Ligpit Pang-Higpit: An AI-Integrated Bolts and Nuts Robotic Sorting System using Deep Learning and Computer Vision

Dave Simon DS. Jacinto, Jyd Rey A. Mercado, Jean Irah M. Ortega, Kimberly Joy V. Osias, Arnel D. Suarez

Electronics Engineering Department
Polytechnic University of the Philippines
Sta. Mesa Manila, Philippines
{davesimondjacinto, jydreyamercado, jeanirahmortega, kimberlyjoyvosias, arneldsuarez} @iskolarngbayan.pup.edu.ph

***Abstract*—This paper presents the development and performance of "Ligpit Pang-Higpit," an AI-integrated robotic sorting system designed to accurately classify bolts and nuts using deep learning and computer vision. The system utilizes the YOLOv8 algorithm for object detection and employs an A4Tech PK-910H 1080P HD webcam for image acquisition alongside a six-degree-of-freedom (6DOF) robotic arm controlled by an Arduino Uno board. The dataset, comprising images from various sources, underwent extensive preprocessing and augmentation to enhance the model's performance. The trained model demonstrated significant accuracy in real-time object classification and sorting, validated through metrics such as precision, recall, and mean average precision. The system's integration of AI enhances productivity, reduces human error, and minimizes repetitive tasks in manufacturing environments, contributing to sustainable industrial automation aligned with SDG 9. This study underscores the potential of AI-powered robotics to advance automation processes within the manufacturing sector.**

Keywords—Robotics Technology, Robotic Arm, Automation, Artificial Intelligence (AI), Deep Learning, Computer Vision, Bolts and Nuts

I. INTRODUCTION

Nowadays, there is an increasing demand for robots that can categorize and sort different objects. Various sectors, such as automotive, construction, and manufacturing, must integrate smoothly with automation or advanced robotic sorting systems [1].

Individuals in the manufacturing sector, particularly those producing bolts and nuts, may encounter human errors while arranging and sorting objects due to their dimensions and quantification. Bolts and nuts are essential hardware fasteners extensively used in various industries for securely joining multiple components [2].

Considering those challenges that may be encountered in the industry, advanced robotic sorting systems with Artificial Intelligence (AI) integration can also enhance productivity, reduce human error, less working time, and minimize repetitive tasks in manufacturing settings [1][3][4].

Artificial Intelligence (AI) is significant in addressing various challenges and enhancing the robotic manipulation of small partEven in unorganized environments, s. AI-powered vision systems can recognize and distinguish small pants. Additionally, AI can refine robotic movements, enabling more delicate grasping and manipulation of tiny objects [4][5].

By integrating AI with robots, manufacturers can develop more sophisticated systems that handle small parts with the precision and flexibility required for a fully automated solution [4][5].

Furthermore, some studies involve developing standardized robotic arm interfaces that can be readily incorporated into existing sorting lines and even designing robot arm movements that operate efficiently [1][3].

Implementing robotic sorting systems with AI should also be economically viable and optimal for industries. Considering that, this paper explores the integration of AI into robotic arms to enhance their capability to determine and sort objects, such as bolts and nuts, based on their dimensions. The project's objectives are to develop a robotic arm with an AI-integrated sorting system, to classify bolts and nuts through the YOLOv8 algorithm based on their appearance, and to determine and sort bolts and nuts into their designated locations.

This project focuses on introducing and developing a robotic arm capable of autonomously identifying and sorting bolts and nuts using the OpenCV and YOLOv8 algorithms. The robotic arm with an AI-integrated sorting system consists of six servo motors, providing it with six Degrees of Freedom (DOF) to sort objects effectively. The camera that will be used is an A4Tech PK-910H 1080P Full HD Web Cam.

The study also addresses several limitations and assumptions relevant to the study. These include restrictions related to hardware availability, algorithm performance influenced by environmental factors, and challenges arising from object variability. Furthermore, operational constraints regarding the robotic arm's speed, precision, and capacity in real-world settings are also considered.

Lastly, this study was aligned with Sustainable Development Goals (SDGs) No. 9: Industry, Innovation, and Infrastructure. It significantly impacts various industrial processes in sectors such as construction, manufacturing, and automotive, where rapid and precise sorting of small objects is important.

II. LITERATURE REVIEW

Tyler Herman, a Project & Process Manager at Fives Intralogistics Corp., discussed the growing use of robotics in warehouse sorting [7]. Herman noted that automation helps warehouses address labor shortages and reduce costs. He highlighted the benefits of robotic sorting, such as increased safety and scalability, while addressing challenges like initial costs and concerns about job displacement. Moreover, Boysen et al. [6] emphasized the importance of decision-making and optimization strategies in robotic sorting systems within large-scale warehouses. Critical areas highlighted include allocating products to orders, assigning orders to collection points, and managing robot transport. These strategies are essential for enhancing accuracy, efficiency, and adaptability in industrial sorting processes.

Selwitz [8] pointed out that warehouse automation strategies closely align with enhancing efficiency and accuracy in industrial sorting. Integrating robotics and AI into warehouse tasks mirrors efforts to address research gaps in sorting processes. This integration is essential to meet modern commerce demands by improving productivity and quality control. With this, a research team [9] developed an automated system to sort different types of fasteners like bolts, nuts, screws, and washers. This system uses a Pi camera and machine learning models to identify each fastener. The system includes a bowl feeder to release the fasteners, a conveyor belt to move them, and a tilting platform to sort them based on the computer's instructions. Images are processed and compared with a dataset using a machine learning model to predict the type of fasteners. Based on the prediction, a servo motor-controlled tilting mechanism directs the fasteners to their designated destinations.

Another study focused on using computer vision to recognize and sort nuts and bolts [10]. Proximity sensors and images acquired by a webcam are processed through MATLAB using the stationary wavelet transform. This method is more efficient than using an artificial neural network for image processing and detection. Tan et al. addressed the challenge of object detection in robotic vision systems, particularly focusing on bolts and nuts [11]. Their evaluation of YOLOv8 models under various conditions demonstrated promising results, with significant mean average precision achieved across different datasets.

The introduction of YOLOv8 represents a significant advancement in computer vision, specifically in object detection, with significant enhancements over YOLOv5. It features an anchor-free detection system, eliminating predefined bounding boxes for better accuracy and flexibility. Additionally, YOLOv8 features modifications to the convolutional blocks, enhancing the model's learning efficiency, feature extraction, and overall performance in object detection and classification. YOLOv8 also uses mosaic augmentation, combining four imageduring training, enhancinges its generalization ability. Interestingly, this augmentation is turned off before the last ten training epochs, which allowed the model to further optimize its learning on standard images for more effective real-world performance. [12].

Furthermore, manipulator capabilities and control systems are crucial in robotic sorting tasks [13]. Aside from sorting and object detection, grasping is an essential capability. Advanced control models allow for quick sorting of objects in space environments, showcasing the adaptability and efficiency of robotic arm intelligent control. Recent research by Mao & Chen presents innovative methodologies for intelligent sorting systems [14]. Their approach integrates image processing, recognition, and positioning to enhance productivity despite limited datasets. This integration reduces delays, enhances training efficiency, and addresses data labeling labor and time constraints, ultimately supporting automation, efficiency, and improved processes in manufacturing.

III. METHODOLOGY

A. *Model Development*

a. Data Preparation

The dataset is collected from various sources, ensuring diobject appearance, background, and condition diversityThis includes images with different lighting, angles, and positions. The study utilized datasets that were sourced from different Roboflow public datasets and proponent-generated datasets (captured using a webcam). The datasets included are photos of bolts and nuts, specifically hex bolts and nuts. The initial number of datasets sourced from Roboflow is 170, while 187 proponents manually captured the images - totaling 352 images. The number of images used for each class was: 244 for bolts, and 220 for nuts - this is due to some images containing 2 or more data.

The dataset was further processed through Roboflow. The images are annotated using the annotation tool in Roboflow to create bounding boxes around objects of interest. Each bounding box is labeled with the corresponding object class. To enhance the model's generalization capabilities, data preprocessing and augmentation techniques are applied, which can be seen in Table 1, to create a more diverse dataset.

TABLE 1. Preprocessing and Augmentation of Dataset through Roboflow

Preprocessing	Auto-Orient: Applied
	Resize: Stretch to 640x640
Augmentations	Flip: Horizontal, Vertical
	90° Rotate: Clockwise, Counter-Clockwise, Upside-Down
	Crop: 0% Minimum Zoom, 10% Maximum Zoom
	Rotation: Between -15° and +15°
	Saturation: Between -20% and +20%
	Brightness: Between -15% and +15%
	Exposure: Between -10% and +10%
	Noise: Up to 0.1% of pixels

After this process, the total dataset was 550 images. The annotated dataset is then split into train set, valid set, and test set with proportions of 72% (369 images), 13% (72 images), and 15% (82 images), respectively. This is to ensure the model can be evaluated on unseen data during training. These generated datasets are now accessible to the public through Roboflow to foster greater reproducibility and utilization.The generated dataset was exported to be utilized for model training.

b. Model Architecture Selection

YOLOv8 (You Only Look Once version 8) was chosen for its state-of-the-art performance in real-time object detection, which was built by Ultralytics [15]. YOLOv8 combines speed and accuracy, making it suitable for various applications. Moreover, Roboflow training utilizes the YOLO-NAS model version.

In this study, the proponents explored various hyperparameters such as batch size (refers to the number of training instances processed before updating the model's internal parameters.), number of epochs (represents one complete pass through the entire dataset), patience value (used in early stopping, to halt training when the model’s performance stops improving), and optimizer (minimizes the loss function) - where values are configured in each training. AdamW (Adaptive Moment Estimation) was the optimizer used in this study, an adaptive optimization algorithm which is the modification of Adam. This AdamW optimizer consists of optimized learning rate (defaults to 0.001), weight decay(0.01), beta values or the decay rates of the first and second-order moment of the optimizer (defaults to 0.9,0.999), epsilon value (defaults to 1e-8), and other parameters [16].

c. Training, Validation, and Testing

The training was done through two different environments, one Roboflow training using a predefined model, and another is using PyTorch utilizing YOLOv8. The Roboflow training is done with one iteration. The YOLOv8 object detection model is trained on the training dataset with three experiments having adjusted epoch and patience values for comparison of results performance. During training, the model's performance is continuously validated on the validation set. Metrics such as Mean Average Precision (mAP), precision, recall, and F1 score are tracked to monitor the model's accuracy and robustness. After training, the final model is evaluated on a separate test set to assess its real-world performance. This test set is unseen during training and validation to provide an unbiased evaluation.

d. Evaluation

i. Performance Matrix

Upon assembling and configuring the system, its performance must be thoroughly tested using a designated dataset. During this phase, metrics such as precision, recall, and mean average precision are generated by comparing the system's output to manual labels on some datasets. The system's strength and reliability are assessed by examining its ability to adapt to environmental changes and unexpected variations in the bolts and nuts characteristics.

The following metrics will be employed to assess the classification outcomes of the system.

$$\frac{TP+TN}{TP+TN+FP+FN}$$

Equation 1. Formula of Accuracy

$$\frac{TP}{TP+FN}$$

Equation 3. Formula of Recall

$$\frac{TP}{TP+FP}$$

Equation 2. Formula of Precision

$$2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Equation 4. Formula of F-1 Score

ii. Evaluation Matrix

The metrics mAP50 and mAP50-95 are commonly used to evaluate the performance of object detection models. These metrics are variants of mean Average Precision (mAP). Specifically, mAP50 measures the mean Average Precision at an Intersection over Union (IOU) threshold of 50% is determined as follows::

$$\text{mAP50} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_{50}(C_i) \qquad (5)$$

Where N is the number of classes, Ci is the i-th class, and AP50 (Ci) is the Average Precision for class Ci at an IOU threshold of 50%.

Intersection Over Union, (IOU), quantifies how much the ground truth and anticipated truth boxes overlap. For a correct detection, there must be at least 50% overlap when the IOU is 50%. The average precision (AP) measured at different IOU thresholds and averaged from 50% to 95% in 5% increment is represented by the mAP50-95 metric:

$$\text{mAP50-95} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{T}\sum_{t=50\%}^{95\%} \text{AP}_t(C_i) \qquad (6)$$

Where N is the number of classes, Ci is the i-th class, T is the number of IOU thresholds (from 50% to 95%, in 5% increments), and APt (Ci) is the Average Precision for class Ci at an IOU threshold of t.

B. Block Diagram

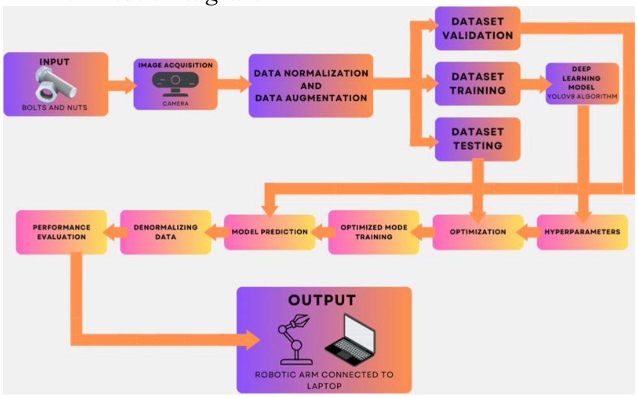

Fig. 1 Block Diagram of the System

Figure 1 shows the block diagram of the system, starting with the input of bolts and nuts. The first step of the process is image acquisition, which uses a camera to capture images of these input objects. These images are then subjected to data normalization and augmentation to prepare them for training. The prepared data is used to train a model with the YOLOv8 algorithm. After training, the dataset undergoes testing and validation to ensure accuracy. The model is then optimized for better performance, and its final performance is evaluated through various metrics. The system's output is classifying and sorting bolts and nuts into their designated

C. Data Analysis

The dataset utilized in this study was sourced and managed through Roboflow, which facilitated efficient annotation and augmentation processes. Roboflow's platform allowed for seamless integration of diverse image datasets, ensuring consistent and high-quality annotations. Various augmentation techniques, including rotation, scaling, and color adjustments, were applied to enhance the robustness of the model.

YOLOv8 was employed for training the object detection model. Known for its speed and accuracy, YOLOv8 processes images in real-time, making it ideal for applications requiring quick decision-making. The model was trained on a dataset comprising bolts and nuts. The training process involved optimizing the model's hyperparameters and leveraging transfer learning to enhance performance.

D. Project Design

This project showcases a design for a system that automatically recognizes and sorts bolts and nuts. This system uses computer vision and robotics to tackle a common task in machine vision - separating objects by their shape. A camera will capture images of objects placed in the working area. A computer program will then analyze these images to find and identify each object. A vital part of this system is using a pre-trained program, similar to YOLOv8, to recognize objects based on their type. Finally, a robotic arm with a claw/gripper will pick up and place the objects in designated plates based on their types (bolts or nuts). The robotic arm is secured to the platform's base while the designated sorted plates, which are classified as bolts and nuts, are located beside it. The webcam is elevated and placed in front of the working area for image acquisition. Also, the laptop was used for programming the system, as shown in Figures 2 and 3.

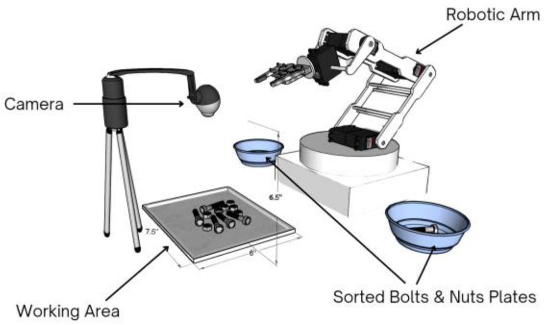

Fig. 2 CAD Design Set-up of the Project

Fig. 3 Actual System Setup

This design offers several benefits, such as faster operation, more accurate sorting, and reduced manual labor. The project will be built in stages: first, designing the system and setting up the hardware; next, developing the intelligent object detection model and control of the robotic arm; and finally, conducting thorough testing to ensure the accuracy and reliability of the system. For seamless operations, ensuring optimal functionality of both hardware and software components is vital. This requires thorough testing and careful calibration of servo motors and grippers, which are important for maintaining precision and accuracy. By carefully considering factors like lighting and calibration, the project aims to deliver a solid and effective solution for automatically recognizing and sorting objects.

E. System Flowchart

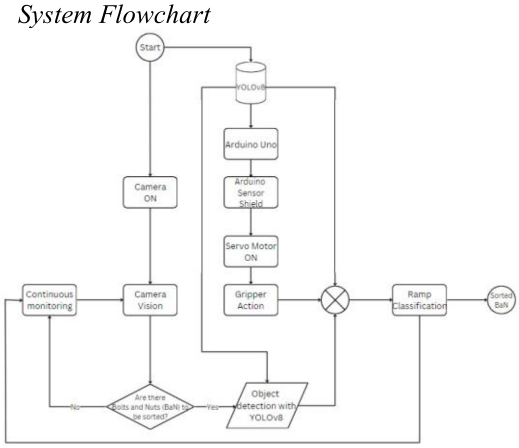

Fig. 4 System Flowchart of Robotic Bolts and Nuts Sorter

The robotic gripper's decision-making concepts and corresponding parameters are illustrated in the flowchart. The YOLOv8 model uses the Arduino Uno and Arduino Sensor Shield to sort out nuts and bolts. The robotic arm's movement is managed by the servo motors, which are powered by the sensor shield. After the camera captures the presence of Bolts and Nuts, it will then be sorted according to its ramp. The combination of computer action and gripper action allows the Bolts and Nuts to be picked up, and finally to be dropped on its respective ramp.

F. Project Development

a. Materials

The essential components for developing a six-degree-of-freedom (6DOF) robotic arm system include a metal robotic arm frame with Tower Pro Digital Robot Servo Motors (MG996R) for movement and an Arduino Uno R3 Board with a Sensor Shield V5 for control. Connectivity is ensured by jumper wire cables, and power is provided by an AC-DC 5V 5A adapter. The system employs a 1080p HD Webcam for visual processing and uses bolts and nuts as the objects that will be sorted in the system. Lastly, plywood and plastic containers are used as the platform and sorting bin.

b. Software

i. Arduino (C++)

The proponents used the C++ language to code the program on the Arduino UNO microcontroller board to control the servo motors for pick and place robotic arm purposes that will be integrated into the Python language by receiving serial input. The language will serve as the program for the actuator of the proposed system.

ii. Python

Python language was used to develop computer vision using a web camera to capture images for data acquisition through the OpenCV library. It was also used for training and performing image classification using the Yolov8 model. The analysis of accuracy and loss analysis of the model was also carried out using Python language.

c. Hardware

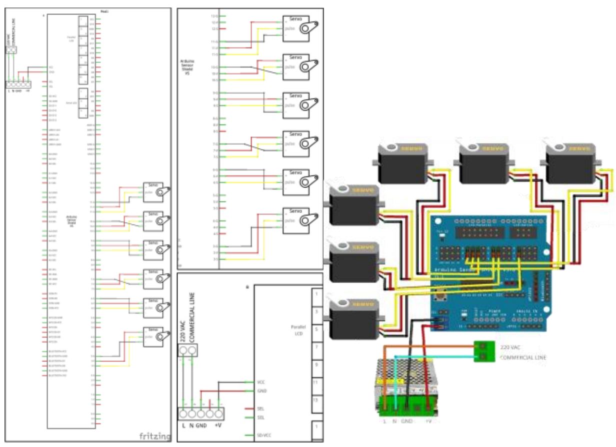

Fig. 5 Schematic Diagram and 3D Diagram of the Robotic Arm

The schematic diagram and 3D diagram connection of the robotic arm were constructed through Fritzing. It shows the wired connection of each component in the robotic arm. The robotic arm is powered by 220 VAC connected to an AC-DC 5V 5A power supply, making it compatible with other components. The Arduino Uno and Arduino Shield (placed atop the Arduino Uno) serve as the central hub of the electronics used in the project. The Arduino Shield's VCC and GND pins are connected to the power supply, and the servo motors are connected to their digital pins – 3, 6, 7, 9, 10, and 11, respectively.

IV. RESULTS AND DISCUSSION

a. Confusion Matrix

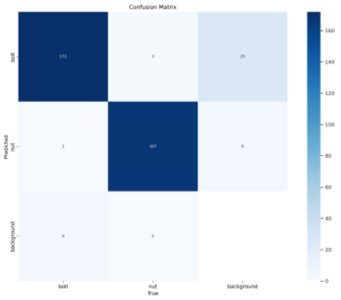

Fig. 6 Confusion Matrix with 25 Epochs Using YOLOv8

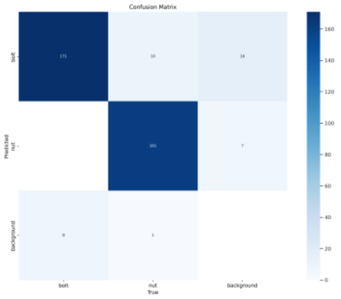

Fig. 7 Confusion Matrix with 100 Epochs Using YOLOv8

The YOLOv8 model with 25 epochs shows higher true positives for bolts and nuts and fewer false positives for bolts. In contrast, the 100 epochs model has fewer false negatives for bolts and performs similarly in false positives for nuts. The 25 epochs model also slightly outperforms in minimizing false positives for background predictions. Overall, the 25 epochs model demonstrates superior performance in most metrics, suggesting greater reliability for accurate classification. However, the 100 epochs model's strength in reducing false negatives for bolts may be crucial for certain applications.

b. Discussion of Findings

The deep learning models in this study were developed to assess and classify whether the object is classified as bolts antouprogramre trained using a dataset that distinguished between these two types. The table below summarizes performance metrics such as precision (P), recall (R), and mean average precision. The study extensively experimented with various training and validation setups, focusing on YOLOv8 models with different epoch and patience configurations, including 15 epochs with patience 3, 30 epochs with patience 5, and 50 epochs with patience 10.

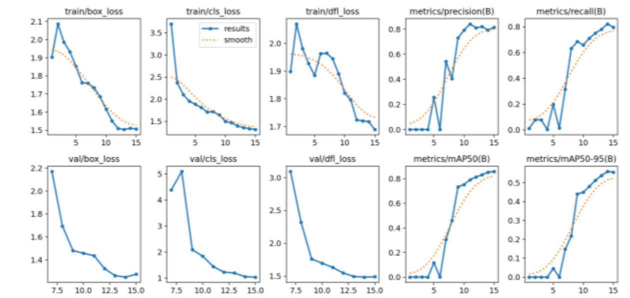

Fig. 8 Training and Validation Metrics for YOLOv8 Over 15Epochs with Patience 3

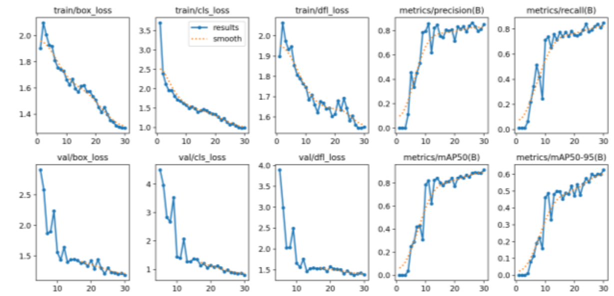

Fig. 9 Training and Validation Metrics for YOLOv8 Over 30 Epochs with Patience 5

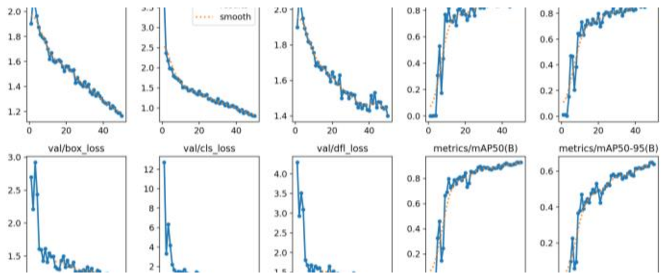

Fig. 10 Training Metrics for the Roboflow Model Over 50 Epochs with Patience 10

The training and validation metrics for the YOLOv8 model across different epochs and patience values reveal significant insights into its learning dynamics and generalization capabilities. Across Figures 8, 9, and 10, training and validation losses (box, class, and DFL loss) consistently decrease, indicating effective learning. At the same time, performance metrics such as precision, recall, mAP50, and mAP50-95 show an increasing trend, demonstrating improved predictive performance. In Figure 8, training losses decrease rapidly and plateau early, with consistent validation loss trends but a risk of premature stopping. Figure 9 shows gradual improvements and more stable validation losses, benefiting from extended training before early stopping. Figure 10 exhibits steady declines in losses and smooth performance metric trends, with significant improvements in mAP metrics, indicating better generalization, though at the cost of increased training time and resources.

CONCLUSION

This paper presents an extensive approach to an AI-integrated bolts and nuts robotic sorting system. The image processing, quality detection, and robotic arm and gripper action are successful. The yaw, pitch, and roll parameters of the system can be adjusted to achieve the accuracy of its movement. The project combines hardware and deep learning algorithms to create an advanced and reliable bolts and nuts robotic sorting system. These components enable automation for improved efficiency and productivity.

Shorter training with lower patience risks early stopping despite quick initial learning. Moderate training with balanced patience ensures stable improvements and consistent validation performance. Extended training with higher patience achieves the best generalization and performance metrics, though at the cost of more training time and computational resources. Balancing these factors is essential for developing effective deep-learning models for object classification.

Recommendation

The researchers have the following recommendations as they summarize and conclude the development, training, and validation of an AI-Integrated Bolts and Nuts Robotic Sorting System.

- It is recommended to evaluate the system's generalization and robustness by testing it on a more extensive and diverse dataset.
- The use of alternative grippers, such as electromagnets or magnetic grippers, is proposed to improve the robotic arm's grip on bolts and nuts during the sorting process.
- Further experimentation with models or algorithms with enhanced precision is suggested to test and identify the most accurate model.
- The system can be further refined and optimized for broader industrial applications and technologies.

Acknowledgment

The authors would like to thank the Department of Electronics Engineering at the Polytechnic University of the Philippines and Engr. Marife A. Rosales, for guiding them throughout this project.