MEKATRONIKA: JOURNAL OF INTELLIGENT MANUFACTURING AND MECHATRONICS
E-ISSN: 2637-0883
VOLUME 8, ISSUE 1, 2026, 1 – 11
DOI: https://doi.org/10.15282/mekatronika.v8i1.13367

RESEARCH ARTICLE

Design and development of an automated robotic system for classification and sorting using machine vision

Fiyinfoluwa David Osokoya[1], Kanyinsola Taiwo Kehinde[1], Kafayat Oluwatoyin Shobowale[1]*, Yekini Bello[1], Muyideen Omuya Momoh[2], Sabur Ajibola Alim[3]

[1]Department of Mechatronics Engineering, Air Force Institute of Technology, Kaduna, Kaduna State 800283, Nigeria
[2]Department of Computer Engineering, Ahmadu Bello University, Zaria, Kaduna State 800001, Nigeria
[3]Department of Mechanical Engineering, Ahmadu Bello University, Zaria, Kaduna State 800001, Nigeria

Abstract - The growing increase in the demand for goods has pushed automation to the forefront of industrial innovation. However, most manufacturing industries still rely on traditional automated means of production, which are slow and inefficient for handling concurrent increases in production. In this study, the design and development of an automated robotic system that integrates a machine vision algorithm through a webcam to detect and classify objects on a conveyor system based on two classes is presented. A 2-degree-of-freedom (DOF) robotic arm controlled by an ESP32 sorts the product into the designated bins based on a specified servo angle. The machine vision algorithm achieved a mean average precision (mAP) of 99.1%. The results of the overall robotic system setup for classification and sorting produced an accuracy of 94% and 92%, respectively. This is a robotic system framework that is geared toward Industry 4.0. Future work will incorporate more classes and optimisation techniques to further improve the system's accuracy.

Article History
Received : 2 November 2025
Revised : 10 January 2026
Accepted : 7 June 2026
Published : 30 June 2026

Keywords
Automation
Image Processing Industry 4.0
Machine vision
Robotic arm
YOLOv11

1. Introduction

Industry 4.0 has placed automation at the forefront of industrial innovation, where machines can perform complex tasks with minimal human intervention. This fourth industrial revolution, which signifies the next phase of manufacturing digitisation, is driven by trends such as data and analytics, the Internet of Things (IoT), minimal human intervention, and improved robotics to increase productivity, reduce errors, and increase flexibility on the shop floor [1]. While Industry 4.0 lays the foundation for smart manufacturing, Industry 5.0 refines this vision by reintroducing the human touch into automation. This industry involves collaboration between humans and machines, combining human creativity and decision-making with intelligent systems [2]. Importantly, Industry 5.0 is a value-driven paradigm that complements, rather than replaces, the technology-driven focus of Industry 4.0. It is about automation, efficiency, and real-time data exchange through cyber-physical systems and IoT. Its three interconnected core values are human-centricity, sustainability, and resilience. This means that technology adapts to human needs, respects planetary boundaries, and safeguards operations against crises. However, Industry 4.0 has not ended, as it continues to evolve with the principles of Industry 5.0, forming a techno-social revolution that combines cutting-edge automation with a purpose-driven agenda [3,4]. Automation aims to perform tasks with minimal or no human intervention. "To achieve a high level of automation combined with flexibility, human skills must be described and transferred to automated machines [5]. Despite the increasing evolution of automation in most modern manufacturing industries, there is still a significant reliance on traditional automation systems, which are slow, inefficient, and challenging to scale [3]. As a result, they fail to keep up with the exponential increase in production during the Fourth Industrial Revolution. These systems lack the adaptability to handle dynamic tasks. To handle the increasing demand for production in this era, a more reliable, efficient, and autonomous method of solving this problem is required [6].

Another key feature of the Fourth Industrial Revolution is the implementation of vision technology. Machine vision, a subfield of computer vision, is a key solution for adaptive automations. Once set up and calibrated, machine vision systems can operate with minimal human supervision for all-day-round operations, increase throughput, and reduce labour costs [7]. Machine vision refers to the use of cameras, sensors, and computer algorithms to allow machines to "see" and interpret visual information [8]. Machine vision uses cameras, sensors, and complex algorithms to achieve this goal. This allows systems to acquire, process, and interpret visual data, mimicking human visual cognition [9]. The proposed solution to the slow and inefficient processes of traditional automation techniques is to create an automated robotic system integrated with machine vision to improve its accuracy, efficiency, productivity, and adaptability. This technology enables real-time object recognition, defect detection, object monitoring, and tracking. Existing implementations of traditional automation rely on hardware, such as light detection and ranging (LIDAR) sensors [10], which are suitable for industrial applications but fail to capture the entire view in large-scale industrial applications.

Many studies have integrated machine vision to perform detection alone or in combination with robotic systems. For instance. For instance, a colour-sorting system presented in [11] used a webcam to capture images of products on a conveyor and performed image processing to determine the dimensions and geometry of the objects by converting the images to binary through thresholding and applying a Gaussian filter to remove noise. The Canny edge detection algorithm in EmguCV was applied for colour detection and object classification. A five-degree-of-freedom (DOF) robotic arm controlled by six servo motors was used to sort the objects on the conveyor after an infrared (IR) sensor detected them. The system successfully sorted the objects with a minimal error rate of 0.08% to 2.653% in classifying and sorting

*CORRESPONDING AUTHOR | K.O. Shobowale | ✉ kshobowale@gmail.com

the colours. These earlier automated systems used classical techniques, such as the MATLAB image processing tool [12] and the Pixy camera for colour-based object detection [13], to improve the operating efficiency of traditional automated systems. Although these classical approaches have improved traditional automation systems, their performance is limited in complex, dynamic industrial environments, creating the need for more robust and intelligent vision-based solutions. Other researchers have explored more advanced methods of computer vision; in the research presented by [14], the researcher demonstrated an evaluation of the performance of a convolutional neural network (CNN) and support vector machines (SVMs) in a waste classification and sorting system. SVM was used as a classifier for the feature extraction algorithm, and CNN was used for classification and feature extraction to identify the garbage image dataset. The system utilised two pre-trained models: AlexNET and VGG19 model with the VGG19 model. The accuracy was 97%. This study provided the best prediction methods and effectively distinguished between different types of waste, with the best results obtained using CNN. The application in the agricultural industry was presented in [15], where a control system for sorting agricultural products on a conveyor based on specified parameters was developed. The system consists of a Dobot magician robotic arm, photoelectric sensor, mass sensor, and colour sensor to help classify the products.

Another study [16] introduced NN-sort, a neural network-enabled sorting method that learns data distribution to efficiently sort large-scale datasets. The NN-sort algorithm is divided into three phases: the input phase, where data are prepared by encoding; the sorting phase, where a trained model organises the data into a sorted array; and the polish phase, which ensures that the sorted data are correct. Their performance experiments showed that NN-sort achieved performance improvements ranging from 2.18 × to 10 × over traditional sorting gear trains, and its high torque output made it suitable for lifting beverage containers and operating a gripper. In the study conducted by [17], an automated computer vision system was developed for monitoring and classifying equipment wear in face milling tools. The system comprises a measurement method for signal acquisition, processing and classification. The ToolWearNET base architecture was employed and optimised to create three wear classification and regression models to predict wear in tools with only 10 incorrect predictions. Various sorting mechanisms were evaluated in a study by [18], who employed three pneumatically powered systems as sorting mechanisms for sweet potatoes. The best results were obtained by integrating machine vision wita linear air cylinder. A further study by [19] explored the feasibility of robotic sorters for separating mixed debris, especially industrial waste, while [20] presented the design of a PLC-controlled conveyor sorter that integrates a YOLOv8-based vision model. Their prototype sorted recyclables (plastic bottles, paper, and metal) with ~90% precision and was engineered for real-time control using a PLC. While these recent advanced computer vision algorithms have produced significantly great results, in this study, a YOLOv11 was employed, which is a more recent computer vision algorithm that has a faster processing speed as a result of its name, “You Only Look Once”, for its object detection in an automated robotic system aimed at classifying objects on a conveyor belt.

2. Materials and Methods

This section discusses the hardware and software components, design, and implementation processes of developing the system.

2.1 System Architecture

In this study, an automated robotic system was designed and developed for classification and sorting using machine vision for industrial applications. The system comprised a webcam for image acquisition, a servo-controlled robotic arm with a two-fingered gripper for sorting, and a conveyor belt driven by a stepper motor. An ultrasonic sensor detected the presence of an object at the end of the conveyor before the gripper received a signal to pick it, and an ESP32 microcontroller for seamless communication and synchronisation between the components. The YOLOv11 model was integrated into the system to detect and classify objects (Milo and Chivita). The system architecture is depicted in Figure 1.

Table 1. Hardware components and specifications of the proposed system

S/N	Component	Model / Type	Key Specifications	Function
1	Microcontroller	ESP32 WROOM-32	Dual-core, Wi-Fi, Bluetooth, 30+ GPIO	System control and communication
2	Servo Motors ×2)	MG996R	4.8–7.2 V, 11.6 kg·cm torque	Robotic arm actuation
3	Ultrasonic Sensor	HC-SR04	2–400 cm range	Object presence detection
4	Buck Converter	DC–DC	4–40 V input, 1.25–37 V output	Voltage regulation
5	Display	I2C 16×2 LCD	I2C interface	Status display
6	Camera	Chusei USB Webcam	720p, USB	Image acquisition
7	Stepper Motor	NEMA 17	1.8° step angle, 1.2–2 A	Conveyor drive
8	Stepper Driver	TB6600	9–40 V input, 4 A max	Motor control
9	Gripper	2-finger aluminium	107 × 98 mm	Pick-and-place
10	Power Source	Li-ion 18650	3.7 V, 2000–3500 mAh	System power
11	Structural Materials	Wood, Acrylic, PU belt	-	Conveyor construction

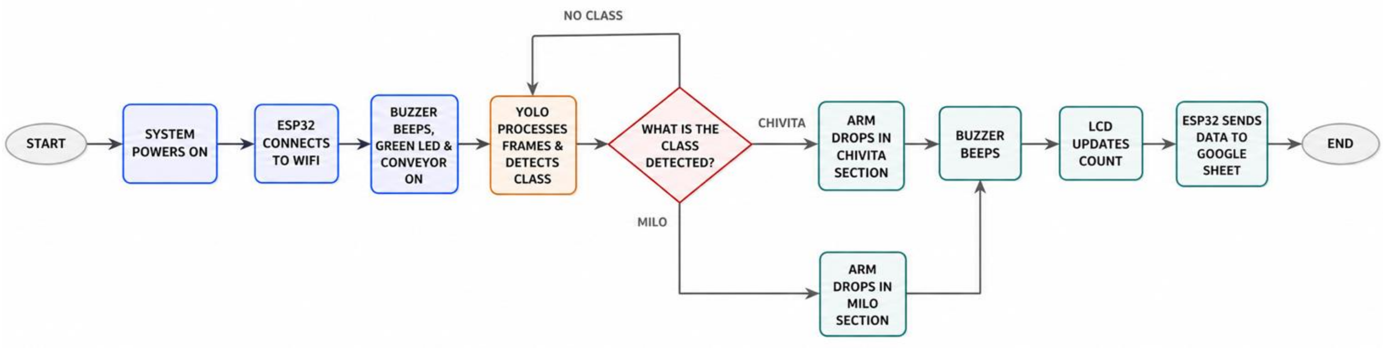


Figure 1. Functional sequence of the classification and sorting system

2.2 Hardware Components

The hardware components and specifications required to build the entire system architecture are described in this section.

ESP32 Microcontroller: The ESP32 is a low-cost, dual-core 32-bit microcontroller with integrated Wi-Fi and Bluetooth, chosen as the system's control unit for its high processing power and wireless connectivity. It provides dozens of GPIO pins and supports I2C/SPI/UART interfaces for sensors and actuators. Its on-board micro-USB connector allows USB power and programming. Its specifications include a dual-core MCU, Wi-Fi, Bluetooth, and 30+ GPIO. The model used is ESP32WROOM- 32

MG996R Servo Motors (×2): Two standard MG996R hobby servos were used for the robotic arm's joints, providing approximately 9–10 kg·cm torque. Each servo's metal gear train and high torque output make it suitable for lifting beverage containers and operating a gripper. The specifications were 4.8–7.2 V, 11.6 kg·cm torque @ 6 V, and 180 rotations.

HC-SR04 Ultrasonic Sensor: The ultrasonic sensor module has two side-by-side transducers for transmitting and receiving ultrasonic pulses. It was chosen for its low cost and wide range (2–400 cm) of non-contact distance measurement, which helps detect package presence or height.

Bulk Converter: serves as a step-down direct current to direct current converter to decrease voltage and increase current. The specifications used are Input: 4–40V, Output: 1.25– 37V, 2A max Output.

I2C 16×2 LCD Module: The display shows the system status, such as the current bin selection. This module was used for its simplicity and compact form factor; the I2C interface conserves GPIO pins on the ESP32 while providing contrast and backlight control.

USB Webcam: A standard USB webcam (Chusei 3D Webcam) was used as the vision sensor to easily capture images for the YOLO machine vision model on the PC. The webcam connects to the PC running the trained YOLO network, providing object detection without burdening the microcontroller. Its compact size makes it practical for monitoring the sorting area. The specification used is a 720p resolution and a USB interface

Status LEDs: Small 5 mm LEDs signal system status (e.g. red when the conveyor halts for sorting and green when it is in motion).

Buzzer: An active buzzer (5V) provides audible alerts for sorting completion or system initialisation.

U-bracket: This is an aluminum frame. The U-bracket connects the base servo, and the L-bracket connects the end effector.

L-bracket: An aluminum frame used as a connector between the U-bracket and gripper, and holds the gripper servo.

Gripper: The aluminum alloy gripper performs the pick-and-place operation of objects with consistent dimensions. The specifications used were 2-fingered, 107 mm length, and 98 mm width.

Stepper motor: A NEMA 17 stepper motor was used to drive the belt in the conveyor system. The specifications used were 5V–12V, 1.8° step angle, 2 phase, and 1.2–2A.

Stepper motor driver: The motor was interfaced with a TB6600 stepper motor driver, which was controlled by the ESP32 microcontroller. The specification used is 9-40V input, micro stepping (up to 1/32), max 4A

Bolts and nuts: The bolts (Stainless steel, Hex.) Nuts (stainless steel, flat, ID 8 mm, OD 16 mm) were used to securely fasten the rollers to the sidebars of the conveyor frame.

Battery: A 3.7v, 2000 3500mAh, Li-on, 18x65mm mm powers the system.

Oak wood: A 0.6–0.9 g/cm^3, straight, coarse texture, light to medium brown colour was used as part of the conveyor design.

Acrylic sheet Plexi glass: a 1.18g/cm3, 5 mm thickness, black colour. was used as part of the conveyor design.

Belt Material: A polyurethane sheet with 1.2m length and 15 cm width was used.

Washers: Washers are placed between the bolt head or nut and the frame surface to distribute the load evenly, prevent damage, and reduce the risk of loosening due to vibration.

2.3 Software Components

Arduino IDE: The Arduino Integrated Development Environment was used to write and upload firmware to the ESP32 microcontroller. The Arduino IDE was used because it offers a straightforward interface and a rich library ecosystem for the embedded devices. Using the IDE's C/C++ environment, the control logic that reads the sensor inputs (such as the ultrasonic sensor) and drives the servos and motors was programmed.

YOLOv11 (Ultralytics): To identify and classify milo and chivita containers on the conveyor, the YOLOv11 object detection framework from Ultralytics was used. YOLO (You Only Look Once) is well-known for its fast, real-time detection capabilities, which are crucial for a continuous sorting system.

Python libraries (OpenCV, torch, serial, time, logging, and ultralytics): The core of the vision and control software was a Python script that bound all the software together. OpenCV (cv2) was used to capture video frames from the camera and perform any necessary image pre-processing (such as resizing) before detection. The torch library (PyTorch) powers the YOLO neural network inference, making it possible to run the object detection model on the frames. The Ultralytics library provided convenient wrappers for loading the YOLOv11 model and running detections with minimal custom code. The serial library allowed the written Python program to send commands over USB for communication. Time was employed to manage delays together with frame rates, and logging to record system events together with its attendant errors during operation.

Google Apps Script: To monitor the system's sorting performance, Google Apps Script was employed to log data to an online spreadsheet. A script was written to receive data from the ESP32 and append this information to a Google Sheet. This allowed for the automatic recording of each sorting event without manual entry.

SolidWorks: SolidWorks CAD software was used to design the mechanical components of the robotic arm and conveyor system prior to fabrication. The arm links, gripper, conveyor frame, and motor mounts weree modelled in 3 using SolidWorksD.

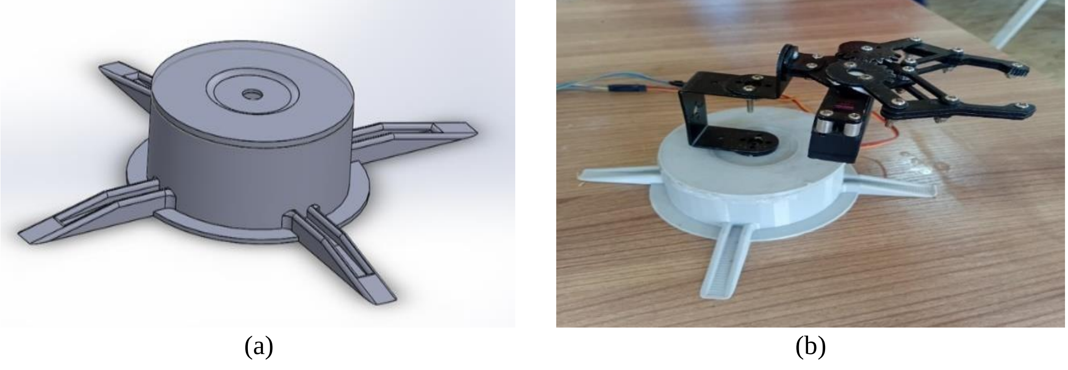
CirKitDesigner (Circuit Designer): To plan the electrical wiring, CirKitDesigner (also known as Circuit Designer) was used to draw the circuit schematic of the system. This tool ensures proper placement of components like the ESP32 board, sensors, motor driver, and power supplies; connecting them logically. The creation of a clear circuit diagram before the connection allows for the simulation of the circuitry and testing it without damaging components.

2.4 Mechanical Design of Robotic System

The mechanical structure of the robotic system comprises a 2-DOF robotic arm and a conveyor belt system as depicted below:

2.4.1 Robotic arm

The 2-DOF robotic arm is an assembly of a 3D-printed base (Figure 2(a)), an off-the-shelf gripper, 2 MG966R servo motors, L-brackets, and U-brackets. The 3D-printed base was designed using SOLIDWORKS. It has a base lid and a base support part. The base lid and base support designs were assembled on SOLIDWORKS to depict the complete base structure of the robotic arm, ensuring proper alignment of the servos and providing the necessary support for smooth operation. The L- and U-shaped brackets were screwed together and connected to the gripper with the servos attached to the base and gripper area, respectively. The assembled structure was then placed on the base through the hole to connect it to the seat to produce the complete assembled robotic arm, as depicted in Figure 2(b).



(a) (b)

Figure 2. Designed and assembled robotic arm

2.4.2 Conveyor System

A CAD model of the conveyor system was designed using SOLIDWORKS (left of Figure 3). Owing to its affordability and availability, the conveyor system was constructed using a wooden structural frame (right of Figure 3). A NEMA 17 stepper motor controlled the conveyor belt using a TB6600 motor driver, producing sufficient torque to cause continuous motion at the desired speed. The polyurethane belt was used on the conveyor because of its elasticity. After the object was classified, it was detected by the ultrasonic sensor. The ultrasonic sensor detected the presence of the product after the YOLO model classified it. This process temporarily stopped the conveyor for the robotic arm to sort the product.

Figure 3. Comparison of the designed CAD model and the assembled conveyor-belt prototype

2.5 Object Detection and Classification

YOLOv11 introduces key architectural enhancements, including the replacement of the C2f module with the more efficient C3k2 block (using smaller convolutions for better computational efficiency) and the addition of the C2PSA attention mechanism after SPPF for improved spatial focus and feature selectivity, particularly beneficial for small or cluttered objects. These changes result in a Pareto improvement: YOLOv11 achieves higher mean Average Precision (mAP) on benchmarks such as COCO while using up to 22% fewer parameters (e.g., in medium variants) and delivering faster inference, especially on CPU/edge devices. Additionally, optimised training pipelines enable faster convergence during fine-tuning, reducing training time and improving precision under constrained conditions typical of industrial setups. This study represents one of the early applications of YOLOv11 in a vision-guided robotic sorting system, demonstrating its suitability for industrial automation tasks.

The YOLOv11 architecture was used for this system's computer vision algorithm. Following the selection of the object detection algorithm. Over 1000 images of the objects to be sorted were hand-taken and then annotated using Roboflow, a computer vision tool that facilitates accurate labelling and dataset management. The annotated dataset comprising the required images is shown in Figure 4. Subsequently, this dataset was used to train the machine learning model to detect the objects on the conveyor belt. An Nvidia T4 GPU was used to train the model, using its Tensor cores for accelerated deep learning computation. The GPU’s 16GB GDDR6 memory and 320 Tensor cores helped the training process, reducing the time required to train the model from over 12 hours on a CPU to 30 minutes (estimated). The model was trained using PyTorch and the Ultralytics YOLOv11 framework with a batch size of 32, an initial learning rate of 0.001, and 200 epochs. To avoid data leakage and potential evaluation bias, the annotated dataset was explicitly divided into three mutually exclusive subsets: training, validation, and testing. A 70:20:10 split was adopted, where the training set was used for model weight optimisation, the validation set was used to monitor training performance and tune hyperparameters, and the testing set was reserved exclusively for final performance evaluation. A Chusei webcam captured the frames, enabling the model to process them in real time using the implemented YOLOv11 model. The model localises and classifies the image based on the Milo and Chivita pouch. The model performs both localisation and classification of each object on the conveyor. Following successful detection, the current class label was sent to the robotic arm to enable it to perform the sorting operation.

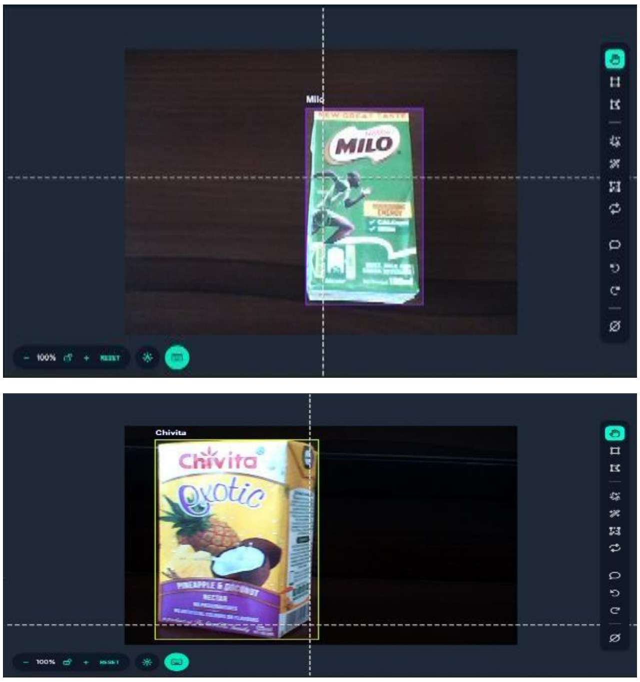


Figure 4. Annotation of various classes for object detection

2.6 Control System Design

The connection of the control system was simulated using CirKitDesigner (Figure 5). The control unit coordinates communication between the vision system, robotic arm system, and conveyor belt mechanism. The ESP32 microcontroller was at the centre of this communication and communicated with the detection system over Wi-Fi. The robotic arm was powered by 3.7V batteries, and the conveyor system was powered by a 12 V adapter connected to the motor driver to control the speed and torque of the stepper motor, which drives the conveyor.

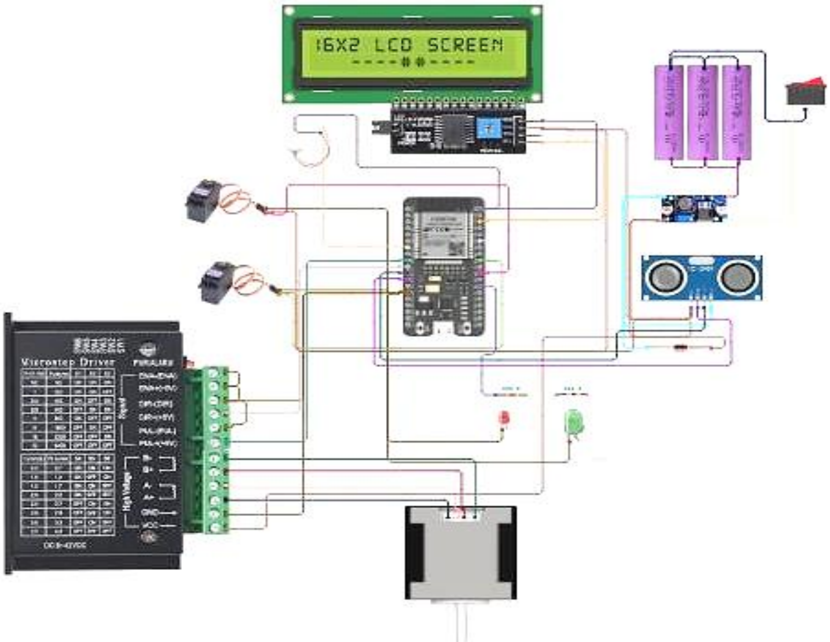


Figure 5. Circuitry design

2.7 Dashboard Integration

The dashboard was developed using Google Apps Script, which receives classification results from the ESP32 microcontroller, such as detection metrics, confidence score, timestamp, product class, and object count. The Google Sheets dashboard records each detection event and can be used for analysis and trend prediction.

2.8 Working Process of the Robotic System

The complete working process of the system is depicted in Figure 6. The developed system begins with objects moving on the conveyor system. In this case, Milo and Chivita are moving on the conveyor system. The webcam detects Milo as it approaches the robotic arm and classifies it, sending the update to the Google Sheets dashboard and sending the classification to the robotic arm. The ultrasonic sensor detects Milo at the end of the conveyor, and the arm picks it up and places it in the bin on the right for Milo. The exact process is repeated for Chivita. The webcam detects Chivita, and the system classifies it as Chivita. It sends the classification to the robotic arm, which senses only when Chivita is at a specific distance from it due to the ultrasonic sensor. It triggers the robotic arm, which then picks and places Chivita in the designated bin on the left.

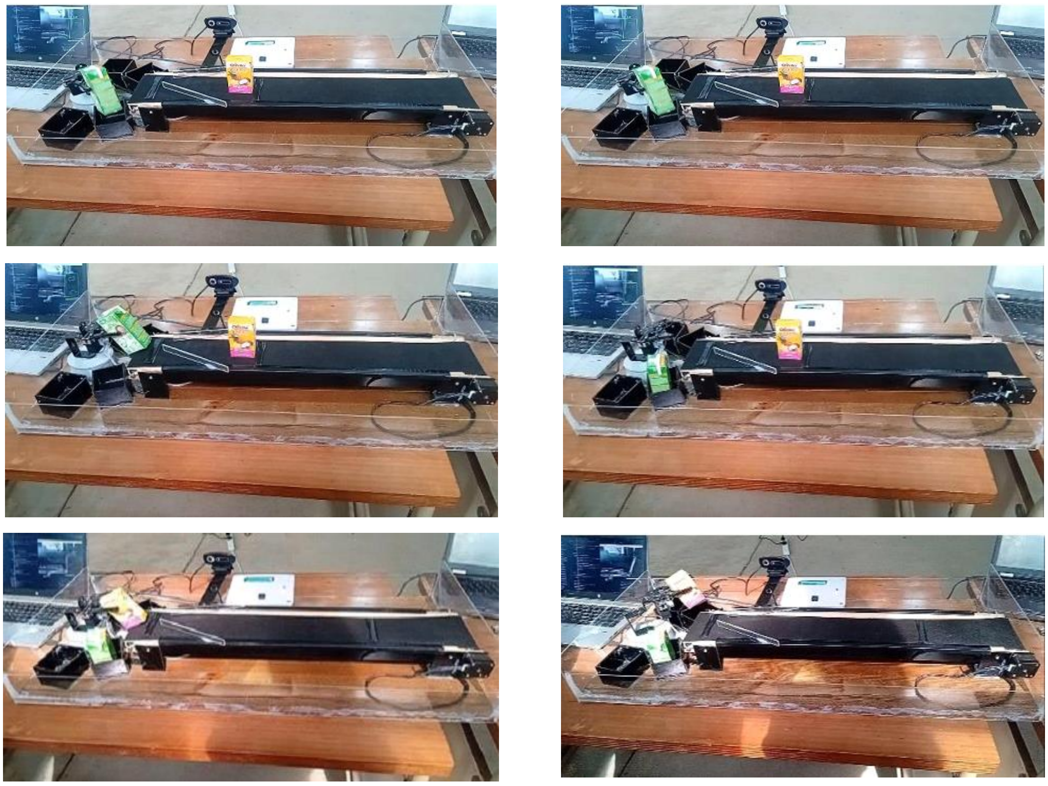


Figure 6. System operation and sorting process

3. Results and Discussion

In this study, an automated robotic system was designed. The proposed system can automatically classify and sort two products (Milo and Chivita) using a 2DOF robotic arm, a conveyor mechanism, and machine vision. The results obtained by the developed system are presented below.

3.1 Sorting Performance

The sorting performance of the system was evaluated by conducting 50 trial runs, and the results are presented in Table I. The table illustrates 50 instances, with C representing correctly sorted and X representing incorrectly sorted.

Table 1. Sorting performance

Trial	Milo	Chivita	Trial	Milo	Chivita	Trial	Milo	Chivita
1	C	C	10	C	X	18	C	C
2	C	C	11	C	C	19	C	C
3	C	C	12	C	C	20	C	X
4	C	X	13	C	C	21	C	C
5	X	C	14	C	C	22	C	X
6	C	C	15	C	C	23	C	C
7	C	C	16	C	C	24	C	C
8	C	C	10	C	X	25	C	C

The complete pick-and-sort system was then evaluated by running beverages on the conveyor. Fifty beverages were fed individually, and the arm attempted to pick and drop each detected object into the bin for the predicted class. The total success and errors were computed as shown in Table 2, enabling calculation of the sorting accuracy. The sorting accuracy is the percentage of objects placed correctly into the correct bin. The overall sorting process produced a sorting success accuracy of 92%, as shown in Table 2. The data in Table 2 show that the robotic arm had no problem matching the classes (Milo and Chivita) to their designated bins, especially under good lighting conditions. However, the few times the robotic arm could not sort the beverages correctly resulted from the vision system not classifying the beverage. This is depicted in Figure 7 for successful classification and Figure 8 for sorting into each class’s designated bins.

Table 2. Sorting accuracy

Object	Trials	Success	Failure	Success rate (%)
Milo	25	24	1	96
Chivita	25	21	4	84
Overall	50	46	5	92

Figure 7. Successful classification of objects into Milo and Chivita

Figure 8. Successful sorting of objects into the respective bins

Table 3. Metrics for detection and results

Class	True Positive (TP)	False Positive (FP)	True Negative (TN)	False Negative (FN)	Precision (%)	Recall (%)	F1 Score (%)
Milo	24	1	23	1	96.00	96.00	96.00
Chivita	23	1	24	2	95.83	92.00	93.86

The detection accuracy of the overall system is calculated using the following equation:

$$Detection\ Accuracy = \frac{\text{TPMilo} + \text{TPChivita x 100\%}}{Total\ Objects} \tag{2}$$

Therefore;

$$Detection\ Accuracy = \frac{23+24}{50} = \frac{47}{50}$$

$$Detection\ Accuracy = 0.94\ \times 100\% = 94\%$$

An overall detection accuracy of 94% was obtained, indicating that the physical experimental setup test provided a precise, accurate, and reliable result.

The confusion matrix, shown in Figure 9, confirmed that the system accurately distinguished between Milo and Chivita under proper lighting conditions. Although Milo had a detection accuracy of 100%, there was a slight misclassification when evaluating its detection accuracy. Based on the confusion matrix, the true positive, false positive, true negative, and false negative values were obtained. After obtaining the confusion metrics in Table 4, from the confusion matrix, we can deduce that the model for Milo achieved a precision of 99.9% and a recall of 98.4% for Milo classification. The Chivita classification, on the other hand, achieved an accuracy of 92.4% with two misclassifications, resulting in a slightly lower recall of 100%, as shown in Table 4. This means that the Milo objects had good repeatability, with only one false positive, resulting in an F1 score of 99.16%, while Chivita's performance was slightly lower, with an F1 score of 96.16%. However, it is essential to note that the few classification errors observed were because of poor or uneven lighting conditions that affected the clarity of the captured images, reduced the model's confidence, or resulted in no detection. Overall, the mean average precision (mAP) for Milo and Chivita is 99.1%, indicating that the system performed significantly better than some existing works. To preliminarily assess robustness, additional qualitative tests were conducted under uneven lighting (e.g., shadows from overhead lamps) and rotated object orientations (±30° from default). Detection confidence dropped by 8–12% in low-contrast regions, leading to 3–5 additional misclassifications out of 20 trials per condition. These observations reinforce the need for enhanced preprocessing (e.g., histogram equalisation) or augmented training data with varied illumination and poses, planned for future iterations.

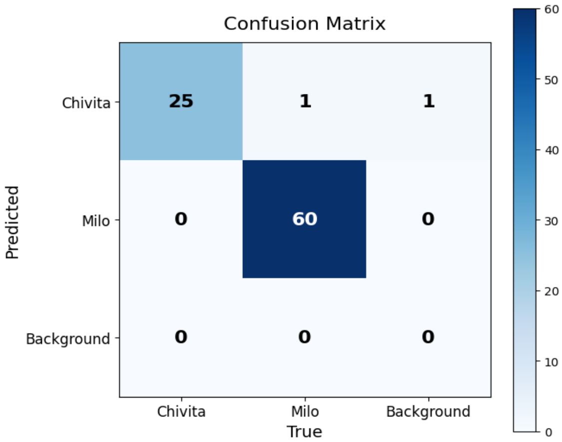


Figure 9. Confusion matrix

Table 4. Confusion metrics

Class	Images	Instances	Precision (P)	Recall (R)	mAP@0.5	MAP@0.5:0.95
Milo	60	61	0.999	0.984	0.983	0.973
Chivita	25	25	0.924	1	0.976	0.964
Overall	73	86	0.962	0.992	0.991	0.968

3.2 System Speed and Timing

Timing analysis was performed for each subsystem. The average time from the beverage being fed onto the conveyor belt to when it is sorted into its designated bin is about 25 seconds. Table 5 shows the average time for each stage of the subsystem. This is depicted in Table 5. The real-time inference performance of the YOLOv11 model was evaluated during system operation. On the NVIDIA T4 GPU platform, the average inference latency per frame was approximately 2.5 seconds, corresponding to an effective processing rate of approximately 0.4 FPS. Although the system did not operate at high-frame-rate video speeds, this inference rate was sufficient for the conveyor-based sorting process, where objects are processed sequentially and at controlled speeds.

Table 5. Average time per stage of each subsystem

S/N	Subsystem	Time (seconds)
1	Conveyor	20.0
2	Vision model	2.5
3	Robotic arm	2.5
4	Total	25.0

4. Conclusions

The project successfully implemented an automated robotic sorting system that met the original objectives. The developed prototype achieved high classification accuracy (on the order of 90–95% in testing) and could pick and place objects into sorted bins with similarly high reliability. This demonstrates that the vision-guided system can correctly identify different object classes and operate the manipulator accordingly. In the future, consistent and uniform lighting of the workspace can increase accuracy, and the number of training images of different classes can be added to expand the vision system. This will allow the vision system to learn to recognise a broader range of objects.

Acknowledgements
The authors acknowledge the contribution of the Department of Mechatronics Engineering, Air Force Institute of Technology, for providing adequate laboratory equipment, technical assistance, consultation, and an overall conducive environment to ensure the success of this research.

Funding
This study was not supported by any grants from funding bodies in the public, private, or not-for-profit sectors.

Declaration of Competing Interest
The authors declare no conflict of interest.

Credit Authorship Contribution Statement
F.D. Osokoya: Conceptualization; Methodology; Data curation; Analysis; Writing - original draft; Resources
K.T. Kehinde: Formal analysis; Methodology; Data curation; Writing - original draft; Resources
K.O. Shobowale: Supervision; Analysis; Writing- review and editing
Y. Bello: Supervision; Analysis and Discussion; Writing- review and editing
M.O. Momoh: Methodology; Analysis; Writing- review and editing
S.A. Alim: Methodology; Analysis; Writing- review and editing

Availability of Data and Materials
The datasets generated and/or analysed during the current study are available from the corresponding author upon reasonable request.

Ethics Declarations
This study did not involve human participants or animal subjects. Therefore, ethical approval was not required.

Generative Artificial Intelligence Declarations
Generative artificial intelligence tools were used solely to improve the language and readability. All scientific content, analyses, interpretations, and conclusions were produced, reviewed, and approved by the authors, who took full responsibility for the manuscript.

References
[1] Ghobakhloo M. Industry 4.0, digitization, and opportunities for sustainability. *Journal of Cleaner Production.* 2020;252:119869. https://doi.org/10.1016/j.jclepro.2019.119869
[2] Apriaskar E, Fahmizal F, Fauzi MR. Robotic technology towards industry 4.0: Automatic object sorting robot arm using kinect sensor. *Journal of Physics: Conference Series*. 2020;1444(1):012030. https://doi.org/10.1088/1742-6596/1444/1/012030
[3] Bahrin MF, Othman MF, Azli NH, Talib MFM. Industry 4.0: A review on industrial automation and robotic. *Jurnal Teknologi (Sciences & Engineering)*. 2016;78(6–13):137-143. https://doi.org/10.11113/jt.v78.9285
[4] Xu X, Lu Y, Vogel-Heuser B, Wang L. Industry 4.0 and Industry 5.0—Inception, conception and perception. *Journal of Manufacturing Systems*. 2021;61:530–535. https://doi.org/10.1016/j.jmsy.2021.10.006
[5] Evjemo LD, Gjerstad T, Grøtli EI, Sziebig G. Trends in Smart Manufacturing: Role of Humans and Industrial Robots in Smart Factories. *Current Robot Reports*. 2020;1:35–41. https://doi.org/10.1007/s43154-020-00006-5
[6] Bortolini M, Faccio M, Galizia FG, Gamberi M, Pilati F. Adaptive automation assembly systems in the industry 4.0 era: A reference framework and full–scale prototype. *Applied Sciences*. 2021;11(3):1256. https://doi.org/10.3390/app11031256
[7] Zhou K, Meng Z, He M, Hou J, Li T. Design and test of a sorting device based on machine vision. *IEEE Access* 2020;8:27178–27187. https://doi.org/10.1109/ACCESS.2020.2971349
[8] Javaid M, Haleem A, Singh RP, Suman R, Gonzalez ES. Understanding the adoption of Industry 4.0 technologies in improving environmental sustainability. *Sustainable Operations and Computers*. 2022;3:203–217. https://doi.org/10.1016/j.susoc.2022.01.008
[9] Szeliski R. *Computer Vision: Algorithms and Applications.* London: Springer; 2011. https://doi.org/10.1007/978-1-84882-935-0
[10] Li S, Zhang B, Zheng J, Wang D, Liu Z. Development of automated 3D LiDAR system for dimensional quality inspection of prefabricated concrete elements. *Sensors*. 2024;24(23):7486. https://doi.org/10.3390/s24237486
[11] Fadhil AT, Abbar KA, Qusay AM. Computer vision-based system for classification and sorting color objects. *IOP Conference Series: Materials Science and Engineering*. 2020;745(1):012030. https://doi.org/10.1088/1757-899X/745/1/012030
[12] Sofu MM, Er O, Kayacan MC, Cetişli B. Design of an automatic apple sorting system using machine vision. *Computers and Electronics in Agriculture*. 2016;127:395–405. https://doi.org/10.1016/j.compag.2016.06.030
[13] Rahman M, Das D, Islam O, Zaman H. A novel design of a robotic object sorter based on color differences using image processing techniques. In: Proceedings of the 2018 International Conference on Computer, Communication,

Chemical, Material and Electronic Engineering (IC4ME2); Feb 8-9; Rajshahi, Bangladesh. Piscataway (NJ): IEEE; 2018. p.1–4. https://doi.org/10.1109/IC4ME2.2018.8465620
[14] Cai H, Cao X, Huang L, Zou L, Yang S. Research on computer vision-based waste sorting system. In: Proceedings of the 5th International Conference on Control, Robotics and Cybernetics (CRC); 2020 Oct 16-18; Wuhan, China. Piscataway (NJ): IEEE; 2020. p. 117–122. https://doi.org/10.1109/CRC51253.2020.9253456
[15] Zakozhurnikov S, Zakozhurnikova G. Development of a control system for sorting agricultural products according to specified criteria. *E3S Web of Conferences*. 2023;390:03019. https://doi.org/10.1051/e3sconf/202339003019
[16] Zhu X, Zhang Q, Zhou W, Liu L. Deep learning service for efficient data distribution aware sorting. In: Proceedings of the 2024 IEEE International Conference on Big Data (BigData);2024 Dec 15-18; Washington, DC, USA. Piscataway (NJ): IEEE; 2024. p. 1508–1515. https://doi.org/10.1109/BigData62323.2024.10825633
[17] Friedrich M, Gerber T, Dumler J, Döpper F. A system for automated tool wear monitoring and classification using computer vision. *Procedia CIRP*. 2023;118:425–430. https://doi.org/10.1016/j.procir.2023.06.073
[18] Xu J, Lu Y. Design and preliminary evaluation of automated sweetpotato sorting mechanisms. *AgriEngineering* 2024;6(3):3058–3069. https://doi.org/10.3390/agriengineering6030175
[19] Kiyokawa T, Takamatsu J, Koyanaka S. Challenges for future robotic sorters of mixed industrial waste: A survey. *IEEE Transactions on Automation Science and Engineering*. 2024;21(1):1023–1040. https://doi.org/10.1109/TASE.2022.3221969
[20] Petersen M, Lakenen B, Chavan K, Waghmare P, Sergeyev A, Rawashdeh NA. Conveyor line color object sorting using a monochrome camera, colored light and RGB filters. *arXiv* [Preprint]. 2025.arXiv:2502.08419. https://doi.org/10.48550/arXiv.2502.08419