

Research paper

Integration of Computer Vision and Robotic Handling for Inventory Management

Integración de visión por computador y manipulación robótica para gestión de inventarios

[Juan C. Guacheta-Alba](#)¹, [Anny Astrid Espitia-Cubillos](#)^{2*}, [Robinson Jiménez-Moreno](#)²

¹ [Universidade Federal do Rio de Janeiro](#), Rio de Janeiro, Brasil. ² [Universidad Militar Nueva Granada](#), Bogotá, Colombia.

*Correspondence E-mail: anny.espitia@unimilitar.edu.co

Received: August 19th, 2025

Modified: May 20th, 2026

Accepted: July 7th, 2026

Abstract

Context: This article presents a development that enables the identification of products on a production line and their subsequent storage using artificial intelligence techniques. This approach facilitates product handling by a robotic arm that picks up the products from the conveyor belt and moves them to a shelf, while controlling the location and inventory level of each product. The system also includes an automatic generator of real-time reports in PDF format.

Method: The design involves five stages: first, the construction of a database of virtual products with different packaging and labels; second, the establishment of the simulation environment, which includes all the components of the work area (robotic arm, conveyor belt, XY platform, and shelf); third, product detection using a You Only Look Once version 2 (YOLO v2) network followed by a Speeded-Up Robust Features (SURF) algorithm; fourth, the integration of the identification algorithms with the robotic arm and an automatic inventory report generator; and fifth, validation.

Results: The simulations showed a classification accuracy of 96.12% and correct placement of the products. The mean detection time with YOLO v2 was 150 ms per object, plus 55 ms for SURF, resulting in an end-to-end cycle time of 32 255 ms, dominated by robot motion in CoppeliaSim. These times confirm the system's capability to operate in near real time in a virtual industrial environment.

Conclusions: The development was tested in a virtual environment created in the CoppeliaSim software, which made it possible to demonstrate the correct implementation of the classification and

distribution algorithm, its ability to classify, locate, and control inventory levels, and its effectiveness in report generation.

Keywords: Artificial intelligence, Inventory control, Robotics, Simulation

Resumen

Contexto: Este artículo presenta un desarrollo que permite la identificación de productos en una línea de producción y su posterior almacenamiento utilizando técnicas de inteligencia artificial. Este enfoque facilita la manipulación de los productos por parte de un brazo robótico que los toma de la banda transportadora y los traslada a una estantería, a la vez que controla la ubicación y el nivel de inventario de cada producto. El sistema también incluye un generador automático de informes en tiempo real en formato PDF.

Método: El diseño consta de cinco etapas: en primer lugar, la construcción de una base de datos de productos virtuales con diferentes envases y etiquetas; en segundo lugar, el establecimiento del entorno de simulación, que incluye todos los componentes del área de trabajo (brazo robótico, banda transportadora, plataforma XY y estantería); en tercer lugar, la detección de productos mediante una red You Only Look Once versión 2 (YOLO v2) seguida de un algoritmo Speeded-Up Robust Features (SURF); en cuarto lugar, la integración de los algoritmos de identificación con el brazo robótico y un generador automático de informes de inventario; y en quinto lugar, la validación.

Resultados: Las simulaciones mostraron una exactitud de clasificación del 96.12% y una ubicación correcta de los productos. El tiempo medio de detección con YOLO v2 fue de 150 ms por objeto, más 55 ms de SURF, lo que da como resultado un tiempo de ciclo completo de 32 255 ms, dominado por el movimiento del robot en CoppeliaSim. Estos tiempos confirman la capacidad del sistema para operar casi en tiempo real en un entorno industrial virtual.

Conclusiones: El desarrollo se probó en un entorno virtual creado en el software CoppeliaSim, lo que permitió demostrar la correcta implementación del algoritmo de clasificación y distribución, su capacidad para clasificar, localizar y controlar los niveles de inventario, y su eficacia en la generación de informes.

Palabras clave: Inteligencia artificial, Control de inventario, Robótica, Simulación

1. Introduction

Recent advances and algorithms associated with artificial intelligence (AI) have boosted developments in intelligent industries [1], [2], with approaches ranging from the use of convolutional

networks (CNN) [3] to the implementation of transformer networks [4]. This has provided clear insights into the benefits and weaknesses of AI in the industry [5], thus evidencing the wide field of applications still under development.

Clustering algorithms for multi-feature identification in product quality [6], machine vision algorithms for product defect detection and automated box classification [7], [8], the use of neural networks for process control in manufacturing [9], and the implementation of deep learning algorithms in the detection of anomalies in industrial products [10] are clear examples of the incursion of AI into industry. Among these, CNNs, as a class of deep learning algorithms, enable image-based analysis in computer vision systems, which is very useful in the industrial environment [11].

You Only Look Once (YOLO) networks, a type of network derived from CNNs, are being used in Industry 4.0 and 5.0 thanks to their versatility in classifying and locating objects in images. This enables applications such as waste recycling [12], defect detection [13], [14], and damage detection [15], as well as applications related to video surveillance and object tracking [16]. Moreover, YOLO networks stand out for their suitability for embedded mobile systems [17], low-cost platforms [18], and devices with limited computational resources [19], which makes them ideal for Internet of Things (IoT) systems.

An Internet-connected system under Industry 4.0 and 5.0 opens up the possibility of managing aspects of the production line such as inventories [20] and end-consumer management [21]. This is linked to the development of smart inventory systems, i.e., inventories based on AI and IoT algorithms [22]. However, the state of the art shows no evidence of industrial environments that integrate the generation of inventories and reports based on the identification and location of specific products through AI.

Therefore, it is necessary to identify specific products in an industrial environment based on their type of packaging and content volume in order to perform real-time inventory queries. This paper presents the use of a YOLO v2 network in conjunction with a Speeded-Up Robust Features (SURF) algorithm [23] for the identification and storage of products, taking as a reference a virtual environment of a cleaning products company, in which a robotic arm picks up the plastic containers of different types of products and stores them. SURF is a computer vision algorithm that detects and describes interest points in images for object identification and is robust to changes in scale, rotation, and lighting. Work combining robots with localization algorithms such as the YOLO network is

common in the state of the art [24], [25]; the contribution of this paper is the use of these tools to manage the inventory of specific products and their storage in a virtual test environment.

This paper addresses a relevant and current topic: the integration of AI algorithms (YOLO v2 and SURF) for computer vision, robotic manipulation, and inventory management in industrial environments. It aims to answer the following research question: How can computer vision algorithms (YOLO v2 and SURF) be integrated with robotic manipulation to automate product identification and inventory management in a virtual industrial environment while generating real-time reports?

2. Materials and Methods

This research begins with the construction of a database of virtual products distinguished by packaging and label; the packaging is associated with the product size, and the label with the product type. With this information, the algorithm that identifies each product is trained using AI [26]. In the second stage, the simulation environment, which includes all the components of the work area, is established in the CoppeliaSim software [27]. There, a robotic arm is configured to move each product from the conveyor belt to the shelf.

In the third stage, a YOLO v2 network [27] followed by a SURF algorithm [29] is used to detect products on the conveyor belt and identify each one by size and item type [27]. In the fourth stage, the identification algorithms are integrated with the robotic arm, which is responsible for defining the routes to move each product, and with an automatic inventory report generator that includes location and quantity. Finally, the designed algorithms are validated by testing the identification of packaging and product labels, the handling of products and their placement on the shelf by the robotic arm, and the generation of reports.

The UR5 manipulator used in this work is a collaborative robotic arm with six degrees of freedom (DOF) that contribute to both the translational and rotational movements of the end effector. The UR5 is characterized by its lightweight, fast, and safe design, which makes it ideal for industrial applications. Figure 1 shows the parameters, reference frames, rotation axes, and joints of the UR5 manipulator, where the X, Y, and Z axes are represented in red, green, and blue, respectively.

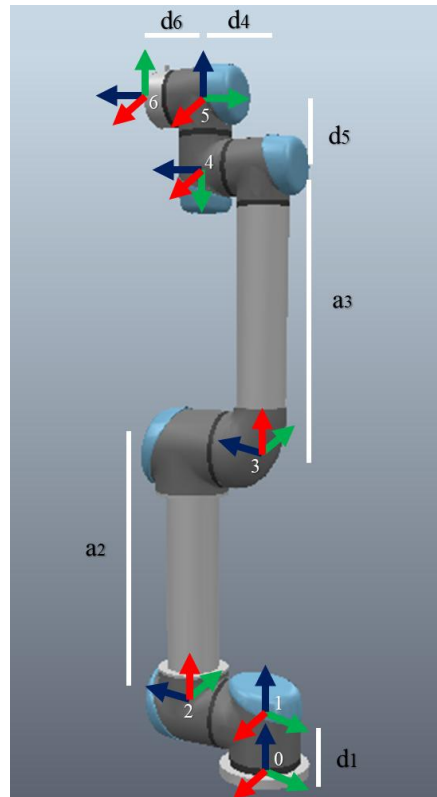


Figure 1. UR5 manipulator in CoppeliaSim

The kinematics of the UR5 manipulator is modeled using the Denavit–Hartenberg (DH) parameters, which define the geometric relationships between its links and enable the calculation of the position and orientation of the end effector. Table I presents the DH parameters of the UR5, including link distances and angles required for kinematic analysis.

Table I
Denavit-Hartenberg Parameters of the UR5 Manipulator

i	a_i [m]	α_i [rad]	d_i [m]	θ_i [rad]
0	0	0	-	-
1	0	$\pi/2$	0.08916	θ_1
2	0.425	0	0	θ_2
3	0.39225	0	0	θ_3
4	0	$\pi/2$	0.10915	θ_4
5	0	$-\pi/2$	0.09456	θ_5
6	-	-	0.0823	θ_6

The position and orientation of the end effector are determined by multiplying the transformation matrices derived from the DH parameters. Equation (1) represents the transformation from the base to the end effector:

$$T_6^0(q) = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 \quad (1)$$

Inverse kinematics aims to compute the joint angles needed to achieve a desired end-effector position and orientation. The process starts by solving for θ_1 and θ_5 based on the reference frame transformations. Using the position of the fifth joint, θ_1 is determined, which yields two possible configurations. Once θ_1 is obtained, θ_5 is computed, followed by θ_6 using spherical coordinates. The remaining angles, θ_2 , θ_3 , and θ_4 , are calculated by considering the parallel axes of joints 2, 3, and 4, which behave as a three-DOF planar arm. The inverse kinematics approach was implemented in MATLAB R2025^a.

3. Simulation Environment

The development begins with the identification and storage of specific products on a production line. These products are packaged in containers that vary in shape, size, and content; Figure 2 shows some examples of real packaging and labels.



Figure 2. Packaging and labels

The real products were digitized for manipulation in the CoppeliaSim virtual environment. Figure 3 illustrates the correspondence between real products and their digitized models, which include five

packaging sizes (500 ml, 800 ml, ½ gal, 1 gal, and 2 gal) and six cleaning product categories (dishwashing liquid, neutral cleaner, multipurpose cleaner, polymeric wax, bleach, and ultra cleaner).



Figure 3. Real and digitized products. Adapted from [26]

To train and validate the YOLO v2 network, a synthetic dataset was generated entirely in CoppeliaSim. Variations in textures (seven container colors and three cap colors) and labels (six designs) produced 55 representative digital objects that closely resemble the real products. Figure 4 shows the initial variations of product images used to build the dataset. These images were rotated through a full 360° in 10° increments, generating 36 views per product. Subsequently, greater random variability was introduced in orientation ($\pm 180^\circ$), scale ($\pm 8\%$, $\pm 10\%$, and $\pm 12\%$), position, light intensity, and illumination direction. This was complemented by data augmentation in MATLAB R2025a (random rotations, horizontal flips, brightness and contrast fluctuations, background changes, and partial occlusions), resulting in a database of 6000 images. Image labeling was automated by extracting product identifiers and projecting their 3D coordinates onto the image plane to obtain bounding boxes, which were then exported to a YOLO-compatible format (based on Pascal VOC/COCO). The final dataset was divided into 70% for training (4200 images), 15% for validation (900 images), and 15% for testing (900 images), ensuring a balanced representation of the classes and consistency in the annotations. All images are in color and were captured by a virtual camera located at the conveyor belt sensor, configured with a resolution of 640×480 pixels, a perspective angle of 60° , near and far clipping planes of 0.01 m and 10 m, and the Legacy OpenGL rendering mode. The network input size was set to 416×416 pixels.



Figure 4. Initial variations of product images for the database

In the CoppeliaSim software, a complete work environment was designed with all the components necessary for product handling, namely a conveyor belt, a UR5 robotic arm, an XY platform, and a storage rack, arranged as shown in Figure 5. In this virtual environment, the conveyor belt automatically moves the objects from the production area to the warehouse area. The UR5 robotic arm was configured to perform precise movements using forward and inverse kinematics algorithms, which allows it to manipulate and sort the products. To increase the robot's flexibility and range of motion, an XY platform was integrated to provide additional movement in the horizontal plane, which allows the robotic arm to be positioned anywhere in the workspace, mainly for storage tasks. The inventory is based on the products located in a storage rack with a capacity of 50 products, arranged in 10 columns and 5 rows.

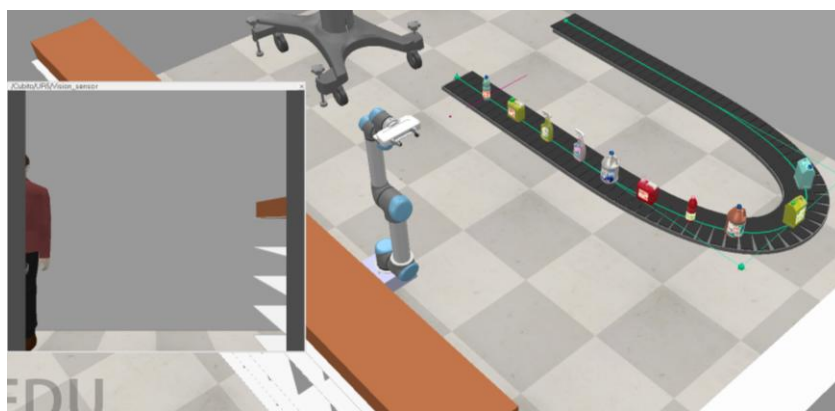


Figure 5. Simulation environment with conveyor belt, robotic arm, XY platform, and storage rack

The identification of each product is performed in two stages. The first stage uses the YOLO v2 network [28], which detects the products along the conveyor belt in a general and accurate manner. This algorithm provides the spatial localization of each object, giving the robotic arm the information

necessary to grasp the product and place it correctly in its designated location. The YOLO v2 network consists of 19 convolutional layers and 5 max-pooling layers, with Leaky ReLU (rectified linear unit) activation functions and batch normalization in all layers. As shown in Table II, YOLO v2 was trained for 10 epochs using the root mean square propagation (RMSprop) optimizer, five anchor boxes ($k = 5$), and an initial learning rate of 0.001. With class balancing, the network achieved an accuracy and F1-score of 92.4%, reflecting its ability to perform reliable detections.

Table III
Training Parameters of the YOLO v2 Network

Parameter	Configuration
Pretrained Network	YOLO v2
InputSize	[416 416 3]
Feature Layer	"leaky_relu_5"
Number of Anchors	5
Optimizer	RMSprop
Initial Learning Rate	0.001
Gradient Decay Factor	0.9
Squared Gradient Decay Factor	0.999
Mini Batch Size	8
Maximum Epochs	10
Learning Rate Schedule	"piecewise"
Learning Rate Drop Period	5
L2 Regularization	0.0005
Hardware Resource	Simple CPU
Validation Frequency	50 iterations
Training Time	193 minutes
Accuracy	92.44%
Macro F1	92.45%
Average Recognition Time	326 ms

Table III presents the metrics by class: the best-recognized product is bleach with 94% accuracy, and the worst is neutral cleaner with 89%, which shows that product types are distinguishable. Figure 6 shows the confusion matrix, which reveals infrequent confusion between classes of product.

Table IIIII
Training Metrics by Class

Class	Precision	Recall	F1	Support
Dishwashing liquid	0.9315	0.9067	0.9189	150
Neutral cleaner	0.8910	0.9267	0.9085	150
Multipurpose cleaner	0.9205	0.9267	0.9236	150
Polymeric wax	0.9379	0.9067	0.9220	150
Bleach	0.9456	0.9267	0.9360	150
Ultra cleaner	0.9226	0.9533	0.9377	150

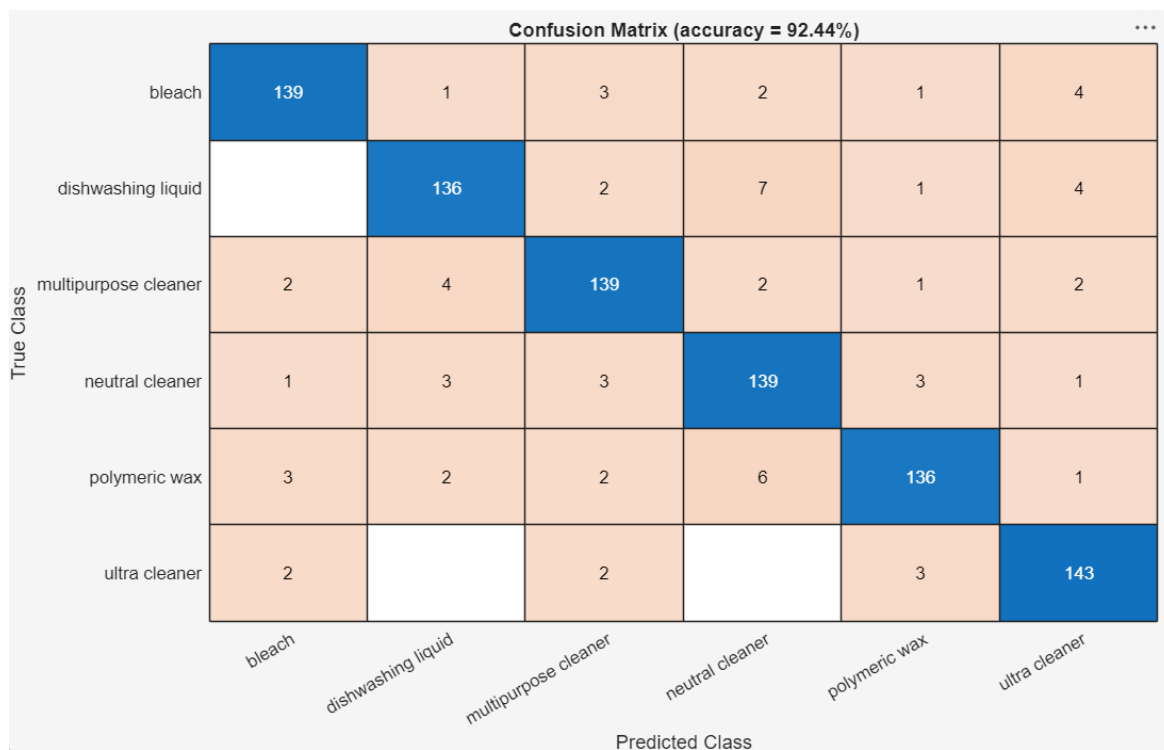


Figure 6. YOLO v2 confusion matrix

The second, more specific stage employs the SURF algorithm [29], which identifies distinctive features of the products detected in the first stage. SURF uses interest points to compare images and classify products accurately, which facilitates their correct placement on the corresponding storage shelf. Figure 7 highlights the 10 most relevant interest points identified by SURF on the real labels of the selected products. The image captured in the CoppeliaSim virtual environment allows the product to be classified correctly and a bounding box to be generated that estimates the label's position. This combination of the YOLO v2 and SURF algorithms aims to ensure a robust and efficient system for product classification and handling in the simulated environment.



Figure 7. Interest points selected by the SURF algorithm on the labels of the six selected products.

4. Results

Tests of the detection algorithm were performed on different objects positioned along the conveyor belt. In these tests, both the types of packaging and the products present in the simulated environment were evaluated. The results of these tests, which include the estimation and classification of the products, are shown in Figure 8, which presents examples of four types of packaging and four different products. Once the objects are captured by the camera, the algorithm detects them, classifies them, and calculates their quantity. These results enable real-time decision-making on the transport and classification of the products and the efficient management of the workflow in the automated system.

Using the class detected by the YOLO v2 algorithm for each product and the coordinates of the shelf, a table is generated that indicates the free positions on the shelf and the class of the product in each cell. First, the algorithm looks for the first free position in the row corresponding to the product class. If it finds one, it assigns the corresponding coordinates and updates the table to mark the position as occupied. Finally, it displays a message with the location assigned to the product and returns the coordinates of the selected position.

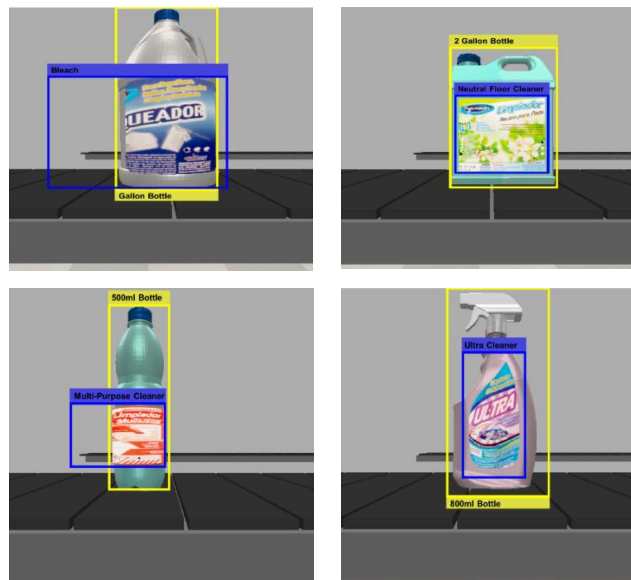


Figure 8. Results of the detection and classification of products on the conveyor belt, showing different types of packaging and products.

These operations are integrated into a broader product manipulation process in the simulated environment, whose structure is shown in the flowchart in Figure 9. Initially, the robot receives the information from the camera and moves toward the position of the detected product. The camera then captures the image, which is processed using the YOLO v2 and SURF algorithms to classify each product and verify its location and the available shelf space. The robotic arm moves to this location, adjusts its gripper to pick up the product, and places it in the designated position on the shelf. Finally, the robotic arm returns to its initial position. The process also includes simulating the robotic arm's trajectory, ensuring precise and efficient movements for correct product positioning and thus improving the overall system performance.

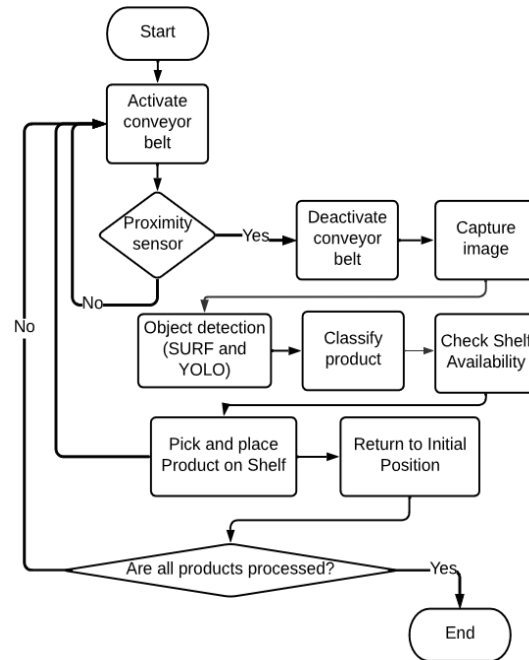


Figure 9. Flowchart of product handling in the simulated environment.

Ten tests were conducted to verify the functionality of the process described above. In these tests, the system had to classify products and place them on a shelf in an automated virtual environment. Figure 10 shows images of the test results, demonstrating that the different products were correctly placed on the shelf regardless of their dimensions, which confirms the robotic arm's ability to handle products of various shapes and sizes. Furthermore, these tests show that the robot can reach all positions on the shelf, ensuring an efficient and accurate distribution of objects without limitations in its range of motion.



Figure 10. Execution of the process of classification and arrangement of products on the shelf.

The classification of the 10 products was simulated effectively; the products were deliberately mixed to include a variety of shapes and sizes. During the simulation, the process was made as dynamic as possible, ensuring that the robot could sort and place the products efficiently on the shelf regardless of their dimensions. Throughout the simulation, the robot demonstrated its ability to adapt to different scenarios, performing precise movements, reaching all shelf positions, and ensuring correct distribution. Figure 11 shows how the 10 products were correctly classified and placed on the storage shelf, confirming the effectiveness of the system in classifying and correctly placing the products in the virtual scenario.

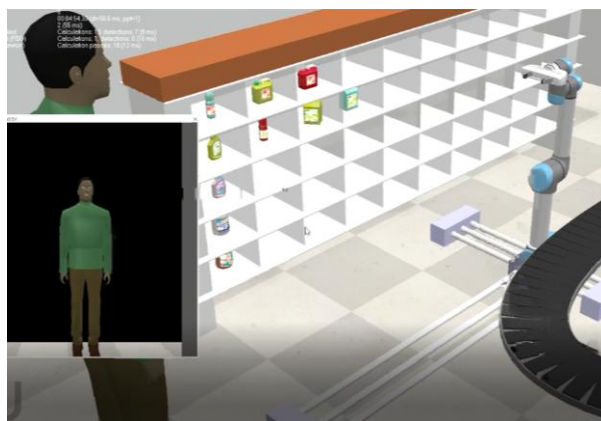


Figure 11. Final test results.

To assess the performance of the automated system more accurately, the execution time of each key process component was measured. The mean, extreme values, and standard deviation for object detection, feature matching, grasp planning, robot movement execution, and the complete cycle are presented in Table IV, which also includes the recognition and placement failure rates used to evaluate system reliability. The reported times correspond to measurements performed in MATLAB R2025a on a CPU (Intel Core i5-10300, 2.5 GHz, 5 GB of RAM), reflecting processing times on standard computational resources.

Table IV
Automated System Metrics

Metric	Mean (ms)	Std (ms)	Min (ms)	Max (ms)
YOLO v2 detection	150	30	100	220
Feature matching	55	20	30	100
Grasp planning	50	20	20	100
Robot motion execution	32 000	5000	15 000	45 000
End-to-end cycle	32 255	5010	15 150	45 420
Recognition failure rate (%)	8%	—	—	—
Placement failure rate (%)	3%	—	—	—

Table IV shows that robot movement is the determining factor in the duration of a complete handling cycle, reflecting the physical constraints and kinematics modeled in CoppeliaSim. In contrast, the computational tasks of YOLO v2-based product detection, SURF feature matching, and grasp planning are executed almost instantaneously. These results demonstrate the efficiency of both the YOLO v2 detector, which provides fast and reliable information for decision-making during product handling, and the SURF algorithm, which offers a lightweight and rapid verification step that complements the initial YOLO v2 detection, enabling bottleneck-free sorting and placement. Grasp planning, implemented through simple inverse kinematics calculations and trajectory generation, contributes minimally to the overall cycle, further emphasizing that computation is not the limiting factor. The low recognition and placement failure rates confirm that the system can reliably and dynamically handle various product types and packaging sizes.

In MATLAB R2025a, a detailed report is created that describes the inventory status, product classification, and storage process. The code automatically generates PDF reports that present the information in tables and graphs, showing the distribution by product type and packaging, along with images of the processed products. The code flow covers data generation, table creation, pie chart insertion, and product image inclusion. Once the process is complete, the PDF is saved, providing a visual and textual summary of the inventory updated in real time.

Using this functionality, a PDF report containing two inventory tables was generated for the simulation with 10 products. Table V presents the product classification by type and size. The rows are organized by product type: multi-purpose cleaner, neutral floor cleaner, ultra cleaner, bleach, and liquid dish soap, while the columns are organized by size: 500 mL, 800 mL, ½ gal, 1 gal, and 2 gal, resulting in 25 combinations. Each cell displays the quantity of products corresponding to that combination. A row and a column with the totals are also included.

Table V
Inventory of product types by packaging size

Product Type	500ml	800ml	½ gal	1 gal	2 gal	Total
Multi-Purpose Cleaner	1	0	2	0	0	3
Neutral Floor Cleaner	1	1	0	0	2	4
Ultra Cleaner	0	1	0	0	0	1
Bleach	0	0	0	1	0	1
Liquid Dish Soap	0	0	0	1	0	1
Total	2	2	2	2	2	10

Table VI presents, in order of classification, the type of product found in each shelf position. The shelf position is expressed using a row and column convention; for example, position C4 indicates that the product is in row C (the third row) and column 4.

Table VI
Location of each product

Product	Shelf Position
500ml Multi-Purpose Cleaner	A1
½ gal Multi-Purpose Cleaner	A2
800ml Neutral Floor Cleaner	B1
800ml Ultra Cleaner	C1

From these tables, two pie charts were generated in the report: one showing the classification of products by type and the other showing their distribution by size. These charts provide a clear view of the proportion of each product class in the simulation. The pie charts for the 10-object simulation are shown in Figure 12.

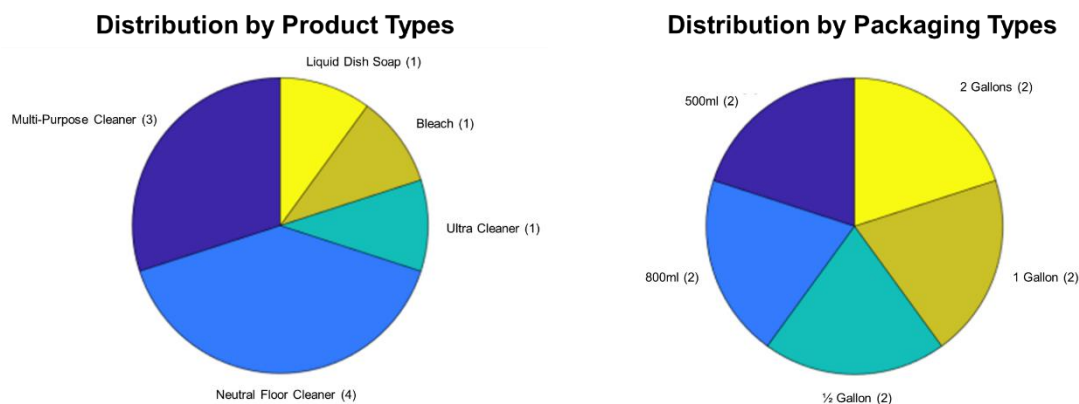


Figure 12. Pie charts for the 10-object simulation, classified by type and size.

5. Discussion

The YOLO v2 network has proven to be a robust architecture for classifying and locating objects in images. Although newer versions have since emerged, the version used is sufficient for the limited number of specific cleaning product classes. Controlled scenarios such as a virtual environment do not reveal key aspects of the algorithm's real-world performance; therefore, the algorithm was also validated with real products to contrast the results obtained. Figure 13a shows the identification by the YOLO v2 network of some of the products used in the virtual environment, demonstrating its functionality. In contrast, Figure 13b shows two extreme cases: direct exposure to a light source, and shadows caused by storage on the lower parts of a shelf. In both cases, the network failed to identify the products correctly, highlighting limitations in the dataset coverage and the algorithm's sensitivity to lighting and occlusion.



(a) Successful identification with YOLO v2. (b) Failed identification with YOLO v2.

Figure 13. Successful and unsuccessful identification results with YOLO v2.

Overall, the system demonstrates robust performance, with estimated recognition and placement failure rates of 8% and 3%, respectively (Table IV), indicating reliability for controlled classification tasks. As reported in [30], different versions of the YOLO network do not significantly affect performance in controlled industrial environments, and the confidence levels reported there are consistent with those obtained in this study. However, the network's performance in industry cannot be assessed without an actual implementation.

In terms of efficiency, the vision modules perform adequately. Detection with the YOLO v2 network, including capture, preprocessing, and postprocessing, takes an average of 150 ms per image, while feature matching on labels with SURF takes 55 ms. Grasp planning takes 50 ms per object. These values are minimal compared to the simulated robot movement, which dominates the entire cycle at approximately 32 s per object. This demonstrates that the perception and planning components of the

system are computationally lightweight, enabling real-time response in practical applications. Additionally, the system achieves 96.12% accuracy in product identification, owing to SURF, which contributes an improvement of 3.72 percentage points. Therefore, it is concluded that SURF complements and strengthens YOLO v2 by providing fast and accurate feature verification, making the integration suitable for product classification and location.

The system demonstrates flexibility in handling multiple product types and packaging sizes. By integrating YOLO v2 detection with SURF feature matching, the method can discriminate subtle variations in textures and labels that traditional rule-based approaches (such as color histograms or geometric heuristics) might miss. This flexibility is particularly important for distinguishing visually similar products or detecting partially occluded objects, confirming the suitability of the integrated approach for automated sorting in diverse scenarios. The vision and planning modules operate efficiently and reliably, supporting the system's viability for real-time automated sorting within the constraints of a controlled virtual environment.

6. Conclusions

This paper presents the design, implementation, and validation of a system for product identification and inventory management that integrates computer vision algorithms (YOLO v2 and SURF) with robotic handling, achieving an overall accuracy of 96.12% and per-class precision, recall, and F1-scores greater than 0.891, 0.9067, and 0.9085, respectively. The system was validated in a virtual environment (CoppeliaSim). The simulations demonstrate its ability to accurately detect, classify, and locate products of varying sizes and packaging types on a shelf, managing up to 10 objects in a dynamic sorting scenario. The use of YOLO v2 for rapid object detection, complemented by SURF for detailed label verification, proved to be an effective strategy, providing robustness against subtle differences in texture and labeling that would be difficult to detect using rule-based methods. It is concluded that, for real-world implementation, environmental lighting must be controlled, as extreme conditions can negatively affect identification.

Performance measurements indicate that the vision and planning modules are highly efficient: the initial detection of each product takes an average of 150 ms (using YOLO v2), feature matching verification (using SURF) takes 55 ms, and grasp planning takes 50 ms. The entire cycle was dominated by simulated robot movement, which takes an average of 32 255 ms per object. These

results show that computational costs are minimal compared to product handling, supporting the feasibility of real-time operation in a virtual industrial environment.

Additionally, the system generates detailed inventory reports in PDF format that provide a clear record of the classification, location, and distribution of product inventory. This functionality gives the proposed approach practical utility for monitoring and managing automated warehouse operations, highlighting the advantages of integrating agile detection, robust verification, and automated report generation. The proposed strategy demonstrates an overall accuracy exceeding 96%, along with flexibility and efficiency in a simulated environment.

The validation of the system in a virtual industrial scenario developed in CoppeliaSim constitutes an applied contribution, complemented by the measurement of execution times and the generation of inventory reports in PDF format, which together show the applied nature of the work.

Although these results are limited to a virtual environment, they constitute a solid foundation and benchmark for future implementation in real-world industrial settings, validating both the algorithmic design and the system integration.

Acknowledgements

This work is a product of the research project “Fortalecimiento de procesos de recepción de pedidos y control de inventario de materias primas soportado en industria 4.0” (INV-ING-4150), funded by the Vice-Rectorate for Research of Universidad Militar Nueva Granada in 2025. The authors, as full-time professors, thank Universidad Militar Nueva Granada for the funding and time allocated to the development of the project.

Author Contributions

J. Guachetá-Alba: Investigation, Data curation, Software, Writing – original draft. **A. A. Espitia-Cubillos:** Conceptualization, Writing – review & editing, Funding acquisition, Project administration, Resources. **R. Jiménez-Moreno:** Methodology, Formal analysis, Software, Writing – review & editing.

Access to Research Data

The data supporting the findings of this study are available at <https://github.com/juankg18/PRODUCT-INVENTORY-CONTROL-USING-ARTIFICIAL-INTELLIGENCE-TECHNIQUES-24027>

Conflict of Interest

The authors declare that there are no conflicts of interest that could have influenced the research or its interpretation.

Statement on Artificial Intelligence

The authors declare that they did not use generative AI or AI-assisted technologies during the preparation of this work.

References

- [1] J.-I. Park and T. Choi, “Regional Factors Associated with Smart Factory Adoption in South Korea,” *Technology in Society*, vol. 77, Art. no. 102509, 2024, <https://doi.org/10.1016/j.techsoc.2024.102509>
- [2] T. Kanangnanon, C. Fongsamut, and P. Jitngernmadan, “Optimal AI Models for Process Data Analysis in Industry 4.0 Regarding Digital Twin Implementation,” in Proc. 2024 8th *International Conference on Information Technology (InCIT)*, Chonburi, Thailand, 2024, pp. 675–680, <https://doi.org/10.1109/InCIT63192.2024.10810585>
- [3] B. Park, T. Kim, and J. Jong-pil, “AI-Smart Factory: Design and Verification of Korean SME AI Smart Factory Using Level Diagnosis System,” *Procedia Computer Science*, vol. 24, pp. 169–179, 2023, <https://doi.org/10.1016/j.procs.2023.09.025>
- [4] X. Fang, Q. Song, Z. Li, X. Wang, H. Ma, and Z. Liu, “Intelligent Monitoring System for Production Lines in Smart Factories: A Hybrid Method Integrating Transformer and Kalman Filter,” *Journal of Manufacturing Systems*, vol. 79, pp. 27–47, 2025, <https://doi.org/10.1016/j.jmsy.2024.12.014>
- [5] S. Hartmann, J. Brock, A. Kühn, and R. Dumitrescu, “Applying Artificial Intelligence in the Smart Factory: Lessons Learned from Real-World Use Cases,” *Procedia CIRP*, vol. 130, pp. 100–105, 2024, <https://doi.org/10.1016/j.procir.2024.10.062>

- [6] C. Liu, Y. Gao, J. Zhang, J. Liu, and S. Feng, “Research on Industrial Product Quality and Safety Emergency Detection Based on Multifeature Fusion,” in Proc. 2021 IEEE *International Conference on Industrial Application of Artificial Intelligence (IAAI)*, Harbin, China, 2021, <https://doi.org/10.1109/IAAI54625.2021.9699941>
- [7] M. Butolo et al., “Advanced Real-Time Surface Defect Detection in Industrial Products Using Machine Vision,” in Proc. 2024 IEEE SmartBlock4Africa, Accra, Ghana, 2024, <https://doi.org/10.1109/SmartBlock4Africa61928.2024.10779491>
- [8] J. C. Guacheta-Alba, A. A. Espitia-Cubillos, and R. Jiménez-Moreno, “Automated Box Classification in a Virtual Industrial Environment Using Machine Vision Algorithms,” in Proc. 12th *International Conference on Control, Mechatronics and Automation (ICCMA)*, London, UK, 2024, <https://doi.org/10.1109/ICCMA63715.2024.10843920>
- [9] V. Sumathi, R. S. C. C. S. Basavaraddi, V. J. I. M. V., and S. Srinivasan, “Advanced Process Control in Manufacturing Using IoT Devices and Artificial Neural Networks,” in Proc. 4th *International Conference on Sustainable Expert Systems (ICSES)*, Kaski, Nepal, 2024, <https://doi.org/10.1109/ICSES63445.2024.10763299>
- [10] Y. Jiang, W. Wang, and C. Zhao, “A Machine Vision-Based Realtime Anomaly Detection Method for Industrial Products Using Deep Learning,” in Proc. 2019 *Chinese Automation Congress (CAC)*, Hangzhou, China, 2019, <https://doi.org/10.1109/CAC48633.2019.8997079>
- [11] S. Chanda, R. D. Prangon, and K. N. Hoque, “A CNN-Based Approach for Room Number Detection Using Drone in Indoor Environment,” in Proc. 2024 IEEE *International Conference on Power, Electrical, Electronics and Industrial Applications (PEEIACON)*, Rajshahi, Bangladesh, 2024, <https://doi.org/10.1109/PEEIACON63629.2024.10800605>
- [12] R. Hamzah, L. Ang, R. Roslan, N. H. I. Teo, K. A. Samad, and K. A. F. A. Samah, “Comparing Modified YOLO V5 and Faster Regional Convolutional Neural Network Performance for Recycle Waste Classification,” in Proc. 2024 IEEE *International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Shah Alam, Malaysia, 2024, <https://doi.org/10.1109/I2CACIS61270.2024.10649835>
- [13] Y. Qi and H. Sun, “Defect Detection of Insulator Based on YOLO Network,” in Proc. 9th *International Conference on Electronic Technology and Information Science (ICETIS)*, Hangzhou, China, 2024, <https://doi.org/10.1109/ICETIS61828.2024.10593675>

- [14] H. Zheng, X. Chen, H. Cheng, Y. Du, and Z. Jiang, “MD-YOLO: Surface Defect Detector for Industrial Complex Environments,” *Optics and Lasers in Engineering*, vol. 178, Art. no. 108170, 2024, <https://doi.org/10.1016/j.optlaseng.2024.108170>
- [15] Y. Ji, A. Zhang, Z. Chen, M. Wei, Z. Yu, and X. Zhang, “Lightweight Road Damage Detection Algorithm Based on the Improved YOLO Model,” in Proc. 5th *International Conference on Artificial Intelligence and Electromechanical Automation (AIEA)*, Shenzhen, China, 2024, <https://doi.org/10.1109/AIEA62095.2024.10692408>
- [16] A. Anish, S. R. A. H. Malini, and T. Archana, “Enhancing Surveillance Systems with YOLO Algorithm for Real-Time Object Detection and Tracking,” in Proc. 2nd *International Conference on Automation, Computing and Renewable Systems (ICACRS)*, Pudukkottai, India, 2023, <https://doi.org/10.1109/ICACRS58579.2023.10404710>
- [17] S. Xu, Y. Ji, G. Wang, L. Jin, and H. Wang, “GFSP-YOLO: A Light YOLO Model Based on Group Fast Spatial Pyramid Pooling,” in Proc. IEEE 11th *International Conference on Information, Communication and Networks (ICICN)*, Xi’an, China, 2023, pp. [start]–738, <https://doi.org/10.1109/ICICN59530.2023.10393445>
- [18] R. Narwaria, A. Ahirwar, A. K. Prajapati, A. Kumar, and A. Tiwari, “Smart Object Detection Using ESP32-CAM Based on YOLO Algorithm,” in Proc. 2nd *International Conference on Intelligent Cyber Physical Systems and Internet of Things (ICoICI)*, Coimbatore, India, 2024, <https://doi.org/10.1109/ICoICI62503.2024.10696374>
- [19] J. Xu, F. Pan, X. Han, L. Wang, Y. Wang, and W. Li, “EdgeTrim-YOLO: Improved Trim YOLO Framework Tailored for Deployment on Edge Devices,” in Proc. 4th *International Conference on Computer Communication and Artificial Intelligence (CCAI)*, Xi’an, China, 2024, <https://doi.org/10.1109/CCAI61966.2024.10602964>
- [20] R. Bose, H. Mondal, I. Sarkar, and S. Roy, “Design of Smart Inventory Management System for Construction Sector Based on IoT and Cloud Computing,” *e-Prime – Advances in Electrical Engineering, Electronics and Energy*, vol. 2, Art. no. 100051, 2022, <https://doi.org/10.1016/j.prime.2022.100051>
- [21] S. Weißhuhn and K. Hoberg, “Designing Smart Replenishment Systems: Internet-of-Things Technology for Vendor-Managed Inventory at End Consumers,” *European Journal of Operational Research*, vol. 295, no. 3, pp. 949–964, 2021, <https://doi.org/10.1016/j.ejor.2021.03.042>

- [22] E. Saha and P. Rathore, “A Smart Inventory Management System with Medication Demand Dependencies in a Hospital Supply Chain: A Multi-Agent Reinforcement Learning Approach,” *Computers & Industrial Engineering*, vol. 191, Art. no. 110165, 2024, <https://doi.org/10.1016/j.cie.2024.110165>
- [23] Y. D. Pranata et al., “Deep Learning and SURF for Automated Classification and Detection of Calcaneus Fractures in CT Images,” *Computer Methods and Programs in Biomedicine*, vol. 171, pp. 27–37, 2019, <https://doi.org/10.1016/j.cmpb.2019.02.006>
- [24] M. Z. Chaari and G. P. Pereira, “Powering a Six-Axis Robotic Arm with AI to Collect Plastic Bottles,” in Proc. 2024 *International Conference on Microelectronics (ICM)*, Doha, Qatar, 2024, <https://doi.org/10.1109/ICM63406.2024.10815860>
- [25] F. J. Aguilar and H. Rodríguez, “Open-Source Integration Environment for Industrial Robot Arm NACHI MZ04 with ROS,” in Proc. 9th *International Engineering, Sciences and Technology Conference (IESTEC)*, Panama City, Panama, 2024, <https://doi.org/10.1109/IESTEC62784.2024.10820217>
- [26] J. C. Guacheta-Alba, A. A. Espitia-Cubillos, and R. Jiménez-Moreno, “Robotic Product-Based Manipulation in Simulated Environment,” *International Journal of Electrical and Computer Engineering*, vol. 15, no. 6, pp. 5894–5903, 2025, <https://doi.org/10.11591/ijece.v15i6.pp5894-5903>
- [27] S. Chakraborty and P. S. Aithal, “A Custom Robotic Arm in CoppeliaSim,” *International Journal of Applied Engineering and Management Letters (IJAEML)*, vol. 5, no. 1, pp. 38–50, 2021.
- [28] O. Caixiao, “Research on Object Grasping System Based on YOLO Algorithm,” in Proc. 2024 *International Conference on Telecommunications and Power Electronics (TELEPE)*, Frankfurt, Germany, 2024, <https://doi.org/10.1109/TELEPE64216.2024.00048>
- [29] S. Bakheet, A. Al-Hamadi, and R. Youssef, R. “A fingerprint-based verification framework using Harris and SURF feature detection algorithms,” *Applied Sciences*, vol. 12, no. 4, p. 2028, 2022, <https://doi.org/10.3390/app12042028>
- [30] L. Fan, Y. Wang, Y. Zhou, J. Zhang, and H. Chen, “YOLO-APGC: A Lightweight Method for BGA Surface Defect Detection Based on Cross-Scale Feature Fusion,” *Microelectronics Reliability*, vol. 175, Art. no. 115906, 2025, <https://doi.org/10.1016/j.microrel.2025.115906>

Authors and biographies

Juan C. Guachetá-Alba received his B.S. in Mechatronics Engineering from Universidad Militar Nueva Granada, Bogotá, Colombia, in 2020, and his M.Sc. in Mechatronics Engineering from the same university in 2022. He is currently working toward a Ph.D. in Mechanical Engineering at Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil. His research interests include robotics, multi-agent systems, and optimization.

Email: juan.camilo@mecanica.coppe.ufrj.br. ORCID: <https://orcid.org/0000-0001-9949-9005>

Anny Astrid Espitia-Cubillos received her B.S. in Industrial Engineering from Universidad Militar Nueva Granada, Bogotá, Colombia, in 2002, and her M.Sc. in Industrial Engineering from Universidad de los Andes, Bogotá, Colombia, in 2006. She is an Associate Professor in the Industrial Engineering Program at Universidad Militar Nueva Granada, Bogotá, Colombia.

Email: anny.espitia@unimilitar.edu.co. ORCID: <https://orcid.org/0000-0002-4791-0250>

Robinson Jiménez-Moreno received his B.S. in Electronic Engineering from Universidad Distrital Francisco José de Caldas, Bogotá, Colombia, in 2002, his M.Sc. in Engineering from Universidad Nacional de Colombia, Bogotá, Colombia, in 2012, and his Ph.D. in Engineering from Universidad Distrital Francisco José de Caldas in 2018. He is an Associate Professor at Universidad Militar Nueva Granada, Bogotá, Colombia, and his research focuses on the use of convolutional neural networks for object recognition and image processing in robotic applications such as human-machine interaction.

Email: robinson.jimenez@unimilitar.edu.co. ORCID: <https://orcid.org/0009-0003-6933-691X>