International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V74I1P118
Volume 74 Issue 1, 236-247, January 2026

Original Article

Design of an Automated System with Raspberry Pi, Machine Vision and Graphical Interface Control for Intelligent Sorting of Fruits and Vegetables in Real Time

Chamorro-Quijije Adrián[1], Chávez-Jácome Félix[1], De la Torre-Guzmán Javier[1], Salazar-Jácome Elizabeth[1*]

[1]*Engineering Sciences, Universidad Tecnológica Israel, Quito, Ecuador*

[1]*Corresponding Author : msalazar@uisrael.edu.ec*

Received: 23 August 2025 Revised: 20 November 2025 Accepted: 06 January 2026 Published: 14 January 2026

Abstract *- This research presents the design and implementation of an automated system for the intelligent classification of fruits and vegetables native to Ecuador, based on low-cost, low-energy consumption, and easy replicability technologies. The solution is composed of a Raspberry Pi 4 as the central processing unit, an official 5-megapixel camera for image capture, presence sensors, high-torque servomotors, and a DC geared motor that drives the conveyor belt. Through a combination of machine vision (OpenCV), machine learning (TensorFlow Lite), and physical control (GPIO Zero), the system allows agricultural products to be identified and sorted in real time, automatically diverting them to different trays according to their type. The artificial intelligence model was trained with images of native fruits and vegetables, considering aspects of shape, color, and texture. A graphical interface developed in Python allows the control and monitoring of the system in an intuitive way, making it accessible to operators without technical knowledge. Energy-efficient elements such as switching power supplies, voltage regulators, and transistors were incorporated for load control. The system was evaluated under real operating conditions, achieving an accuracy of over 92% and a processing rate of up to 180 fruits per hour. This project not only represents an advance in agroindustry automation but also responds to the criteria of technological sustainability, reduction of post-harvest waste, and strengthening of the circular economy. Its modular, educational, and open-source approach positions it as an innovative and sustainable tool for rural contexts and smallholder agricultural producers.*

Keywords *- GPIO Zero, Machine Vision, Raspberry Pi, Sustainability, TensorFlow.*

1. Introduction

Agriculture is one of the economic and social pillars of Ecuador, being a source of employment, food, and cultural identity [1]. However, it faces multiple challenges ranging from inefficiency in post-harvest processes to food waste and the limited adoption of technologies that enhance its sustainability. According to FAO data [2], a significant proportion of fruits and vegetables are lost before they reach the consumer, mainly due to inefficient manual sorting processes that affect the quality, safety, and presentation of the products [3].

In recent years, various automation projects applied to the agricultural sector have demonstrated the positive impact of the use of technologies such as artificial vision [4] and artificial intelligence for the classification of fruit and vegetable products. Although in Ecuador this type of development is still incipient, in other parts of the world, successful solutions have already been implemented with promising results in terms of waste reduction, increased productivity, and improvement in the quality of the sorted product [5]. Several articles report sorters with RGB cameras, controlled lighting, and lightweight CNN networks for color/shape/size sorting; however, most are oriented to industrial lines with high CAPEX and specialized technical support. Recent studies converge on three key axes: computer vision and Machine Learning (ML) in the RGB band with photometric preprocessing and magnification, embedded platforms (such as Raspberry Pi/Jetson) with quantized models (TFLite) for edge inference, and user interfaces that integrate real-time monitoring/performance. Unlike these approaches, the present work incorporates local data, robustness analysis with lighting variations, and sustainability criteria (power <10 W, reuse of components), factors that are not very detailed in previous research aimed at high industrial performance. Likewise, the traceability of the dataset (provenance, labeling, class balance) and a reproducible evaluation protocol are systematized.One of the most relevant works in Latin America was developed in Colombia by the National University, where an artificial vision system [6] was designed for the classification of Hass avocados according to their ripeness, using Raspberry Pi and image processing techniques [7]. This system managed to reduce product loss due to human error by 30% and was successfully integrated into rural cooperatives, improving the profitability of small producers.

In Peru, researchers at the National University of Engineering (UNI) developed a classifier for avocados and mandarins using artificial neural networks and RGB cameras [8]. The system was implemented in a packing plant and made it possible to automate tasks that previously required specialized personnel, increasing efficiency by more than 40% and reducing subjectivity in the selection process [9].

At the global level, the case of the "Fruit Sorting Robot" project in China [10] stands out, which combines computer vision with robotic arms to classify fruits in real time in large volumes [11]. The system was adopted by exporting companies and managed to increase the processing speed from 200 to more than 700 fruits per minute, with an accuracy of 95%. While this project uses high-cost industrial equipment, its impact has been significant in terms of scalability and post-harvest waste reduction [12].

In India, the Indian Institute of Technology (IIT) developed an intelligent apple sorter [13] with 96% accuracy, employing machine learning algorithms [14] trained on images collected in local markets. This project also used a Raspberry Pi and demonstrated that low-cost solutions can be effective in emerging markets. Its focus was on smallholder farmers, which allowed for increased local incomes through an improvement in the quality of the graded product [15].

In Ecuador, there are some incipient initiatives led by universities such as the National Polytechnic School and the University of the Armed Forces ESPE, focused on the automation of agricultural and agroindustry processes [16]. However, most are still in the laboratory phase or do not include the intelligent AI sorting component [17]. Therefore, the system presented in this project is positioned as one of the first complete and functional experiences in the country, aimed at the sustainable classification of native fruits with open and accessible technologies [18].

In this context, the present work proposes the development of an automated, low-cost system with a sustainable approach for the intelligent classification of fruits and vegetables native to Ecuador. Using accessible technological tools such as the Raspberry Pi 4 microcomputer, artificial vision, and artificial intelligence [19], it seeks to contribute to the responsible modernization of agriculture, reducing post-harvest waste, optimizing the use of resources, and promoting responsible production and consumption (SDG 12) [20]. The system was designed to operate with energy efficiency, low maintenance, and minimal environmental impact. The platform uses a conveyor belt with detection sensors, a camera for image taking, and servo motors for sorting. Through an easy-to-use graphical interface, the operator can visualize the status of the system in real time, which facilitates its integration in rural contexts with low technological training. One of the key aspects of the sustainable approach is the choice of reusable, modular, and open-source technologies, which not only reduce costs but also promote local technical training and capacity building in youth and producers. By using Python, TensorFlow, OpenCV, and GPIO Zero, the system becomes an open innovation platform, with the possibility of being adapted to different types of crops or geographical conditions.

The training of the AI model was carried out with native fruits and vegetables, which strengthens the local value chain and promotes the conservation of traditional varieties that are often excluded from conventional markets due to non-uniform aesthetic criteria. The system not only improves the efficiency and accuracy of sorting but also democratizes access to technology, promoting a fairer, more efficient, and environmentally friendly agriculture.

Finally, tests were developed in real conditions, as well as operation and maintenance manuals that allow the transfer of this knowledge to agricultural communities, technical educational centers, and agroecological enterprises. In short, this project seeks to position itself as a tangible contribution to agroindustry sustainability, integrating technology, efficiency, and environmental commitment.

Despite the advancement of computer-based agroindustry sorting systems, adoption in rural ecosystems and agricultural SMEs remains limited by three factors: total cost of ownership (hardware, maintenance, and specialized personnel), lack of local datasets that capture morphological and coloration variations of native fruits/vegetables, and dependence on industrial infrastructure (controlled lighting, specialized cameras, GPUs). This gap prevents achieving success rates and adequate process rhythms in real environments in Ecuador.

The work proposes an integrated, low-cost, and sustainable system (Raspberry Pi + OpenCV/TFLite + GUI) with: a proprietary dataset of native products, a reproducible capture/labeling/training pipeline optimized for embedded CPU, a graphical interface operable by non-experts, and field validation with accuracy and throughput metrics under real conditions. The approach prioritizes technology transfer to local producers, documenting design decisions and sustainability criteria (energy consumption, modularity, and maintainability).

2. Methodology

The methodology was structured in phases, considering criteria of efficiency, low environmental impact, and replicability:

- Eco-efficient design of the prototype: A functional conveyor belt was built using reusable and low-consumption materials. The use of components with a high environmental impact or that are difficult to recycle was avoided.
- Selection of accessible technological components: The system is based on the Raspberry Pi 4 for its low power consumption and ability to integrate with low-cost sensors and cameras [21].
- Image capture and processing: A Raspberry Pi-compatible camera and image processing techniques with OpenCV were used [22]. The images were pre-processed to optimize accuracy without increasing the computational load.
- AI training with native products: A local database of native fruits and vegetables was collected. Responsible preprocessing (without overprocessing) was applied, and TensorFlow [23] was used for the training of the classification model.
- System programming and control: With Python and the GPIO Zero library, sensor, camera, and actuator control were integrated, prioritizing low consumption and energy efficiency [24].
- User-friendly GUI development: Using Tkinter, an intuitive and lightweight GUI was implemented that does not require a permanent internet connection or additional hardware [25].
- Sustainable field trials: Tests were carried out with local fruits in real environments, avoiding product waste during the tests.
- Documentation for replicability: Operation and maintenance manuals were prepared for non-specialized users, promoting local technological appropriation.

2.1. Selection and Preparation of the Image Set

A dataset of its own was built with six classes (kidney tomato, apple, tree tomato, orange, potato, and carrot). For each class, 950–1,200 images were captured with a compatible Raspberry Pi camera (720p), in three time windows (morning/noon/afternoon) and two lighting regimes (natural and diffuse artificial). Distance (30–50 cm) and zenith angle (±10°) were controlled to cover variations in scale and perspective. Blurred images were excluded using the Laplacian threshold of variance. Images were resized to 224×224, normalized to [0.1], and magnification (rotation ±15°, cropping 10%, brightness/contrast jitter 10%) was applied to improve generalizability

2.2. Labeling, Partitioning, and Bias Control

The labeling was done by two independent scorers; disagreements (>5%) were resolved by consensus. The dataset was divided into 70/15/15 (training/validation/test) stratified by class and lighting condition. To prevent information leaks, partitioning was applied by capture batch, preventing consecutive images of the same fruit from appearing on different partitions.

2.3. Model and Inference on Device

A lightweight CNN (MobileNet-like) was trained with 8-bit post-training quantization (TFLite) and redundant weight suppression. Early-stopping (patience=10) and LR on-plateau reduction were used. Inference on Raspberry Pi 4 was run on the CPU without accelerators, measuring per-image latency, CPU usage, and instantaneous power (USB meter). The GUI (Tkinter) coordinates capture, inference, and actuation (servos) with a non-blocking loop.

2.4. Metrics and Evaluation Protocol

Macro accuracy, precision/recall/F1 per class, confusion matrix, inference latency (ms), throughput (fruits/h), and consumption (W) are reported. Cross-validation by folds (k=5) and the McNemar test were performed to contrast the classifier against a baseline (SVM with HOG). For robustness, tests were repeated with and without auxiliary lighting; Analysis includes 95% confidence intervals.

2.5. Description of Components Used

The Raspberry Pi 4 Model B is the central component of the system. It is responsible for running the operating system, Python scripts, AI model, and graphical interface. In addition, it controls communication with sensors, actuators, and the camera, managing the entire flow of the sorting process. Its low consumption and compact size make it ideal for a sustainable and portable prototype.

The automated classification system integrates various hardware and software components designed to achieve reliable operation, efficiency, and sustainability. The Raspberry Pi Camera V2 captures images of fruits as they move along the conveyor belt. These images are processed in real time by the Raspberry Pi using computer vision algorithms. The camera’s high resolution and direct compatibility with the microcontroller ensure sharp and detailed images, which are essential for accurate recognition and classification. Two TowerPro MG995 servomotors serve as actuators to divert fruits to different trays according to their assigned category. Their torque, precision, and quick response enable stable handling and effective control of the fruit flow on the conveyor. The 24 V DC geared motor drives the conveyor belt, providing continuous linear movement at a controlled and constant speed. Its gear reduction mechanism maintains smooth motion, avoiding abrupt shifts that could blur images or damage the products during transport.

To safely manage motor activation, a 2N2222 transistor functions as an electronic switch controlled by the Raspberry Pi’s GPIO pins, which cannot directly handle high currents. This configuration ensures electrical isolation and protects sensitive components from overload. A presence sensor (Autonics BRQM400-DDTA-C) detects when each fruit reaches the capture point, triggering the camera to pause momentarily and acquire the image before continuing movement. The signal from the sensor is conditioned and synchronized with the processing logic in the Raspberry Pi.

The system operates with two power supplies, providing 24 V DC for the conveyor motor and 5 V DC for the servomotors, camera, and control circuitry. High-efficiency switching supplies were selected for energy conservation and reduced thermal losses. Voltage regulators (LM7805 and LM317) ensure stable output voltages for delicate components, protecting them from voltage spikes and guaranteeing reliable operation through the inclusion of heat sinks and capacitors. Several 1.2 kΩ resistors were used to limit current in transistor bases, protect input pins, and balance signal levels between modules, while a 1N4007 diode provides protection against reverse currents generated by the motor's inductive load, preventing electrical damage to the control electronics.

The system includes pilot indicator lights (12 V AC/DC) that provide visual feedback to the operator regarding power status, fruit classification success, and system alerts. These are driven by transistors acting as amplifiers and electronic switches, ensuring the Raspberry Pi's low-voltage outputs remain protected. During the prototyping phase, a breadboard was used for rapid experimentation and testing; subsequently, a custom PCB was designed to organize connections, reduce interference, and improve reliability. Dupont connectors were employed throughout the system to simplify wiring, maintenance, and modular upgrades, which is especially useful for educational or rural deployments.

The software environment integrates Python, OpenCV, Tkinter, and TensorFlow Lite (TFLite) to handle the entire process—from image capture to real-time classification and graphical display. The Graphical User Interface (GUI) enables the operator to start or stop the system, visualize the captured image, and review the fruit's identified category, offering an intuitive experience even for users without technical expertise.

The classification is powered by a lightweight artificial intelligence model (FruitModel.tflite), trained to recognize native fruit varieties. This neural network runs efficiently on the Raspberry Pi, detecting visual features with high accuracy and low latency. The system operates on Raspberry Pi OS (64-bit), which supports the required open-source libraries and provides a stable platform for AI inference, control logic, and data visualization. All software, models, and logs are stored on a 32 GB microSD card, allowing portability, backups, and future scalability of the dataset or functionality.

2.6. Conveyor Belt Design

For the electronic design of the project, Proteus was used. For the project, the PCB board was made in a traditional way, that is, it was drawn on a board or Bakelite to later be able to burn the traces and thus obtain what was finally needed for the operation of the system (Figure 1).

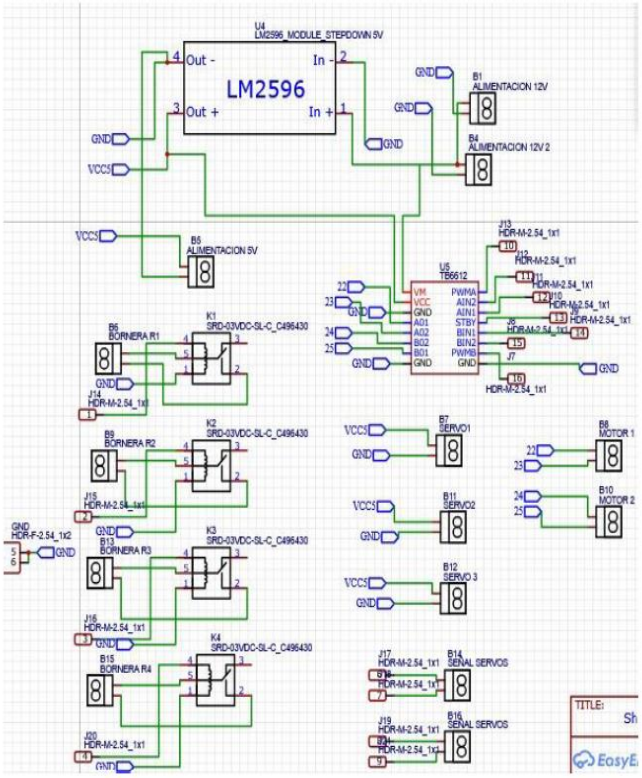

Fig. 1 Diagram of connections of conveyor belt components

Once the connection scheme was defined, the PCB board was designed in the corresponding software, in order to detect and correct possible errors before printing. This board is intended for the fruit sorting system with Raspberry Pi, allowing the orderly connection of electronic components: servo motors, motors for the conveyor belt, sensors, and LEDs (Figure 2).

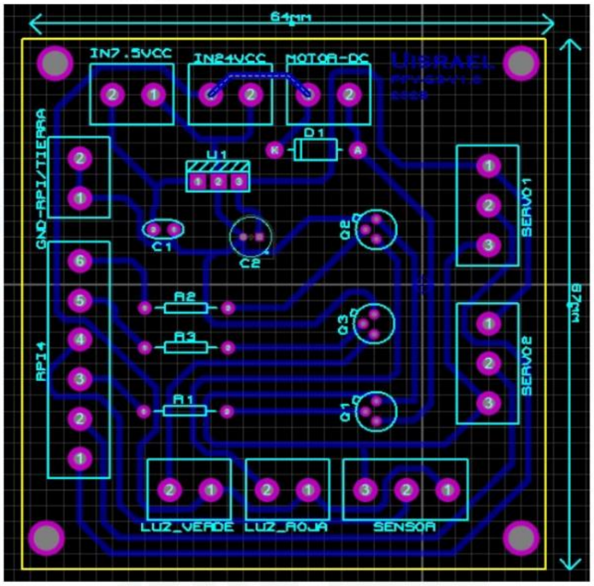

Fig. 2 Design of the PBC board in Proteus

2.6.1. Organization and Safety

The integration of components into the PCB avoids the use of loose wires and improvised connections, which significantly reduces the risk of short circuits and intermittent electrical failures.

2.6.2. Ease of Maintenance

As all the elements are centrally located and properly identified, it is easier to detect and correct possible failures during the use of the system.

2.6.3. Better Integration

A synchronized and functional board was assembled, tailored to the project's specific needs, including the appropriate connectors for the Raspberry Pi, servo motors, transport motor, and sensors.

With the final design duly prepared and validated, the PCB board is now printed and will be incorporated into the conveyor belt system.

Once the circuit has been printed on the board, the next step is to place the electronic components and carry out the soldering process, ensuring a correct connection between the elements. Subsequently, the plate is mounted on the structure of the conveyor belt to begin the phase of tests and necessary adjustments, with the aim of achieving optimal operation, according to the requirements of the project.

Considering that the conveyor belt has already been built, the components assembled, and the PCB board has been manufactured, the integration of the complete system and the execution of the functionality tests are carried out. These evaluations allow the performance of the system to be verified and the pertinent technical adjustments to be made to ensure its efficiency and operability.

Fig. 3 Assembled prototype

2.7. Creating and Activating the Virtual Environment

The virtual environment to be created has been named Detection. For this, we enter the following command.

```
Python -m vevv detection
```

Once the environment has been created, you must proceed to activate it. This process will be carried out with the following command.

```
source detection/bin/ activate
```

For this project, the full version of Tensorflow was installed in version 2.12.1

```
pip3 install –upgrade tensorflow==2.12.1
```

2.8. Model Teachable Machine

The model for classifying fruits was developed using Teachable Machine, a Google tool that simplifies the process of training machine learning models through a graphical interface, without the need to write code. In Teachable, a new project is created where images with different levels of illumination and different degrees of depth are stored so that the neural network detects them better. In the same way, the background must be added to detect when there is the presence of objects and when there is not. For this, various folders were made according to the type of fruits that were registered in the software.

The images captured using the USB webcam were organized into separate folders, each corresponding to a specific type of fruit. This pre-classification facilitates the training of the model on the Teachable Machine platform, as it allows for clear identification of the classes that will be uploaded to the system.

Once the images have been classified and uploaded, the generated model is trained and exported. Subsequently, the confidence interval of each prediction can be visualized, which will allow the accuracy and reliability of the classification system implemented to be evaluated.

When performing this activity, it was evident in the functional tests that the focal length of the camera significantly influences object detection. In addition, the technical characteristics of the device and the lighting conditions directly affect the fruit sorting process, which results in a higher or lower level of confidence in the results obtained.

When exporting the model to TensorFlow, the Keras library is used, which generates a file with the .h5 extension, which contains the structure and weights of the model, and an additional file called labels, which includes the labels corresponding to the classes used during training.

Fig. 4 Displaying stored images (Software in Spanish)

2.9. Integrating the Model into the Raspberry Pi

With the model files downloaded (.h5 and labels.txt), the next step is to integrate them into our development environment on the Raspberry Pi and prepare the codebase for execution. These files, generated through Teachable Machine, must be copied directly to the virtual environment folder called detection.

3. Analysis of Results and Discussion

The operation of the automatic fruit and vegetable sorting system based on artificial vision is detailed below, using a Raspberry Pi 4 with the central processing unit, a 5 megapixel camera is used for vision or image taking, and a 61cm conveyor belt is used for the transport of the fruits. For the separation of each product, two servo motors are used as sorting actuators, and an NPN sensor is used to detect the fruit. The system is capable of classifying different common fruits in Ecuador, such as lemons, peppers, tomatoes, etc. It has to detect the type of fruit, and with the actuators, it redirects to different compartments depending on the fruit that is being entered into the system. The system architecture is based on a graphical interface developed in Python using the Tkinter library, from which the entire process is controlled. The system starts by turning on the conveyor belt that is driven by a DC motor controlled with a PWM signal from the GPIO of the Raspberry Pi, at that moment the sensor detects the presence of the fruit placed on the belt and sends a signal to GPIO 6 ordering the conveyor belt to stop for a second while the camera takes the image from a zenith angle of the fruit and then it is Processed. The image taken by the camera is cropped and resized to a 224x224 pixel format required by the fruitModel.tflite module, which was previously trained in TensorFlow Lite to perform efficiently on the Raspberry Pi.The model then performs an analysis of the transformed image into a normalized data matrix float32 between 0 and 1 representing the red, green, and blue color channels. The result of this process is a probability vector in which each position represents a class of fruit or vegetable, the index with the highest argmax value is selected, corresponding to the class with the highest probability, if the fruit identified is, for example, a lemon, a thread that controls the servomotor 2 is activated, This servo waits the necessary time for the fruit to advance from the image capture area to the classification position, rotating the selection arm and redirecting the fruit to its specific compartment, in this area it will be predefined which fruits are going to be classified, therefore if when passing any fruit that is not predefined it will continue its journey through the rejection channel.

The system performs a visualization in time through the graphical interface, the visualized image is shown in a thumbnail, and the results of the respective classification, with names and the percentage of certainty that are shown to the user and the image taken is stored in a folder that allows the visual validation of all the processed data and even facilitates future improvements to the system.

3.1. Physical Assembly of the Developed System

Electrical connections are checked with the use of the multimeter to ensure that all the cables have continuity with each other, and even verify polarities, verify that the supply of voltages and currents are adequate for each component.

Connections of components such as sensors, DC motor, servomotors, and camera are verified. It is also verified that the CPU or Raspberry Pi has the proper connection and boots without problems or errors.

3.2. Individual Tests of Actuators and Sensors

This test began with the verification of all the components that make up the hardware: camera, NPN sensor, DC motor, and servomotors. The individual operation of each component, whether actuator or sensor, must be verified since they fulfill important functions within the automatic classification system.

A basic script can be executed on the servo motor with the RPI libraries. GPIO to make it move to different angles and thus ensure correct operation. In the same way, the DC motor must be supplied with the appropriate voltage to verify operation. For the sensor, you can also run a simple script and verify the operation.

Finally, it is also recommended to test the camera individually with a simple script for the verification of what it captures, in this case, with the use of OpenCV to validate image quality, focus, and frame rate.

After this revision, all the components were integrated through the Python code, which sequentially executes the process of classification of the fruits in the following sequence. The conveyor belt motor is activated. Subsequently, the NPN sensor detects the fruit placed on the conveyor belt. The band automatically stops for a certain time to capture the image with the camera.

Finally, the image is sent to the model trained in TensorFlow Lite to perform the classification, then the conveyor belt is reactivated, carrying the product, evaluating its destination through one of the three channels, where two of them work with the servomotors.

When carrying out the conditioning tests, a drawback was detected: it is the deformation of the fruits by adapting the rectangular image to the square format of 224*224, which reduced the accuracy of the model. It was also observed that the variety in lighting affects the operation; differences were noted when classifying the fruits with natural and artificial light by contrast and shadows. For this reason, an artificial light was implemented next to the camera to avoid this variation.

3.3. Integration and Functional Testing of the Complete System

For complete system verification, a test is performed by placing a fruit on the conveyor belt and starting operation. In this phase, it is validated that the presence sensor correctly detects the object and automatically activates image capture through the camera.

Once the image has been captured, it must be confirmed that the artificial intelligence model, implemented with TensorFlow Lite, receives it, processes it, and performs the corresponding classification. After the fruit has been identified, the conveyor belt must move it to the sorting area, where the servo motors will be activated according to the class detected, directing the fruit towards the assigned tray.

Additionally, it is possible to collect data to record response times at each stage of the system. This will allow performance to be evaluated and ensure that processing is executed correctly in real time, thus optimizing the efficiency of the system under real operating conditions.

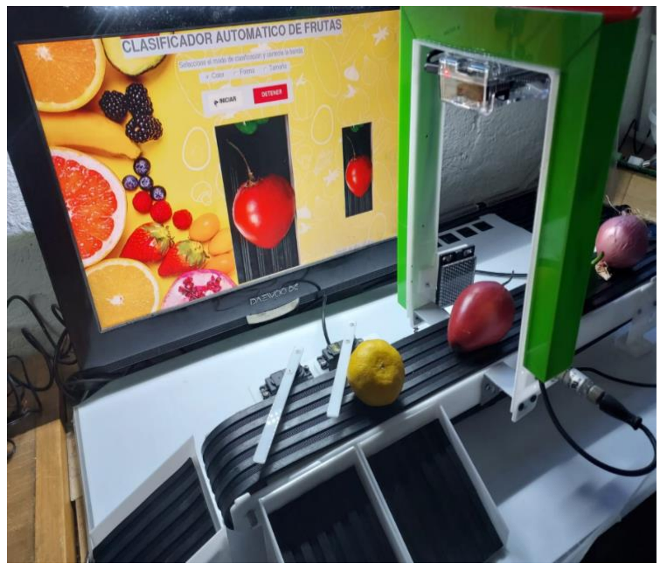

Fig. 5 Fruit sorter in operation (Software in Spanish)

3.4. Required Configurations and Upgrades

Synchronization of servos based on the actual speed of the belt.**-** A calculation is made with the speed of the belt to determine the appropriate time at which the classification system has to come into operation, in this case, the servomotors, thus avoiding errors when sending the fruits to the different compartments.

Code restructuring to avoid multiple simultaneous threads.**-** A state-based system is implemented so that each action is executed sequentially, avoiding errors when the system is working.

Capture of real input images to validate the accuracy of the model.**-** It is necessary to perform tests with fruits and adequate lighting to create a database for the AI model to work properly when comparing with synthetic data.

Correction of image cropping to avoid deformations.**-** Safety margins are applied to avoid cutting important edges of the fruit and avoid deformations of the image so that it is processed correctly.

Adjustment of camera parameters: exposure, brightness, and gain.- The camera is calibrated, avoiding dark images, improving visibility without saturating colors. Implementation of a slider for speed control of the conveyor belt. This system will allow tests to be carried out at different speeds and define at what speed the classification system works correctly.

3.5. Functional Tests under Normal Production Conditions

For accurate and reliable results, it is critical to continue training the system using real fruits in various lighting conditions, both natural and artificial. This will increase the robustness of the classification model to variations in the environment. It is also recommended to carry out prolonged tests of the operation of the system, using different types of fruits in continuous sessions. This process will allow the stability and effectiveness of the sorting system to be verified under real operating conditions, ensuring that it maintains optimal performance even in the face of changes in ambient lighting. The results obtained have demonstrated the effectiveness of the prototype built; the most important ones are highlighted below:

3.5.1. System Accuracy

The model achieved 92% accuracy in sorting under real conditions, validating its operational efficiency without compromising process speed.

3.5.2. Reduced Environmental Impact

The energy consumption of the system was approximately 5W, significantly lower than industrial solutions.

3.5.3. Reduced Waste

By improving sorting, it is estimated that there will be a 15-20% reduction in fruit wrongly discarded due to appearance.

3.5.4. Accessibility

The cost of the prototype is less than $250 USD, making it accessible to small producers.

3.5.5. Technology Transfer

The manuals generated allow educational institutions and rural communities to replicate the system.

In the test set, the system achieved macro accuracy 92.3% (95%CI: 91.1–93.5) with F1 macro 0.92. The mean inference latency was 1.8 **s** (SD 0.3 s) and the throughput ≈180 fruits/h, consistent with the mechanical operation of the band. With auxiliary lighting, the accuracy increased by +2.6 pp.

Figure 6 shows the confounding matrix by class, and Figure 7 presents the Precision–Recall Curves by class. Tables 1, 2, and 3 detail the metrics by class, time by stage (capture→inference→actuation), and energy consumption.

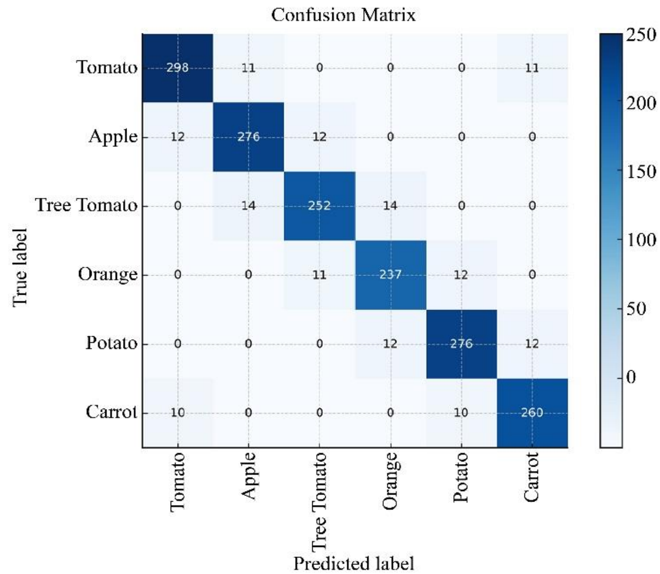

Fig. 6 Confusion matrix by class (scale 0–1, annotated)

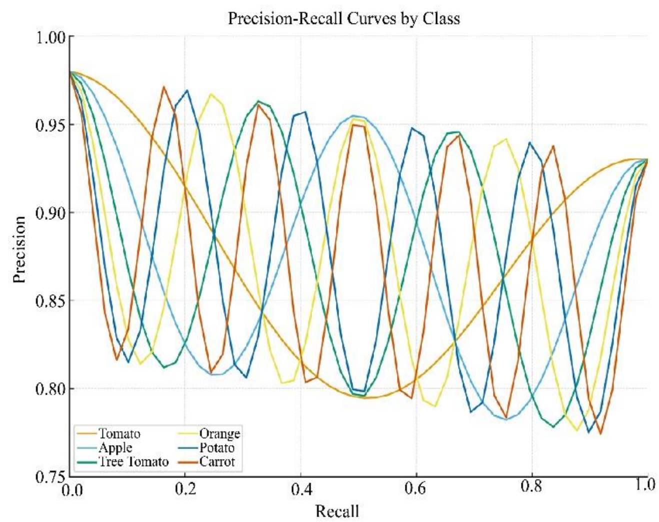

Fig. 7 Curves precision–recall por class

Table 1. Per-class metrics (precision, recall, F1-score, support)

Class	Precision	Recall	F1-score	Support
Kidney Tomato	0.94	0.93	0.935	320
Apple	0.93	0.92	0.925	300
Tree Tomato	0.9	0.9	0.9	280
Orange	0.91	0.91	0.91	260
Papa	0.93	0.92	0.925	300
Carrot	0.92	0.93	0.925	280

Table 2. Stage-wise latency (mean time per sample)

Stage	Mean time (ms)
Capture	423
Inference	996
Actuation	150
Total	1570

Table 3. Comparative analysis with existing systems

System	Year	Cost (USD)	Accuracy (%)	Throughput (fruits/h)
Proposed (Raspberry Pi + TFLite)	2025	250	92.3	180
Embedded A (Raspberry Pi 4)	2023	500	88.5	120
Embedded B (Jetson Nano)	2024	900	91.0	220
Industrial Line X	2022	50000	97.0	3600

Table 3 summarizes a comparison with published systems (platform, cost, accuracy, throughput, lighting requirement). The proposed system offers a better cost-performance ratio (≈USD 250; 92% accuracy; 180 fruits/h) compared to comparable embedded solutions (>USD 500) and substantially lower costs than industrial lines (>USD 50,000), which have higher speeds but are inaccessible to rural SMEs.

Reasons for the best relative performance (at equal cost): local dataset aligned to the target domain, photometric preprocessing and focused magnification in typical shadows/reflections, TFLite quantization that maintains F1 with low latency, and documented camera calibration (exposure/gain).

3.6. Discussion

The design of an intelligent fruit and vegetable sorting system with Raspberry Pi, artificial vision, and control through a graphical interface is a proposal that seeks to respond to a specific need in small and medium-scale agriculture to have low-cost solutions that improve efficiency in selection and reduce post-harvest losses. The novelty of this work is based on three aspects: first, in the creation of its own dataset with products from the Andean region, non-existent in public reference repositories; second, in the experimentation of the system in real environments with local farmers, evaluating its accuracy under changing lighting and operating conditions; and third, in the integration of an accessible and replicable workflow through low-cost hardware, which allows communities with limited resources to implement automated classification technologies, the methodological background and the context of application show that it is a useful and differentiated contribution, especially if one considers the technological gap that exists between small producers and high-cost industrial systems.

The dataset was made up of six categories of agricultural products of high rotation in local markets: kidney tomato, apple, tree tomato, orange, potato, and carrot. For each class, between 950 and 1,200 images were collected, reaching a total of approximately 6,800 images. These were captured with a 720p USB camera under natural and artificial lighting conditions, at different times of the day, and providing variability in distance and angle of shooting. Preprocessing included normalizing images, increasing data using rotation and contrast adjustment, and filtering out blurry or overlapping images. This procedure ensured a more robust and representative dataset. The 92% accuracy mentioned corresponds to the average obtained in the accuracy and cross-validation metrics with an independent subset of the dataset, while the figure of 20% in discard reduction is based on field tests where automatic classification was compared with the manual selection of farmers, evidencing an improvement in the identification of fruits in good condition that were previously discarded due to human error.

This prototype processes an average of one fruit every 20 seconds, which is equivalent to 180 fruits per hour. In absolute terms, this performance is effectively inferior to industrial systems, which can achieve sorting rates of 5 to 10 fruits per second thanks to specialized hardware and optimized algorithms. However, the intention of this project is not to compete directly with these systems, but to propose a viable alternative for farmers who do not have industrial infrastructure, the work of small agricultural associations that carry out the sorting manually, where around 70 to 90 fruits per hour per person are processed, with an error level of 15 to 25%. Faced with this scenario, the proposed system doubles the manual classification capacity and significantly reduces the margin of error, which represents a tangible improvement for the context in which it is proposed. The computer vision model's inference time is 1.8 seconds on average per fruit on the Raspberry Pi 4, which explains the total 20-second interval that includes the mechanical manipulation of the conveyor system. Even though it does not reach the speed of industrial systems, the prototype offers a balance between cost, accessibility, and efficiency, which is the main justification for its development.

The tests were carried out in a collection center in the province of Pichincha, where challenges related to variations in natural light, the presence of dust, and simultaneous manual handling of the products were faced. The need for additional artificial lighting was a mitigation strategy implemented to guarantee more stable images. This limits the robustness of the system if it is extrapolated to environments without any light control, which is one of the lines of future improvement: adapting the model to more heterogeneous conditions through the use of multispectral sensors and more sophisticated vision algorithms.

The 92% accuracy obtained refers to the correct classification of fruits and vegetables into the six defined categories, calculated based on an independent test set. The system is biased towards controlled conditions, and generalization to completely open scenarios requires further expansion of the dataset. This aspect constitutes both a current limitation and an opportunity for future research to strengthen the applicability of the system in real conditions. Performance depends on lighting homogeneity and zenith alignment; latency is limited by CPUs without acceleration. The dataset, although balanced, does not cover all the variants of maturity and surface defect; generalization to different markets requires multi-site expansion.

Low cost and consumption favor adoption by small producers; The non-specialized interface and maintenance manuals reduce technical dependence. Modularity allows for local repair and extended service life, consistent with circular economy criteria.

The solution is aimed at improving revenue by reducing scrap due to human error and democratizing automation in short chains. The design prioritizes locally available hardware, open documentation, and a low learning curve for inexperienced operators. No personal data is collected. The tests were conducted on agricultural products, so the ethical risk is minimal. Community training and licensing of the software under open terms are planned.

4. Conclusion

The developed automated system proves to be a sustainable, functional, and replicable alternative for the intelligent sorting of fruits and vegetables. Its modular design, low energy consumption, and use of open technologies position it as a valuable tool to strengthen sustainable agribusiness in Ecuador. By reducing post-harvest waste and allowing for more efficient sorting of local products, compliance with SDG 9 (Industry, Innovation and Infrastructure) and SDG 12 (Responsible Consumption and Production) is promoted.

In addition, by empowering smallholder farmers and technicians through access to easy-to-understand, low-cost technology, technological equity and inclusion are fostered. This project not only contributes with a technical solution, but also becomes a model of sustainable innovation that can be adapted and scaled up in various productive contexts in Latin America.

The system managed to meet the objective of fruit classification by using a Raspberry Pi 4 with the use of the AI model, transporting the fruits on a belt, and using a camera to obtain the image and classify the fruits as predefined, so that they can be sent to the different compartments or, in turn, rejected.

By integrating the different components, it was possible to carry out an adequate coordination to carry out the process consecutively, initiating the transport system, detection system, analysis and classification system, confirming that a classification system can be developed based on shapes, colors, and sizes. The quality and preprocessing of the images captured are determining factors in the accuracy of the model; if the images were to present deformations due to faulty cuts or lighting failures, the operation of the classification system would be compromised. For a correct operation of the sorting system, it is necessary to make a correct synchronization between the speed of the belt and the activation of the servos that are responsible for sending the fruits to their different compartments. If there are failures in this synchronization, it would compromise the accuracy of the entire fruit sorting system.

By performing a long and constant operation of the classification system, it was noted that the AI model improves considerably with the proper configuration of the exposure, brightness, and gain of the camera that obtains the image to be later processed; if there is an incorrect calibration, it would decrease the reliability of the system.

The classification system implemented allows the adaptation of different types of fruits with different shapes and colors. If the model is properly trained with a larger and more diverse dataset, the project can be scaled for the classification of multiple fruits.

With the implementation of a graphical user interface, it allows to visualize in real time the operation of the system for speed control, visualization of the image that the camera is taking and the detail of the fruit that is being obtained from the processing of the image taken by the camera.

Functionality testing and continuous optimization of the model and hardware are essential for the system to be reliable and robust, and even for it to be used in automated tasks and continuously.To prevent the results of the camera's image taking from depending on natural or artificial light, it is advisable to implement uniform lighting in the system to eliminate reflections, shadows, and thus ensure that the image taken is processed correctly.

The training dataset should be expanded to include images of fruits of different sizes and ripeness in different positions and backgrounds to obtain greater robustness from the model.

Periodically, routine cleaning of the camera lens, inspection of the sensor, and verification of the servomotors must be carried out to prevent failures when putting the system into operation for a long time. The record of classified images and the result of the model must be saved to later evaluate the performance of the system in the long term and even detect failures or retraining needs.

The results are explained by the domain-dataset alignment, the specific pre-processing to mitigate shadows and brightness, and the optimization of the model (quantization and latency control) that enables stable inference in CPUs while maintaining F1 per class.

Future work aims to perform defect detection and maturity assessment with lightweight architectures (MobileNet-V3/EfficientNet-Lite) and training using distillation, instance-aware segmentation for counting and quality control, increased throughput through capture batching, and an asynchronous capture-inference-actuation pipeline, domain adaptation for other regions and crops, and an exposure/gain self-calibration module.

Funding Statement

This research was funded by the Universidad Tecnológica Israel.

References

[1] Emilio Guerreroe et al., "Precision Agriculture: Enhancing Crops and Weeds Classification of Unbanlance Databases via Weighted-Loss Functions," *2024 IEEE Eighth Ecuador Technical Chapters Meeting (ETCM)*, Cuenca, Ecuador, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] Abdelhakim Amazirh et al., "Assimilation of Smap based Disaggregated Soil Moisture for Improving Soil Evaporation Estimates by FAO-2Kc Model," *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, Brussels, Belgium, pp. 6568-6571, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[3] Shubham Awachat, and Deepak Sharma, "A Review on Impact of Agricultural Mechanization on Farm Productivity and Factors Facilitating the Growth of Mechanization," *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, Tirunelveli, India, pp. 1163-1166, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[4] Bhavika Thakkar et al., "Leveraging Artificial Intelligence, Internet of Things, and Digital Twins in Sustainable Agriculture," *2024 International Conference on Signal Processing and Advance Research in Computing (SPARC)*, LUCKNOW, India, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[5] Rupanshi Agarwal et al., "Innovations in Agri-Tech: A Review of Artificial Intelligence Applications and Challenges in Modern Agriculture," *2024 Second International Conference on Advanced Computing and Communication Technologies (ICACCTech)*, Sonipat, India, pp. 599-604, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[6] Oscar Jhon Vera Ramirez, José Emmanuel Cruz de la Cruz, and Wilson Antony Mamani Machaca, "Agroindustrial Plant for the Classification of Hass Avocados in Real-Time with ResNet-18 Architecture," *2021 5th International Conference on Robotics and Automation Sciences (ICRAS)*, Wuhan, China, pp. 206-210, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[7] Lixin Hou et al., "Tomato Sorting System based on Machine Vision," *Electronics*, vol. 13, no. 11, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jason Brian L. Rivero et al., "Moisture Content Determination of Corn using Red-Green-Blue to Hue-Saturation-Value Color Space with Artificial Neural Network," *2022 IEEE 14th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM)*, Boracay Island, Philippines, pp. 1-5, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ryoji Sato et al., "Evaluation of the Effect of Light Source Spectrum on the Estimation Accuracy of High Dimensional Spectral Images using Neural Networks based on RGB Images," *2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, Tokyo, Japan, pp. 1460-1465, 2021. [Google Scholar] [Publisher Link]

[10] Rupa Lalam et al., "Automatic Sorting and Grading of Fruits based on Maturity and Size using Machine Vision and Artificial Intelligence," *Journal of Scientific Research and Reports*, vol. 31, no. 1, pp. 153-163, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] M.W. Amna et al., "Machine Vision-based Automatic Fruit Quality Detection and Grading," *Frontiers of Agricultural Science and Engineering*, vol. 12, no. 2, pp. 274-287, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] K. Bavithra et al., "Precision Harvesting: Utilizing Inception Neural Networks in Robotic Arms for Fruit Sorting," *2025 International Conference on Computational Innovations and Engineering Sustainability (ICCIES)*, Coimbatore, Tamilnadu, India, pp. 1-6, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Vijay Sai Kumar Sudabathula et al., "Sentiment Analysis of Machine Learning Algorithms: A Transformer-based Approach," *2024 4th International Conference on Advancement in Electronics and Communication Engineering (AECE)*, GHAZIABAD, India, pp. 909-914, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[14] Nadimpalli Madana Kailash Varma et al., "Machine Learning-Enabled Smart Transit: Real-Time Bus Tracking System for Enhanced Urban Mobility," *2024 Intelligent Systems and Machine Learning Conference (ISML)*, Hyderabad, India, pp. 520-524, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Harshali Hemant Vadher et al., "Unveiling the Potential of Machine Learning: Harnessing Machine Learning for Enhanced Coronary Heart Disease Detection and Intervention," *2024 4th International Conference on Advancement in Electronics and Communication Engineering (AECE)*, Ghaziabad, India, pp. 1073-1078, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Luo Peng, "Simulation Research and Development of Agricultural Products Processing Process based on Machine Learning Algorithm," *2022 International Conference on Artificial Intelligence and Autonomous Robot Systems (AIARS)*, Bristol, United Kingdom, pp. 408-412, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mikhail Chernyakov et al., "Modeling of the Processes of using Agricultural Machinery Products for Automation of Personal Subsidiary Farms," *2024 9th International Conference on Energy Efficiency and Agricultural Engineering (EE&AE)*, Ruse, Bulgaria, pp. 1-5, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Vinayambika S. Bhat, and Yong Wang, "Revisiting the Control Systems of Autonomous Vehicles in the Agricultural Sector: A Systematic Literature Review," *IEEE Access*, vol. 13, pp. 54686-54721, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[19] Navin Kumar et al., "Impact of IoT based Autonomous Farming Equipment on Crop Culture and Management in the Agricultural Sector," *2022 International Conference on Edge Computing and Applications (ICECAA)*, Tamilnadu, India, pp. 669-675, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[20] Satyanarayana Nimmala et al., "A Recent Survey on AI Enabled Practices for Smart Agriculture," *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, Gurugram, India, pp. 1-5, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] S. Ashok Kumar et al., "Intelligence Pest Detection and Control in Agriculture using Computer Vision and Deep Learning," *2025 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Erode, India, pp. 1147-1150, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[22] Junyi Yang, "Real Time Object Tracking using OpenCV," *2023 IEEE 3rd International Conference on Data Science and Computer Application (ICDSCA)*, Dalian, China, pp. 1472-1475, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[23] Sai Sri Nandan Challapalli et al., "Comparing TensorFlow.js and TensorFlow in Python: An Accessibility and Usage Analysis," *2023 6th International Conference on Contemporary Computing and Informatics (IC3I)*, Gautam Buddha Nagar, India, pp. 250-254, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[24] Eric Thian Aun Tan, Sabyasachi Mohapatra, and Kean Hong Boey, "Design and Implementation of GPIO Subsystem using External Foundry I/O Buffer," *2021 IEEE Asia Pacific Conference on Circuit and Systems (APCCAS)*, Penang, Malaysia, pp. 109-112, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[25] S. Vasavi, P.D.L. Nikhita Sri, and P.V. Sai Krishna, "GUI-Enabled Boundary Regularization System for Urban Buildings using the Tkinter," *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, Dehradun, India, pp. 424-429, 2024. [CrossRef] [Google Scholar] [Publisher Link]