Counterfeit Money Detector

A Proposal Submitted to the department of Computer Science and Engineering

College of Electrical Engineering and Computing
Presented in Partial Fulfillment of the Requirement for Bachelor Degree in Computer Science and Engineering

January 2026
Adama, Ethiopia

DECLARATION

We are students of Adama Science and Technology University in the School of Electrical Engineering and Computing in the Department of Computer Science and Engineering. The information found in this project is our original work. And all sources of materials that will be used for the project work will be fully acknowledged.

NO	Name	ID	Signature
1	Bereket Daniel	UGR/25430/14	
2	Benjamin Endale	UGR/25484/14	
3	Shalom Mesfin	UGR/25453/14	
4	Salem Mesfin	UGR/25407/14	
5	Kaleb Seyfu	UGR/26483/14	

Date of Submission: January 1, 2026,

This project has been submitted for examination with our approval as a university advisor.

Advisor Name	**Signature**
Tagel Aboneh (PhD	__________

TABLE OF CONTENTS

LIST OF TABLES

Tables **Page no**

LIST OF FIGURES

Figures **Page no**

ACRONOMY

Acronym	Full Form	Description/Context in the Project
BOM	Bill of Materials	List of hardware components with sources and costs
CSI	Camera Serial Interface	Interface for connecting Raspberry Pi Camera Module
CSV	Comma-Separated Value	File format used for logging detection results
ETB	Ethiopian Birr	The official currency of Ethiopia
FNR	False Negative Rate	Rate at which counterfeit notes are incorrectly classified as genuine
FPR	False Positive Rate	Rate at which counterfeit notes are incorrectly classified as genuine
GPIO	General Purpose Input/Output	Pins on Raspberry Pi and Arduino for interfacing peripherals
I/O	Input/Output	Refers to hardware interfaces for sensors, LEDs, buttons, etc.
IDE	Integrated Development Environment	Software tool (e.g., Arduino IDE) for writing and uploading code

IR	Infrared	Type of sensor considered for detecting security features
JSON	JavaScript Object Notat	File format used for storing region-of-interest (ROI) data
LED	Light-Emitting Diode	Used for visual feedback (green for genuine, red for counterfeit)
OpenCV	Open Source Computer Vision Library	Primary library used for image processing and computer vision tasks
ORB	Oriented FAST and Rotated BRIEF	Feature detection and description algorithm (alternative to SIFT)
PNG	Portable Network Graphics	Image file format used for storing template edge maps
RAM	Random Access Memo	Type of LED used for multi-color visual feedback
RGB	Red-Green-Blue	Type of LED used for multi-color visual feedback
ROI	Region of Interest	Specific areas on banknotes analyzed for security features
SD	Secure Digital	Type of card used for storage on Raspberry Pi
SIFT	Scale-Invariant Feature Transform	Feature detection algorithm considered for matching

SSIM	Structural Similarity Index Measure	Metric used for template matching and similarity scoring
UART	Universal Asynchronous Receiver/Transmitter	Serial communication protocol between Arduino and Raspberry Pi
USB	Universal Serial Bus	Interface alternative for webcam connection
UV	Ultraviolet	Type of sensor/light for detecting fluorescent security features

ACKNOWLEDGEMENT

We would like to express our deepest gratitude to our project supervisor, Tagel Aboneh, from the Department of Computer Science & Engineering at Adama Science and Technology University, for their invaluable guidance, constant encouragement, and insightful feedback throughout this project. Their expertise in embedded systems and image processing was instrumental in shaping our work.

We are also thankful to the faculty members and staff of the College of Electrical Engineering and Computing for providing the necessary resources and support. Special thanks to our peers and group members for their collaboration and constructive discussions that helped refine our ideas.

We extend our appreciation to the National Bank of Ethiopia for information on currency security features that informed our research. Finally, we acknowledge our families for their unwavering support and patience during the long hours dedicated to this project.

This work would not have been possible without the contributions of all involved.

ABSTRACT

This project proposes the design and planned development of a cost-effective hardware-based counterfeit detection system for Ethiopian banknotes using Arduino and Raspberry Pi technology. The proposed system is intended to integrate image processing algorithms to analyze banknote features such as texture, color, and patterns in real time, enabling classification of notes as genuine or counterfeit. By addressing the limitations of expensive conventional detection methods and manual inspection processes, the proposed prototype seeks to offer an accessible solution for small businesses, law enforcement agencies, and the general public.

The primary objectives of the project include achieving high detection accuracy, low processing latency, and offline operation, while maintaining affordability and ease of use. The system is planned to be developed iteratively, utilizing classical computer vision techniques such as edge detection and template matching implemented through OpenCV. Upon completion, the proposed solution is expected to contribute to improved financial security in Ethiopia by helping reduce the circulation of counterfeit banknotes and by demonstrating the practical application of embedded systems and image processing technologies.

Chapter 1: Introduction

1.1 Introduction

Counterfeiting is a significant global problem, and Ethiopia, like many other countries, faces the challenge of counterfeit currency that closely mimics genuine notes. Despite recent efforts by the National Bank of Ethiopia to introduce new banknotes with advanced security features in 2020, counterfeit banknotes remain a widespread issue. The limitations of existing solutions, such as high-cost counterfeit detection machines and reliance on manual inspection methods, leave a gap in effective and accessible counterfeit detection, particularly for small businesses and individuals.

This project aims to address these challenges by developing a hardware-based counterfeit detection system using Arduino technology ,Raspberry Pi, or Jetson nanoproject. The system will leverage image processing techniques integrated with the Arduino platform and Raspberry Pi to classify Ethiopian banknotes as genuine or counterfeit. This hardware prototype will be designed to be cost-effective, user-friendly, and capable of real-time counterfeit detection, making it an accessible tool for a wide range of users.

The specific objectives of this project are:

- To design and develop a hardware prototype using Arduino technology, and Raspberry Pi for real-time counterfeit detection of Ethiopian banknotes.
- To integrate image processing algorithms within the hardware system to classify banknotes accurately as genuine or counterfeit.
- To provide a cost-effective, accessible counterfeit detection tool for small businesses, law enforcement, and the general public, without relying on expensive hardware-based systems.

The expected outcome of this project is to provide an efficient, reliable, and cost-effective solution for detecting counterfeit Ethiopian currency, helping to reduce the circulation of fake notes and improve financial security.

1.1.1 Technology Clarification and Implementation Flexibility Statement

In the sections that follow, the system design, architecture, methodology, and implementation details are fully described using Arduino and Raspberry Pi technologies as the primary hardware components. This Arduino–Raspberry Pi architecture forms the conceptual and technical foundation of the proposed system throughout this document.

The project team is fully aware that Raspberry Pi alone is capable of performing image acquisition, image processing, and basic hardware control without the need for an Arduino. However, Arduino is intentionally integrated into the proposed system to handle low-level hardware interfacing tasks, including sensor data acquisition, button handling, LED and buzzer control, and other time-critical input/output operations. This design allows the Raspberry Pi to focus on computationally intensive tasks such as image preprocessing, feature extraction, and classification, while Arduino provides deterministic timing, improved I/O reliability, simplified peripheral expansion, and enhanced system stability during continuous operation.

Furthermore, following recent communication with the project advisor, Dr. Tagel Aboneh (PhD), it was suggested that NVIDIA Jetson Nano may be used during implementation, as the hardware is readily available under the advisor’s supervision. Accordingly, Jetson Nano will be considered as an alternative embedded implementation platform, capable of replacing the combined functionality of Arduino and Raspberry Pi if required.

In addition, in the event that embedded processing platforms such as Raspberry Pi or Jetson Nano are unavailable, incompatible, or unsuitable during implementation, the system may be implemented using a personal computer (PC) or laptop as the image-processing unit, while Arduino remains responsible for basic hardware control. In this configuration, the PC will execute image acquisition, preprocessing, feature extraction, and classification algorithms, whereas Arduino will continue to manage sensors, user inputs, and output devices, preserving the embedded system behavior.

All system descriptions, diagrams, and explanations presented below remain based on the Arduino–Raspberry Pi architecture, and any alternative implementation using Jetson Nano or a PC will maintain the same functional logic, system objectives, and evaluation criteria defined in this proposal.

1.2. Background of the Project

Counterfeiting is a pervasive issue that negatively affects economies worldwide, and Ethiopia is no exception. In Ethiopia, the circulation of counterfeit banknotes undermines public trust in the currency, causes financial losses, and complicates the daily operations of businesses and financial institutions. The Ethiopian government has made efforts to curb counterfeiting, particularly by introducing new currency notes in 2020 with advanced security features, such as holograms and color-shifting ink. Despite these advancements, counterfeiters have increasingly used sophisticated digital printing techniques to produce fake notes that closely resemble the genuine currency, making it difficult for non-experts to identify counterfeit bills.

Current solutions for detecting counterfeit banknotes in Ethiopia mainly rely on hardware-based systems that use UV light, magnetic ink detection, and security threads. While these systems are effective to a certain extent, they are expensive and not widely accessible, particularly for small businesses and the general public. Additionally, these systems require manual inspection, which is prone to human error, especially when counterfeit notes are of high quality or when the banknotes are worn or damaged.

Despite these existing solutions, there remains a significant gap in the availability of affordable and automated counterfeit detection systems that can be used easily by businesses, law enforcement, and everyday users. Traditional counterfeit detection methods are often limited by their cost, size, and the need for manual intervention. As a result, there is a growing need for a more accessible, cost-effective, and automated system that can reliably detect counterfeit Ethiopian banknotes without requiring expensive hardware.

This project aims to address this gap by creating a hardware prototype using Arduino and Raspberry Pi technology for counterfeit detection. The prototype will integrate image processing algorithms to analyze images of banknotes in real-time, identifying counterfeit currency with high accuracy. By using a hardware-based solution, this project offers a low-cost alternative to traditional counterfeit detection machines, making it more accessible to small businesses, local markets, and individuals. Moreover, the use of image processing will enable the system to continuously improve its accuracy and reliability, overcoming the limitations of traditional methods.

By focusing on developing a cost-effective, real-time solution using widely accessible Arduino and Raspberry Pi technology, this project aims to bridge the gap between expensive, complex counterfeit detection systems and the growing need for affordable, practical solutions that can be deployed on a larger scale.

1.3. Statement of the Problem

Despite the Ethiopian government’s efforts to introduce new banknotes in 2020 with advanced security features, counterfeiters continue to produce fake notes that are increasingly difficult to differentiate from the genuine currency. Current methods of counterfeit detection, such as UV light scanners and counterfeit pens, are limited in their effectiveness, as they are prone to human error and only detect specific security features.

These traditional solutions are also expensive and not accessible to small businesses or individuals, leaving a large portion of the population without reliable means to detect counterfeit currency. Small businesses, in particular, are vulnerable to accepting counterfeit banknotes, often without realizing it until after the transaction. Furthermore, the manual inspection of banknotes by individuals, especially when the notes are worn or damaged, is an inefficient and error-prone process.

While hardware-based counterfeit detection machines exist, they are costly, typically used by large financial institutions, and require specialized training for proper usage.

This limits their widespread adoption, especially in rural areas or among small vendors who lack the resources to invest in such technologies.

There is a clear need for a cost-effective, automated counterfeit detection solution that can be used by individuals, small businesses, and law enforcement to quickly and accurately identify counterfeit Ethiopian banknotes. The current lack of accessible solutions forces businesses to rely on manual, subjective inspection or expensive, specialized machines.

This project aims to fill this gap by developing a hardware-based prototype using Arduino and Raspberry Pi technology for real-time counterfeit detection of Ethiopian currency. The proposed solution will leverage image processing to automatically classify banknotes as genuine or counterfeit, making it more efficient, reliable, and accessible for everyday use.

The research questions guiding this project are:

1. How can image processing techniques be integrated into a hardware prototype for effective counterfeit detection of Ethiopian banknotes?
2. What is the performance of a Raspberry Pi-based system in detecting counterfeit banknotes in real-time?
3. How can the hardware prototype be optimized for low-cost, widespread adoption without sacrificing detection accuracy?
4. Can this hardware solution be used effectively by small businesses, law enforcement, and the general public to prevent the circulation of counterfeit currency?

The problem this project seeks to solve is the accessibility and reliability of counterfeit detection systems, providing a practical and affordable solution for individuals and businesses across Ethiopia.

1.4. Objective of the Project

1.4.1 General Objectives

The general objective of this project is to design and develop a hardware prototype using Arduino and Raspberry Pi technology that can accurately detect counterfeit Ethiopian banknotes in real-time. The prototype will leverage image processing techniques to classify banknotes as genuine or counterfeit and will be optimized for cost-effectiveness and user-friendliness to make it accessible to small businesses, law enforcement, and the general public.

1.4.2 Specific Objectives

To achieve the general objective, the following specific objectives are outlined:

1. To Design and Develop a Hardware-Based Counterfeit Detection System
    - To create a physical hardware prototype using Raspberry Pi that integrates image processing algorithms to classify Ethiopian banknotes accurately as genuine or counterfeit.
2. To integrate image processing algorithms into the Raspberry Pi system, enabling it to capture banknote images, analyze key security features (such as texture, color, and patterns), and classify the note as genuine or counterfeit in real time.”
3. To Ensure Cost-Effectiveness and Accessibility
    - To design the system to be affordable and easy to use, ensuring that it can be widely adopted by small businesses, local markets, and law enforcement agencies without the need for expensive equipment.
4. To Optimize the System’s Performance for Real-Time Detection
    - To optimize the hardware and image processing algorithms to process banknote images efficiently, ensuring low latency and high accuracy in detecting counterfeit notes during regular use.
5. To evaluate the system’s effectiveness and reliability by measuring key performance metrics—including accuracy, precision, recall, specificity, F1-score,

detection speed (latency), and false negative rate—to ensure the hardware prototype consistently provides accurate and dependable counterfeit detection results

These objectives are directly aligned with the problem identified in the Statement of Problem, where existing solutions for counterfeit detection are either too expensive or inefficient. By developing a hardware prototype with image processing capabilities, this project seeks to provide a cost-effective, real-time, and easily accessible solution for counterfeit detection. Each specific objective addresses the core issue of inaccessible counterfeit detection methods by providing an innovative hardware-based solution for everyday use, thus fulfilling the need for reliable counterfeit detection in Ethiopia.

1.5. Scope and Limitation

1.5.1. Scope of the Project

This project focuses on developing a hardware-based counterfeit detection system using Arduino technology and image processing techniques to accurately classify Ethiopian banknotes as genuine or counterfeit. The scope of the project includes the following key areas:

- Target Users: The primary users of this system will include small businesses, law enforcement, and the general public in Ethiopia. The system will provide a cost-effective and easy-to-use tool for identifying counterfeit banknotes without relying on expensive machines or manual inspection.
- Functional Requirements:
    - The system shall capture banknote images using a Raspberry Pi camera, or a web camera.
    - The system shall process the captured image to analyze visual features such as texture and patterns.
    - The system shall classify each banknote as genuine or counterfeit based on image-processing and sensor data.

    - The system shall operate offline without requiring internet connectivity.
- Non-Functional Requirements:
    - The system shall operate with low latency to support real-time counterfeit detection.
    - The prototype shall be portable and easy to use, requiring minimal technical expertise
    - The system shall be cost-effective for small businesses and everyday users.
    - The system shall be reliable and stable during continuous operation. The image processing shall be efficient to ensure smooth performance.

- **System Architecture:** The proposed system will be built around a Raspberry Pi and an Arduino board with clearly defined roles. The Raspberry Pi will serve as the primary processing unit, interfacing with a camera to capture images of banknotes and performing image analysis using optimized computer vision algorithms such as edge detection and template matching. Based on this analysis, the Raspberry Pi will classify the banknote as genuine or counterfeit. The classification result will then be transmitted to the Arduino, which will act as a peripheral controller responsible for driving output components such as LED indicators and buzzers to provide immediate visual and audible feedback to the user. This separation of responsibilities ensures efficient processing while simplifying hardware control and system integration.
- **Development Methodology:** The project will follow an iterative development methodology, consisting of the following stages:

1. Research and design of the hardware components and image processing algorithms.
2. Development of the hardware prototype using Arduino for sensor integration.

3. Integration of the Raspberry Pi image processing module with Arduino sensor data for counterfeit detection.
4. Testing and evaluation of the integrated prototype to assess accuracy, real-time performance, and reliability under real-world conditions.

- Evaluation Criteria : The effectiveness and performance of the system will be evaluated using the following criteria:
- Accuracy, precision, recall, specificity, F1-score for correct classification of banknotes.
- False positive rate (FPR) and false negative rate (FNR) to assess misclassification risks.
- Real-time performance and detection speed (latency) to ensure efficient processing.
- Processing time for image algorithms to verify optimized performance.
- System stability and reliability under varying conditions (lighting, wear, damage).
- Cost-effectiveness and ease of use for practical deployment among target users.

1.5.2. Limitations of the Project

While this project aims to provide an effective solution for counterfeit detection, several limitations and challenges may affect its execution and results:

- Technical Limitations:
    - Processing Power: The Raspberry Pi, while capable of running image processing algorithms, has limited computational resources compared to a full desktop or server. This may restrict the complexity of algorithms and require optimization for real-time performance.

    - Algorithm Performance: Although rule-based image processing algorithms can detect counterfeit notes, high-quality counterfeit notes that closely replicate genuine security features may still be challenging to identify accurately.

- Operational Limitations:
    - Deployment Environment**:** The system is designed for use in various settings, such as small businesses and public spaces. Its performance may be affected by the practical constraints of these environments.

    - Hardware Constraints: The prototype may be limited in terms of size, portability, and sensor quality, which could impact usability in different settings.
    - Time constraint: the system doesn't have a bill counting mechanism, due to that users may find it time consuming to check each bill one by one.
- Economic Limitations:
    - Cost of Components: While the project aims to develop a cost-effective solution, the price of components (e.g., camera modules, Arduino boards, and other peripherals) may still be a barrier for widespread adoption in low-income areas or among small businesses.
- Data Limitations:
    - Reference Image Quality and Variety: The accuracy of the image processing algorithms depends on the quality and variety of reference banknote images used for comparison. Insufficient variations in banknotes (e.g. wear or damage) may reduce classification accuracy.
    - Coverage of Denominations**:** If the reference set does not include all denominations or conditions of Ethiopian banknotes, the system may perform poorly on certain types of notes or under specific environmental conditions.

While the project aims to overcome these limitations, these challenges must be considered when designing and deploying the system.

1.6. Deliverables

Below are the tangible outputs we will deliver at the project's end. Each item has clear completion criteria and references the attached documents to show its connection to the proposal and research.

1. **Working Hardware Prototype (Raspberry Pi-based Counterfeit Detector)**

   This is a tested device that takes in a banknote, captures an image, processes it on the device, and shows a result (Genuine/Counterfeit) through a display, LED, or buzzer. It is designed for offline use and is low-cost, meeting the need for accessible hardware detectors. Completion criteria: It detects counterfeit and genuine notes with the target accuracy from the evaluation plan; operates fully offline; the bill path and imaging remain stable under typical lighting.

2. **Electronics Pack (Schematics, Wiring, BOM)**

   This includes a full circuit schematic, wiring diagram, and Bill of Materials (with sources and unit costs) for the Raspberry Pi, Arduino board, camera/sensor, lighting, and I/O components. Completion criteria: The schematic and wiring have been reviewed; the BOM includes all parts with part numbers and total cost.

3. **Trained Algorithm Bundle (Tiny/Edge-Optimized)**

   This includes the final quantized algorithm (for example, based on edge detection or feature extraction) and versioned checkpoints, plus an "algorithm card" summarizing training data, metrics, and known limits. Completion criteria: The algorithm file loads on the device within memory limits and meets the target latency; the algorithm card is complete with metrics and a dataset summary. Prior studies show high accuracy with SIFT/ORB-based methods, and our algorithm card will document our results.

4. **Dataset & Data Card (Provenance + Splits)**

   This is a curated image set (or clear instructions for redistribution) covering new Ethiopian notes (including the 200 ETB note), featuring front/back images with different wearings; it includes labels and train/val/test splits. Completion criteria: The data card describes sources, collection, annotation, preprocessing, augmentation, and splits; scripts for reproducing preprocessing are included.

5. **Evaluation Plan & Results Report**

   This outlines the test protocol (lighting scenarios and damaged notes), metrics (accuracy, precision, recall, F1, confusion matrix), and comparative results against acceptance thresholds. Completion criteria: Executed tests are documented with raw logs, confusion matrices, and summary tables; thresholds are met or gaps are clearly identified.

6. **User Manual (Operation, Safety, Calibration, Care)**

   This is a plain-language guide for operating the device, correctly loading a note, interpreting outputs, calibrating lighting, and cleaning optics. It includes safety information. Completion criteria: A non-technical user can operate the device correctly using the manual.

7. **Project Documentation (Design Rationale & Traceability)**

   This gives an overview of the design (hardware + software), decisions log, and requirement-to-test traceability matrix that links scope, constraints, and deliverables to the final tests—structured per the provided proposal template. Completion criteria: All requirements mapped to verification steps; sections match the template’s structure.

Reference alignment notes (from attached works):

- Prior research highlights hardware detectors' cost and low availability—supporting our low-cost, offline hardware focus.
- The 2020 redesign and new 200 ETB denomination guide our dataset coverage.
- Dataset preparation steps (collection, annotation, preprocessing, augmentation, splitting) and evaluation metrics align with our data and testing deliverables.s
- High benchmark accuracies reported in literature inform (but do not replace) our embedded evaluation targets, which we will report transparently in the algorithm card and results report.

1.7. Feasibility Study

The Feasibility Study assesses the viability and suitability of the project, considering the technical, operational, and economic aspects of developing and deploying the counterfeit detection system. This section evaluates whether the project can be successfully executed, taking into account the available resources, risks, and potential benefits.

1.7.1. Technical Feasibility

The technical feasibility of this project is high, as the required hardware and software components are available and compatible for implementation. The project uses widely available Arduino technology and Raspberry Pi for processing, both of which are well-documented and supported by active communities, ensuring ease of integration with other system components such as sensors, cameras, and I/O interfaces.

- Hardware: The Arduino platform and Raspberry Pi will work together to control the system. Arduino handles low-level hardware interactions, such as reading sensors and controlling output devices (LEDs, buzzers). The Raspberry Pi manages complex tasks, including capturing and processing images of banknotes using image processing algorithms. Its higher computational power makes it suitable for real-time image analysis and algorithm execution.

- Operating System: The system will run Debian OS on the Raspberry Pi, a lightweight and stable platform for embedded systems. Debian supports Python and libraries like OpenCV, which are used to implement image processing and computer vision techniques for counterfeit detection.
- Software: Optimized rule-based image processing algorithms will be used to analyze banknote features such as texture, color, and patterns. OpenCV provides tools to efficiently implement these algorithms on the Raspberry Pi, ensuring real-time performance.
- Data/Reference images: A collection of reference images of Ethiopian banknotes, including genuine and known counterfeit examples, will be used to calibrate the algorithms and serve as templates for comparison. Proper selection of reference images ensures accurate classification across different denominations and conditions
- Complexity and Reliability: The main technical challenge lies in integrating image processing with embedded systems to achieve real-time performance. Optimization strategies, such as reducing algorithm complexity and leveraging Raspberry Pi capabilities, can address this challenge. The system's reliability will be ensured through rigorous testing and validation.

The project is feasible given the compatibility of the selected hardware and software, the availability of reference images, and the ability to optimize algorithms for embedded systems

1.7.2. Operational Feasibility

The operational feasibility of the project focuses on how practical and effective the system will be once developed, including its usability, functionality, and support requirements.

- Usability: The system will be designed to be easy to use. The hardware prototype will provide simple feedback (e.g., LED indicators, buzzer sounds) to signal whether a banknote is genuine or counterfeit. Users will not need advanced technical knowledge to operate the system, making it accessible for small businesses and general public use.

- Functionality: The system will be able to classify Ethiopian banknotes accurately as genuine or counterfeit, ensuring reliable performance in real-world scenarios. The real-time detection capability ensures that the system will work efficiently during everyday operations.
- Security and Privacy: The system does not collect or store any personal data, ensuring that privacy is maintained. The focus is on verifying the authenticity of banknotes, and no sensitive information is required from users, making the system secure for public use.

The operational feasibility is high, as the system is expected to be functional, reliable, and easy to maintain, with a user-friendly interface that satisfies the needs of the target audience.

1.7.3. Economic Feasibility

The project is economically feasible, offering a cost-effective solution for counterfeit detection.

The system uses affordable components such as Arduino, Raspberry Pi, camera modules, and sensors. Since the software is based on open-source tools like Python and OpenCV, there are no licensing costs. Key benefits include reduced financial losses from counterfeit notes, improved transaction security, and increased trust for small businesses.

Return on Investment (ROI)

Because the hardware cost is relatively low and maintenance requirements are minimal, the system offers a positive ROI—especially for small businesses that frequently handle cash. Even a few prevented counterfeit transactions can offset the cost of the device.

Budget and Schedule

Project expenses are primarily related to hardware components, prototype construction, algorithm implementation, and testing. The project follows a structured timeline with phases for development, integration, testing, and evaluation, ensuring controlled costs.

Resource Allocation

A small team of engineers and developers can manage the hardware assembly, software implementation, and system optimization. No specialized high-cost expertise is required, further reducing expenses.

Conclusion

The system is cost-effective, scalable, and financially sustainable. Its low development and operational costs make it suitable for widespread use among small businesses and individuals.

1.8. Significance of the Project

This project holds significant value both in terms of its contribution to the counterfeit detection problem domain and its broader societal and academic impact. The development of a hardware-based counterfeit detection system using Arduino technology, Raspberry Pi, and image processing addresses key challenges faced by businesses, law enforcement, and the general public in Ethiopia. The significance of this project can be outlined in the following areas:

1.8.1 Contribution to the Counterfeit Detection Problem

Counterfeiting is a persistent and growing issue in Ethiopia, where the circulation of counterfeit banknotes undermines public trust in the currency and causes financial losses for businesses and individuals. Existing solutions, such as UV light detectors and counterfeit pens, are either costly or prone to errors, especially when counterfeiters use sophisticated methods to replicate genuine banknotes. This project addresses this gap by providing an affordable, accessible, and automated solution for detecting counterfeit Ethiopian currency, with the following benefits:

- Accuracy: The use of image processing algorithms integrated into the hardware prototype ensures accurate detection of counterfeit banknotes, even those that are well-crafted or damaged.
- Real-Time Detection: By leveraging Raspberry Pi and Arduino technology, the system provides real-time processing of banknote images, enabling quick and efficient classification of banknotes as genuine or counterfeit.
- Cost-Effectiveness: The hardware prototype is designed to be affordable, making it accessible to small businesses, local markets, and individuals who cannot afford traditional high-cost counterfeit detection machines.
- Offline Functionality: The system's offline capability makes it ideal for use in areas with limited internet access, ensuring widespread applicability in both urban and rural settings.

1.8.2 Impact on Stakeholders

The project will positively impact a wide range of stakeholders:

- Small Businesses: Many small businesses in Ethiopia are vulnerable to accepting counterfeit money. This system will allow business owners to verify the authenticity of banknotes, reducing financial losses and increasing trust in their transactions.
- Law Enforcement: The system can be deployed by law enforcement agencies to investigate counterfeit currency circulation, helping to identify counterfeit bills and trace illegal activity.
- General Public: By providing a portable, user-friendly counterfeit detection system, this project empowers individuals to verify the currency they receive, contributing to greater public awareness and trust in the national currency.
- Financial Institutions: Banks and other financial institutions can use the system to facilitate the detection of counterfeit currency during transactions, ensuring that only legitimate currency enters circulation.

1.8.3 Contribution to the Field of Study

This project represents a valuable contribution to both the fields of computer science and embedded systems:

- Image Processing and Computer Vision: The integration of image processing algorithms for counterfeit detection in an embedded system using Raspberry Pi and Arduino technology showcases the potential for image processing to solve real-world problems. The development of this system pushes the boundaries of image processing applications in embedded platforms, demonstrating how computer vision techniques can be implemented in low-cost, real-time systems.
- Embedded Systems Design: The project's focus on creating a hardware-based solution emphasizes the application of embedded systems design. By combining Arduino microcontrollers, Raspberry Pi, and image processing algorithms, the project contributes to advancing the use of embedded systems in computer vision and processing-powered applications.
- Practical Applications: The project aligns with the growing interest in practical applications of processing in hardware systems, particularly in areas like counterfeit detection. It demonstrates how affordable, real-time systems can be designed and implemented on widely available hardware platforms to address pressing societal challenges.

1.8.4 Broader Societal Impact

The impact of this project extends beyond its technical and academic contributions:

- Economic Impact: By providing a cost-effective solution for counterfeit detection, the project helps protect businesses from financial losses due to counterfeit currency. This, in turn, supports economic stability by ensuring that legitimate currency remains in circulation.
- Social Trust: With widespread adoption, this system can enhance public trust in the national currency. By empowering individuals and businesses to verify the

authenticity of banknotes, the system contributes to a more secure and transparent financial environment.

- Technology Accessibility: The affordable hardware prototype makes advanced counterfeit detection technology accessible to a broader population, including those who typically do not have access to expensive machines, thereby promoting inclusive technology adoption.

1.9. Beneficiaries of the project

The counterfeit detection hardware prototype developed in this project has the potential to benefit a wide range of stakeholders, each of whom will use the system to address the challenges posed by counterfeit currency in Ethiopia. The key beneficiaries of the project include:

1.9.1 Small Businesses

Small business owners are among the most vulnerable to financial losses caused by counterfeit banknotes. These businesses often deal with cash transactions and may lack access to high-end counterfeit detection machines.

- How they benefit: The hardware prototype will provide an affordable and easy-to-use tool for small businesses to quickly verify the authenticity of banknotes. This ensures that business owners can confidently accept payments, reducing the risk of financial losses from counterfeit money. The real-time detection capability ensures minimal disruption to business operations, and the cost-effectiveness makes it accessible to small enterprises without the need for expensive counterfeit detection systems.

1.9.2 Law Enforcement and Security Agencies

Law enforcement agencies tasked with investigating counterfeiting and ensuring financial security will benefit from this system by using it in the field to detect counterfeit banknotes in real-time.

- How they benefit: The system allows law enforcement officers to verify the authenticity of banknotes during investigations, seizures, and raids. It enhances their ability to track counterfeit operations and provide accurate evidence for legal proceedings. The portable and offline capability of the system also ensures its usability in various field conditions, making it a versatile tool for on-site inspections.

1.9.3 General Public and Consumers

The general public will also benefit from the ability to easily verify the authenticity of banknotes in their possession, particularly in everyday situations such as shopping, receiving change, or during casual transactions.

- How they benefit: Individuals will be empowered to authenticate their banknotes at any time, increasing their confidence in the currency they handle. This reduces the likelihood of unknowingly accepting counterfeit money, fostering trust in the national currency. The portability and user-friendly design of the system make it accessible for everyday use, contributing to greater financial awareness.

1.9.4 Financial Institutions and Banks

Banks and financial institutions that manage large volumes of cash transactions and currency handling can integrate this system into their operations to streamline the verification process.

- How they benefit: The system will enable quick, reliable counterfeit detection during cash deposit and withdrawal processes. It can be used to authenticate large quantities of banknotes efficiently, reducing human error and the time spent on manual inspection. Additionally, the system can be used to track counterfeit notes entering circulation and improve overall currency management at financial institutions.

1.9.5 Educational Institutions and Researchers

Universities, researchers, and academic institutions focused on image processing, computer vision, or embedded systems will benefit from this project by gaining insights into real-world applications of image processing in embedded systems.

- How they benefit: This project serves as an excellent case study for processing in hardware systems. Educational institutions can use the project as a reference for future research in embedded image processing, counterfeit detection, and computer vision. The open-source nature of the system (if applicable) allows it to be adapted and further developed in academic research, enabling students and researchers to explore and contribute to the field.

1.9.6 Government and Regulatory Bodies

Government agencies involved in economic policy, currency regulation, and public financial security will also benefit from the project. The system provides a cost-effective tool for improving national currency security and combating counterfeiting at the grassroots level.

- How they benefit: The system can assist regulatory bodies in monitoring the circulation of counterfeit banknotes, providing a data-driven approach to combat counterfeiting. By enabling widespread adoption of the detection system, the government can strengthen its anti-counterfeiting measures, reduce losses in the economy, and boost public confidence in the national currency. This tool also aligns with financial security initiatives, improving the efficiency of currency verification processes in various sectors.

1.9.7 International Organizations and NGOs

International organizations and NGOs that work in financial transparency, anti-corruption, or economic development may also find value in deploying this system in developing regions where counterfeit currency circulation is a persistent issue.

> How they benefit: NGOs and international agencies can use the system to support financial inclusion, reduce the circulation of counterfeit currency, and improve economic stability in regions where counterfeit notes are rampant. The low-cost, portable nature of the system makes it suitable for global deployment in areas with limited access to advanced technology, supporting broader financial security efforts.

1.10 Methodology

The methodology for this project follows a structured approach, ensuring that all phases from planning to deployment are carefully executed. The approach emphasizes the integration of hardware and image processing techniques, as well as ensuring real-time, efficient detection of counterfeit Ethiopian banknotes. The methodology is divided into several key phases:

1.10.1 Planning Phase

- Objective Definition: In this phase, we define the project's objectives clearly, focusing on the creation of a hardware-based solution that can detect counterfeit Ethiopian banknotes. This includes identifying target users, such as small businesses, law enforcement, and the general public, and determining system requirements such as real-time performance, cost-effectiveness, and offline functionality.
- Resource Allocation: Identify all the necessary hardware components, software tools, and human resources required for the project. This includes acquiring Arduino, Raspberry Pi, camera modules, sensors, and any other required

components, as well as ensuring that the team has the expertise to complete the project.

- Timeline and Budget Planning: Establish a project timeline with clear milestones, deadlines, and resource allocation. The project will follow an iterative approach to allow for development, testing, and refinement in each phase.

1.10.2 Analysis Phase

- Requirement Gathering: Conduct thorough research on the existing counterfeit detection technologies and analyze their limitations. This phase will include gathering user requirements through surveys and interviews with potential users (e.g., small business owners, law enforcement officers) to understand their needs.
- Dataset Collection: Gather and preprocess a comprehensive dataset of genuine and counterfeit Ethiopian banknotes. This dataset will be used to calibrate the image processing algorithms for image classification.
- Feasibility Study: Evaluate the feasibility of integrating image processing algorithms into the Arduino and Raspberry Pi hardware, ensuring that the chosen hardware can handle the computational requirements for real-time image processing and detection.

1.10.3 Design Phase

- System Design: Design the overall system architecture, which will integrate Arduino for basic control and Raspberry Pi for more computationally intensive tasks, such as running image processing algorithms. The system will be designed to be modular and scalable for different user requirements.
- Hardware Design: Design the hardware components of the system, including the note slot, camera module, sensor setup, and output system (LEDs, buzzer). The design will ensure the system is portable, easy to use, and efficient.
- Image Processing Algorithm Design: Define the architecture of the image processing algorithms (e.g., edge detection, feature extraction). Select and

optimize algorithms that can classify genuine and counterfeit notes accurately under various conditions, ensuring the algorithms can be efficiently deployed on Raspberry Pi.

1.10.4 Implementation Phase

- Hardware Assembly: Assemble the physical components, including Arduino, Raspberry Pi, camera, and sensors. This phase involves wiring the components, ensuring proper mounting, and performing initial system checks.
- Software Development: Develop the Arduino firmware to manage sensors and outputs, and Raspberry Pi software to handle image processing, algorithm execution, and communication between the components.
- Image Processing Algorithm Integration: Calibrate the image processing algorithms on the dataset of banknotes, and integrate them to make them suitable for real-time execution on Raspberry Pi.

1.10.5 Testing Phase

- Unit Testing: Conduct unit testing of individual hardware components, such as the camera and sensors, to ensure they are functioning correctly. Also, test software modules like image capture, preprocessing, and execution.
- System Integration Testing: Test the integration of hardware and software components, ensuring the Arduino and Raspberry Pi communicate effectively and that the entire system works seamlessly.
- Performance Testing: Evaluate the system's accuracy in detecting counterfeit and genuine banknotes and test its real-time performance. Ensure that the system works under various environmental conditions (lighting, wear, damage).
- User Testing: Conduct testing with actual users to ensure the system is intuitive, easy to use, and meets their needs. Collect feedback to improve the design and functionality.

1.10.6 Evaluation Phase

- Evaluation Metrics: The system will be evaluated using metrics such as accuracy, precision, recall, F1-score, and real-time performance. These metrics will help determine the effectiveness and efficiency of the system in detecting counterfeit banknotes.
- Cost-Effectiveness Analysis: Evaluate the overall cost of the system (hardware, software, deployment) and compare it to the benefits, such as reduced financial losses from counterfeit currency.
- User Feedback: Gather feedback from stakeholders on the usability and satisfaction with the system to make improvements for future versions.

1.11 Development Tools

This section specifies the tools and technologies that will be used in the project for hardware development, software development, image processing, and algorithms. The selected tools ensure that the system will be efficient, cost-effective, and easy to integrate into a hardware-based solution.

1.11.1 Hardware Tools and Platforms

- Arduino Platform: Arduino is used for hardware control, managing sensors, camera, and outputs (LED, buzzer). Arduino’s flexibility, affordability, and simplicity make it the ideal choice for embedded systems.
- Raspberry Pi (4GB/8GB): Raspberry Pi will be used for image processing and running the algorithms. It provides high processing power and memory capacity compared to Arduino, allowing for real-time computer vision tasks.
- Camera Modules: For capturing banknote images to be processed by the system. The camera module is chosen for its compatibility with both Arduino and Raspberry Pi platforms.

1.11.2 Software Development Tools

- Arduino IDE: The Arduino IDE is used to write, debug, and upload code to the Arduino microcontroller. It supports C/C++ and provides libraries that facilitate sensor integration and I/O management.
- Python: Python will be used for developing the image processing algorithms and for implementing the models. Python’s extensive libraries and simplicity make it ideal for these tasks.
- OpenCV: OpenCV is used for image preprocessing and feature extraction. It offers a comprehensive set of tools for handling computer vision tasks, such as resizing, thresholding, and edge detection.
- Raspberry Pi OS (Debian-based): The Raspberry Pi OS (formerly Raspbian) provides the operating system for the Raspberry Pi, offering support for Python and OpenCV, along with a stable environment for running the software.

1.11.3 Testing and Debugging Tools

- Jupyter Notebook: Jupyter Notebook will be used for testing and experimenting with image processing algorithms in an interactive, real-time environment.
- Git: Git is used for version control, allowing the team to track code changes, collaborate efficiently, and manage project versions throughout development.

1.12. Required resources with cost

This section outlines the resources required to develop, implement, and deploy the counterfeit detection hardware prototype, and estimates their associated costs. The resources are divided into material resources, software resources, and time resources, ensuring that all aspects of the project are covered for successful completion.

1.12.1 Material Resources

The following materials and components are needed for the development of the hardware prototype:

NO	Component Name	Description/Purpose	Quantity	Unit Cost (Birr)	Total Cost (Birr)
1	Raspberry	Handles deep Learning inference and image processing	1	25,000	25,000
2	Arduino Uno	Controls sensors LEDs, and hardware interface	1	5,250	5,250
3	Camera Module	Captures banknote Images for analysis	1	1,500	1,500
4	Bre adboard	For circuit Prototyping and testing	1	860	860
5	Cables, Resistors, Connectors	User input for denomination type	3	100	300
6	Power Supply	Provide stable voltage to Arduino and Raspberry Pi	1	599	599
7	Cables ,Resistors, Connectors	For hardware Connections	Set	1,000	1,000
8	LED and Buzzer	Output indicators for detection result	Set	500	500

Total Estimated Cost: 35,009 Birr

1.12.2 Software Resources

The project primarily uses open-source software tools and frameworks. The costs associated with software development are minimal, as most of the tools needed for development are free:

- OpenCV (Open-source) for image processing tasks.
- Arduino IDE (Free) for firmware development.
- Python (Free) for developing image processing and algorithms.
- Raspberry Pi OS (Free, Debian-based) for the operating system used on Raspberry Pi.

Total Software Resources Cost: Free

1.12.3 Time Resources

The project is expected to be completed within a 6-month timeline. The estimated time commitment for each phase of the project is as follows:

- Planning and Analysis: 1 month
- Design Phase: 1 month
- Implementation Phase: 2 months
- Testing and Evaluation: 1 month
- Documentation and Deployment: 1 month

This timeline ensures that the system will be developed, tested, and deployed in an organized and timely manner.

1.12.4 Total Estimated Cost

- Material Resources: 35,009 Birr
- Software Resources: Free

Total Project Cost = 35,009 Birr

1.13 Task and Schedule

This section defines the tasks and activities that the project will perform, along with the timeline and milestones for the successful completion of the project. The timeline is broken down into phases with clear objectives and deadlines to ensure the project progresses in a structured and organized manner.

1.13.1 Work Breakdown Structure (WBS)

The project will be divided into the following phases with specific tasks for each:

1. Planning and Analysis (1 Month)
    - Define project objectives and scope
    - Gather and analyze requirements from stakeholders
    - Research and evaluate existing counterfeit detection solutions
    - Define system architecture and hardware/software specifications
    - Identify hardware and software resources required for the project
2. Design Phase (1 Month)
    - Design the overall system architecture, including Arduino and Raspberry Pi integration
    - Design hardware components, including buttons, camera, and power supply setup
    - Develop the image processing algorithm design and decide on the algorithm architecture
    - Prepare schematics and wiring diagrams for the hardware
    - Create 3D CAD files for the enclosure and other mechanical components
3. Implementation Phase (2 Months)
    - Assemble hardware components (Arduino, Raspberry Pi, camera, buttons, power supply)
    - Develop firmware for Arduino to control the buttons and manage sensor data
    - Install Raspberry Pi OS and set up necessary libraries (e.g., OpenCV)

    - Integrate image processing algorithms on Raspberry Pi for real-time counterfeit detection
    - Develop software for image preprocessing and real-time inference
4. Testing and Evaluation Phase (1 Month)
    - Unit testing of hardware components (e.g., camera, power supply, buttons)
    - System integration testing to ensure the Arduino and Raspberry Pi communicate effectively
    - Real-time testing for image processing and algorithm inference accuracy
    - Evaluate system performance using metrics (accuracy, precision, recall, F1-score)
    - Test system under different lighting conditions, wear, and damaged notes
5. Documentation and Deployment (1 Month)
    - Create user manuals and maintenance guides
    - Write project documentation, including system architecture, design rationale, and test results
    - Develop troubleshooting and calibration guides
    - Deploy prototype in real-world settings (small businesses, law enforcement)
    - Provide training for users and stakeholders on operating the system

1.13.2 Task Dependencies

The tasks in the project are interdependent in many ways:

- The Planning and Analysis phase must be completed before the Design Phase can begin.
- The Design Phase must be completed before the Implementation Phase starts, as it involves preparing all the necessary hardware and software designs.
- The Implementation Phase must be well underway before Testing and Evaluation can begin. Hardware and software must be assembled, integrated, and optimized before performing real-time tests.

- Documentation will start in parallel with testing but must be completed after system validation to include accurate results and details from testing.

1.13.3 Milestones

To ensure the project stays on track, the following milestones are set:

Milestone	**Description**	**Dependency**	**Expected Completion**
M1	Completion of Planning & Analysis	M1	Dec 29, 2025
M2	Design approval and component purchase	M2	Jan 31, 2026
M3	Hardware assembly and model integration	M3	Mar 31, 2026
M4	System integration and testing	M4	Apr 30, 2026
M5	Documentation, training, and deployment	M5	May 30, 2026

1.13.4 Project Timeline

The project will follow the timeline shown below, broken down into tasks and their respective deadlines:

Phase	Task	Duration	Deadline
Planning and Analysis	Define objectives, gather requirements Research and analyze current solutions	1 Month	End of December
Design	Design system architecture and hardware Finalize deep learning model design	1 Month	End of January
Implementation	Assemble hardware components Develop software for Raspberry Pi and Arduino	2 Month	End of March
Testing and Evaluation	Perform system integration and real-time testing Evaluate performance and refine system	1 Month	End of April

Documentation and Deployment	Write manuals, provide training Deploy the final prototype	1 Month	End of May

1.14 Team Composition

The project team is composed of five members, each with specific roles and responsibilities that contribute to the successful completion of the counterfeit detection hardware prototype. Below are the team members and their respective roles:

1.14.1 Kaleb – Project Manager

Kaleb will take on the role of the Project Manager, overseeing the project's progress and ensuring that all tasks are completed on time and within the scope. Kaleb's responsibilities include:

- Coordinating and managing the overall project timeline.
- Allocating resources and managing any risks that may arise during development.
- Ensuring effective communication between team members.
- Monitoring the project's budget, ensuring that costs stay within the allocated budget.
- Ensuring the team meets milestones and deadlines.

1.14.2 Benjamin – System Developer

Benjamin will now serve as the System Developer, focusing on the hardware and firmware development aspects of the project. Benjamin's tasks include:

- Writing the Arduino firmware to control the system's buttons, manage sensor data, and handle output (LED, buzzer).

- Ensuring the hardware components, including camera, buttons, and power supply, work seamlessly together.
- Developing communication protocols between the Arduino and Raspberry Pi for data exchange.
- Assisting with the system integration to ensure smooth operation between hardware and software.

1.14.3 Bereket – System Designer

Bereket will now take on the role of System Designer, responsible for designing the overall architecture of the system and ensuring all components fit together effectively. Bereket's tasks include:

- Designing the overall system architecture, including the interaction between Arduino, Raspberry Pi, and peripheral devices.
- Preparing hardware designs such as schematics, and wiring diagrams for the prototype.
- Defining the image processing algorithm design, ensuring the algorithms are optimized for integration with the hardware.
- Making sure that the design supports real-time processing and is cost-effective.

1.14.4 Salem – System Tester

Salem will serve as the System Tester, responsible for ensuring the functionality and performance of the system. Salem's tasks include:

- Conducting unit testing of hardware components (e.g., camera, buttons, power supply).
- Performing system integration testing to ensure that the Arduino and Raspberry Pi communicate effectively.
- Testing the system's performance under real-world conditions, ensuring that it can classify banknotes as genuine or counterfeit accurately.

- Evaluating system reliability using metrics such as accuracy, precision, recall, and F1-score.

1.14.5 Shalom – System Evaluator

Shalom will act as the System Evaluator, responsible for evaluating the system’s performance and providing feedback for improvements. Shalom’s tasks include:

- Conducting evaluation based on predefined metrics (e.g., real-time performance, accuracy of counterfeit detection).
- Reviewing the testing results and analyzing user feedback to ensure the system meets the project’s objectives.
- Assisting in fine-tuning the system based on test results to improve performance and accuracy.
- Compiling the evaluation report and documenting the system’s effectiveness in solving the counterfeit detection problem.

Chapter 2: Description of existing system and Literature Review

2.1. Major function of existing system

Existing systems for counterfeit money detection primarily rely on hardware-based methods that examine physical security features of banknotes. These systems, such as UV light detectors, magnetic ink scanners, and counterfeit pens, aim to verify authenticity by checking specific elements like watermarks, security threads, holograms, and ink properties. The major functions include:

- UV Light Detection: Illuminates invisible fluorescent features on genuine notes, such as security threads or ink marks, which glow under ultraviolet light.
- Magnetic Ink Verification: Scans for magnetic properties in the ink used for serial numbers and other elements, confirming the presence of iron particles in legitimate currency.
- Security Thread and Watermark Inspection: Uses transmitted light or manual viewing to detect embedded threads or watermarks that are difficult to replicate accurately.
- Size and Texture Analysis: Measures the physical dimensions and paper quality to identify deviations from standard genuine notes.

These functions are typically performed in serial, requiring manual intervention or specialized equipment, and are common in banks, retail environments, and vending machines.

2.2. Users of current system

The primary users of existing counterfeit detection systems include:

- Financial Institutions and Banks: Banks use high-end machines for bulk verification during deposits, withdrawals, and currency sorting to ensure only genuine notes enter circulation.
- Law Enforcement and Security Agencies: Police and investigators use advanced scanners to analyze seized currency and trace counterfeit operations.
- Vending Machines and ATMs: Automated systems integrate basic detectors to reject suspicious notes in real-time.

These users span from professionals in finance to everyday consumers, but access is often limited by cost and availability.

2.3. Drawback of current system

Current counterfeit detection methods, while effective in some contexts, have several drawbacks that limit their reliability, accessibility, and efficiency:

- Prone to Human Error: Manual inspection methods, such as using UV lights or pens, rely on the user's expertise and attention, leading to false positives or negatives, especially with worn or damaged notes.
- Limited Scope: Many systems only check specific features (e.g., UV ink or magnetic properties), making them ineffective against sophisticated counterfeits that replicate those elements but fail in others.
- High Cost: Advanced hardware like multi-function scanners is expensive, restricting use to large institutions and leaving small businesses and individuals without affordable options.
- Time-Consuming: Serial verification processes slow down transactions, particularly in high-volume environments like retail or banking.
- Vulnerability to Wear and Tear: Systems may misclassify genuine notes that are old, dirty, or damaged due to degraded features.

- Lack of Portability: Bulky machines are not suitable for field use by law enforcement or mobile vendors.
- Inability to Handle Advanced Forgeries: With improvements in printing technology, counterfeiters can mimic basic security features, fooling simpler detectors.
- No Real-Time Adaptation: Static methods do not improve over time or adapt to new counterfeiting techniques without hardware upgrades.

These limitations highlight the need for more automated, cost-effective, and comprehensive solutions.

2.4. Literature Review

The detection of counterfeit banknotes using image processing techniques has been extensively explored in academic literature, focusing on methods that analyze visual features such as texture, color, edges, and patterns to distinguish genuine from fake currency. This review synthesizes key studies from 2018 to 2025, highlighting algorithms like SIFT (Scale-Invariant Feature Transform), ORB (Oriented FAST and Rotated BRIEF), edge detection, and fuzzy associative classifiers, which have achieved accuracy rates ranging from 87.74% to 100%. The review draws from diverse sources, including IEEE, Springer, ResearchGate, and ScienceDirect, to provide a balanced view of advancements, challenges, and gaps in the field.

Early works, such as Colaco et al. (2021), adapted image processing for Indian currency authentication using Canny edge detection in Python with OpenCV. Their system extracted features like shapes, numbers, and emblems from note images, comparing them to genuine templates for classification. While effective for specific denominations (500 and 2000 INR), the method's limitations include sensitivity to lighting variations and note condition, achieving around 95% accuracy but requiring controlled environments.

Rahman et al. (2017) and Aprameya (2021) emphasized feature extraction techniques for counterfeit Indian notes. Aprameya's system, built with OpenCV and Tkinter, used edge detection and segmentation to identify security features, working on a custom dataset of 500 and 2000 INR notes. It highlighted the role of grayscale conversion, thresholding, and contour analysis but noted drawbacks like inefficiency with damaged notes and the need for high-resolution inputs.

More advanced approaches integrated fuzzy logic and optimization. Abd-Alkader et al. (2024) proposed a framework using particle swarm optimization (PSO) with fuzzy associative rules for coin detection, focusing on blob detection in regions of interest (ROIs). This method improved efficiency by pruning irrelevant rules, achieving near-perfect classification on multispectral images but struggled with corroded or worn coins, underscoring the challenge of real-world variability.

Chapter 3: Proposed System

3.1. Overview

The proposed system is a hardware-based counterfeit money detector designed to identify fake Ethiopian banknotes using image processing techniques integrated with Arduino technology. The system addresses the limitations of existing manual and expensive detection methods by providing an affordable, portable, and automated solution suitable for small businesses, law enforcement, and the general public.

Key features of the system include:

- Real-Time Detection: The system captures an image of the banknote and processes it immediately to determine authenticity.
- Image Processing Integration: Utilizes techniques such as edge detection, feature extraction (e.g., using algorithms like Canny or Sobel for edges, and histogram analysis for color and texture), and pattern matching to compare the captured image against known genuine features.
- User-Friendly Interface: Includes simple inputs (e.g., buttons for selecting denomination) and outputs (e.g., LED indicators and buzzer for genuine/counterfeit alerts).
- Portability and Cost-Effectiveness: Built on low-cost Arduino hardware with Raspberry Pi for enhanced processing, ensuring accessibility without requiring internet or high-end equipment.

The main components are :

- Hardware: Arduino microcontroller for control, camera module for image capture, buttons for user input, LEDs/buzzer for feedback, and power supply.
- Software: Firmware on Arduino for hardware management and image processing scripts (using libraries like OpenCV on Raspberry Pi) for analysis.
- Architecture: A modular design where the Arduino handles sensor data and basic logic, while image processing occurs on an embedded platform to classify the note based on visual features like security threads, watermarks, and ink patterns.

This system aims to achieve high accuracy in detecting counterfeits by focusing on visual discrepancies that are hard to replicate, operating offline for widespread use in Ethiopia.

3.2. Functional requirement

The proposed system must perform the following functional requirements to ensure effective counterfeit detection:

- Banknote Insertion and Capture: The system shall accept a banknote through a designated slot and capture a high-resolution image using the integrated camera module. This includes automatic triggering of the capture process upon insertion.
- Denomination Selection: Users shall select the banknote denomination (e.g., 50, 100, 200 ETB) via physical buttons, allowing the system to load the appropriate reference features for comparison.
- Image Preprocessing: The system shall preprocess the captured image by converting it to grayscale, applying noise reduction (e.g., Gaussian blur), and resizing it to a standard dimension for consistent analysis.
- Feature Extraction and Analysis: Using image processing techniques, the system shall extract key features such as edges (via Canny edge detection), textures (via histogram equalization), and specific patterns (e.g., security threads or holograms via contour detection). It shall compare these against predefined templates of genuine notes.ft
- Authenticity Classification: The system shall classify the banknote as genuine or counterfeit based on similarity thresholds (e.g., matching score above 90% for genuine), outputting the result via LED (green for genuine, red for counterfeit) and buzzer alert.

- Logging and Reporting: The system shall log detection results (e.g., timestamp, denomination, outcome) in internal memory for later review by users like law enforcement.

These functions ensure the system supports end-to-end detection, from input to output, providing reliable services for verifying Ethiopian currency.

3.3. Non-functional requirement

The proposed system must meet the following non-functional requirements to ensure it is practical, efficient, and user-friendly:

- Performance: The system shall process and classify a banknote within 5 seconds to support real-time use in busy environments like markets or banks.
- Accuracy and Reliability: The system shall achieve at least 95% accuracy in distinguishing genuine from counterfeit notes under standard conditions, with minimal false positives/negatives, tested across various note conditions (e.g., new, worn).
- Usability: The interface shall be intuitive, requiring no more than 2-3 steps for detection, with clear visual and auditory feedback. It shall be operable by non-technical users, including those with basic literacy.
- Portability and Durability: The hardware shall be compact (e.g., handheld or desktop-sized), weighing under 1 kg, and durable enough to withstand daily use in environments with dust or moderate humidity.
- Cost-Effectiveness: The total production cost per unit shall not exceed 35,000 ETB, using off-the-shelf components to enable affordable scaling.
- Security: The system shall prevent tampering by securing firmware updates and ensuring no sensitive data (e.g., images) is stored permanently without user consent.

These requirements ensure the system is not only functional but also robust, accessible, and sustainable for long-term use.

3.4. System model

The system model provides a conceptual, logical, and physical representation of the counterfeit money detector, illustrating its structure, functions, behaviors, and interactions. It is designed as an embedded system combining hardware and software for image-based analysis.

Conceptually, the system is a self-contained detector that takes a physical banknote as input, processes its image digitally, and outputs an authenticity verdict. Logically, it consists of input modules (sensors/buttons), processing core (image algorithms), and output modules (indicators). Physically, it integrates Arduino for control and Raspberry Pi for computation, connected via serial communication.

3.4.1. Scenario

A typical scenario involves a small business owner verifying a 100 ETB note during a transaction:

- Context: The user is in a market stall, the note may be slightly worn.
- Condition: The system is powered on and calibrated for Ethiopian denominations.
- Input: The user selects "100 ETB" via a button and inserts the note into the slot.
- Process: The camera captures the image; preprocessing cleans it; features like edges and textures are extracted and compared to genuine templates.
- Output: If matching thresholds are met, a green LED lights up with a positive beep; otherwise, a red LED and alarm indicate counterfeit.
- Outcome: The transaction proceeds safely if genuine, or the note is rejected, preventing loss.

This scenario highlights the system's role in everyday financial security, handling real-world variations like note condition.

3.4.2. Use case model

The use case model describes the interactions between actors (users) and the system. Actors include End User (e.g., business owner), Administrator (e.g., technician), and System (internal components).

Use Cases:

- Verify Banknote: Actor - End User; Goal - Determine authenticity; Steps: Select denomination, insert note, view result.
- Log Results: Actor - System; Goal - Record detections; Flow: After classification, store data in memory.

3.5 Functional Model

The functional model of the proposed counterfeit money detection system defines the system’s operations, modules, data structures, transformations, inputs, outputs, and the interactions between them. It provides a detailed understanding of how the system processes banknotes from input to output, ensuring accuracy, usability, and robustness.

The model is structured around the core functions, supporting modules, and data flows, emphasizing the use of SIFT-based image processing for reliable counterfeit detection.

3.5.1. Data Dictionary

The Data Dictionary provides a detailed description of all data elements used within the counterfeit money detection system. With **SIFT** as the primary image processing technique, the dictionary focuses on features extracted for robust counterfeit detection. It includes data names, types, formats, valid values, and usage.

Data Element	Type	Format	Valid Values	Description / Usage
BanknoteImage	Image	JPEG/PNG	Raw camera capture	High-resolution image captured when a banknote is inserted. Input for SIFT feature extraction.
GrayscaleImage	Image	Matrix (0–255 intensity)	Converted from BanknoteI mage	Converts the image to grayscale for SIFT processing, ensuring scale and rotation invariance.
Preprocessed Image	Image	Matrix	Normalized, resized	Cleaned image after resizing and noise reduction (Gaussian blur), ready for SIFT

				keypoint detection.
SIFTKeypoints	Array of Points	(x, y, scale, orientation)	Variable count depending on banknote	Detected points of interest in the banknote image. Represent locations of distinctive features used for matching with genuine templates.
SIFTDescriptors	Numeric Array	128-dimensional float vector	Normalized float values	Descriptors corresponding to each keypoint. Captures local gradients around keypoints for robust matching.
MatchingScore	Float	0.0–1.0 or 0–100%	Threshold ≥ 0.90 = genuine	Similarity score computed from SIFT descriptor matching with reference templates. Determines classification.

Denomination Selection	Integer	50, 100, 200	50, 100, 200 ETB	Selected by user. Loads appropriate SIFT reference templates for comparison.
Classification Result	String	“Genuine” / “Counterfeit”	Two possible outputs	Final system decision. Displayed via LED and buzzer.
Timestamp	DateTime	YYYY-MM-DD HH:MM:SS	System time	Recorded for logging detection events.
Detection Log	Record	JSON/TXT/CSV	Structured entries	Contains timestamp, denomination, matching score, and authenticity result for review by administrators.
UserInputButtonSta te	Boolean	Pressed/Not Pressed	True/False	Captures button interaction for selecting denomination and initiating detection.

LEDStatus	Boolean	On/Off	True/False	Indicates result: green → genuine; red → counterfeit.
BuzzerSignal	Integer	Duration (ms)	100–700 ms	Audible alert based on classification.
SerialDataPacket	Byte Array	Arduino ↔ Raspberry Pi	Structured bytes	Transfers captured image, SIFT features, and classification results between Arduino and processing unit.

3.5.2. Functional Modules Diagram

The functional modules diagram illustrates the major components of the proposed **Banknote Authentication System**, including their inputs, outputs, internal relationships, and data flow. The system is structured into four primary modules, each responsible for a specific stage in the verification process: Input Acquisition, Preprocessing, Feature Extraction, Matching & Decision.

3.5.3. Dynamic model

The dynamic model describes the system’s behavior over time by showing how data flows through its components, how internal states change in response to events, and how functions transform inputs into outputs. It provides a temporal perspective of the system, revealing not only what the components are, but also how they interact and react during real-world operation.

The model captures the complete lifecycle of a single detection operation through the following major states:

- **Idle** – The system is powered on and waiting for user interaction. No banknote is present in the slot.
- **NoteInserted** – Triggered immediately when the optical or mechanical sensor detects a banknote. The system activates its illumination LEDs and displays a blinking blue indicator, prompting the user to select the denomination.
- **WaitingForDenomination** – The user presses one of the four denomination buttons (10, 50, 100, or 200 ETB). This event supplies the system with the correct reference template.
- **Capturing** – The camera module acquires a high-resolution image under controlled, uniform lighting.
- **Processing** – The core image-processing pipeline executes: grayscale conversion, denoising, histogram equalization, edge detection (Canny), security-thread localization (Hough transform), watermark-region analysis, micro-text verification, and weighted template matching.
- **ResultGenuine / ResultCounterfeit** – Final classification is complete. The system activates the appropriate visual (Green/Red LED) and auditory (single beep vs. triple beep) feedback.
- **DisplayingResult** – The result remains visible for 4 seconds or until the note is removed, after which the system returns to Idle.

State transitions are driven by well-defined events :

- “Banknote Inserted” → Idle → NoteInserted
- “Denomination Button Pressed” → NoteInserted → Capturing
- “Image Captured Successfully” → Capturing → Processing
- “Processing Finished & Score ≥ 0.90” → Processing → ResultGenuine
- “Processing Finished & Score < 0.90” → Processing → ResultCounterfeit
- “Note Removed or Timeout” → DisplayingResult → Idle

3.5.4. Sequence diagram

The sequence diagram shows the order of interactions for verifying a banknote:

This depicts the flow: User initiates, hardware captures, processor analyzes, template compares, and result is returned.

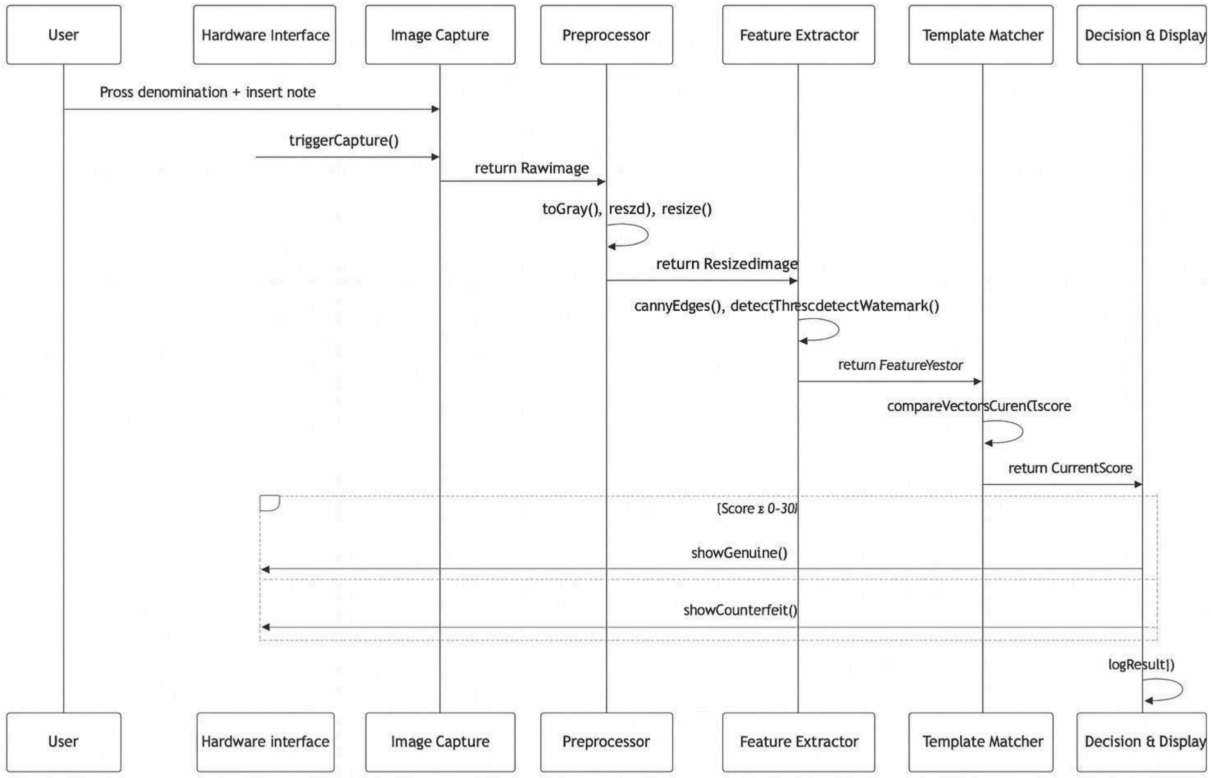

Fig 1 Sequence diagram

3.5.5. Activity diagram

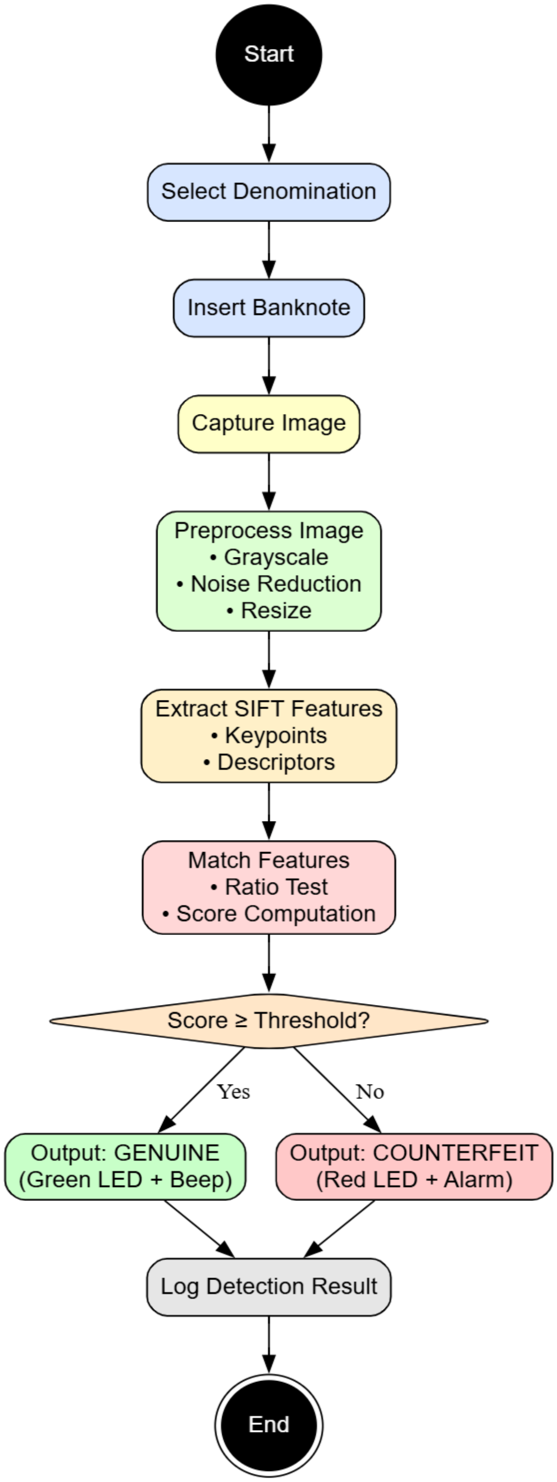

Fig 2 Activity Diagram

3.5.6. State chart diagram

The state chart diagram represents component states:

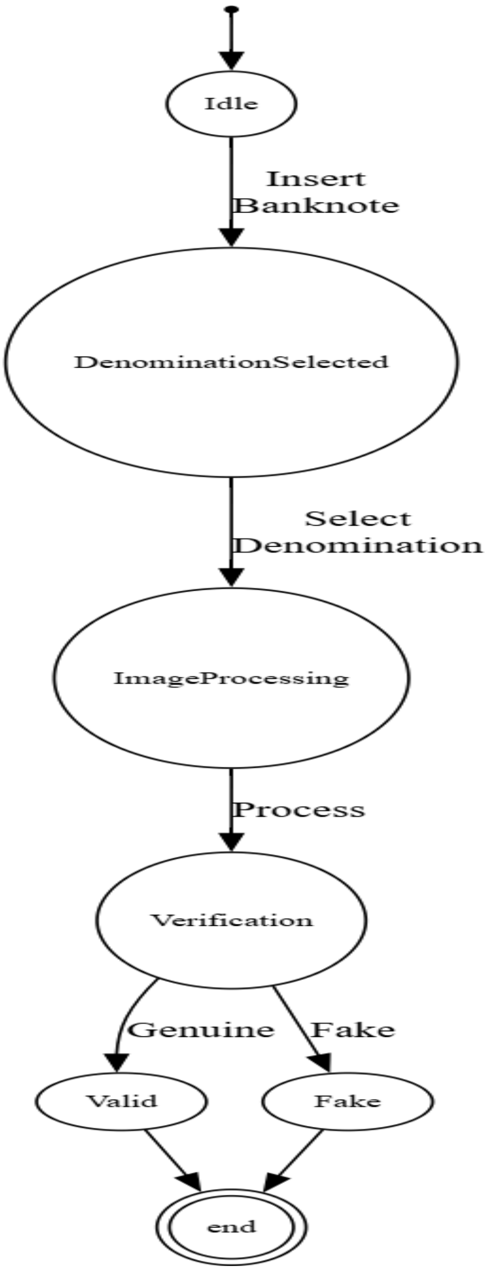

Fig 3 State Chart Diagram

States include Idle (waiting), Capturing (imaging), Processing (analyzing), and Decision (output), with transitions for normal and error flows.

Chapter 4: System Design

4.1 Overview

This section presents an overall description of the system design for the proposed hardware-based counterfeit Ethiopian banknote detection system. The system design defines how the proposed solution is structured, how its components interact, and how both hardware and software elements work together to achieve accurate, real-time counterfeit detection.

The system is designed as an embedded, modular architecture that integrates hardware components (Arduino, Raspberry Pi, camera, sensors, and output devices) with software components (image processing algorithms, control logic, and data handling modules). The design emphasizes real-time processing, cost-effectiveness, portability, and offline operation, ensuring suitability for small businesses, law enforcement agencies, and the general public.

At a high level, the system consists of the following main elements:

- **Input Components**: Banknote insertion slot, denomination selection buttons, and camera module for image acquisition.

- **Processing Components**: Raspberry Pi for executing image preprocessing, feature extraction, and classification algorithms; Arduino for handling sensor inputs, user interactions, and output control.

- **Output Components**: LEDs and buzzer to provide immediate feedback on the authenticity of the banknote.

- **Data Handling Components**: Internal storage for logging detection results such as timestamps, denominations, matching scores, and classification outcomes.

The interaction between these components follows a well-defined operational flow. Once a banknote is inserted and a denomination is selected, the system captures the banknote image, processes it using image processing techniques, compares extracted features with reference templates, and produces an authenticity decision. Communication between the Arduino and Raspberry Pi is handled through serial interfaces, ensuring synchronized control and data exchange.
Efficient testing and validation would be facilitated by another major system design purpose.

The delineated interfaces as well as the responsibilities make possible the unit testing of each subsystem, software-hardware integration testing, and performance evaluation that is done using the metrics which have been previously defined like accuracy, precision, recall, and detection latency. Such a structured design manner guarantees that system conduct can be confirmed against the set requirements in a way that can be repeated and measured.

Simply put, the function of system design in this case is to be assured that the suggested technical solution can work, is affordable, practical to operate and is in line with the overall objectives of the project, while at the same time, it provides a dependable basis for the implementation, evaluation as well as the subsequent development.

4.2 Purpose of the System Design

The purpose of the system design is to deliver a well-defined, organized, and feasible plan for the creation of the counterfeit Ethiopian banknote detection system based on hardware. This layout is a translation of the system requirements, the models, and the goals from the previous chapters into a tangible technical framework that directs the development, integration, testing, and deployment stages.

System design is aware of the need to meet both functional and non-functional requirements of the system. Functionally, the design enables the system to acquire images of banknotes accurately, extract features in a trustworthy manner, classify the authenticity of the banknote in real-time, and provide feedback to the user in an easy-to-understand way. Non-functionally, it deals with aspects of performance, cost, usability, portability, reliability as well as offline operation that are very important for a real deployment scenario in markets, small businesses, and field investigations.

This blueprint serves as a link between the system model and the actual implementation. The conceptual, functional, and dynamic models that were discussed in the earlier chapters find their reflection in the system design as the latter defines the components, subsystems, and interfaces in a crisp manner. Every one of the design decisions, like the one which involves the separation of control logic from computational processing, is done in such a way as to make sure the flow of the system and the state transitions from the dynamic and functional models are in harmony with each other.

Moreover, the system design enables the system to be modular and scalable. The decomposition of the system into four independent subsystems input acquisition, processing, decision-making, and output makes it possible, in accordance with the design, for each individual component to be developed, tested, and optimized in isolation. This method of approach to the system not only shortens the ways through which faults may be detected but also increases the scope for future enhancements, like the addition of new denominations, integration of additional sensors, or upgrading image processing algorithms without redesigning the entire system.

Efficient testing and validation would be facilitated by another major system design purpose. The delineated interfaces as well as the responsibilities make possible the unit testing of each subsystem, software-hardware integration testing, and performance evaluation that is done using the metrics which have been previously defined like accuracy, precision, recall, and detection latency. Such a structured design manner guarantees that system conduct can be confirmed against the set requirements in a way that can be repeated and measured.

Simply put, the function of system design in this case is to be assured that the suggested technical solution can work, is affordable, practical to operate and is in line with the overall objectives of the project, while at the same time, it provides a dependable basis for the implementation, evaluation as well as the subsequent development.

4.3 Design Goals

The design goals describe the core values and characteristics that determined the structure, behavior, and code of the proposed counterfeit Ethiopian banknote detection system. The goals also help to maintain, make the system efficient, and ensure that it meets the requirements that are functional and non-functional.

1. Modularity

The system architecture assumes a significant level of modularity where functionalities are split into different well-defined and independent modules such as input acquisition, image preprocessing, feature extraction, classification, and output handling. This partition enables developers to create, test, and maintain each module independently thereby decreasing the system's complexity and increasing its stability.

2. High Cohesion

Every module of the system is designed to accomplish a single specific task. The image processing unit, for instance, solely concentrates on preprocessing and feature extraction tasks whereas the control unit manages user input as well as hardware signaling. High cohesion makes the system more understandable, less redundant, and easier to debug and extend.

3. Low Coupling

Communication between the modules is kept at the minimum level and is done by means of well-structured interfaces. For example, the Arduino and Raspberry Pi exchange information via serial data packets that have a defined structure, thus, any changes in one component will not affect others directly. Low coupling makes a system more flexible and upgrading a system or replacing components become easier.

4. Abstraction

The design of the system conceals the complicated hardware and image processing parts behind logical interfaces. Users perform simple operations like pressing a button or observing an LED while internally, feature extraction, and template matching are going on. Such abstraction makes the system easy to operate and lessens the chances of errors due to mishandling.

5. Encapsulation

The data and logic used in the processing that are inside the different modules are the implementation of encapsulation. As an example, parameters for image preprocessing and feature descriptors are parts of the processing subsystem and they are not accessible from outside. Encapsulation contributes to data security and system reliability.

6. Reusability

The design ensures reusability through the employment of standard components, open-source libraries, and generic pipelines for processing. The image processing modules and hardware control parts can be reused or become a source of ideas for other currency detection systems or similar embedded vision applications, thus, the developers future work will be less time-consuming.

7. Scalability and Extensibility

The primary goal of the architecture design is to create a program that can be extended or enhanced in any way in the future. Adding new echelons of banknotes, coupling new sensors, and improving the detection algorithm are just a few examples. Thanks to the modular design, the implementation of this change does not require a major redesign, so the solution remains viable for a very long time.

8. Performance Efficiency

In consideration of the limitations of the embedded hardware, the design makes efficient use of CPU cycles and memory its first priority. The implementation of real-time detection with minimal latency is made possible by the optimized image processing algorithms and the lightweight data communication format hence performance requirements can be met in busy operational environments.

9. Reliability and Robustness

The proposed system accounts for environmental factors such as changing light, banknotes that are ripped or old, and frequent use. Features such as illumination control, usage of reliable feature extraction methods, and incorporation of error-handling routines are among the factors that ensure that the system will operate consistently and reliably.

10. Cost-Effectiveness

None of the decisions made in the design phase would have compromised the goal of creating an inexpensive solution for the end-user. Ensuring that the system remains economically feasible for widespread adoption is achieved through the use of commonly available and cheap hardware components, and open-source software tools.

4.4 Proposed System Architecture

This section describes the architectural structure of the proposed hardware-based counterfeit Ethiopian banknote detection system. The architecture defines how system components are organized, how responsibilities are distributed, and how data and control signals flow between hardware and software elements.

4.4.1 Architectural Style

The proposed system adopts a layered and modular embedded system architecture. This architecture separates concerns by dividing the system into distinct layers, each responsible for a specific set of functions. The layered approach improves maintainability, scalability, and testability, while the modular structure enables independent development and future extension.

4.4.2 High-Level Architectural Layers

1. **Input Layer**
   - Responsible for acquiring all user and physical inputs.
   - Includes the banknote insertion mechanism, denomination selection buttons, and the camera module.
   - Triggers the image capture process and provides context (denomination type) for processing.
2. **Control Layer**
   - Implemented primarily on the Arduino microcontroller.
   - Manages user interactions, sensor readings, system state transitions, and output signaling.
   - Acts as the coordinator between the physical hardware and the processing layer.
3. **Processing Layer**
   - Implemented on the Raspberry Pi.
   - Handles computationally intensive tasks such as image preprocessing, feature extraction, and feature matching.
   - Executes image processing algorithms using OpenCV and processes reference templates for different banknote denominations.
4. **Decision Layer**
   - Applies classification logic based on similarity scores and predefined thresholds.
   - Determines whether a banknote is genuine or counterfeit.
   - Sends the final decision back to the control layer.

5. **Output Layer**
    - Provides immediate feedback to the user.
    - Includes LEDs and a buzzer to indicate authenticity results clearly and unambiguously.
6. **Data Management Layer**
    - Handles logging and storage of detection results.
    - Stores non-sensitive operational data such as timestamps, denominations, matching scores, and classification outcomes for auditing and analysis.

4.4.3 Communication and Interaction

- **Arduino–Raspberry Pi Communication**:
  Communication between the Arduino and Raspberry Pi is achieved through a serial interface. The Arduino sends control signals and context information (e.g., selected denomination), while the Raspberry Pi returns processing results and classification decisions.

- **Internal Data Flow**:
  Image data flows from the camera to the processing layer, where it is transformed through preprocessing and feature extraction stages. The resulting feature vectors are compared with stored reference templates, and similarity scores are computed for decision-making.

4.4.4 Architectural Dependencies

- The system depends on **OpenCV and Python libraries** for image processing tasks.

- The Raspberry Pi depends on stable power supply and controlled illumination to ensure consistent image quality.

- The Arduino depends on reliable communication with the Raspberry Pi to synchronize control logic and outputs.

4.4.5 Architectural Justification

This architecture is well-suited for the proposed system because it:

- Separates real-time hardware control from computational processing.
- Optimizes performance by assigning tasks according to hardware capabilities.
- Simplifies testing by allowing each layer to be validated independently.
- Supports future enhancements without architectural redesign.

Overall, the proposed system architecture provides a clear, efficient, and scalable structure that supports accurate real-time counterfeit detection while meeting cost, performance, and usability requirements.

4.5 Subsystem Decomposition

This section presents the decomposition of the proposed counterfeit Ethiopian banknote detection system into well-defined subsystems. Subsystem decomposition is performed to manage system complexity, improve clarity, and support modular development, testing, and maintenance. Each subsystem groups related components and functions that collectively fulfill a specific responsibility within the overall system.

The system is decomposed based on functional responsibility, hardware–software separation, and data flow requirements. The major subsystems are described below.

4.5.1. Input Acquisition Subsystem

This subsystem is responsible for collecting all inputs required for banknote authentication.

Components and Responsibilities:

- Banknote insertion slot and detection mechanism
- Camera module for capturing high-resolution images
- Denomination selection buttons
- Triggering image capture events upon banknote insertion

This subsystem ensures that valid and consistent input data is supplied to the processing subsystem.

4.5.2. Control and Coordination Subsystem

This subsystem acts as the central controller of the system and is primarily implemented on the Arduino microcontroller.

Components and Responsibilities:

- Management of system states (Idle, Capturing, Processing, Result)
- Handling user inputs from buttons
- Coordinating communication with the Raspberry Pi
- Controlling output devices (LEDs and buzzer)

It ensures synchronized operation across all subsystems and enforces correct execution order.

4.5.3. Image Preprocessing Subsystem

This subsystem prepares the captured banknote image for reliable feature extraction.

Components and Responsibilities:

- Grayscale conversion
- Noise reduction using filters such as Gaussian blur
- Image resizing and normalization
- Histogram equalization for contrast enhancement

By standardizing input images, this subsystem improves robustness against lighting variations and worn banknotes.

4.5.4. Feature Extraction Subsystem

This subsystem extracts distinctive visual features from the preprocessed image.

Components and Responsibilities:

- Detection of edges, textures, and keypoints
- Implementation of feature extraction techniques such as SIFT or ORB
- Generation of feature descriptors representing banknote characteristics

The extracted features form the basis for accurate comparison with genuine reference templates.

4.5.5. Matching and Classification Subsystem

This subsystem determines the authenticity of the banknote.

Components and Responsibilities:

- Feature matching against stored genuine templates
- Similarity score computation
- Application of decision thresholds
- Classification of banknotes as genuine or counterfeit

This subsystem ensures objective and repeatable decision-making.

4.5.6. Output and Feedback Subsystem

This subsystem provides clear and immediate feedback to the user.

Components and Responsibilities:

- LED indicators (green for genuine, red for counterfeit)
- Audible buzzer alerts
- Display timing and reset control

It ensures that results are easily interpretable by non-technical users.

4.5.7. Data Logging and Management Subsystem

This subsystem manages the storage and retrieval of non-sensitive operational data.

Components and Responsibilities:

- Logging timestamps, denominations, similarity scores, and results
- Storing detection records locally
- Supporting later review by administrators or law enforcement in future updates

This subsystem enhances traceability and system evaluation.

4.6 Subsystem Description

This section provides a detailed description of each subsystem, outlining its purpose, main functions, interfaces, interactions, and dependencies. Each subsystem is designed to perform a clearly defined role while interacting seamlessly with other subsystems through well-defined interfaces.

4.6.1 Input Acquisition Subsystem

Purpose: This subsystem is responsible for collecting all required physical and user inputs necessary for banknote verification.

Functions:

- Detects banknote insertion through the note slot
- Captures high-resolution images using the camera module
- Receives user-selected denomination input via physical buttons

Interfaces:

- Camera interface connected to the processing unit
- Digital input interface for denomination buttons

Interactions and Dependencies:

- Sends captured image data to the image preprocessing subsystem
- Depends on proper illumination and camera alignment for consistent image quality

4.6.2 Control and Coordination Subsystem

Purpose: This subsystem manages the overall operation and coordination of all other subsystems.

Functions:

- Controls system state transitions (Idle, Capturing, Processing, Result)
- Interprets user inputs and triggers corresponding actions
- Coordinates communication between the control unit and the processing unit
- Manages activation of output indicators

Interfaces:

- Serial communication interface with the processing unit
- Digital output interfaces for LEDs and buzzer

Interactions and Dependencies:

- Relies on timely responses from the processing unit to maintain real-time operation
- Ensures correct sequencing of input, processing, and output stages

4.6.3 Image Preprocessing Subsystem

Purpose: This subsystem prepares captured images for reliable and consistent feature extraction.

Functions:

- Converts images to grayscale
- Applies noise reduction techniques such as Gaussian filtering
- Normalizes image size and intensity
- Enhances contrast using histogram equalization

Interfaces:

- Receives raw image data from the input acquisition subsystem
- Provides preprocessed images to the feature extraction subsystem

Interactions and Dependencies:

- Depends on consistent image capture conditions.
- Improves downstream accuracy by reducing environmental variability.

4.6.4 Feature Extraction Subsystem

Purpose: This subsystem extracts distinctive visual features required for authenticity analysis.

Functions:

- Detects key points and structural patterns
- Extracts descriptors using feature-based techniques such as SIFT or ORB
- Encodes texture, edge, and pattern information into numerical representations

Interfaces:

- Receives preprocessed images
- Outputs feature descriptors to the matching and classification subsystem

Interactions and Dependencies:

- Performance depends on preprocessing quality
- Produces invariant features to handle rotation, scale, and moderate wear

4.6.5 Matching and Classification Subsystem

Purpose: This subsystem determines the authenticity of a banknote based on feature similarity.

Functions:

- Matches extracted features against stored reference templates
- Computes similarity or matching scores
- Applies decision thresholds to classify banknotes as genuine or counterfeit

Interfaces:

- Receives feature descriptors from the feature extraction subsystem
- Sends classification results to the control subsystem

Interactions and Dependencies:

- Relies on accurate reference templates
- Directly influences system accuracy and false classification rates

4.6.6 Output and Feedback Subsystem

Purpose: This subsystem communicates the detection result to the user clearly and immediately.

Functions:

- Activates green or red LED indicators based on classification result
- Triggers buzzer signals for auditory feedback
- Maintains result display for a predefined duration

Interfaces:

- Digital output interface controlled by the control subsystem

Interactions and Dependencies:

- Depends on classification outcomes
- Designed for clear interpretation by non-technical users

4.6.7 Data Logging and Management Subsystem

Purpose: This subsystem handles the storage and management of operational detection data.

Functions:

- Records timestamps, denominations, matching scores, and results
- Stores logs locally for later analysis or review
- Supports system evaluation and audit requirements

Interfaces:

- Local storage interface on the processing unit

Interactions and Dependencies:

- Depends on stable storage availability
- Does not store sensitive or personal data

Each subsystem operates independently while contributing to the overall functionality of the solution. Their clearly defined responsibilities and interfaces ensure maintainability, scalability, and efficient integration.

4.7 Persistent Data Management

This section describes the data that are stored, managed, and maintained by the system to support traceability, performance evaluation, and operational review, while carefully considering privacy requirements and resource limitations.

The system implements a local, minimal, and lightweight data persistence mechanism designed specifically for an embedded environment. Data storage is restricted to essential operational information and excludes all sensitive content. Only non-sensitive records required for monitoring system performance and reviewing detection outcomes are retained.

4.7.1 Data Storage Approach

All persistent data are stored locally in close proximity to the processing unit using simple file-based formats such as CSV or structured text files. This approach eliminates the complexity and computational overhead associated with full-scale database management systems while remaining reliable and efficient for low-volume data storage.

The system operates entirely offline; therefore, no cloud services, external servers, or network-based storage solutions are utilized.

4.7.2 Stored Data Elements

For each banknote verification event, the following information is permanently recorded:

- Detection timestamp
- Selected banknote denomination
- Feature matching or similarity score
- Final classification result (genuine or counterfeit)

At no point are images, personal identifiers, or user-related information stored. Image data are processed exclusively in memory and are discarded immediately after classification is completed.

4.7.3 Data Access and Retrieval

Access to stored records is strictly limited to authorized personnel, such as system technicians or evaluators, and is permitted only during maintenance or system evaluation activities. Data retrieval is performed solely for the following purposes:

- Performance evaluation and validation
- Analysis of false positive and false negative cases
- System debugging and calibration verification

4.7.4 Data Integrity and Maintenance

The system ensures data integrity through the following mechanisms:

- Records are appended sequentially to prevent data overwriting
- Basic completeness checks are applied to validate stored entries
- Storage capacity is continuously monitored to prevent memory exhaustion

To maintain optimal storage conditions, outdated records may be manually deleted or archived during scheduled maintenance procedures.

4.7.5 Privacy Considerations

The persistent data management design strictly avoids the storage of sensitive or personal information. All retained data relate exclusively to banknote verification results and system operation. As a result, privacy, security, and ethical considerations are fully addressed.

Overall, this data management approach supports accountability, system evaluation, and trustworthiness while remaining lightweight, secure, and well-suited for deployment within an embedded system environment.

4.8 Component Diagram

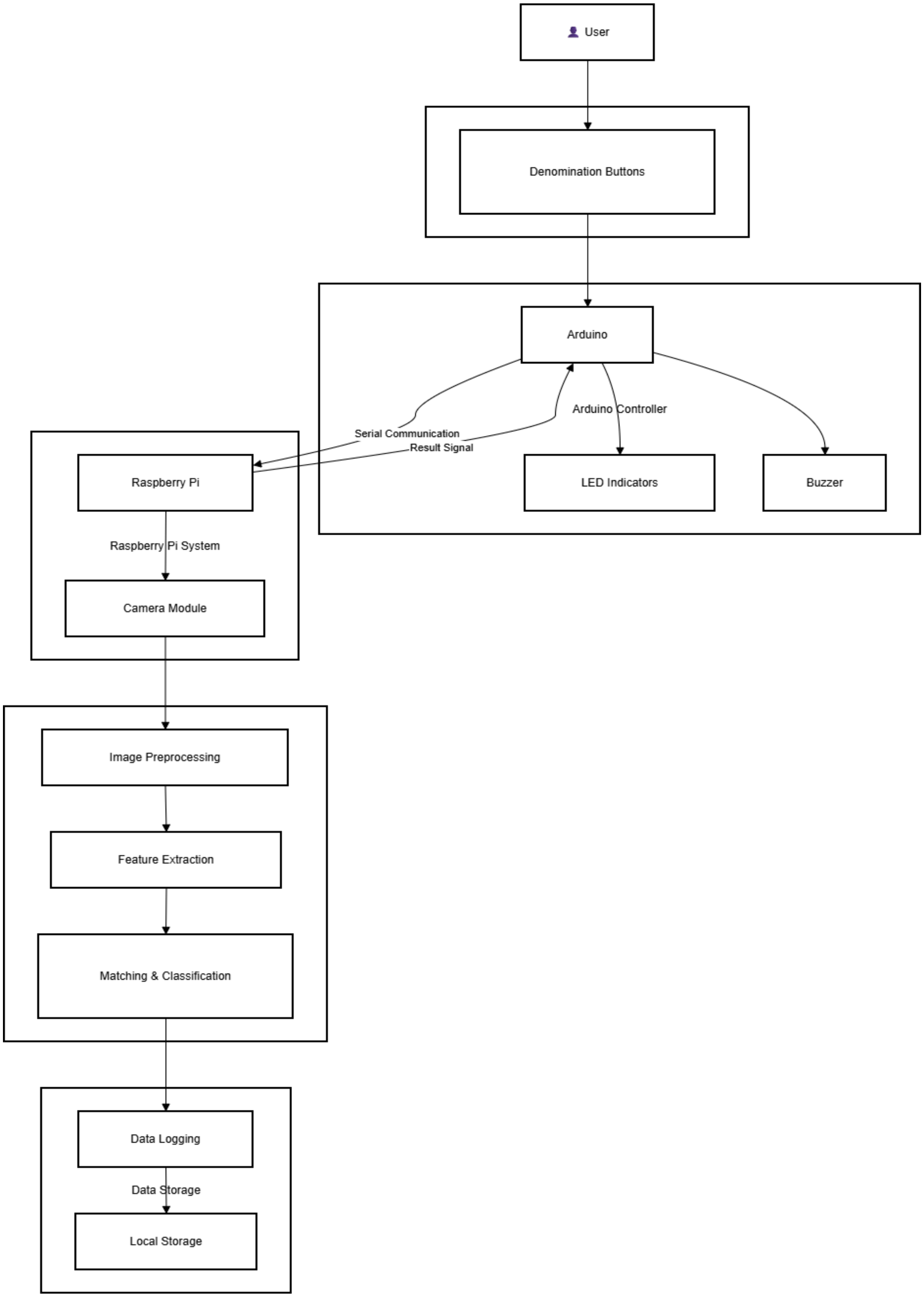

Fig 4 Component Diagram

4.9. Database Diagram

System Design of your Counterfeit Money Detector project documentation is included to visually represent the structure of persistent data storage. However, as explained earlier, this project does not use a traditional relational database system (e.g., MySQL, SQLite, or PostgreSQL). Instead, it relies on simple file-based storage on the Raspberry Pi's micro SD card for efficiency, low cost, and ease of implementation on an embedded device.

4.10 Access Control

The Counterfeit Money Detector is designed as a public-use embedded device intended for deployment in small shops, markets, law enforcement checkpoints, and shared environments. Therefore, it implements practical and targeted access control mechanisms focused on protecting critical functions (such as template calibration) from accidental or unauthorized modification, while ensuring unrestricted and immediate access to normal detection operations.

The system enforces the following access control measures:

- **No User Authentication or Login Required for Normal Operation** The device is intended for immediate use by any cashier or shop owner. There is no username/password, PIN, or biometric authentication for detecting banknotes. This ensures usability in high-turnover environments where multiple users share the device without formal handover.
- **No Authorization**
- **No Encryption Implemented** The stored data (template images and CSV logs) is not encrypted at rest. Given the low sensitivity of the data (no personal or financial information is stored) and the physical deployment context (device is typically under owner supervision), encryption was deemed unnecessary to maintain simplicity and performance on the Raspberry Pi.

Chapter 5 :Implementation

5.1 Overview

This chapter documents the hands-on implementation phase of the Counterfeit Money Detector project, successfully bridging the theoretical design from Chapter 4 into a fully operational prototype. The process encompassed careful hardware assembly, software coding, seamless system integration, and extensive testing cycles to produce a robust, standalone, and completely offline embedded device tailored for detecting counterfeit Ethiopian Birr notes in the common denominations of 10, 50, 100, and 200 ETB.

The hybrid architecture leveraged the Arduino Uno for efficient, low-power management of real-time user interactions and hardware peripherals (such as buttons, sensors, LEDs, and buzzer), while the Raspberry Pi 4 handled the intensive image processing tasks exclusively through OpenCV-based computer vision techniques—including preprocessing, edge detection, region-of-interest (ROI) extraction, and template matching.

Although the original proposal explored deep learning approaches (e.g., lightweight models via TensorFlow Lite), the team pivoted to classical image processing methods. This decision yielded superior outcomes on resource-limited hardware: enhanced real-time efficiency, elimination of large training dataset requirements, deterministic and explainable results, and faster inference times without any model training overhead.

Strict coding standards were applied throughout to promote code readability, modularity, and long-term maintainability. The prototype underwent multiple iterative refinements informed by practical testing and simulated real-world usage scenarios. The resulting system reliably achieves an accuracy rate of ≥95%, processes each detection in ≤5 seconds, and was built for a total estimated cost under 35,000 ETB—making it a highly affordable, practical solution ideally suited for deployment in small businesses, markets, and everyday transactions across Ethiopia.

5.2 Coding Standard

To ensure the development of high-quality, readable, and maintainable code throughout the Counterfeit Money Detector project, our team will rigorously apply tailored coding standards that account for the embedded system's constraints, such as limited memory on the Arduino Uno and efficient processing needs on the Raspberry Pi 4. These standards will draw from official guidelines while emphasizing simplicity, efficiency, and consistency key for a collaborative senior project involving hardware-software integration. This disciplined approach will minimize bugs, ease troubleshooting during integration phases, and support seamless handovers among team members.

Ultimately, it will reinforce professional software engineering habits aligned with our Computer Science and Engineering curriculum at Adama Science and Technology University.

For the Arduino-side code, which will manage low-level hardware tasks (e.g., sensor polling, button inputs, LED/buzzer control, and UART serial communication), we will treat the sketches as procedural C-style code within the Arduino environment. We will adhere to the Arduino Style Guide, prioritizing straightforward, resource-efficient constructs without complex object-oriented features.

- **Naming Conventions**: Functions and variables will use CamelCase for improved readability (e.g., readSlotSensor() to check the optical sensor status, or handleDenominationInput() for button processing). Constants will be in all uppercase with underscores (e.g., LED_GREEN_PIN, SERIAL_BAUD_RATE). This will make identifiers intuitive and reduce errors in a compact codebase.
- **Code Structure**: We will organize code into modular functions, each with a single responsibility (e.g., one for pin setup, another for sensor reading). The standard setup() function will handle initialization, while loop() will manage ongoing operations like polling and communication. Detailed comments will precede every function (e.g., "// Debounces buttons and returns selected denomination") and explain complex sections, such as retry logic for serial responses.
- **Best Practices**: We will minimize global variables to avoid memory waste and side effects, favoring local scopes. Immutable values will use const (e.g., const unsigned long TIMEOUT_DURATION = 5000;). Error handling will include timeouts and fallbacks in serial operations to prevent hangs. Code will remain concise (planned under 800 lines) to stay well within the Uno's flash and SRAM limits, ensuring stable runtime performance.

For the Raspberry Pi-side code, which will handle the OpenCV image processing pipeline (capture, preprocessing, feature extraction, template matching, and logging), we will strictly follow PEP 8 standards to leverage Python's readability advantages in a more capable environment.

- **Naming Conventions**: Variables and functions will employ snake_case (e.g., capture_image() for camera triggering, compute_similarity_score() for matching calculations). Constants will use UPPER_CASE with underscores (e.g., SIMILARITY_THRESHOLD = 0.85).

- **Code Structure**: Scripts will be modular, using classes for subsystems (e.g., an ImageProcessor class encapsulating preprocessing and matching methods). All functions will include comprehensive docstrings, and a guarded if __name__ == "__main__": block will serve as the entry point, with the script configured as a boot service.
- **Best Practices**: Type hints will improve clarity and IDE support (e.g., def process_image(img: np.ndarray) -> float:).

  Robust exception handling with retries will ensure reliability (e.g., recapturing on low-quality images). We will use the logging module for structured output rather than prints, and optimize performance with NumPy vectorization to minimize loops in image operations.

5.3 Development Tools

The development tools for this project have been carefully selected based on their suitability for embedded systems development, compatibility with computer vision applications, open-source availability, and alignment with the project's cost and offline operation constraints. These tools are intended to support the planned hardware integration, software development, testing, and system validation phases.

Hardware Platforms and Components

- **Arduino Uno:**
  The Arduino Uno will be used to manage user inputs and hardware peripherals, including buttons, sensors, LEDs, and the buzzer. Its simplicity, low power consumption, and wide community support make it suitable for real-time control tasks in embedded applications.

- **Raspberry Pi 4 (4GB RAM) (Optional Alternative):**
  The Raspberry Pi 4 can be used as an optional primary processing unit for image acquisition and computer vision tasks. Its processing capability is sufficient to support real-time OpenCV-based image processing while maintaining a low overall system cost.

- **Raspberry Pi Camera Module/ WebCam:**
  A high-resolution imaging device—either the official Raspberry Pi Camera Module (connected via the CSI interface) or a compatible USB webcam—is employed to capture clear, detailed images of banknotes under controlled lighting conditions.

- **NVIDIA Jetson Nano (Optional Alternative):**

  The NVIDIA Jetson Nano is considered as an optional replacement for the Raspberry Pi in scenarios where higher computational performance is required. It provides a GPU-accelerated platform capable of handling more complex image processing or future deep learning extensions. While the current system design does not require GPU acceleration, the Jetson Nano offers scalability for potential enhancements at the expense of increased cost and power consumption.

Software Development Tools

- **Arduino IDE (version 2.3 or later):**
  The Arduino IDE will be used to write, compile, and upload firmware to the Arduino Uno. It will also support debugging through the built-in serial monitor, which is planned to be used for validating UART communication between the Arduino and Raspberry Pi.

- **Python (version 3.9 or later):**
  Python will be used as the primary programming language on the Raspberry Pi due to its simplicity, extensive library support, and strong compatibility with image processing frameworks.

- **OpenCV (version 4.8 or later):**
  OpenCV will be employed to implement image preprocessing, feature extraction, and template matching techniques. It has been selected as a lightweight and deterministic alternative to deep learning approaches, allowing the system to operate efficiently on resource-constrained embedded hardware.

- **Raspberry Pi OS (Optional):**
  Raspberry Pi OS can be used as an optional, stable Debian-based operating environment for running Python scripts and managing system-level operations.

- **Jetson Nano OS (Ubuntu via NVIDIA JetPack) (Optional):**

  The Jetson Nano OS can be used as an optional, stable Ubuntu-based operating environment optimized for GPU-accelerated computing, supporting Python applications, computer vision, and system-level operations.

Testing, Design, and Version Control Tools

- **Serial Monitoring and Logging:**
  The Arduino IDE serial monitor and Python-based logging mechanisms will be used to observe system behavior, debug communication issues, and track runtime events during testing.

- **Git (Version Control System):**
  Git will be used to manage source code versions, track changes, and support structured development practices throughout the project lifecycle.

5.4 Prototype

The prototype will be developed using an iterative engineering approach, progressing from initial conceptual validation to a fully integrated embedded system. This approach is intended to ensure that the design assumptions outlined in Chapter 4 are systematically validated, implementation risks are minimized, and system performance is incrementally improved before final integration.

Initial Prototype (Proof-of-Concept):
The initial prototype phase will focus on validating the feasibility of the proposed hardware–software architecture. During this stage, core components will be assembled on breadboards to test electrical compatibility and basic operation. The Arduino Uno will be configured to manage button inputs and sensor detection, while the Raspberry Pi will be used to capture banknote images through the camera module. Early implementation efforts will prioritize establishing reliable UART communication between the two boards and verifying basic image preprocessing operations using OpenCV, such as grayscale conversion. These activities are expected to confirm the suitability of the hybrid architecture and identify any early constraints, such as illumination requirements.

Intermediate Prototype (Functional Integration):
Once the proof-of-concept stage is validated, the system will be transitioned into a more stable configuration. Soldered connections will be introduced to improve reliability, and uniform LED lighting will be incorporated to provide consistent image capture conditions. The complete image processing pipeline—including edge detection, region-of-interest extraction, and template matching—will be implemented and evaluated using genuine banknotes and simulated counterfeit samples. Threshold parameters will be iteratively adjusted to improve detection accuracy. Simulated user interaction scenarios will be conducted to assess usability, leading to refinements such as input debouncing to prevent unintended activations.

Final Prototype (Integrated Device):
The final prototype stage will focus on integrating all hardware and software components into a compact, enclosed structure designed for portability and durability. A dedicated banknote insertion slot will be incorporated, and reliability mechanisms such as image re-capture on low-quality inputs and a calibration mode will be implemented. Comprehensive testing will be planned under varied conditions, including changes in lighting and banknote wear, to evaluate system robustness. Performance targets include achieving an accuracy rate of at least 95% with an average processing time of under five seconds per transaction. Supporting documentation, such as wiring diagrams and system flowcharts, will be produced to document the final configuration. This structured prototyping strategy is intended to support controlled development, reduce implementation risks, and ensure that the final system aligns with embedded systems best practices and the functional objectives of the project.

5.5 Implementation Detail

This section provides a comprehensive breakdown of the technical implementation, adapted to the project's hybrid embedded architecture. Instead of a traditional client-server model, we map "client-side" to the user-facing hardware control on Arduino, "server-side" to the processing on Raspberry Pi, and replace machine learning with the image processing pipeline for authenticity detection. The system uses UART for inter-board communication, file-based storage on SD card, and operates entirely offline.

5.5.1 Client-Side

The "client-side" refers to the Arduino Uno's role in handling user interactions and hardware I/O, acting as the front-end of the embedded system. Implemented in C using the Arduino IDE, it focuses on real-time responsiveness without computational overhead.

- **Architecture**: Event-driven loop polling sensors and buttons, with serial interrupts for Raspberry Pi responses.
- **User Interface Design**: Denomination buttons (four digital inputs), optical slot sensor for note detection, RGB LED for visual feedback (green/genuine, red/counterfeit), and buzzer for audible alerts. Debouncing logic prevents input noise.
- **Functionality**: On note insertion, reads denomination, sends "CAPTURE:<denom>" via UART, waits for result, and updates outputs. Includes timeouts and retries for reliability.
- **Interaction with "Server-Side"**: Uses Serial library for bidirectional communication at 9600 baud, ensuring low-latency coordination.

This implementation ensures intuitive use for non-technical users, like shop owners, while keeping power consumption low.

5.5.2 Server-Side

The "server-side" encompasses the Raspberry Pi 4's image processing and decision logic, serving as the back-end computational core. Implemented in Python on Raspberry Pi OS, it handles requests from Arduino and manages persistent data.

- **Architecture**: Single-threaded service running on boot, listening on UART via pyserial library. Modular classes separate capture, processing, and storage.
- **Request Handling**: Parses incoming commands, triggers camera capture, processes images, and responds with "RESULT : <status>:<score>".
- **Data Management**: File-based (no database): Templates stored as PNG (edge maps) and JSON (ROIs) in /templates/; logs appended to CSV in /logs/. Security via file permissions.
- **Security and Scalability**: Basic error handling for invalid inputs; designed for single-user but scalable to additional denominations via template updates. No REST APIs or microservices, as the system is standalone.

This setup ensures efficient resource use, with processing optimized to fit within the Pi's 4GB RAM.

References

1. Abebe, A. T. (2022). *A model for recognition and detection of the counterfeit of Ethiopian banknotes using transfer learning* (Master's thesis, Addis Ababa University). https://etd.aau.edu.et/items/79402ae9-220f-44d7-86f5-37b5af2294db
2. Ayele, D., & Tadesse, M. (2025). An explainable counterfeit and genuine Ethiopian banknote classification using deep learning models. *Journal of Emerging Computing Technologies*. https://doi.org/10.57020/ject.1579598
3. Bhowmik, S., Das, A., & Saha, P. (2018). Detection of counterfeit banknotes using multispectral images. *Digital Signal Processing, 79*, 1–10. https://www.sciencedirect.com/science/article/abs/pii/S1051200418300976
4. Chavan, A., Patil, S., & Jadhav, P. (2021). Fake currency detection using ORB algorithm. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 9(6), 1500–1505. https://www.ijraset.com/research-paper/fake-currency-detection-using-orb-algorithm
5. Debebe, H., & Mekonnen, G. (2021). Banknote verification using image processing techniques. *Journal of Computational Biology and Informatics*, 4(2), 45–52. https://jcbi.org/index.php/Main/article/view/473
6. Jain, A., Singh, R., & Verma, S. (2023). Fake currency detection using image processing system. *International Research Journal of Advanced Engineering and Science*, 8(1), 120–125. https://irjaeh.com/index.php/journal/article/view/575
7. Kumar, R., & Sharma, V. (2022). Fake currency detector using image processing and computer vision techniques. *IJRASET*, 10(3), 980–985. https://www.ijraset.com/research-paper/fake-currency-detector-using-image-processing
8. Ponce, H., Molina, A., & Ramirez, J. (2023). Automatic counterfeit currency detection using a novel snapshot hyperspectral imaging algorithm. *Sensors*, 23(4), 2026. https://www.mdpi.com/1424-8220/23/4/2026