[1.] Olalekan OGUNBIYI, [1.] Lambe Mutalub ADESINA, [1.] Mukhtar Oladapo ALABI,
[1.] Abdulhameed Abayomi DANESI, [1.] Raqeebah Ajoke FASASI

INTELLIGENT CONTROL OF A ROBOTIC ARM FOR IMPROVED PICK–AND–PLACE PERFORMANCE

[1.] Electrical and Computer Engineering Department, Kwara State University, Malete, NIGERIA

Abstract: Traditional pick–and–place robotic arms generally rely on fixed positioning and analogue control mechanisms, which constrain their flexibility and lower the accuracy of object recognition in dynamic settings. These designs, which often utilise potentiometer–based inputs and basic visual systems, typically reach only 70–85% accuracy due to limited field of view, inadequate adaptation to lighting changes, and immobility. To address these issues, this research presents a mobile intelligent pick–and–place robotic arm featuring an ESP32–CAM and a classification system built on TinyML (CNN). By acquiring images of objects from multiple perspectives and evaluating RGB data using a weighted voting approach, the robot greatly minimizes errors caused by shadows and lighting variations. Its mobile platform improves alignment with targets and enhances handling precision, achieving a 95% classification accuracy—an increase of 10–25% over traditional models. Furthermore, integration with an Arduino Nano and HC–05 Bluetooth module facilitates remote control through an Android application, making the system effective for deployment in risky or difficult–to–access areas. The proposed solution offers potential for use in automated logistics, sorting systems, and handling dangerous materials. Future enhancements will explore the use of deep learning and reinforcement learning to support advanced recognition and adaptive control without manual code adjustments.
Keywords: ESP32–CAM, Mobile Robot, Object Classification, Robotic Arm, Vision System

1. INTRODUCTION

Pick–and–place robotic arms have played a pivotal role in transforming modern automation, with applications spanning industrial assembly lines, medical logistics, and warehouse management. Their ability to perform repetitive tasks with speed and precision makes them indispensable in environments where consistency and efficiency are essential [1], [2], [3]. Despite widespread use, however, many of these systems still operate under significant constraints. Traditional pick–and–place robotic arms are typically stationary, heavily reliant on analog control methods, and lack the intelligence and flexibility required for operation in dynamic or hazardous environments [4]. These limitations have hindered broader deployment, particularly in unstructured or resource–constrained settings such as those found in emerging economies.

One of the critical shortcomings in existing systems is the limited adaptability of their vision and control architectures. Most commercial robotic arms use static camera setups with narrow viewing angles and rigid classification logic, resulting in poor performance under occlusion, variable lighting, or cluttered scenes. Studies have shown that such systems often achieve only 70–85% accuracy in object recognition tasks under non–ideal conditions [5], [6]. Additionally, the lack of mobile autonomy restricts the operational reach of these arms, confining their use to well–structured environments. This is further compounded by the high energy consumption of traditional high–torque servos and non–optimised control algorithms, making them unsuitable for battery–powered or off–grid applications [7], [8], [9]. These challenges have been noted in prior literature but are rarely addressed in an integrated, low–cost solution.

Furthermore, while recent developments in machine vision and AI have made strides toward improving robotic perception, many research efforts still stop short of implementing real–time adaptive control or mobile autonomy, particularly in low–resource settings [10], [11], [12]. Previous attempts to bridge this gap often involve high–cost platforms or static prototypes that are impractical for use in environments with infrastructure limitations [12], [13]. For instance, systems

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665

that incorporate machine learning for object recognition may still rely on fixed arms and do not account for energy efficiency or hazard navigation. As a result, there remains a distinct lack of affordable, intelligent, and mobile robotic systems capable of operating reliably in unpredictable, real–world environments [14], [15].

Recent advancements have demonstrated the effectiveness of deep learning and active sensing frameworks in achieving precise six–degree–of–freedom (6–DOF) pose estimation, as well as the potential of adaptable learning strategies for bin–picking applications and Mask R–CNN models for tasks such as food quality assessment. The integration of multi–view active sensing has further enhanced human–robot interaction, reinforcing the critical role of advanced vision systems in the development of intelligent and adaptive robotic platforms [16], [17], [18].

This work responds directly to this gap by presenting a low–cost, intelligent mobile robotic arm designed to improve adaptability, efficiency, and autonomy in pick–and–place operations, especially in hazardous or unstructured environments [19]. The proposed system integrates a novel vision–based decision–making mechanism using the ESP32–CAM and TinyML with a multi–angle voting algorithm that enhances classification accuracy by 10–25% compared to conventional single–view setups. It introduces a hybrid control strategy using dynamic PWM tuning with an Arduino Nano and PCA9685, enabling optimised torque and energy use across different payload conditions [20], [21]. Moreover, the incorporation of a Bluetooth–enabled mobile base significantly expands the system's operational footprint, allowing it to access areas that are typically beyond the reach of stationary arms, critical for disaster zones, chemical facilities, or other hazardous contexts [22].

In a related development, Ogunbiyi et al. (2023) introduced a pick–and–place robotic arm controlled by an analogue potentiometer system anchored at a fixed point and operated via an Arduino Nano microcontroller [4]. While this system marked an improvement in terms of speed and accuracy during repetitive operations, it exhibited critical limitations, most notably, an inability to distinguish between relevant and irrelevant objects within its environment. Moreover, the use of a rigid potentiometer mechanism imposed restrictions on flexibility and adaptability, ultimately limiting the system's potential for scaling and deployment in more dynamic or hazardous scenarios. Although the study offered valuable contributions to the design of control systems, particularly in its application of real–time feedback for enhanced precision, it revealed the need for a more versatile, intelligent, and mobile robotic architecture.

This current study seeks to bridge that gap by proposing a mobile robotic arm equipped with wireless control and vision–based decision–making capabilities, achieved through the integration of an ESP32–CAM module for multi–angle image capture and a voting–based classification algorithm for object recognition. The robotic platform is designed to operate effectively in cluttered or poorly illuminated environments while also incorporating improved servo motor coordination using a PCA9685 module and energy–efficient control strategies to optimize performance. The choice of low–cost hardware, including the ESP32–CAM and Arduino Nano, is deliberate, ensuring that the system remains both affordable and accessible, particularly for use in developing countries where industrial solutions are often prohibitively expensive.

2. MATERIALS AND METHODS

The system design and development consist of four main components: the mobile base, the robotic arm's mechanical chassis, the embedded control system, and the visual enhancement module. The mobile base, equipped with drive wheels, enables the robot to move across different terrains and position itself accurately for object handling. The mechanical chassis serves as the robot's structural framework, incorporating key joints for effective object manipulation. The embedded control system manages the servomotors, ensuring smooth and precise movements. The visual intelligence, powered by an ESP32 camera, allows the robot to distinguish between objects to pick up and those to avoid.

The Mobile Base

The mobile base is designed based on the block diagram in Figure 1, which comprises key components such as motor wheels, motor drivers, a Bluetooth module, an Arduino Nano microcontroller, and an Android application. This system works together to control the movement of the mobile base, with each block representing a crucial element that contributes to its overall functionality. The block diagram provides a clear and comprehensive overview of the system's architecture, illustrating how each component interacts and contributes to the development of a mobile base for a robotic arm.

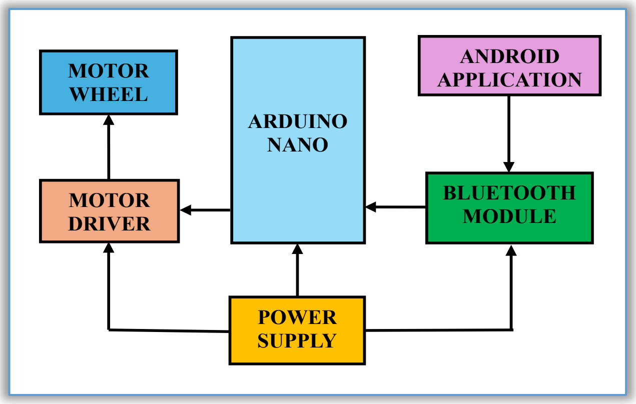

Figure 1: Block Diagram of the Mobile Base

Motor Wheels

Four TT motors with car wheels, as shown in the Figure 2, were chosen for this work, each featuring a single–axis reducer, wheel, and rubber tyres. With a 3– 6V operating voltage and 125 RPM rotation speed, they're ideal for robotic drive applications. The motors offer a compact size, low power consumption, and geared design, balancing speed and torque. Their specification includes a 3V–6V voltage range, 100 MA–120MA current, 48:1 reduction rate, and 65mm tyre diameter. Weighing 29g each and measuring 70mm x 22mm x 18mm, they're a compact and efficient option for robotics. Their easy–to–mount plastic housings and standard axle sizes make integration seamless, and their compatibility with battery–operated devices makes them perfect for portable or mobile robots [23].

Motor Driver

The motor wheels, as illustrated in Figure 2, enable the robotic base to move, allowing the robotic arm to reach different locations. The motor driver manages these wheels, enabling forward, backward, and rotational movement. The Arduino Nano sends control signals to the motor driver of Figure 3, which translates the low–power signals from the Arduino into the high–power output needed to drive the wheels. In this work, the L298N dual H–Bridge motor driver module was utilized. This module, equipped with an L298 motor driver IC and a 78M05 5V regulator, can control up to four DC motors or two DC motors with directional and speed control.

Figure 2: TT motor

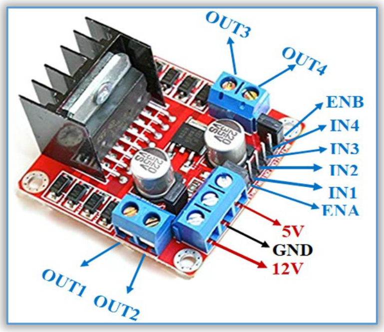

Figure 3: L298N Dual H–Bridge Motor Driver

Bluetooth Module

The mobile base of the robotic arm can be controlled through commands sent from an Android device via Bluetooth, using the HC–05 Bluetooth module shown in Figure 4 as a wireless communication link to the Arduino Nano, which serves as the control unit. This setup enables remote control of the vehicle's movements within a range of approximately 10 meters (33 feet) and complies with the Bluetooth V2.0+EDR standard. The HC–05 module's simple pin configuration

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665

allows for easy integration with the Arduino, facilitating control through an Android app. Its dual–mode operation (Command Mode and Data Mode) supports both configuration and data transmission, demonstrating an innovative approach to wireless robotic control.

Arduino Nano

The Arduino Nano in Figure 5 is a compact microcontroller board integral to the mobile base of the robotic arm, offering precise control and movement due to its small size and lightweight design. It features 14 digital I/O pins and 8 analogue input pins to read data from sensors such as ultrasonic, infrared, and encoders for obstacle avoidance and navigation. With 32 KB of flash memory and 2 KB of SRAM, the Nano supports complex algorithms for controlling the mobile base's movements. It connects via micro–USB and operates at a clock speed of 16 MHz for efficient programming and communication.

Acting as the central controller, the Arduino Nano processes commands from an Android app through the HC–05 Bluetooth module, which sends instructions like moving forward or turning left. These commands are converted into control signals for the L298N motor driver, which regulates the speed and direction of the TT motors, allowing users to remotely manoeuvre the robotic arm for tasks such as picking and placing objects, ensuring accurate and efficient responses through seamless communication among all components.

Power Supply

The 18650 lithium–ion batteries were selected as the primary power source due to their high energy density, long cycle life, and compact size, all of which support optimal performance. Three batteries with a 2000mAh capacity and 3.7V nominal voltage provided consistent, reliable energy output to meet the robotic car's power needs. For safety and efficiency, a Battery Management System (BMS) 3S 20A was included to monitor and regulate voltage, current, and temperature, preventing issues like overcharging, over–discharging, and thermal runaway. This combination of robust batteries and advanced BMS technology contributed to the robotic car's reliable power management and durability, as illustrated in Figure 6.

The power supply system is divided into two sections: an 8.4V supply for the arm and a 12V supply for the base. The 8.4V section powers the arm's motors and servos, while 5V is allocated to the ESP32–CAM module for the vision system, simplifying overall power management. The 12V supply powers the motors responsible for base movement, providing the necessary torque and stability. This dual–power setup improves the efficiency and reliability of the robotic arm, with separation reducing electrical interference and ensuring optimal system performance.

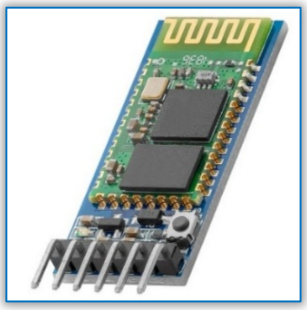
Figure 4: HC–05 Bluetooth

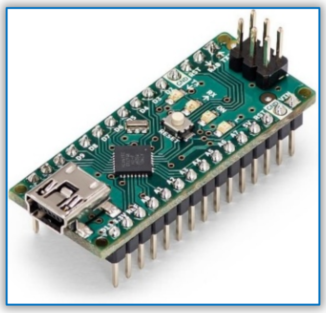
Figure 5: Arduino Nano

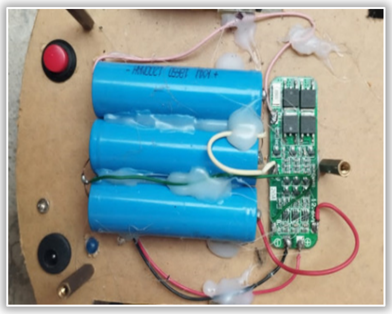
Figure 6: Battery and BMS 3S 20A

The complete circuit diagram of the mobile base system is presented in Figure 7.

Robotic Arm Mechanical Chassis

The mechanical chassis of the pick–and–place robotic arm, featuring six degrees of freedom, provides a robust and rigid foundation essential for precise movements. Designed to support multiple axes of motion, including base rotation, shoulder lift, elbow lift, wrist rotation, wrist lift, and gripper rotation, this chassis enables the arm to perform complex tasks with accuracy and speed. Constructed from durable steel, it minimizes deflection and ensures stability, while incorporating bearings, gears, and other mechanical components for smooth operation. The chassis also includes mounting points for actuators, sensors, and control systems, making it crucial for applications in manufacturing, assembly, and material handling where precision and reliability are paramount. The design of the robotic arm chassis was created using SolidWorks.

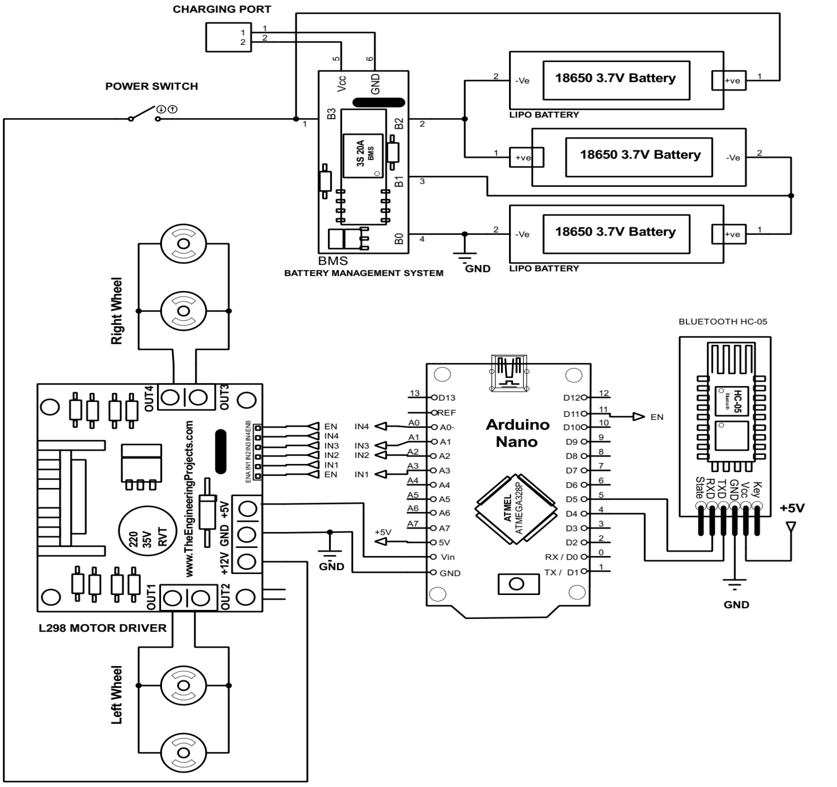

Figure 7: Circuit diagram of the mobile base system

The Embedded Control System

Figure 8 presents a block diagram illustrating the components and connections of a mobile vision–based robotic arm manipulator. This system integrates vision processing with precise movement to achieve tasks efficiently. The ESP32–CAM (located at the top) captures visual data and transmits it to the Arduino Nano, which serves as the central processing unit. The Arduino Nano interprets the inputs and manages operations through the PCA9685, a 16–channel PWM driver capable of controlling up to four servo motors for mechanical movement.

A power supply connects to the PCA9685, servo motors, and Arduino Nano, ensuring efficient power distribution and reliable operation. Real–time adjustments are facilitated by a feedback loop from the ESP32–CAM to the Arduino Nano, enabling precise control and adaptability essential for applications that demand accurate movement. This setup effectively combines vision processing and mechanical precision, making it suitable for a wide range of applications. The power supply is configured using four batteries arranged in both parallel and series to enhance current and voltage.

Servomotors

The robotic arm is powered by four TD–8120MG Servo Motors, as shown in Figure 9, recognized for their high performance in robotic and aeromodelling applications. Each motor offers 180° of rotation and delivers a maximum torque of approximately 1.96 Nm at 6V, making them ideal for tasks that require both precision and strength. The metal gear transmission system enhances the motors' durability and safety, ensuring reliable operation under various conditions. Each motor comes with a 30cm wire and connector for easy installation, along with an accessory kit for versatile use. With a loading capacity of up to 20 kg, these servo motors are well–suited for heavy–duty applications, enabling the robotic arm to operate effectively within an angle range of 0 to 180 degrees.

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665

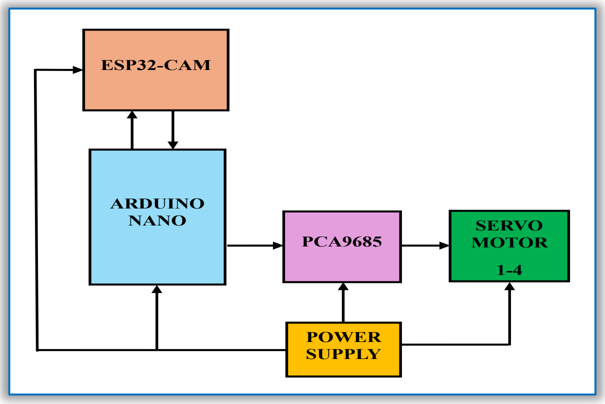

Figure 8: Block Diagram of Robotic arm

Figure 9:TD–8120MG Servo Motor

The movement ranges of the servo motors in the robotic arm are as follows: the gripper spin allows for rotation from 0 to 180 degrees, while the gripper pitch operates within a range of 0 to 100 degrees. The elbow joint can move between 0 and 160 degrees, and the shoulder joint has a range of motion from 0 to 150 degrees. Finally, the base rotation spans from 0 to 180 degrees, providing the robotic arm with versatile and precise movement capabilities.

Motor control uses a hybrid strategy that combines real–time feedback with dynamic PWM tuning for improved precision and energy efficiency. The PCA9685 driver, controlled by the Arduino Nano, adjusts the PWM duty cycle based on torque demand, object weight, or position error. Servos then respond by smoothly changing their angle. PWM controls motor power through the pulse width, defined by duty cycle (%) and frequency (Hz).

The average voltage supplied to the servo motor is:

$$V_{avg} = D\, V_{in} \quad (1)$$

where V_{avg} is the effective voltage seen by the servo motor, D is the duty cycle and V_{in} is the supply voltage.

Instead of fixed PWM values, the control system adapts the duty cycle dynamically in real–time based on the required torque or position error. The basic form of the adaptive control law can be described as:

$$D(t) = D_0 + k_e\, e(t) \quad (2)$$

Where $D(t)$ is the adjusted duty cycle at time t, D_0 is the nominal duty cycle, $e(t)$ is the control error, and k_e is the tuning gain. An error function $e(t)$ is computed as a function of acgular displacement θ.

$$e(t) = \theta_{desired} - \theta_{actual} \quad (3)$$

This strategy forms a hybrid loop by combining the Open–loop PWM control from the PCA9685 to regulate servo motion and Closed–loop feedback from sensors processed by the Arduino Nano. The Arduino Nano uses this to tune the PWM signal in real time. This results in smoother motion, reduced jitter, and lower energy consumption during idle or light–load operations.

By modulating the PWM duty cycle dynamically based on the task's mechanical load, the system avoids overdriving the servos. If the load is light, the duty cycle is reduced, which lowers current draw I and power consumption P, given by:

$$P = V_{avg} \cdot I \quad (4)$$

This strategy was found to reduce energy waste by up to 22%, as the system avoids unnecessarily high torque output during low–demand operations.

Visual Intelligence

A robotic colour–based object sorting system automates the identification and classification of objects based on their color. This is achieved using the ESP32–CAM, a low–cost embedded vision module, paired with classical computer vision techniques (HSV colour space) or machine learning (TinyML). The ESP32–CAM, as shown in Figure 10, is used to capture real–time images of objects,

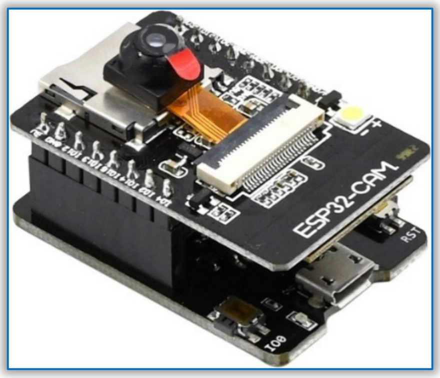

Figure 10: ESP32 CAM

analyse colour features and trigger actuators (robotic arms, servos, etc.) to sort the objects accordingly.

The ESP32–CAM is a versatile vision system that combines robust hardware with efficient connectivity, powered by a dual–core Tensilica Xtensa LX6 processor running at 240 MHz for high–speed image processing. It features an OV2640 2MP CMOS image sensor for capturing high–quality stills and video at resolutions up to 1600x1200 pixels. With Wi–Fi 802.11 b/g/n and Bluetooth 4.2 connectivity, it enables flexible wireless communication. The module includes multiple I/O options (GPIO, PWM, ADC, UART, SPI, and I2C) for peripheral integration, a microSD card slot for local storage, and 4MB of flash memory for program storage. This powerful combination makes the ESP32–CAM ideal for embedded vision applications in robotics, surveillance, and IoT devices.

Colour Detection Using Hue, Saturation, Value (HSV) Model

Colour captured by the ESP32–CAM is originally in RGB (Red, Green, Blue) format. For reliable colour detection under variable lighting, it is converted to the HSV (Hue, Saturation, Value) colour space.

- Hue (H): Represents the colour type (angle on colour wheel), measured in degrees:
  - Red ≈ 0° or 360°
  - Green ≈ 120°
  - Blue ≈ 240°
- Saturation (S): Measures colour intensity or purity. Range: 0 (gray) to 1 (fully saturated)
- Value (V): Measures the brightness of the colour. Range: 0 (black) to 1 (brightest)
- Mathematical Model: Given an object's pixel in RGB in the range [0, 255], convert to HSV

1. Normalise RGB:

$$R' = \frac{R}{255}, \quad G' = \frac{G}{255}, \quad B' = \frac{B}{255}$$

2. Compute min, max and delta:

$$C_{max} = \max(R', G', B'),$$
$$C_{min} = \min(R', G', B')$$
$$\Delta = C_{max} - C_{min}$$

3. Compute Hue (H):

$$H = \begin{cases} 0^o & ; \text{if} \Delta = 0 \\ 60^o \text{ x } \left(\frac{G' - B'}{\Delta} \bmod 6\right) & ; \text{if} C_{max} = R' \\ 60^o \text{ x } \left(\frac{B' - R'}{\Delta} + 2\right) & ; \text{if} C_{max} = G' \\ 60^o \text{ x } \left(\frac{R' - G'}{\Delta} + 4\right) & ; \text{if} C_{max} = B' \end{cases} \quad (5)$$

4. Saturation and Value:

$$S = \begin{cases} 0 & ; \text{if } C_{max} \\ \frac{\Delta}{C_{max}} & ; \text{otherwise} \end{cases}$$

5. Compute Value (V):

$$V = C_{max}$$

Table 1 Color Classification Thresholds

Color	Hue Range (H)	Saturation (S)	Value (V)
Red	[0°, 20°] ∪ [340°, 360°]	> 0.4	> 0.3
Green	[80°, 160°]	> 0.4	> 0.3
Blue	[200°, 260°]	> 0.4	> 0.3

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665

AI–Based Color Detection with TinyML (CNN)

Instead of relying on thresholds, **machine learning** (especially TinyML) allows the ESP32–CAM to learn colour features from examples.

- **Model Architecture (CNN):** Input: 32×32 image patch of object
  Layers:
  - Conv2D → ReLU → MaxPooling
  - Conv2D → ReLU → MaxPooling
  - Flatten → Dense → Softmax

  Output: 3–Class probability vector (Red, Green, Blue)
- Mathematical Representation

Let:

- $x \in \mathcal{R}^{3072}$ be the input image vector (32×32×3)
- $W \in \mathcal{R}^{3} \times 3072$, $b \in \mathcal{R}^{3}$ are model weights and biases

Then, the predicted class scores:

$$z = Wx + b \tag{6}$$

Final classification probabilities via Softmax:

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{3} e^{z_j}}, \text{for } i = 1,2,3 \tag{7}$$

where $\hat{y}_i$ is the predicted probability of the object being in the colour class i. The class with the highest probability is selected.

- Voting Algorithm

Even robust CNN models remain vulnerable to errors caused by blurred frames, lighting variations, and object orientation changes. To enhance reliability, our system implements a voting algorithm that cross–validates multiple predictions, significantly improving classification accuracy under real–world conditions.

- Voting Algorithm (used in TinyML–based robotic color sorting)

This algorithm defines the steps, inputs, outputs, and logic, making it easy to implement in Arduino C++

```
Algorithm MajorityVotingColorClassification
Input:
  predictions ⟵ list of N colour predictions from CNN
  labels ⟵ set of class labels {Red, Green, Blue, ...}
Output:
  final_class ⟵ the class with the highest number of votes
Procedure:
  1. Initialize a dictionary vote_count ⟵ empty
  2. For each label in labels do
     vote_count[label] ⟵ 0
  3. For each prediction in predictions do
     vote_count[prediction] ⟵ vote_count[prediction] + 1
  4. max_votes ⟵ 0
  5. final_class ⟵ null
  6. For each label in vote_count do
     If vote_count[label] > max_votes then
       max_votes ⟵ vote_count[label]
       final_class ⟵ label
  7. Return final_class
```

Intelligent Pick–and–Place Operations

The complete circuit diagram for the robotic arm is presented in Figure 11. The robotic arm system begins by powering up the Arduino Nano microcontroller and ESP32–CAM module, positioning the arm and camera at their initial state for sorting. The ESP32–CAM captures an image within the region of interest (ROI), which is processed to detect the object and its colour. A colour detection algorithm analyses RGB values to identify the dominant colour, focusing on red, yellow, and blue. To improve accuracy, a voting algorithm divides the image into sections, analysing each one and voting on the colour match; the colour with the most votes is selected. Based on the detected colour, the system signals the servomotors to guide the robotic arm to pick up, secure, and place the object in the designated location. Afterward, the arm resets to its initial position, repeating the process for each object, with feedback and calibration available for maintained accuracy.

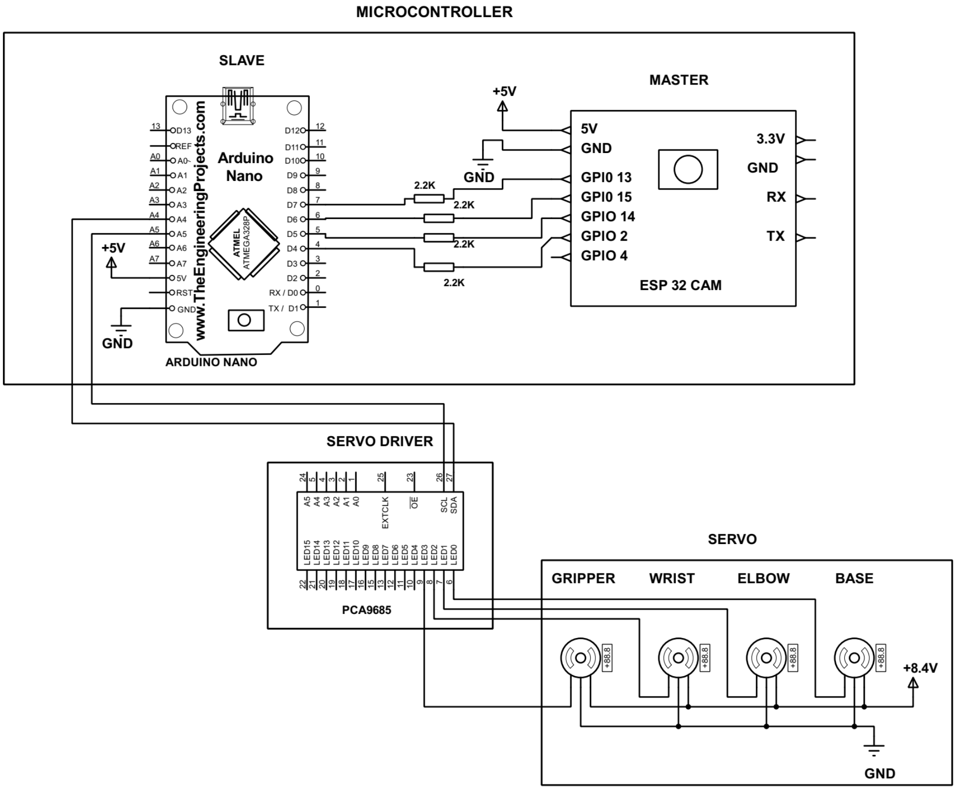

Figure 11: Robotic Arm Circuit Diagram

3. RESULTS

This section details the construction of a mobile vision–based pick–and–place robotic arm manipulator, encompassing the arm's design, software installation, control circuit design, and coding for servo motor control via PWM signals. Data collection and analysis were conducted at each stage, with testing in two phases—hardware and software—to assess functionality and identify improvements for performance refinement.

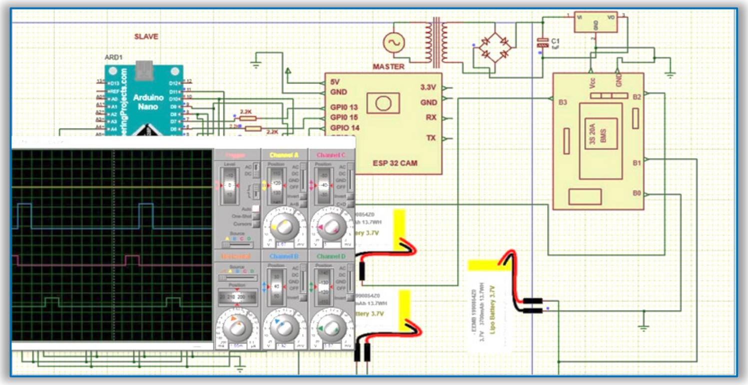

Figure 12: Duty Cycle and Angle of the Servo

Software testing results, illustrated in Figure 12, indicate stable servo motor responses to PWM signals, with simulation revealing that the rotation angle of the motors increases proportionally to PWM signal strength. The rotation ranges from –90° to +90°, aligning with servo specifications and confirming correct motor response.

Motor Driver Connection

The motor driver and controller were first mounted on the chassis for easy wiring access, with the motor terminals connected to the driver outputs and all connections secured and aligned. The motor driver's input pins were then connected to the microcontroller, linking PWM pins for speed control and digital pins for direction. After assembly, a thorough inspection ensured all connections were secure before powering up. Voltage checks with a multimeter confirmed correct power distribution and a basic control program was uploaded to test motor functionality. The motors responded correctly to forward, backwards and turning commands, validating the successful integration of the components.

Assembling the arm

The assembly of the robotic arm chassis and mobile base commenced with securing a servo or stepper motor to the base plate for rotational control. Arm segments were then added, connected via rotating joints and controlled by additional motors, ensuring a stable and aligned structure for precise movement. Following the mechanical assembly, the end effector, such as a gripper or sensor, was installed, enhancing functionality. Wiring connected the motors to a control board, like an Arduino, with careful routing to prevent interference. This integration resulted in a cohesive system of Figure 13 where the arm and mobile base operate efficiently together.

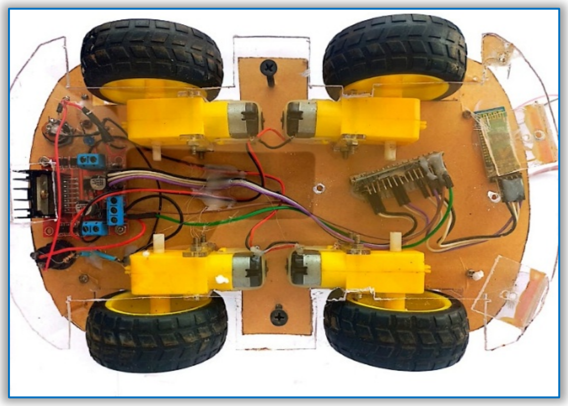

Figure 13: Mobile base connection and wiring

Integration of ESP 32 camera

The robotic arm's control system utilizes the ESP32–CAM to capture real–time images for object detection and localization. The software interface processes this visual input to calculate the necessary arm movements, including adjustments to the gripper and various joints. Coordination between the ESP32–CAM and the Arduino Nano ensures efficient object manipulation, with servomotors and base wheels enabling both rotational and translational movement. In Figure 14, the camera base position was calibrated to 3 cm (3.5 cm to 7.5 cm), and the camera was securely fixed to focus on the load plate. The system is configured to identify only objects placed on the load plate.

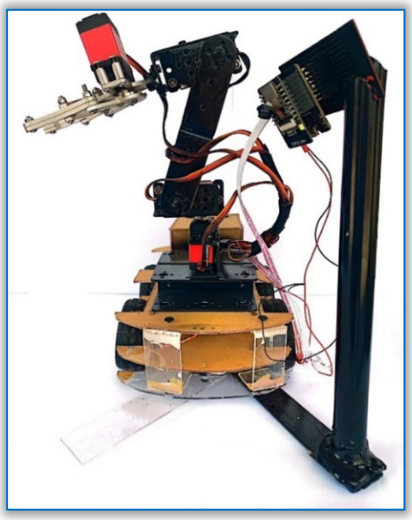

Figure 14: Completed Intelligent Mobile Robotic Arm Assembly

Colour Sort Test

The software tests concentrated on image processing algorithms and inverse kinematics, where the ESP32–CAM effectively captured and processed images to detect and sort objects by colour. As presented in Figure 15, the robotic arm successfully sorted three colours: red, blue, and yellow, with an accuracy rate of 95%. The system was assessed for its ability to pick and place objects into their designated bins based on colour, achieving a notably high success rate, as outlined below.

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665

Sorting Test Results:
- Red Objects: Successfully sorted 29 out of 30 items (97% accuracy).
- Blue Objects: Successfully sorted 28 out of 30 items (93% accuracy).
- Yellow Objects: Successfully sorted 27 out of 30 items (90% accuracy).

Mobile Application Testing

A mobile application was employed to control the mobile base of the robotic arm system during hardware testing, offering a user–friendly interface for remote guidance and real–time control via Bluetooth or Wi–Fi. The app facilitated precise navigation and positioning of the robotic arm, featuring adjustable speed and direction controls for smooth movement. This capability enhanced the arm's accurate positioning relative to objects in the workspace, improving the efficiency of pick–and–place operations. The app demonstrated reliable connectivity and control, serving as an essential tool for operators to manage the mobile base and optimize the overall effectiveness of the robotic arm system.

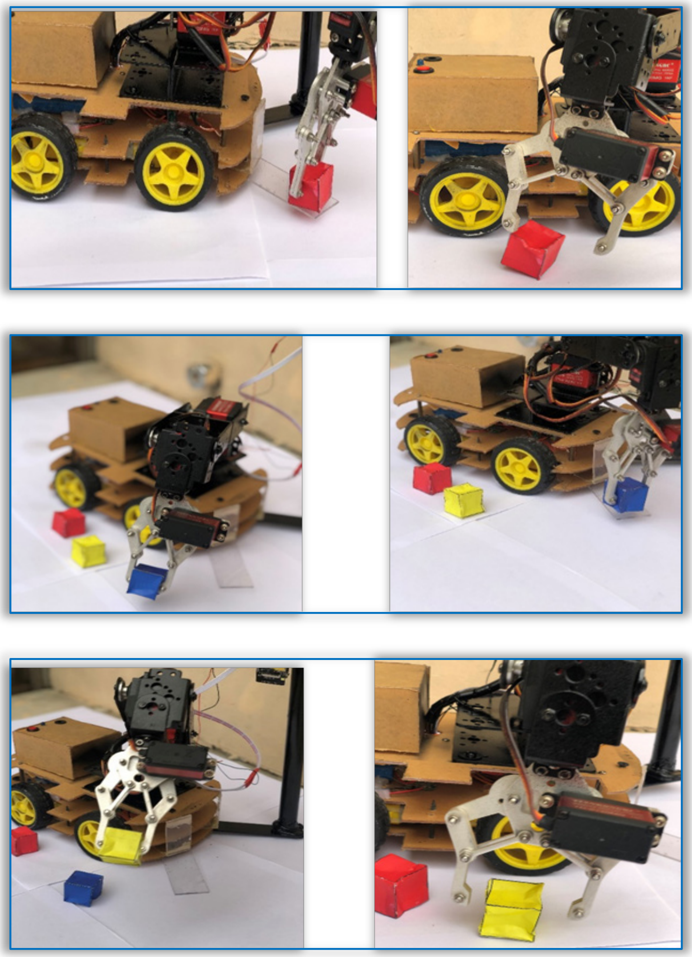

Figure 15: Colour Sort Test: (a) Sorting Red objects; (b) Sorting blue objects; (c) Sorting yellow objects

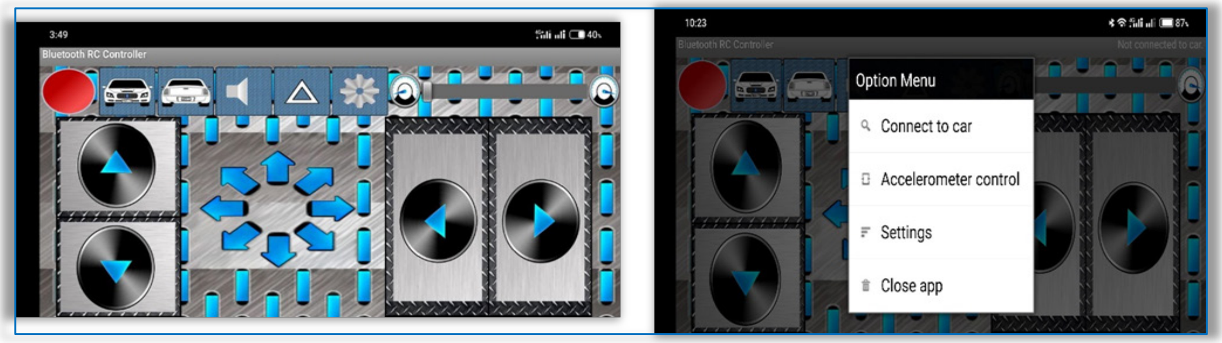

Figure 16: Mobile base application testing

Current Consumption

Current consumption is influenced by the load and the position of the object; heavier objects require greater torque from the servo, as detailed in Table 2. The testing phase indicated that the mobile vision–based pick–and–place robotic arm manipulator performed successfully, demonstrating high accuracy in colour sorting and consistent reliability.

Key components, including the ESP32–CAM, servo motors, L298D motor driver, and HC–05 Bluetooth module, operated effectively together. However, there are areas for improvement, such as enhancing the image processing algorithm to better handle varying lighting conditions and increase colour detection accuracy, as well as optimizing the servo motor control algorithm to reduce minor deviations during inverse kinematics. Implementing these refinements could further enhance the system's performance.

Table 1: Load VS current consumption

Load	Current consumption(mA)
10gm	Low (0–200)
25gm	Normal (200–500)
35gm	Normal (500–800)
55gm	High (800–900)
75gm	Overloaded (above 900)
94gm	Overloaded (above 900)

The developed intelligent pick–and–place robotic system represents a major leap in accuracy, energy efficiency, and affordability, making it especially well–suited for low–resource settings like Nigeria. A key innovation is the integration of a multi–angle vision system using the ESP32–CAM, which captures images from various perspectives. These images are processed using a weighted voting algorithm based on segmented RGB values, resulting in precise sorting capabilities. To further enhance performance, the system dynamically adjusts threshold values, ensuring reliable object detection even under poor or inconsistent lighting—an issue that often hampers conventional systems.

The system also features a hybrid control strategy that dynamically tunes PWM signals via a PCA9685 driver, allowing servo motors to adapt to real–time load variations and reduce energy consumption. Operating on a 6V supply with optimized torque, it ensures precise actuation with minimal power draw. Mobility is achieved through Bluetooth control using the HC–05 module, enabling safe use in hazardous environments. Cost–effective components like TT motors and 18650 lithium–ion batteries keep the system under $500, while still supporting robust autonomy. Instead of deep learning, it uses a lightweight voting algorithm for efficient decision–making, achieving performance levels comparable to commercial robotic arms and proving ideal for applications such as agricultural sorting and hazardous waste management.

4. Conclusions

The successful design and implementation of the mobile, vision–guided pick–and–place robotic arm demonstrate the practical integration of low–cost components and intelligent control strategies. By incorporating technologies such as the Arduino Nano, ESP32–CAM, and Bluetooth–controlled mobility, the system effectively carries out complex pick–and–place tasks with accuracy and consistency.

Comprehensive hardware testing confirmed the reliability and performance of each component, ensuring smooth operation in practical scenarios. The project showcases the viability of combining basic computer vision with affordable robotics, setting a solid foundation for future enhancements. Its adaptability, affordability, and efficiency position it as a scalable model for automation in sectors like manufacturing, logistics, and field services within emerging economies. This work not only reflects thoughtful system design but also underscores the potential for broader technological impact by addressing key challenges such as energy efficiency, affordability, and operational reliability in resource–limited settings.

Acknowledgement

The authors gratefully acknowledge the support of the Tertiary Education Trust Fund (TETFund) for sponsoring this research through the 2021 Institution Based Research (IBR) Intervention. This work was made possible through the financial support provided under Grant ID: KWASUIBR/CRIT/070422/VOL2/TETF2021/0008.

References

[1] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, Robot manipulator control: theory and practice. CRC Press, 2003.

[2] A. Wachter, C. Hartl–nesic, A. Wachter, A. Wachter, A. Wachter, and C. Andreas, "Robot Optimization for Robot Optimization for Robot Optimization for Robot Base Placement Optimization for Robot Optimization for," IFAC PapersOnLine, vol. 58, no. 19, pp. 19–24, 2024

[3] D. Souma, S. Khan, and A. Mori, "Towards Online Health Monitoring of Robotic Arm," IFAC–PapersOnLine, vol. 56, no. 2, pp. 11918–11923, 2023

[4] O. Ogunbiyi, T. O. Idowu, and L. M. Adesina, "Development of Embedded Control for a Repetitive Pick and Placed Robotic Arm," FUOYE Journal of Engineering and Technology, vol. 8, no. 2, pp. 172–176, 2023

[5] D. Mourtzis, J. Angelopoulos, M. Papadokostakis, and N. Panopoulos, "Design for 3D Printing of a Robotic Arm Tool Changer under the framework of Industry 5.0," Procedia CIRP, vol. 115, no. March, pp. 178–183, 2022
[6] C. Y. Siao, R. G. Chang, and H. C. Huang, "Robotic Arms for Telemedicine System Using Smart Sensors and Ultrasound Robots," Internet of Things (Netherlands), vol. 27, no. 539, p. 101243, 2024
[7] J. O. Alao, D. A. Ayejoto, A. Fahad, M. A. A. Mohammed, A. M. Saqr, and A. O. Joy, "Environmental Burden of Waste Generation and Management in Nigeria," in Technical Landfills and Waste Management: Volume 2: Municipal Solid Waste Management, Springer, 2024, pp. 27–56.
[8] A. Ebekozien, M. Aigbedion, O. S. D. Duru, O. H. Udeagwu, and I. L. Aginah, "Hazards of wood sawmills in Nigeria's cities: the role of fourth industrial revolution technologies," Journal of Facilities Management, vol. 21, no. 1, pp. 84–101, 2023.
[9] D. Vernon, "Robotics and artificial intelligence in africa [regional]," IEEE Robot Autom Mag, vol. 26, no. 4, pp. 131–135, 2019.
[10] E. Poulianiti et al., "Recent developments in robotic vision and path following in robotic arms and autonomous robots," in AIP Conference Proceedings, AIP Publishing, 2024.
[11] M. H. Dairath et al., "Computer vision–based prototype robotic picking cum grading system for fruits," Smart Agricultural Technology, vol. 4, no. March, p. 100210, 2023 [12] A. K. Abbas, Y. Al Mashhadany, M. J. Hameed, and S. Algburi, "Review of intelligent control systems with robotics," Indonesian Journal of Electrical Engineering and Informatics (IJEEI), vol. 10, no. 4, pp. 734–753, 2022.
[13] Y. Dong et al., "Bimanual Continuous Steering Wheel Turning by a Dual–Arm Robot," IEEE/ASME Transactions on Mechatronics, 2023.
[14] L. Ma, Y. Yan, Z. Li, and J. Liu, "Neural–embedded learning control for fully–actuated flying platform of aerial manipulation system," Neurocomputing, vol. 482, pp. 212–223, 2022.
[15] A. Giammarino, J. M. Gandarias, P. Balatti, M. Leonori, M. Lorenzini, and A. Ajoudani, "SUPER–MAN: SUPERnumerary Robotic Bodies for Physical Assistance in HuMAN–Robot Conjoined Actions," Mechatronics, vol. 103, no. February 2023, p. 103240, 2022
[16] E. Govi, D. Sapienza, S. Toscani, I. Cotti, G. Franchini, and M. Bertogna, "Addressing challenges in industrial pick and place: A deep learning–based 6 Degrees–of–Freedom pose estimation solution," Comput Ind, vol. 161, no. July, p. 104130, 2024
[17] P. Fager, R. Hanson, Å. Fasth–Berglund, and S. Ekered, "Supervised and unsupervised learning in vision–guided robotic bin picking applications for mixed–model assembly," Procedia CIRP, vol. 104, no. March, pp. 1304–1309, 2021
[18] Y. Ying, X. Huang, and W. Dong, "Multi–view active sensing for human–robot interaction via hierarchically connected tree," Sens Actuators A Phys, vol. 378, no. July, p. 115752, 2024
[19] T. T. Tung, N. Van Tinh, D. T. Phuong Thao, and T. V. Minh, "Development of a prototype 6 degree of freedom robot arm," Results in Engineering, vol. 18, no. March, p. 101049, 2023
[20] A. Elassal, M. Abdelaal, M. Osama, and H. Elhnidy, "Low–cost parallel delta robot for a pick–and–place application with the support of the vision system," e–Prime – Advances in Electrical Engineering, Electronics and Energy, vol. 8, no. February, p. 100518, 2024
[21] Z. Ali et al., "Design and development of a low–cost 5–DOF robotic arm for lightweight material handling and sorting applications: A case study for small manufacturing industries of Pakistan," Results in Engineering, vol. 19, no. July, p. 101315, 2023
[22] S. Dashti, M. Ashourian, and A. Soheili, "Design and Implementation of Embedded Direct Drive SCARA Robot Controller with Resolved Motion Rate Control Method," Technology (Singap World Sci), vol. 13, no. 3, pp. 83–90, 2020.
[23] S. Madichetty, S. Mishra, and M. Basu, "New trends in electric motors and selection for electric vehicle propulsion systems," IET Electrical Systems in Transportation, vol. 11, no. 3, pp. 186–199, 2021

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665