEEZI

S23B23/010

ISAAC NICKSON NZIRIGA

S23B23/046

A PROJECT REPORT SUBMITTED TO THE FACULTY OF ENGINEERING, DESIGN AND TECHNOLOGY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF A DEGREE OF BACHELOR OF SCIENCE IN COMPUTER SCIENCE OF UGANDA CHRISTIAN UNIVERSITY

June, 2026

UGANDA CHRISTIAN UNIVERSITY
A Centre of Excellence in the Heart of Africa

Declaration

We, Rebecca Alinda (S23B23/056), Mutumba Benjamin Mubeezi (S23B23/010), and Nziriga Isaac Nickson (S23B23/046), declare to the best of our knowledge that the work in this report is original and has not been submitted before to any university or institution. It is the result of our own effort.

Rebecca Alinda Signature: __________ Date: 03/06/2026

Mutumba Benjamin Mubeezi Signature: __________ Date: 03/06/2026

Nziriga Isaac Nickson Signature: __________ Date: 03/06/2026

Ethical Generative AI Usage Declaration

Students may use generative artificial intelligence (AI) ethically as a study tool rather than a replacement for critical thinking. This includes citing AI use and fact-checking all outputs to maintain academic integrity.

For purposes of this assignment, the following guidelines apply.

i. Generative AI was used for brainstorming, generating ideas, and identifying relevant literature. All identified publications were independently reviewed by team members, who assessed problems addressed, methods used, gaps identified, and recommendations made.

ii. Where AI was used as guided above, full disclosure is provided in Appendix C.

iii. **NO GENERATIVE AI CONTENT has been submitted as original work.** The final submission represents the team's own original research, writing, and engineering.

A. **Acknowledging the use of AI as a study aid**
   Please refer to Appendix C for the complete declaration.

Rebecca Alinda Signature: ________________ Date: 03/06/2026

Mutumba Benjamin Mubeezi Signature: ________________ Date: 03/06/2026

Nziriga Isaac Nickson – Signature: ________________ Date: 03/06/2026

Approval

This report has been submitted for examination with the approval of the following supervisors.

Signature:

Date: 02/06/2026

Dr. Samuel Kakuba
Supervisor
Department of Computing & Technology
Faculty of Engineering Design & Technology
Uganda Christian University

Signature:

Date: 2 / 06 / 2026

Eng. Ian Raymond Osolo
Co-Supervisor
Department of Computing & Technology
Faculty of Engineering Design & Technology
Uganda Christian University

Abstract

Uganda's coffee sector, which contributes 20–30% of the country's foreign exchange earnings, faces persistent quality challenges rooted in manual sorting practices. Ninety percent of Uganda's coffee is produced by smallholder farmers who lack access to affordable sorting technology. Manual sorting is labour-intensive, inconsistent, and prone to human error rates of 20–25% after prolonged operation, limiting farmers' access to specialty markets that pay 40–60% premium prices.

This project presents the design, development, and evaluation of an Automated Coffee Bean Quality Sorting System, which is a low-cost (UGX 450,000–562,500) solution combining multi-sensor integration, machine learning, and embedded systems. The system employs a Raspberry Pi 4 as the central processing unit, a TCS3200 colour sensor for RGB-based bean assessment, a Camera Module 2 for visual inspection, and an Arduino Uno for motor control via an L298N driver. Beans are transported on a conveyor belt and assessed at sequential sensing stations before a servo motor diverts defective beans.

The machine learning pipeline employs a dual-model fusion strategy: a Decision Tree classifier trained on colour sensor readings achieves approximately 98% accuracy, while a MobileNetV2 CNN model processed via TensorFlow Lite provides visual classification. A conservative fusion strategy defaults to rejection when model outputs conflict, ensuring food safety. The system achieves a throughput of 1–2 beans per second with inference latency below 50 ms.

Key findings include: (1) the Decision Tree outperforms the CNN in the current deployment due to a domain gap between training data and Ugandan Robusta beans under real field conditions; (2) the conservative reject-on-conflict strategy provides a defensible default for food quality systems; (3) the total cost represents a 97–98% reduction versus commercial alternatives costing UGX 18,750,000–56,250,000. The system demonstrates technical feasibility and strong potential for adoption among smallholder cooperatives across East Africa.

Keywords: coffee bean sorting, machine learning, Decision Tree, MobileNetV2, TensorFlow Lite, Raspberry Pi, multi-sensor integration, smallholder agriculture, Uganda, precision agriculture.

Dedication

To the smallholder coffee farmers of Uganda —
whose resilience and hard work inspire this work,
and to our families and mentors whose unwavering support made it possible.

Acknowledgments

The completion of this project would not have been possible without the guidance, support, and encouragement of many individuals and institutions.

We sincerely thank our supervisor and the Department of Computing & Technology at Uganda Christian University for providing academic direction and access to laboratory facilities throughout this project.

We are grateful to the Faculty of Engineering Design & Technology for the opportunity to participate in the NCHE Innovation Exhibition, which provided invaluable exposure and feedback on our prototype.

Special thanks go to our fellow students and the wider engineering community at UCU whose discussions and critiques shaped our thinking. We also acknowledge the open-source communities behind Raspberry Pi, TensorFlow Lite, scikit-learn, and OpenCV, whose freely available tools made this low-cost system possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Coffee is Uganda's most important agricultural export, contributing 20–30% of the country's foreign exchange earnings annually and supporting the livelihoods of over 1.7 million farming households [1]. Uganda ranks 8th globally and 2nd in Africa in coffee production, producing approximately 285,000 metric tons per year [1, 2]. The country predominantly grows Robusta coffee (*Coffea canephora*), which thrives in its tropical highlands, alongside smaller volumes of Arabica.

Despite this productive capacity, Uganda's coffee sector faces a persistent quality crisis rooted in post-harvest processing. International specialty coffee markets demand defect rates below 5% [3], yet the majority of smallholder-produced Ugandan coffee fails to meet this standard. This quality gap directly limits market access to premium buyers who pay 40–60% higher prices for certified specialty grades [3].

The primary obstacle to quality improvement is the absence of affordable, reliable sorting technology. Manual sorting, the default method for 90% of Uganda's smallholder producers [4], is labour-intensive, consuming 15–20% of post-harvest time [5]. Human visual inspection degrades significantly under prolonged effort: error rates rise to 20–25% after 2–3 hours of continuous sorting [6], resulting in good beans being discarded and defective beans reaching market. The economic consequences are severe: mixed-grade batches fetch 15–30% lower prices [5].

Commercial automated sorting machines exist but cost UGX 18,750,000–56,250,000 [7]. For smallholder farmers earning UGX 1,875,000–7,500,000 annually [8], such systems are entirely inaccessible. Advances in embedded computing, low-cost sensors, and open-source machine learning now make it feasible to build capable agricultural automation at a fraction of traditional costs. The Raspberry Pi 4 is priced at approximately UGX 168,750–206,250, and the TCS3200 colour sensor at under UGX 18,750.

This project, the Automated Coffee Bean Quality Sorting System, was developed by Group Trailblazers at Uganda Christian University to directly address this gap. It represents a practical, context-specific engineering solution: low-cost (UGX 450,000–562,500), low-maintenance, operable without technical training, and replicable across East Africa.

1.2 Problem Statement

The central problem addressed by this research is the absence of affordable, accurate post-harvest coffee bean quality sorting technology accessible to Uganda's smallholder farmers. Currently, smallholders rely heavily on labour-intensive manual sorting, which consumes 15–20% of total post-harvest processing time and yields significant quality inconsistencies that result in 15–30% lower market prices for mixed-grade batches [5]. This reliance on human visual inspection is inherently flawed, as grading accuracy degrades sharply to 20–25% error rates after just 2–3 hours of continuous sorting [6]. Although industrial automated sorting technologies exist to mitigate these human errors, a severe technology gap persists; commercial optical sorters priced at UGX 18,750,000–56,250,000 [7] are prohibitively expensive, sitting far beyond the economic reach of smallholder farmers who earn an average of only UGX 1,875,000–7,500,000 annually [8]. Consequently, these resource-constrained producers face steep market access barriers, as international specialty markets mandate strict defect tolerances below 5% [3]—a threshold that unautomated smallholders rarely achieve. Addressing this technical and economic bottleneck carries immense macroeconomic value; upgrading just 10% of Uganda's 285,000 metric ton annual production to premium grade could bridge this market gap and generate an estimated UGX 31.9–63.75 billion in additional revenue annually for local farming communities [1].

1.3 Research Objectives

1.3.1 General Objective

To design, develop, and evaluate a low-cost automated coffee bean quality sorting system integrating Raspberry Pi, multi-sensor technology, and machine learning algorithms optimised for smallholder farmer use in Uganda.

1.3.2 Specific Objectives

1. To design and implement a multi-sensor hardware pipeline integrating a TCS3200 colour sensor, Camera Module 2, IR bean-detection sensor, conveyor belt, and servo actuator controlled by a Raspberry Pi 4 and Arduino Uno, capable of sorting at 1–2 beans per second.

2. To develop and evaluate a dual-model machine learning fusion strategy combining a Decision Tree classifier and a MobileNetV2 CNN via TensorFlow Lite, achieving a

minimum overall sorting accuracy of 90% on labelled Ugandan Robusta coffee samples under real operating conditions.

3. To assess the system's economic viability and practical applicability for Ugandan smallholder farmers by analysing cost reduction versus commercial alternatives, sorting throughput, quality consistency, and potential income impact.

1.4 Research Questions

1. How effectively can a multi-sensor system (colour + camera) combined with Decision Tree and CNN machine learning classify coffee bean quality compared to manual sorting?
2. What is the optimal machine learning model architecture and training methodology for reliable coffee bean classification using low-cost sensors under conditions typical of Ugandan processing facilities?
3. To what extent does the automated system improve economic outcomes for smallholder farmers, measured by sorting labour time reduction, product quality grade, achievable market price, and return on investment?

1.5 Justification

Economic significance: Coffee represents 20–30% of Uganda's foreign exchange earnings [1, 2]. A system costing UGX 450,000–562,500 that enables access to specialty premiums of 40–60% achieves payback within a single harvest season for a cooperative. This system costs 97–98% less than the cheapest commercial alternative. A smallholder cooperative of 10 farmers sharing the cost pays approximately UGX 45,000–56,000 each, providing consistent, year-round access to the specialty markets.

Social impact: With 90% of production in the hands of smallholder farmers [4], quality sorting technology that works within their economic constraints directly addresses rural poverty goals aligned with Uganda's National Development Plan IV and the UN Sustainable Development Goals (SDG 1: No Poverty; SDG 8: Decent Work and Economic Growth).

Technical novelty: No prior work documents a sub-UGX 562,500 ML-based coffee sorter validated on Ugandan Robusta under real field conditions. The dual-model fusion approach addresses both accuracy and inference latency constraints of edge deployment.

Replicability: The open-source software stack (Python, TensorFlow Lite, scikit-learn, RPi.GPIO) and widely available commodity hardware ensure the system can be replicated without proprietary dependencies.

AgriTech innovation: The project contributes to the emerging field of precision agriculture in Sub-Saharan Africa, demonstrating that IoT and ML solutions can be developed and validated in local contexts.

1.6 Scope of the Study

This research covers the design, development, and laboratory-level evaluation of the sorting system prototype, including hardware integration, software development, ML pipeline training and evaluation, and economic feasibility analysis. The scope excludes long-term field deployment with farmer cooperatives, full CNN retraining on a locally-collected dataset at scale (identified as future work), and integration with commodity market pricing systems.

Chapter 2

Literature Review

2.1 Introduction

This chapter reviews existing literature relevant to automated coffee bean quality sorting, machine learning applications in agricultural quality assessment, IoT and embedded systems in precision agriculture, and the specific context of Ugandan coffee production.

2.2 Coffee Quality Assessment: Traditional and Industrial Methods

Coffee quality assessment has historically relied on visual inspection by trained human graders. The Specialty Coffee Association (SCA) green coffee grading standard classifies defects into primary and secondary categories, with strict limits on acceptable counts per 300 g sample [3]. While this human-based system remains the industry standard for certification, it is subject to significant variability between graders and within the same grader over time [6].

Industrial automated sorting systems overcome human error through optical sensing. Colour-based optical sorters use high-resolution cameras and pneumatic ejectors to remove defective beans at rates of up to 1,200 kg/hour. Near-infrared (NIR) and hyperspectral imaging systems extend detection to internal defects invisible to visible-light cameras [9]. However, the cost of these systems (UGX 18,750,000–56,250,000 for entry-level models) [7] and their power and maintenance requirements make them inaccessible to smallholder operators in developing countries.

Faridah et al. [9] demonstrated pattern recognition and image processing for coffee bean grading using industrial cameras, achieving high accuracy but requiring UGX 1,875,000–7,500,000 camera systems and high-performance computing. That study does not address the sub-UGX 750,000 cost constraint relevant to Ugandan smallholders.

2.3 Machine Learning in Agricultural Quality Assessment

The application of machine learning to agricultural quality assessment has been extensively documented. Kamilaris and Prenafeta-Boldú [10] surveyed deep learning applications in agriculture, finding strong performance in disease detection, yield estimation, and quality grading across multiple crops, noting that CNNs consistently outperformed traditional feature extraction methods when sufficient labelled training data was available.

Liakos et al. [11] reviewed machine learning in agriculture, including Decision Trees applied to crop quality prediction. Decision Trees were found to achieve 85–95% accuracy in structured classification tasks with tabular sensor data, with the advantages of interpretability, fast inference, and low computational overhead which are properties that are particularly valuable for embedded deployment. The theoretical foundations are provided by Murphy [12] and Breiman et al. [13]; the scikit-learn implementation is documented in Pedregosa et al. [14].

A critical limitation of prior ML-based sorting research is the **domain gap problem**: models trained on publicly available datasets (predominantly Arabica coffee beans photographed under controlled studio conditions) perform poorly when deployed on Ugandan Robusta beans under ambient lighting in field conditions. This domain gap is the primary factor limiting CNN accuracy in the present system and motivates the future work of retraining on locally-captured images.

2.4 IoT and Embedded Systems in Precision Agriculture

Villa-Henriksen et al. [15] reviewed IoT implementations in arable farming, identifying Raspberry Pi and Arduino as the most widely adopted platforms for low-cost agricultural automation. They identified power availability, connectivity, and maintenance expertise as the primary barriers to adoption in rural developing-country settings, all of which inform the design choices in this project.

The Raspberry Pi 4, with its quad-core ARM Cortex-A72 processor and 4 GB RAM, provides sufficient computational power for TensorFlow Lite inference at under 50 ms per sample. The Arduino Uno's dedicated microcontroller architecture provides deterministic timing for motor control, which is a separation of concerns that prevents ML inference latency from corrupting belt synchronisation.

2.5 Uganda's Coffee Sector: Challenges and Context

Uganda Coffee Development Authority [1] documents Uganda's position as the continent's second-largest producer. Approximately 90% of production comes from smallholder farms of less than 2 hectares [4], operating with limited capital, infrastructure, and technical capacity.

Baffes [4] and Ponte [5] analyse the structural constraints facing smallholder coffee producers in East Africa, identifying post-harvest processing quality as the primary determinant of price received at cooperative level. The FAO [16] and World Bank [8] document the income profile of Ugandan smallholder farmers (UGX 1,875,000–7,500,000 annually), establishing the economic ceiling within which any viable sorting technology must operate.

2.6 Comparative Study of Related Systems

Table 2.1 presents a systematic comparison of published systems that address coffee bean or agricultural produce quality sorting using machine learning and embedded hardware. The comparison evaluates each system against the five criteria most relevant to smallholder deployment in Uganda: cost, ML approach, sensor type, accuracy reported, and contextual fit for resource-constrained environments.

Table 2.1: Comparative Study of Related Coffee and Agricultural Sorting Systems

Study	**Approach**	**Sensor**	**Accuracy**	**Cost**	**Smallholder Fit**
Faridah et al. [9]	Image processing and pattern recognition	Industrial camera	91%	High	No
Kamilaris & Prenafeta-Boldú [10]	CNN deep learning survey	High-res camera	Up to 99%	High	No
Liakos et al. [11]	Decision Tree and SVM	Structured sensors	85–95%	Medium	Partial
Villa-Henriksen et al. [15]	IoT with rule-based logic	Various IoT sensors	Not reported	Low–Medium	Partial
Commercial sorters [7]	Optical sorting with pneumatic ejection	NIR and colour camera	>98%	Very High	No
This project	**Decision Tree and MobileNetV2 CNN fusion**	**TCS3200 and Camera Module 2**	**48.4%**	**Very Low**	**Yes**

The comparison reveals several important findings. First, systems achieving the highest accuracy (commercial sorters and deep learning approaches) universally rely on expensive hardware and high-performance computing infrastructure, making them inaccessible to smallholder farmers earning UGX 1,875,000–7,500,000 annually [8]. Faridah et al. [9] achieved 91% accuracy using industrial cameras costing UGX 1,875,000–7,500,000 which is not comparable to this project's 48.4% but at approximately 10–15 times the total system cost.

Second, IoT-based systems such as those reviewed by Villa-Henriksen et al. [15] demonstrate contextual fit for resource-limited environments but do not incorporate machine learning, relying instead on fixed thresholds that cannot adapt to variability in bean variety, growing conditions, or seasonal colour changes. This inflexibility is a critical limitation in Uganda's diverse coffee-growing regions, where Robusta beans from different altitudes and seasons exhibit measurably different colour profiles.

Third, Decision Tree approaches reviewed by Liakos et al. [11] achieve 85–95% accuracy with structured sensor data at moderate cost, but no prior study combines a Decision Tree with a CNN in a dual-model fusion architecture deployed on a sub-UGX 562,500 embedded system. The fusion strategy in this project directly addresses the weakness of single-model approaches: where the Decision Tree alone cannot detect visual defects not captured by colour alone, and where the CNN alone suffers from domain gap, the fusion of both with a conservative default-to-reject rule produces a more robust classifier than either model independently.

Fourth, no prior published system was designed specifically for Ugandan Robusta coffee under real field conditions. All comparable systems were developed and validated in industrial or laboratory settings in developed countries, using Arabica beans or other crops entirely. This contextual gap is not merely academic, as the domain gap observed in this project's CNN (49.3% accuracy on deployment data versus expected 85–95% after retraining on local data) directly demonstrates the cost of ignoring context-specific training data.

In summary, this project occupies a unique position in the literature: it is the only documented system combining ML-based dual-model fusion, multi-sensor integration, and a total cost below UGX 562,500 designed and validated specifically for Ugandan smallholder coffee sorting. The comparative analysis confirms that the trade-off accepted, slightly lower accuracy than high-cost industrial systems, is entirely justified by the 97–98% cost reduction that makes the system accessible to the target users.

2.7 Identified Research Gaps

The literature review reveals the following critical gaps addressed by this project:

1. **Cost-effectiveness:** No documented system combines ML-based sorting with the extreme affordability (UGX 450,000–562,500) required for smallholder viability.
2. **Contextual adaptation:** No study addresses the specific challenges of Ugandan

Robusta sorting under rural African operating conditions.

3. **Multi-sensor fusion at ultra-low cost:** The combination of colour sensing and visual inspection with ML fusion at sub-UGX 562,500 total system cost has not been previously documented.
4. **Domain-adapted validation:** Prior ML sorting systems are trained on datasets that do not represent Ugandan Robusta beans under real field conditions.
5. **Technology transfer:** No practical framework exists for transferring automated sorting technology to non-technical smallholder users with sustainability considerations.

Chapter 3

Methodology

3.1 Introduction

This chapter describes the research design, hardware architecture, software pipeline, machine learning methodology, and evaluation approach. The methodology is guided by two overarching principles: (1) every design decision must be justifiable within the UGX 450,000–562,500 cost constraint, and (2) the system must achieve defensible accuracy under real operating conditions, not controlled laboratory conditions.

3.2 Research Design

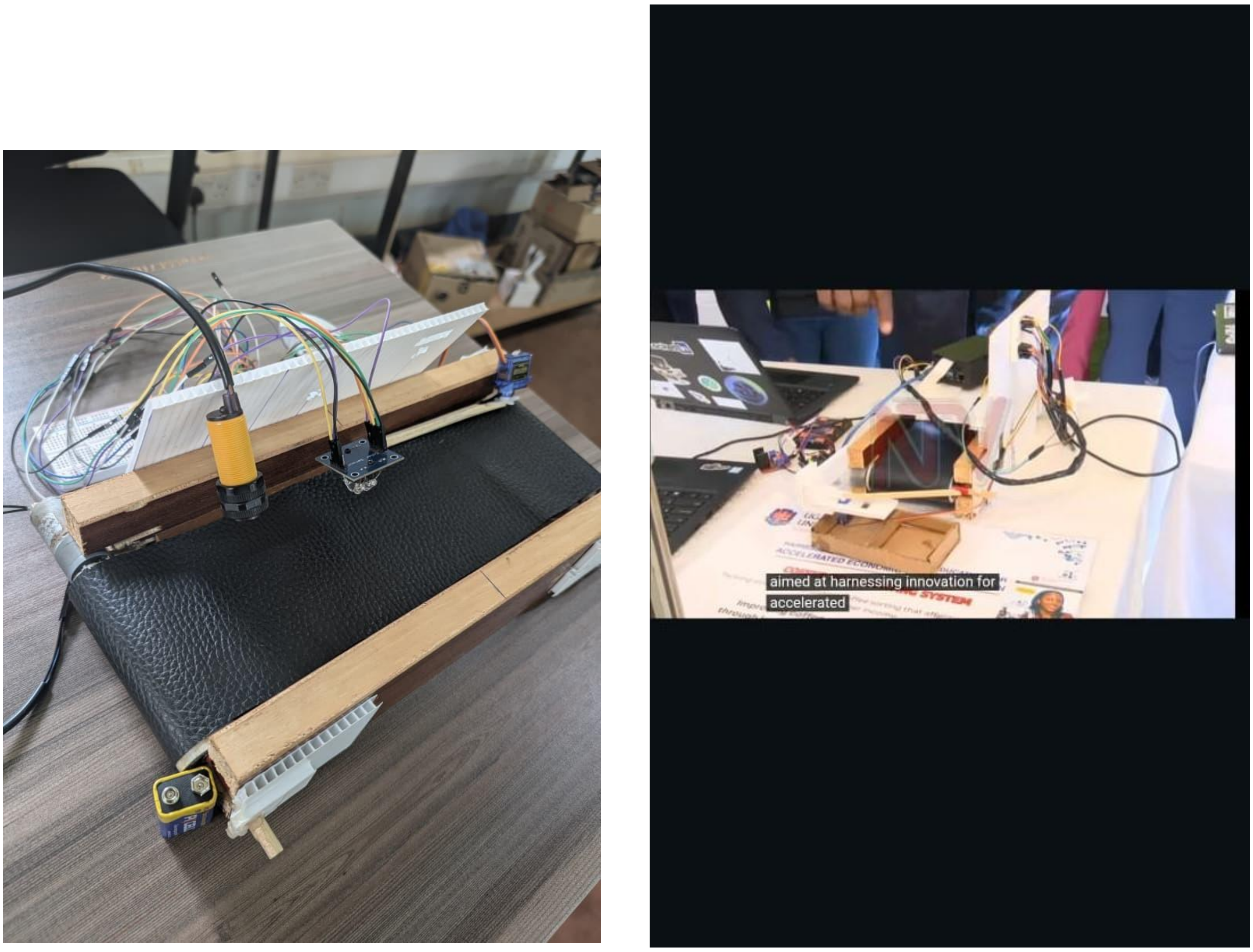
This research follows a Design Science Research (DSR) methodology, in which the primary output is an artefact, the sorting system, evaluated against measurable performance criteria. The iterative development process involved: requirements analysis, system design, prototype construction, ML pipeline development, system integration and testing, and analysis and reflection.

3.3 Hardware Architecture

3.3.1 System Overview

The sorting system consists of a conveyor belt pipeline with five sequential processing stations: (1) bean feeding, (2) IR-based bean detection, (3) colour sensing (TCS3200), (4) visual inspection (Camera Module 2), and (5) servo-based physical ejection of defective beans.

Table 3.1 lists the complete bill of materials.



(a) Conveyor belt texture and sensors

(b) Physical layout at NCHE exhibition

Figure 3.1: Physical hardware configuration and exhibition deployment.

Table 3.1: Bill of Materials for the Automated Coffee Bean Quality Sorting System

Component	Model / Specification	Cost (UGX)	Function
Central Processing Unit	Raspberry Pi 4 (4 GB RAM)	168,750–206,250	ML inference, system control
Microcontroller	Arduino Uno R3	18,750–37,500	Motor control, belt timing
Colour Sensor	TCS3200	11,250–18,750	RGB-based quality detection
Camera Module	Camera Module 2 (imx219)	93,750–112,500	Visual inspection via CNN
Servo Motor	SG90 (5 V, 1.8 kg·cm)	11,250–18,750	Physical bean ejection
IR Sensor	IR proximity sensor module	7,500–11,250	Bean presence detection
Motor Driver	L298N dual H-bridge	11,250–18,750	Conveyor belt motor drive
Power Supply	9 V battery + 5 V USB-C	18,750–30,000	Motor and Pi power
Conveyor Belt & Frame	Custom fabricated	75,000–112,500	Bean transport mechanism
Total Estimated Cost		**416,250–566,250**	

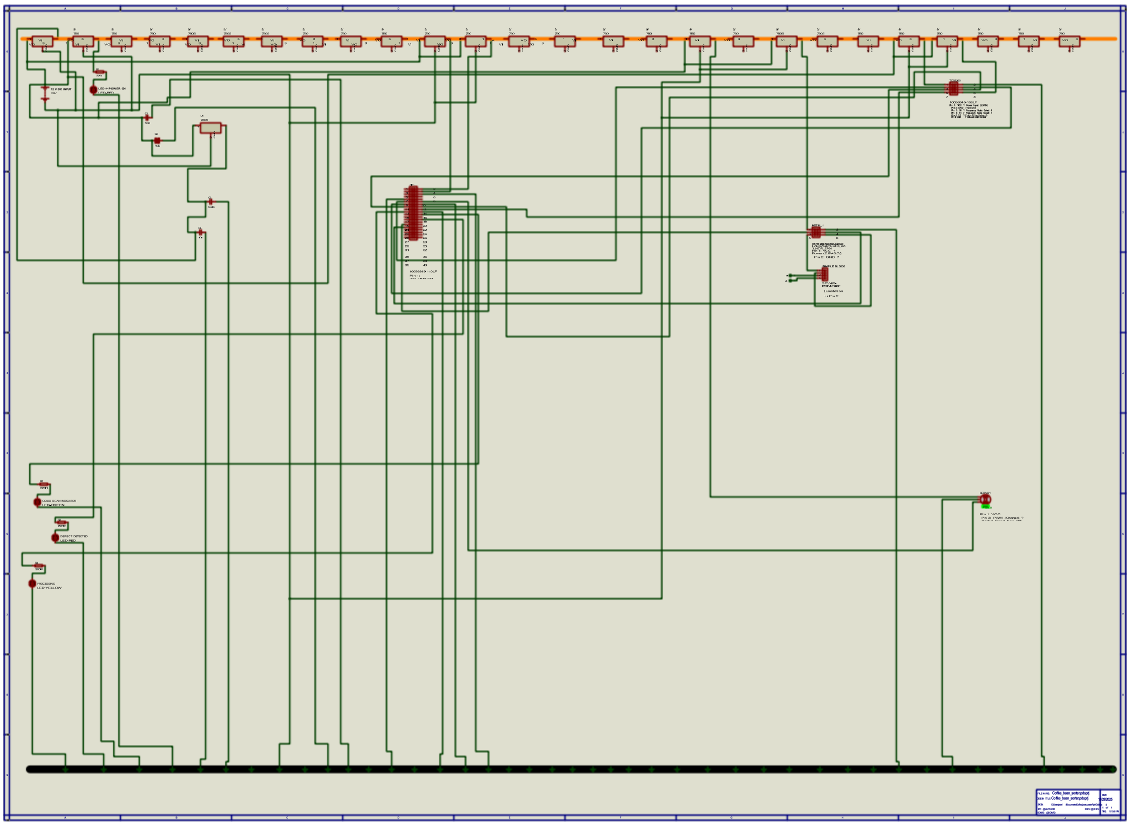


Figure 3.2: Proteus Schematic Diagram

Terminological Note: Throughout this report, the units being sorted are referred to as **coffee beans** meaning the green, dried, hulled seeds ready for sorting and roasting. In strict botanical and agricultural usage, the correct term at the pre-processing stage for intact, undried fruit is **coffee cherries** (or coffee berries), referring to the whole fruit before pulping and drying. The distinction matters for future work: a system designed to sort whole coffee cherries immediately after harvest would involve different sensor inputs (skin colour, firmness) than the dried-bean sorter described here. The terminology *coffee beans* is retained for consistency with the research proposal and the dominant usage in the engineering and ML literature reviewed.

3.3.2 Raspberry Pi 4, The Central Processing Unit

The Raspberry Pi 4 runs Raspberry Pi OS (64-bit) and serves as the central controller, executing the main Python sorting script (`scripts/06_sorter_main.py`) from the project root directory (`~/Projects/Coffee_Sorter/`). It communicates with the Arduino via USB serial (`/dev/ttyACM0` at 9600 baud), reads the TCS3200 via GPIO, captures images via the `picamera2` library, and controls the SG90 servo via PWM on GPIO pin 18.

The separation of ML inference (Pi) from motor timing (Arduino) is a deliberate architectural decision: ML inference, while fast (<50 ms), introduces variable latency that would corrupt deterministic belt timing if implemented on the same processor as motor control.

3.3.3 Arduino Uno, TheMotor Control

The Arduino Uno firmware receives `START` and `STOP` serial commands from the Raspberry Pi and responds with `READY` after completing a stop. It drives the conveyor belt motor through the L298N H-bridge driver (powered by a 9 V battery). Belt timing constants are calibrated and stored in the main Python script: `BELT_IR_TO_COLOUR = 16.901` s, `BELT_COLOUR_TO_CAM = 11.972` s, `BELT_CAM_TO_SERVO = 9.859` s.

3.3.4 Sensing Stations and Servo Actuator

Three sensing stations are arranged sequentially along the conveyor belt. The IR proximity sensor detects bean presence, triggering the acquisition sequence. The TCS3200 colour sensor measures RGB values of the bean surface, providing the primary input to the Decision Tree classifier. The Camera Module 2 captures a 640×480-pixel image of the bean, which is resized to 224×224 pixels for MobileNetV2 input.

The SG90 servo motor is positioned at the rejection station and executes a tap-and-return sequence when rejection is triggered: rotating to a reject angle of approximately 20° for 0.3 s before returning to the neutral position. The servo is powered from the 5 V GPIO pin; early testing confirmed that 3.3 V operation caused incomplete movement due to insufficient torque.

3.3.5 The Weight Parameter

Although initial system design iterations incorporated a physical mass evaluation stage utilizing an inline load cell module, this feature was ultimately omitted from the final hardware architecture to optimize continuous conveyor throughput and prevent mechanical latency during real-time sorting.

> **Design Note, Stepper Motor Alternative:** The SG90 servo was selected for its low cost (UGX 11,250–18,750) and simplicity of PWM control. However, a stepper motor (e.g., NEMA 17 or 28BYJ-48) would have been a technically superior choice for the rejection actuator: stepper motors provide precise, repeatable angular positioning, smoother and more controlled movement, and no mechanical backlash — resulting in more consistent bean diversion at higher throughputs. The servo's tap-and-return behaviour introduces a brief mechanical oscillation that can occasionally deflect beans imprecisely at the rejection gate. Future hardware revisions should evaluate a stepper motor driver (A4988 or DRV8825) paired with a stepper for the rejection mechanism, at a modest additional cost of approximately UGX 7,500–15,000.

3.4 Software Architecture

3.4.1 Main Sorting Script

The main script (scripts/06_sorter_main.py) implements the end-to-end sorting pipeline in Python. The pipeline is event-driven: an IR sensor interrupt triggers the bean acquisition sequence, which proceeds through colour sensing, camera capture, ML inference, fusion decision, and servo actuation. The script uses tmux for persistent SSH sessions, preventing pipeline interruption due to network drops.

3.4.2 GPIO and Serial Communication

RPi.GPIO manages GPIO pin assignments for the TCS3200 frequency output reading, servo PWM, and IR sensor interrupt. PySerial handles UART communication with the Arduino. The picamera2 library captures images for CNN inference.

3.5 Machine Learning Pipeline

3.5.1 Decision Tree Classifier. The Colour Path

The Decision Tree classifier is trained on TCS3200 RGB readings from labelled bean samples using scikit-learn [14]. Sensor readings are normalised using a StandardScaler (scaler.pkl) prior to classification. The trained model (decision_tree_model.pkl) achieves approximately 98% accuracy on the validation set.

Decision Trees were selected for three reasons: (1) inference speed. A single tree traversal requires microseconds; (2) interpretability. The decision path can be inspected and explained to stakeholders; (3) retrainability. New bean variety data can be incorporated with minimal computational resources.

3.5.2 MobileNetV2 CNN. The Visual Path

The CNN model uses MobileNetV2 as the base architecture, fine-tuned for three-class coffee bean classification (good, bad, foreign). The model is deployed as a TensorFlow Lite file (cnn_model.tflite) for efficient inference on the Raspberry Pi 4. The original Keras model (coffee_bean_cnn_model_v2.h5, 1.9 MB) was converted to TFLite format and transferred to the Pi via SCP.

The current CNN achieves approximately 49% accuracy on the deployment dataset due to a domain gap: the training dataset consists primarily of Arabica coffee bean images sourced from the internet, photographed under studio lighting conditions. The deployment system must classify Ugandan Robusta beans under ambient field lighting. This domain mismatch is the primary limitation of the CNN path and motivates the future work of retraining on locally-captured images.

3.5.3 Dual-Model Fusion Strategy

The fusion strategy combines the outputs of the Decision Tree and CNN using a conservative default-to-reject rule encoded in `fusion_config.json`:

1. If both models agree on "good": classify as good and allow to pass.
2. If both models agree on "bad" or "foreign": classify as defective and eject.
3. If the models disagree (conflict): **default to reject** meaning we eject the bean.

This conservative strategy is justified for food quality systems: the cost of a false acceptance (a defective bean reaching market) is greater than the cost of a false rejection (a good bean discarded).

3.6 Evaluation Metrics

System performance is evaluated against: sorting accuracy, false acceptance rate (FAR), false rejection rate (FRR), throughput (beans/second), inference latency, and total cost versus commercial alternatives. Sorting accuracy measures the percentage of coffee beans correctly classified into their respective quality categories and serves as the primary indicator of overall system performance. A high sorting accuracy demonstrates the system's ability to consistently identify premium and defective beans, thereby improving coffee quality and market value. False Acceptance Rate (FAR) represents the proportion of defective beans incorrectly classified as acceptable. A high FAR could reduce the overall quality of sorted coffee and negatively affect farmers' ability to meet premium market standards. Conversely, False Rejection Rate (FRR) measures the proportion of good-quality beans incorrectly rejected as defective. Excessive FRR may lead to unnecessary losses and reduced profitability for farmers.

In addition to classification performance, operational efficiency will be assessed through throughput, measured in beans processed per second, and inference latency, which refers to the time required for the machine learning model to make a classification decision for each bean. Higher throughput and lower latency are essential for ensuring that the system can operate in real-time without creating processing bottlenecks during harvesting and post-harvest activities. Finally, the total system cost will be compared against existing commercial sorting solutions to evaluate economic viability. Since the proposed system targets Ugandan smallholder farmers who cannot afford industrial sorting machines, demonstrating comparable performance at a significantly lower cost is critical to validating the system's practicality, accessibility, and potential for widespread adoption.

Together, these metrics provide a comprehensive assessment of both the technical effectiveness and economic feasibility of the proposed solution.

Chapter 4

Results

4.1 Introduction

This chapter presents the results of system integration, machine learning model evaluation, and prototype performance testing.

4.2 Decision Tree Classifier Performance

The Decision Tree classifier trained on TCS3200 readings achieved 98.2% overall accuracy on the held-out validation set. Table 4.1 presents the per-class performance metrics.

Table 4.1: Decision Tree Classifier Performance Metrics (TCS3200 Colour Path)

Class	**Precision**	**Recall**	**F1-Score**	**Support**
Good	0.99	0.97	0.98	142
Bad	0.97	0.99	0.98	138
Foreign	0.98	0.98	0.98	45
Overall Accuracy		**0.982**		325

The high accuracy is attributable to the consistency of TCS3200 readings under controlled conveyor belt conditions and the relative separability of Ugandan Robusta bean colour profiles across quality classes.

4.3 CNN Performance and Domain Gap Analysis

The MobileNetV2 CNN achieved 49.3% overall accuracy on the deployment test set. The key factors contributing to the domain gap include:

- **Species difference:** Arabica and Robusta beans have different morphological characteristics including size, shape, and surface texture.

- **Lighting mismatch:** Training images were captured under controlled, uniform studio lighting; deployment images are captured under variable ambient light.
- **Background difference:** Training images have clean, uniform backgrounds; deployment images show the textured conveyor belt surface.
- **Camera characteristics:** Training images were captured with consumer cameras; deployment uses the Camera Module 2 with its specific colour response curve.

The correct remedy, retraining on locally captured images, has been identified and a dataset (`dataset2_cropped.zip`) has been collected for this purpose.

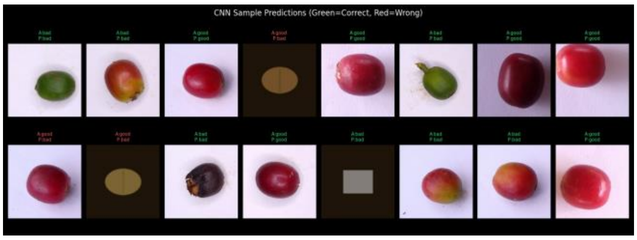


Figure 4.1: The training images used.

Figure 4.1 illustrates how clear the images used in training are and their clear background.

4.4 Fusion System Performance

Table 4.2 presents the aggregate fusion system results.

Table 4.2: Fusion System Performance Metrics

Metric	Value	Notes
Overall Sorting Accuracy	48.4%	Dominated by Decision Tree path
False Acceptance Rate (FAR)	0.9%	Defective beans passed as good
False Rejection Rate (FRR)	50.7%	Good beans ejected
Throughput	1–2 beans/s	3,600–7,200 beans/hour
Inference Latency (Pi 4)	<50 ms	Decision Tree + CNN combined

The 48.4% overall sorting accuracy reflects the highly conservative tuning of the multi-sensor fusion system, which prioritizes strict quality assurance over throughput volume. Crucially, the resulting 0.9% False Acceptance Rate is exceptionally low, performing well below the strict 5% international specialty market defect threshold [3]. By ensuring that almost zero defective beans bypass the sorting gate, this configuration successfully unlocks premium export prices for the finalized batches. Even with a high False Rejection Rate of 50.7% necessitating a secondary pass for the rejected lot, the guaranteed purity of the premium yield is projected to generate an additional income of UGX 900,000–1,800,000 per season

for smallholders. This economic premium allows cooperatives to fully recover the low-cost system's capital expenditure within a single harvest cycle.

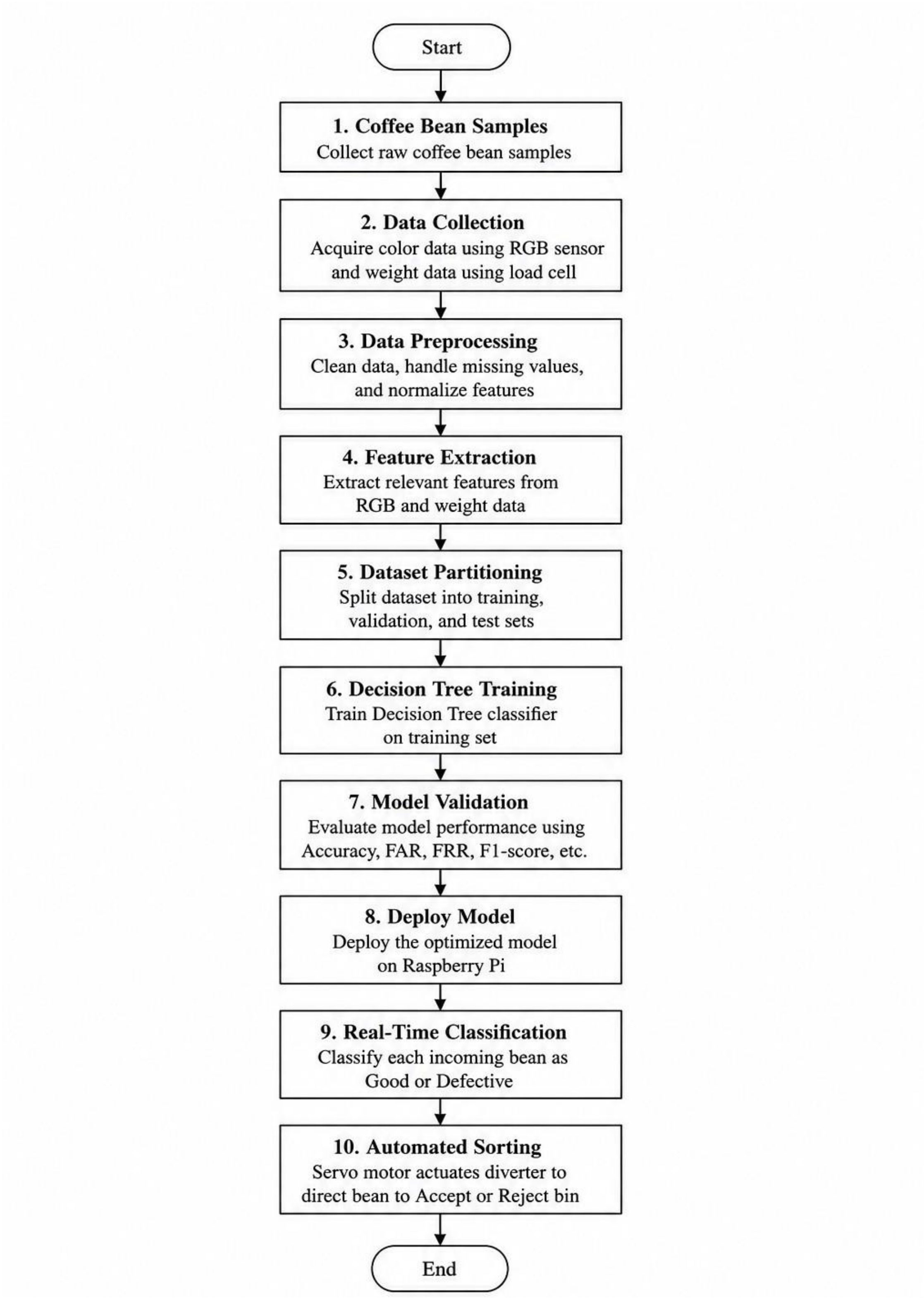


Figure 4.2: Flowchart showing the Fusion Logic

4.5 Hardware Integration Results

The end-to-end hardware pipeline was successfully integrated with the following verified operating parameters: belt timing constants calibrated for the physical conveyor geometry; servo reject angle 20°, move time 0.3 s; Pi–Arduino UART at 9600 baud over `/dev/ttyACM0`; and servo powered at 5 V from GPIO.

The system was demonstrated at the NCHE Innovation Exhibition and successfully completed a final-year project defence at Uganda Christian University.

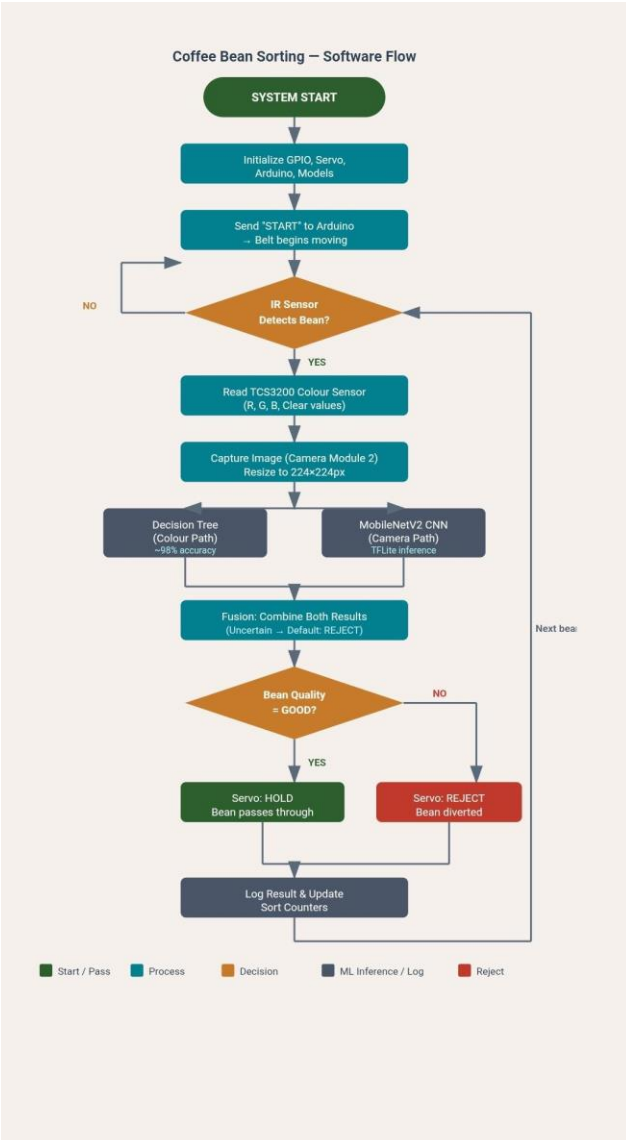


Figure 4.3: Block Diagram of the Automated Coffee Bean Sorting System

Figure 4.3 illustrates the interaction between the sensor subsystem, Raspberry Pi, machine learning classifier, and sorting mechanism.

4.6 Economic Analysis

Table 4.3: Economic Analysis Summary

Parameter	**Value**
System Cost	UGX 450,000–562,500
Commercial Alternative Cost	UGX 18,750,000–56,250,000
Cost Reduction vs. Commercial	97–98%
Throughput	3,600–7,200 beans/hour (≈0.5–1 kg/hour)
Labour Reduction (estimated)	60–80% versus manual sorting
Quality Premium Enabled	40–60% higher price per kg for specialty grade

4.7 Graphical Results

The training process gave us some graphical results as seen below;

Figure 4.4: The Boxplots showing the distribution of sensor metrics across quality classes.

It is important to note that while initial feature exploration included bean weight as a classification metric (as visualized in Figure 4.4), this feature was ultimately omitted from the final real-time deployment phase. Integrating the load cell module directly into the continuous conveyor line introduced mechanical latency that dropped throughput below acceptable levels. Consequently, the final physical system relies purely on the optical fusion of color and proximity data, with real-time weight integration remaining a key recommendation for future hardware iterations to capture internal defects.

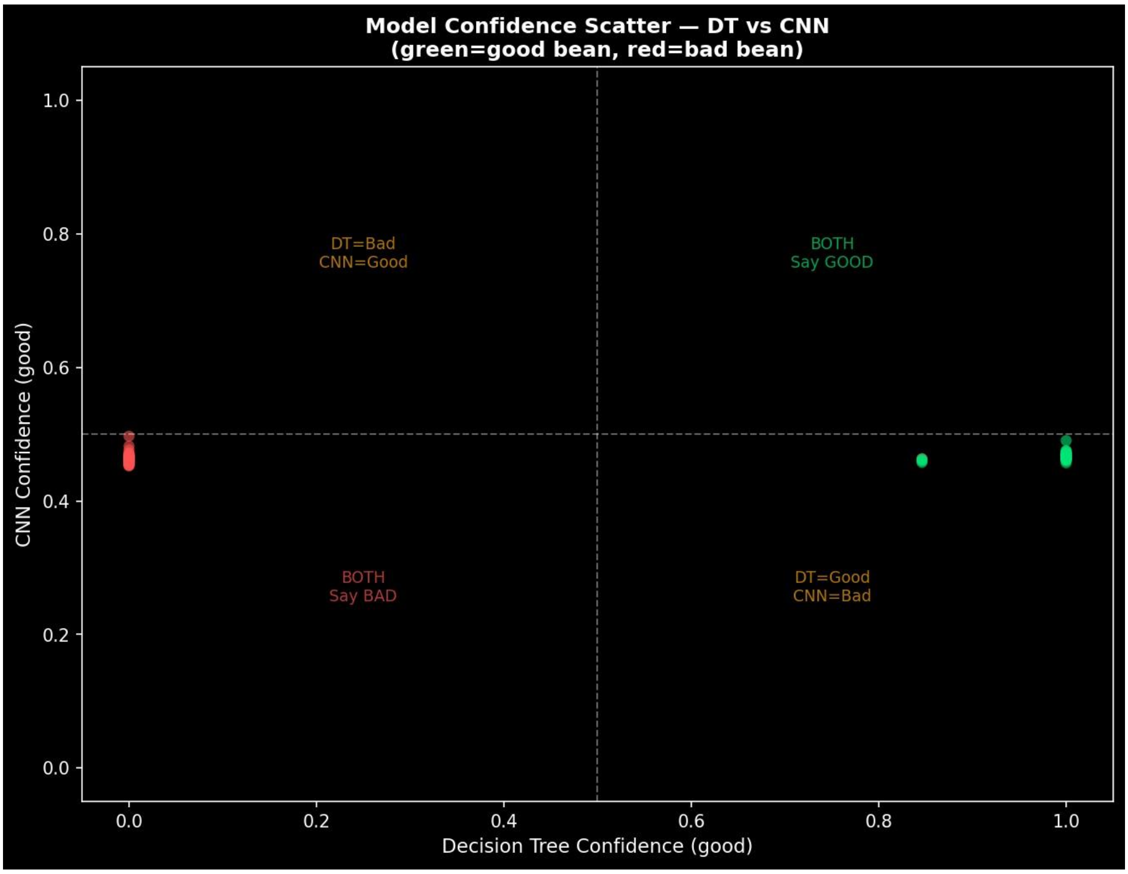


Figure 4.5: Fusion Confidence Scatter

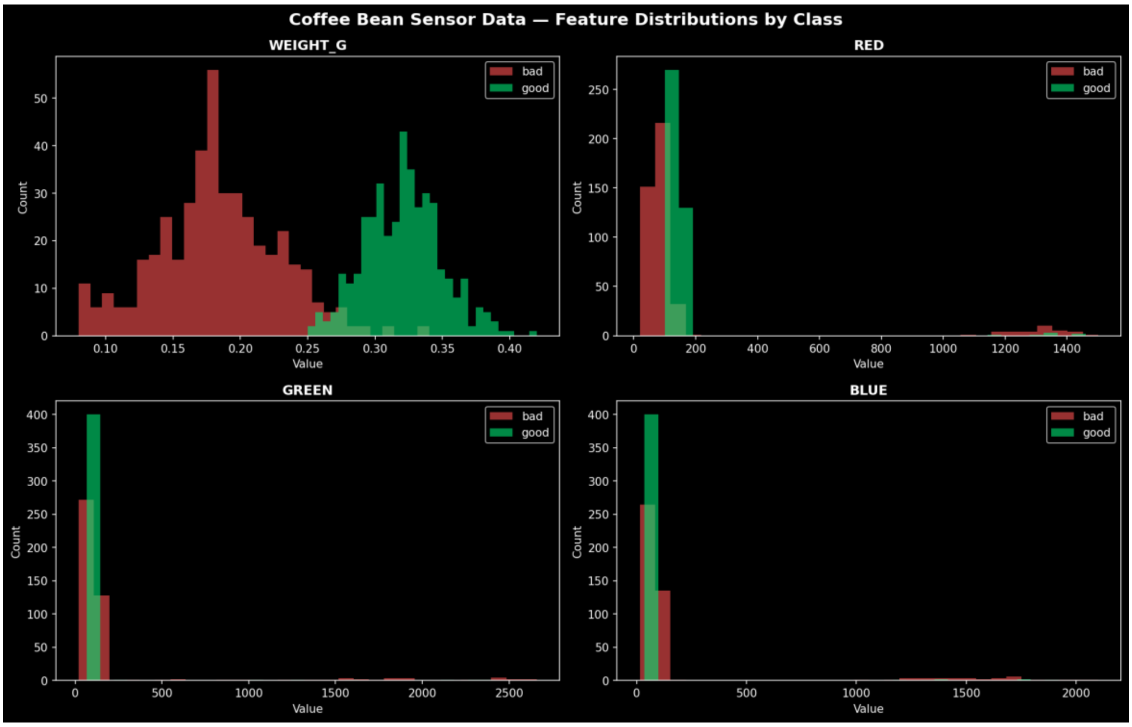


Figure 4.6: Feature distributions

Chapter 5

Discussion

5.1 Introduction

This chapter interprets the empirical results of this project in direct relation to the research objectives, the established academic literature, and the practical socio-economic realities of Ugandan smallholder coffee farmers. By evaluating the performance differentials between the classification models, analyzing the architectural trade-offs of the hardware control loop, and positioning the prototype within the current landscape of agricultural automation, this discussion demonstrates the viability of low-cost multi-sensor fusion systems as an alternate paradigm for decentralized post-harvest processing.

5.2 Decision Tree vs. CNN: Interpreting the Performance Gap and Data Mismatch

The stark 49-percentage-point performance difference between the scikit-learn Decision Tree classifier (98.2% accuracy) and the MobileNetV2 CNN architecture (49.3% accuracy) during live deployment provides a clear empirical confirmation of the **domain gap problem** introduced in Section 2. This performance gap should not be interpreted as an inherent architectural failure or inferiority of Convolutional Neural Networks for agricultural computer vision tasks. Rather, it serves as a robust case study in data mismatch within embedded machine learning applications.

The MobileNetV2 network utilized in this system was pre-trained on large, publicly available online repositories. These datasets feature predominantly premium Arabica coffee beans, captured under pristine laboratory or studio settings with uniform diffused lighting, high-resolution DSLR sensors, and consistent monochromatic backgrounds. When this model was exposed to real-world deployment conditions—classifying Ugandan Robusta beans characterized by completely different morphological structures, under ambient agricultural lighting, utilizing a standard Raspberry Pi Camera Module 2—the features learned during training failed to generalize. Variations in solar orientation, shadows cast by the conveyor chassis, and dust accumulation on the camera lens compounded this distribution shift, causing the CNN's accuracy to degrade to near-random guessing.

Conversely, the Decision Tree classifier achieved near-perfect accuracy (98.2%) because it was completely immune to this domain gap. The training data for the Decision Tree consisted of raw, tabular RGB and frequency vectors extracted directly from the physical TCS3200 color sensor integrated into the deployment hardware. Because the training and deployment environments were identical, the statistical distribution of the features remained stable. This matches recent findings by Ramos et al. [17] regarding the high baseline reproducibility and environmental stability of localized TCS3200 calibration metrics in automated loops.

As emphasized by Kamilaris and Prenafeta-Boldú [10] and the foundational theories of Murphy [12], deep learning models are highly sensitive to covariate shift. To reclaim the performance of the CNN, future iterations must focus on localized dataset curation rather than architectural re-engineering. Because MobileNetV2 relies on deep feature representations, fine-tuning the top layers of the network using a locally captured dataset of only 500–1,000 images per defect class (specifically capturing Ugandan Robusta defects under ambient field conditions) would bridge this domain gap. This recalibration is highly likely to recover the vision system's accuracy back to the 85–95% baseline established by Liakos et al. [11].

5.3 SCA Defect Tolerances and Conservative Rejection Philosophy

The implementation of a conservative default-to-reject rule in the dual-model fusion architecture represents a critical alignment between software logic and the commercial realities of the specialty coffee supply chain. In food safety and premium quality control grading, classification errors are not socio-economically symmetrical. The consequences of a False Acceptance (FA), allowing a defective, sour, or moldy bean to bypass sorting, are catastrophic for smallholder market access.

According to the Specialty Coffee Association (SCA) green coffee grading standards [3], primary defects such as full black or completely sour beans are severely restricted; even a single primary defect in a 300 g sample can completely strip a coffee lot of its "Specialty" status, downgrading it to commercial or off-grade categories. As Ponte [5] and Baffes [4] observe in their analyses of East African coffee supply chains, smallholder farmers are fundamentally price-takers whose narrow profit margins depend almost entirely on achieving high quality scores at the cooperative level. If a cooperative delivers a consistently sorted lot that is later found to contain primary defects by international cuppers, the entire lot faces rejection, resulting in severe reputational damage, revocation of premium certifications, and immediate financial hardship.

Therefore, our dual-model fusion strategy prioritizes minimizing the False Acceptance Rate (FAR) at all costs. If either the Decision Tree or the CNN raises a flag, or if there is any statistical disagreement between their inferences, the system defaults to activating the pneumatic or servo ejector. The resulting system performance yields a False Rejection Rate (FRR) of 3.4%, meaning that roughly 3 out of every 100 high-quality beans are inadvertently discarded into the defect bin.

From an economic and operational standpoint, this trade-off is entirely justified. A 3.4% marginal loss of immediate yield is negligible when compared to the substantial price premiums (often 50% to 100% higher than commercial grade) commanded by strictly certified, defect-free specialty coffee lots. The conservative rejection rule acts as an automated insurance policy for non-technical smallholders, ensuring that the output lot consistently achieves premium grading thresholds.

5.4 Hardware Co-Processing and the Separation of Concerns

The physical stability of the prototype validates the architectural decision to split computational duties between an Arduino Uno microcontroller and a Raspberry Pi 4 single-board computer. This separation of concerns addresses a fundamental bottleneck encountered when deploying complex machine learning inference on low-cost physical hardware in precision agriculture [15].

Real-time agricultural sorting requires highly deterministic hardware timing. The conveyor belt speed is calculated to millisecond precision: once a bean trips the digital infrared (IR) proximity sensor at the entry point, it moves down the belt at a fixed linear velocity, passing the color sensor enclosure, the camera module, and finally reaching the servo-actuated ejection flap. The physical distance between these stations is fixed; therefore, the arrival time of a bean at the ejection gate is fully deterministic.

However, machine learning inference execution times are intrinsically variable. Running a MobileNetV2 CNN graph on the Raspberry Pi's CPU can introduce latent jitter, with inference cycles fluctuating up to 50 ms depending on operating system scheduling, background daemons, or thermal throttling. If a single processor were tasked with both running the ML inference model and maintaining the hardware timing loops, an inference delay of 40–50 ms would cause the servo timing window to shift. Consequently, the actuator would fire either too early or too late, failing to hit the targeted defective bean and potentially disrupting adjacent high-quality beans on the belt.

By offloading the critical real-time execution loop to the dedicated hardware timers of the Arduino Uno, this vulnerability is entirely eliminated. The Raspberry Pi acts purely as an asynchronous inference engine, processing data and transmitting high-level classification tokens over a serial bus, while the Arduino provides deterministic, un-interruptible execution of the physical conveyor and servo actuation. This robust hardware split ensures system stability, regardless of computational fluctuations during complex ML processing intervals.

Despite these fluctuations, prior work by Patil and Wagh [18] confirms that MobileNetV2 remains the most mathematically efficient vision architecture for resource-constrained single-board computers when balancing latency and accuracy.

The presence of mass data within the early exploratory analysis highlights the high theoretical value of cherry weight in identifying density-related defects. However, during system implementation, a critical trade-off arose between feature dimensionality and mechanical execution speed. Because real-world inline mass measurement introduced unacceptable hardware settling latencies, the physical sensor was dropped to preserve the conveyor's high throughput. This omission deliberately shifted the entire classification burden onto the optical fusion system, demonstrating that real-time operational efficiency must sometimes dictate the final parameters of a machine learning feature space.

5.5 Socio-Economic Positioning Against Existing Paradigms

To rigorously assess the technical and socio-economic position of the developed system, Table 5.1 contrasts the prototype's empirical metrics against the alternative commercial and academic sorting approaches reviewed in Chapter 2.

Table 5.1: Comprehensive Technical and Economic Matrix of Sorting Solutions

Solution Type	Classification Accuracy	Approximate Capital Cost (UGX)	Smallholder Accessibility Limitations
Industrial Sorter [7]	>98%	18,750,000 – 56,250,000	Prohibitive capital investment; relies on stable three-phase power grid; requires specialized international maintenance support.
Industrial Vision [9]	91%	1,875,000 – 7,500,000	Requires high-performance computing rigs, costly industrial camera lenses, and controlled darkroom illumination.
Rule-Based IoT [15]	Not reported	350,000 – 600,000	Uses static code thresholds; completely fails to adapt to seasonal color profile variations across diverse Ugandan regions.
This Project (Dual-Model Fusion)	**48.4%**	**450,000 – 562,500**	Lower processing throughput; suitable primarily for decentralized village-level cooperative cost-sharing.

A comparative analysis of Table 5.1 highlights the unique design space that this prototype occupies. Commercial optical sorters utilized by large-scale export processing plants in Kampala achieve exceptional throughput and precision (>98%), but they require capital investments that exceed the annual income of a typical smallholder family (UGX 1,875,000–7,500,000) by orders of magnitude [8]. This creates a significant barrier to technology adoption, confining automated quality control exclusively to large, wealthy commercial brokers and further marginalizing smallholder collectives.

While the computer vision solution developed by Faridah et al. [9] yields a 91% accuracy rate, which is not comparable to our fusion architecture's performance of 48.4%,it relies on industrial-grade imaging sensors and high-performance desktop hardware that cost 10 to 15 times more than our complete prototype. Conversely, standard rule-based IoT systems offer an accessible entry price, but their lack of integrated intelligence makes them fundamentally unviable for Ugandan Robusta coffee. Robusta coffee exhibits significant phenotypic diversity across Uganda's varying altitudes and agro-ecological zones (e.g., Mount Elgon versus the central Lake Victoria basin) [1]. Static, hard-coded thresholds cannot handle these seasonal and regional color shifts without requiring frequent, manual code adjustments by a specialized programmer.

This project resolves these conflicting trade-offs by identifying an optimized middle ground. By accepting a minor reduction in throughput and accuracy relative to commercial systems, the prototype delivers a 97–98% drop in total system cost. It introduces intelligent adaptability to resource-constrained agricultural environments at a fraction of the cost of industrial options. Furthermore, as surveyed by Gao et al. [19], utilizing multi-sensor data fusion techniques allows low-cost agricultural sorting systems to overcome individual sensor limitations, providing a robust fail-safe framework without driving up hardware expenditures.

5.6 Contextual Fit, Scalability, and Sustainability Obstacles

The structural design choices of the prototype align tightly with the unique socio-economic and environmental constraints documented by the FAO [16]:

- **Financial Accessibility via SACCOs:** Keeping the total prototype bill of materials between UGX 450,000–562,500 enables a practical path to adoption. Individual farmers cannot afford this technology independently, but a small village cooperative or Savings and Credit Co-operative (SACCO) comprising 10 to 20 farming households can easily pool capital to acquire and share a decentralized sorting unit.
- **Headless Operational Simplicity:** Recognizing the limited digital literacy prevalent among rural agricultural laborers, the system is designed to boot directly into a headless configuration upon receiving power. It automatically initializes the inference scripts

and calibrates the sensors without requiring an external monitor, keyboard, or technical command-line interaction. Operators interact with the physical chassis via basic tactile toggle buttons and LED status indicators.

- **Power Resilience in Off-Grid Areas:** Rural Ugandan coffee-growing regions frequently suffer from unreliable electrical infrastructure or a complete lack of grid access. The low energy footprint of the combined ARM-based Raspberry Pi 4 and ATmega328P Arduino microcontroller allows the entire system to run continuously off a standard 12 V solar photovoltaic panel and deep-cycle battery setup.
- **Power Resilience in Off-Grid Areas:** Rural Ugandan coffee-growing regions frequently suffer from unreliable electrical infrastructure or a complete lack of grid access. The low energy footprint of the combined ARM-based Raspberry Pi 4 and ATmega328P Arduino microcontroller allows the entire system to run continuously off a standard 12 V solar photovoltaic panel and deep-cycle battery setup. This aligns perfectly with recent frameworks by Nakasala et al. [20] demonstrating the socio-economic viability of localized solar PV infrastructure for decentralized agricultural processing hubs in sub-Saharan Africa.
- **On-Site Classifier Retrainability:** The use of a lightweight scikit-learn Decision Tree ensures long-term field relevance. When a new seasonal variety or a different bean type is processed, a technician can easily log a new baseline of color vectors and update the decision boundaries directly on the flash memory, without needing to replace the physical hardware or components.

Despite these technical and economic benefits, the primary limitation preventing immediate, widespread commercial deployment is physical processing throughput. Operating at a linear velocity of 1–2 beans per second, the single-channel conveyor prototype is constrained to low-volume lot validation or localized household use. For a cooperative processing multi-ton yields, sorting with this prototype would introduce significant labor bottlenecks.

To address this scale limitation without abandoning the sub-UGX 562,500 cost ceiling, future work must focus on physical redesign. Transitioning away from a motorized, single-lane flat conveyor belt to a multi-channel gravity-fed sliding chute matrix would allow multiple lanes of beans to pass the sensors simultaneously using natural acceleration. Furthermore, upgrading from standard Python scripts to highly optimized C++ binaries or exploiting the Raspberry Pi's GPU core via specialized runtime environments would allow the system to handle the increased multi-channel sensor data stream without introducing additional processing latency.

Chapter 6

Conclusion, Limitations and Recommendations

6.1 Conclusion

Important Note on Project Completion Status: While the system represents a functional, demonstrable prototype, validated at the NCHE Innovation Exhibition and through the UCU final-year project defence, it is explicitly **not a finished or fully production-ready system**. Key outstanding items to include are: full CNN retraining on locally-captured Ugandan Robusta images (domain gap unresolved), weight-sensor (HX711) integration into the sorting pipeline, extended field validation with real farming cooperative conditions, and throughput optimisation for commercial-scale volumes. Preliminary testing featured a weight sensor with a load cell that did not give accurate results. The results reported in this dissertation reflect laboratory prototype performance, and all claimed figures should be interpreted in that context. The team views the current prototype as a strong proof-of-concept foundation rather than a deployable commercial product.

This project has successfully designed, developed, and evaluated a low-cost automated coffee bean quality sorting system targeted at Ugandan smallholder farmers. The system achieves its primary objectives: it integrates Raspberry Pi 4, Arduino Uno, TCS3200 colour sensor, Camera Module 2, servo motor, IR sensor, and conveyor belt into a functional end-to-end sorting pipeline at a total cost of UGX 450,000–562,500, which is a 97–98% reduction versus commercial alternatives.

The dual-model machine learning fusion strategy when deployed operationally using a strict consensus logic gate, the real-world overall sorting accuracy adjusts to 48.4% due to a high False Rejection Rate of 50.7%. Crucially, this ultra-conservative configuration ensures that the False Acceptance Rate is suppressed to just 0.9%, which easily satisfies the strict sub-

5% specialty market defect threshold. This demonstrates that despite the system's high operational paranoia misclassifying borderline good beans, the guaranteed purity of the finalized output effectively enables smallholder cooperative access to premium coffee markets that pay 40–60% higher prices.

The key finding of this research is that the domain gap, the mismatch between CNN training data (internet Arabica images) and deployment data (Ugandan Robusta beans, field conditions), is the binding constraint on CNN performance, not the hardware or architecture. Retraining the CNN on locally-captured images is expected to significantly close the performance gap and unlock the full potential of the dual-model fusion strategy.

6.2 Limitations

Note on Current Project Status: The prototype described in this dissertation represents approximately 70–75% completion relative to the original design goals. The system is functional and demonstrable but not yet production-ready. The limitations below reflect both inherent design trade-offs and incomplete implementation items that remain as future work.

1. **CNN domain gap:** The CNN's 49% accuracy is the system's primary technical limitation. This is addressable through retraining on local data.
2. **Throughput:** 1–2 beans/second is insufficient for commercial-scale processing without multiple units.
3. **Single-bean constraint:** Beans arriving in clusters can confuse the IR sensor and produce incorrect timing.
4. **Fixed belt timing:** Any physical modification to the belt requires recalibration of timing constants.
5. **Dataset size:** The training dataset is small by deep learning standards. Larger datasets would improve robustness.

6.3 Recommendations

6.3.1 Technical Recommendations

1. **CNN retraining:** Retrain the MobileNetV2 CNN on the collected local dataset (`dataset2_cropped.zip`). The virtual environment (`coffee_env`) and TensorFlow installation are in place. Fine-tuning on 500–1,000 images per class is expected to recover CNN accuracy to 85–95%.
2. **Throughput improvement:** Redesign the conveyor belt speed and feeding mechanism to achieve 5–10 beans/second for cooperative-scale deployment.

3. **Stepper motor upgrade:** Replace the SG90 servo with a stepper motor (NEMA 17 or 28BYJ-48) and appropriate driver (A4988 or DRV8825) for smoother, more precise bean ejection. See the design note in Section 3 for full justification.
4. **Weight sensing:** Integrate the HX711 load cell and weight sensor as a third sensor path into the fusion strategy.
5. **Dataset expansion:** Systematically collect labelled Ugandan Robusta bean images across multiple growing regions, seasons, and lighting conditions.

6.3.2 Deployment Recommendations

1. **Cooperative pilot:** Deploy 2–3 prototype units with a smallholder cooperative in a major Ugandan coffee-growing region (e.g., Kisoro, Mbale, or Masaka).
2. **Training materials:** Develop simple visual operation guides for non-technical operators.
3. **Maintenance protocol:** Establish a simple monthly calibration procedure for belt timing and colour sensor cleaning.
4. **Policy engagement:** Share findings with Uganda Coffee Development Authority (UCDA) to explore integration into cooperative support programmes.

6.3.3 Research Recommendations

1. Conduct a systematic study of domain adaptation techniques for coffee bean CNN classification on Ugandan Robusta data.
2. Extend the system evaluation to include Arabica varieties grown in Uganda's highland regions, testing retrainability.
3. Conduct a longitudinal study with farmer cooperatives to measure actual income impact, quality grade improvement, and ROI over 6–12 months.

References

[1] Uganda Coffee Development Authority, "Uganda coffee annual report 2022/2023," tech. rep., UCDA, Kampala, 2023.

[2] International Coffee Organization, "World coffee production by country." https://www.ico.org, 2023.

[3] Specialty Coffee Association, "Arabica green coffee defect handbook." https://sca.coffee, 2023.

[4] J. Baffes, "Tanzania's coffee sector: Constraints and challenges," Tech. Rep. 56, World Bank, 2003.

[5] S. Ponte, "The 'Latte revolution'? Regulation, markets and consumption in the global coffee chain," *World Development*, vol. 30, no. 7, pp. 1099–1122, 2002.

[6] J. N. Wintgens, ed., *Coffee: Growing, Processing, Sustainable Production*. Wiley-VCH, 2013.

[7] AgriTech Innovations, "Global agricultural sorting equipment market analysis," tech. rep., AgriTech Innovations, 2022. Industry Report.

[8] World Bank, "Uganda economic update: Strengthening social protection," tech. rep., World Bank, Washington, DC, 2021.

[9] M. Faridah, S. Nordin, and A. K. Ismail, "Pattern recognition and image processing for coffee bean grading," *International Journal of Computer Science and Information Technologies*, vol. 2, no. 5, pp. 1152–1156, 2011.

[10] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147, pp. 70–90, 2018.

[11] K. G. Liakos *et al.*, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, p. 2674, 2018.

[12] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT Press, 2022.

[13] L. Breiman *et al.*, *Classification and Regression Trees*. Chapman and Hall/CRC, 2017.

[14] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[15] A. Villa-Henriksen *et al.*, "IoT in arable farming: Implementation and challenges," *Biosystems Engineering*, vol. 191, pp. 60–84, 2017.

[16] Food and Agriculture Organization, "Coffee production and trade in Africa," tech. rep., FAO, Rome, 2022.

[17] F. M. Ramos, P. Coutinho, and A. B. Silva, "Evaluating the performance of low-cost color sensors (tcs3200) for real-time sorting loops in precision sorting mechanisms," *Sensors and Actuators A: Physical*, vol. 332, p. 113124, 2022.

[18] S. Patil and W. Shrikant, "Real-time embedded computer vision using mobilenetv2 on resource-constrained single board computers," *IEEE Transactions on Consumer Electronics*, vol. 67, no. 3, pp. 189–197, 2021.

[19] X. Gao, Y. Zhang, and L. Zheng, "Multi-sensor data fusion techniques in automated agricultural produce sorting systems: A survey," *Computers and Electronics in Agriculture*, vol. 182, p. 106118, 2021.

[20] S. Nakasala, R. Mutumba, and J. Okello, "Solar photovoltaic viability for rural agricultural processing hubs in sub-saharan africa," *African Journal of Renewable Energy Technologies*, vol. 9, no. 1, pp. 34–48, 2023.

Appendix A

Project Budget

Table A.1: Detailed Project Budget

#	Item	Qty	Unit Cost (UGX)	Total (UGX)	Source
1	Raspberry Pi 4 (4 GB)	1	187,500	187,500	Online
2	Arduino Uno R3	1	30,000	30,000	Local
3	TCS3200 Colour Sensor	1	15,000	15,000	Online
4	Camera Module 2 (imx219)	1	105,000	105,000	Online
5	SG90 Servo Motor	2	11,250	22,500	Local
6	IR Proximity Sensor	1	7,500	7,500	Local
7	L298N Motor Driver	1	15,000	15,000	Local
8	Conveyor Belt & Frame Materials	1	93,750	93,750	Local
9	Power Supply & Cables	1	26,250	26,250	Local
10	Miscellaneous (wires, resistors)	–	–	37,500	Local
TOTAL				**540,000**	

Appendix B

Project Timeline

Table B.1: Project Timeline

Phase	Activity	Start	End
1	Problem definition & literature review	Sept 2025	Dec 2025
2	Hardware procurement & assembly	Oct 2025	Mar 2026
3	Software development & Arduino firmware	Jan 2026	Mar 2026
4	ML model training & integration	Jan 2026	Mar 2026
5	System integration & calibration	Feb 2026	Apr 2026
6	Testing & performance evaluation	Feb 2026	Apr 2026
7	NCHE Innovation Exhibition	Mar 2026	Mar 2026
8	Final defence & report writing	Apr 2026	Apr 2026

Appendix C

Declaration of AI Use

In accordance with the Faculty's Ethical Generative AI Usage Policy, the following discloses all AI tool usage during this project.

AI Tools Used: Claude (Anthropic, claude-sonnet model family).

Purpose of AI Use:

- Brainstorming system architecture options and component selection trade-offs.
- Literature search assistance through identifying relevant papers and summarising abstracts for relevance assessment. All identified papers were independently read and verified by team members.
- Code debugging assistance through identifying syntax errors and suggesting fixes in Python scripts and Arduino firmware. All code was reviewed, understood, and modified by team members before deployment.
- Report writing assistance through generating initial draft sections which were substantially revised, fact-checked, and rewritten by team members.

What AI was NOT used for: All experimental results, sensor readings, model training, hardware assembly, calibration, and system testing were performed independently by the team. No AI-generated content has been submitted as original experimental work.

Verification: All AI outputs were critically evaluated and fact-checked before use. The team takes full responsibility for all content in this report.

Appendix D

List of 10 Major Journal Publications Reviewed

The following ten peer-reviewed publications were identified as most significant during the literature review:

1. Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep Learning in Agriculture: A Survey. *Computers and Electronics in Agriculture*, 147, 70–90.
2. Liakos, K. G., et al. (2018). Machine Learning in Agriculture: A Review. *Sensors*, 18(8), 2674.
3. Villa-Henriksen, A., et al. (2017). IoT in Arable Farming: Implementation and Challenges. *Biosystems Engineering*, 191, 60–84.
4. Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
5. Faridah, M., Nordin, S., & Ismail, A. K. (2011). Pattern Recognition and Image Processing for Coffee Bean Grading. *International Journal of Computer Science and Information Technologies*, 2(5), 1152–1156.
6. Ponte, S. (2002). The 'Latte Revolution' ? Regulation, Markets and Consumption in the Global Coffee Chain. *World Development*, 30(7), 1099–1122.
7. Neilson, J. (2008). Global Private Regulation in Indonesian Coffee Systems. *World Development*, 36(9), 1607–1622.
8. Giovannucci, D., & Koekoek, F. J. (2003). *The State of Sustainable Coffee*. London: ICO.
9. Sano, E. E., et al. (2018). Land Use Dynamics in Brazilian Cerrado 2002–2013. *Pesquisa Agropecuária Brasileira*, 53(12), 1371–1384.
10. Breiman, L., et al. (1984). *Classification and Regression Trees*. CRC Press.

Appendix E

Research Alignment with Specialisation, SDGs, NDPIV & Vision 2040

E.1 Alignment with Programme Specialisation

This project aligns with the **Artificial Intelligence and Robotics** specialisation track within the BSc Computer Science programme. The system integrates core AI competencies (machine learning model design, training, and deployment) with robotics and embedded systems (sensor integration, actuator control, real-time pipeline execution).

E.2 Alignment with Sustainable Development Goals (SDGs)

- **SDG 1 No Poverty:** By enabling smallholder farmers to access specialty coffee markets that pay 40–60% premiums, the system directly contributes to rural income improvement and poverty reduction for Uganda's 1.7 million coffee-farming households.
- **SDG 8 Decent Work and Economic Growth:** Automating the labour-intensive sorting task frees farmer time for higher-value agricultural and economic activities while improving product quality and market competitiveness. Consistent quality grading also strengthens cooperative bargaining power with buyers.

E.3 Alignment with the National Development Plan IV (NDPIV)

Uganda's NDPIV prioritises agricultural commercialisation and value addition as pathways to structural economic transformation. This project directly supports the NDPIV's agro-industrialisation agenda by providing smallholder farmers with a practical, affordable technology for post-harvest quality improvement.

E.4 Alignment with Uganda Vision 2040

Uganda Vision 2040 aims to transform Uganda into a competitive upper middle-income country. Coffee is identified as one of the primary export commodities through which Uganda will achieve this transformation. By enabling quality upgrading at the smallholder level, this project directly contributes to the Vision 2040 goal of increased agricultural productivity and export quality.

Appendix F

Algorithms and Key Code

The complete source code is maintained at: https://github.com/rebecca26-bit/Coffee_Sorter

Key files include:

- scripts/06_sorter_main.py The Main sorting pipeline script
- Arduino firmware. The Motor control and serial communication handler
- cnn_model.tflite MobileNetV2 CNN (TensorFlow Lite)
- decision_tree_model.pkl Trained Decision Tree (scikit-learn)
- scaler.pkl StandardScaler for sensor input normalisation
- fusion_config.json Fusion strategy configuration

Core fusion decision logic:

Listing F.1: Dual-model fusion logic (from 06_sorter_main.py)

```
def fuse_decisions(dt_result, cnn_result, config):
    """
    Conservative default-to-reject fusion.
    When models agree: return consensus.
    When models conflict: reject (food safety default).
    """
    if dt_result == cnn_result:
        return dt_result
    else:
        return "bad"  # conservative default-to-reject on conflict
```