SiliAI: AI-Driven Robotic System for Efficient Chili Pepper Sorting Using YOLOv8

Hannah Rea S. Abarquez[1], Angelo Miguel N. Agsolid[1], James A. Oliveros[1], Alexandra Ysabel F. Sebastian[1], Sean Lenard L. Transfiguracion[1], Jo-Ann V. Magsumbol[1, 2], and Marife A. Rosales[1,2]

[1]*Department of Electronics Engineering, Polytechnic University of the Philippines*
[2]*Research Institute for Strategic Foresight and Innovation,*
Polytechnic University of the Philippines, Sta Mesa Manila, Philippines
abarquezhannahrea@gmail.com, aagsolid51@gmail.com, alexysabelsebastian@gmail.com, jelalfernandez2@gmail.com, javmagsumbol@pup.edu.ph, marosales@pup.edu.ph

***Abstract*— Addressing the increasing demand for chili peppers, this study develops the SiliAI system, an advanced AI-driven robotic arm tailored for sorting peppers into red, green, and rotten categories. The system integrates YOLOv8, a state-of-the-art object detection algorithm, with a 6-degree-of-freedom (6DOF) robotic arm and a 2-finger angular gripper, which collectively enhance sorting efficiency and accuracy. The SiliAI system achieves a promising peak F1 score of 93% and perfect precision 100% across all classes at specific confidence thresholds, demonstrating high reliability in classification. Precision-Recall Curves reveal impressive precision rates for GREEN (99.5%), RED (97.0%), and ROTTEN (91.3%) categories, with a macro-average precision of 95.9% at an Intersection over Union (IoU) of 0.5. These metrics underscore the system's ability to accurately sort peppers based on ripeness. Training metrics indicate consistent performance, supported by a balanced dataset and effective tool integration, including Arduino IDE, Roboflow, and Python. The SiliAI system not only promotes operational efficiency but also aligns with Sustainable Development Goals (SDGs) related to food security. Future plans involve expanding the dataset and refining the gripper to further enhance field performance. This research highlights the potential of AI-driven robotic systems to significantly improve productivity and reduce labor costs in agricultural sorting tasks.**

Keywords–chili peppers, AI-driven sorting, robotic arm, YOLOv8, sustainable agriculture, food security, automation.

I. Introduction

Chili peppers, the fruits of Capsicum pepper plants, are renowned for their distinctive hot flavor and have gained immense popularity in both native regions and international markets. As global diets evolve towards healthier and more diverse eating habits, the consumption of chili peppers has increased due to their rich content of vitamins, minerals, and antioxidants. However, manual sorting of chili peppers is time-consuming and poses significant challenges for the industry [1].

To address these challenges, a technological solution involving a robotic arm with a gripper and an integrated artificial intelligence model is proposed. This system is designed to accurately categorize chili peppers into red, green, and rotten, ensuring a seamless separation process [2]. This approach not only enhances operational efficiency but also aligns with multiple United Nations Sustainable Development Goals (SDGs): SDG 2 (Zero Hunger), SDG 9 (Industry, Innovation, and Infrastructure), and SDG 3 (Good Health and Well-Being). By reducing manual sorting tasks, it contributes to creating a safer working environment for agricultural workers [3].

Agricultural automation has advanced significantly, particularly in sorting and processing systems for fruits and vegetables. However, challenges specific to sorting chili peppers, which vary in ripeness and require precise handling, have not been thoroughly addressed. Existing literature predominantly covers general fruit sorting or specific crops like tomatoes and apples [4], highlighting a gap in knowledge regarding automated sorting systems specifically designed for chili peppers.

In the Philippines, chili peppers are integral to the culinary tradition, valued for their flavor and health benefits in local dishes. Research highlights their economic significance to Filipino farmers and the difficulties they encounter in manual sorting and grading [5]. These challenges are compounded by the variability in ripeness and the chili's delicate nature, posing risks of substantial post-harvest losses if not managed properly.

Robot arms integrated into agricultural sorting machines represent a significant advancement in automating fruit and vegetable sorting. In sili sorting machines, robot arms accurately classify red, green, and rotten sili based on physical and quality attributes [6]. Equipped with advanced sensors and actuators, these arms enable precise handling and manipulation during sorting [7]. They employ computer vision algorithms to detect color, size, and defects in real-time, ensuring consistent sorting accuracy and optimizing yield while minimizing waste.

Furthermore, the importance of robot arm design in sili sorting machines is stated by [8], emphasizing adaptive gripping mechanisms and motion planning algorithms to handle delicate sili variants gently while maintaining high throughput and efficiency.

AI technologies have revolutionized these machines by enabling autonomous decision-making and adaptive learning. AI algorithms, including convolutional neural networks (CNNs) and machine learning models, classify sili based on intricate visual and structural features. Recent advancements highlighted by [9] show AI's capability in discerning subtle differences between ripe, unripe, and rotten sili.

Additionally, AI models achieve high accuracy in distinguishing sili varieties through iterative training on large datasets, surpassing traditional sorting methods in reliability and speed. Research highlighted by [10] demonstrates AI's role in improving sorting machine efficiency through predictive maintenance and anomaly detection. By analyzing sensor data from robot arms and sorting components, AI algorithms preemptively identify potential faults, reducing downtime and maintenance costs.

While existing studies on AI-driven robotic arms for fruit and vegetable sorting [11] provide insights, they often overlook the unique characteristics of chili peppers. These include their small size, variable ripeness colors (red, green), and sensitivity to handling. Automated

sorting systems typically rely on visual and tactile assessments to classify produce based on ripeness. However, chili peppers require precise algorithms capable of discerning subtle differences in color and texture that indicate ripeness stages [12].

Addressing this research gap is crucial for improving efficiency and minimizing waste in chili pepper production and processing. AI-driven sorting technologies can optimize agricultural operations, improve product quality assurance, and meet consumer demands more effectively [13]. This study will emphasize identifying key hardware components for automating sorting with a robotic arm gripper, selecting suitable software for detecting red, green, and rotten chilis, and implementing rigorous testing to enhance sorting accuracy and reliability.

The research focuses on sorting chili peppers into three main categories—green, red, and rotten—as typically found in supermarkets or retail stores. However, it is constrained by the gripper's design and functionality, limiting the study to sorting within these specified classifications.

By addressing these aspects, this research aims to contribute to technological advancements in the agricultural sector, increasing productivity for farmers and improving the quality of produce. It also addresses food security by promoting sustainable practices and enhancing efficiency in chili pepper cultivation.

II. METHODOLOGY

A. *Project Design*

The project focuses on a systematic approach to applied science for assessing different types of chilis using Roboflow. Roboflow simplifies building and deploying computer vision models by providing tools for dataset management, annotation, and model training, making it accessible for developers to integrate AI-driven vision capabilities [14]. The system uses Roboflow to recognize raw images of chilis, which are then converted into annotated datasets (Green, Red, and Rotten Chilis). These datasets are preprocessed to enhance quality and precision for training, testing, and validation of the AI algorithm. After training, the AI model is integrated into a robotic arm to sort the chilis into designated bins. The results will demonstrate the effectiveness of the automated chili sorting system.

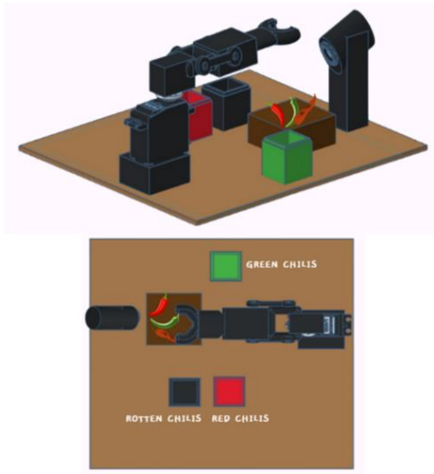

Fig. 1. Front View and Top View of SiliAI Robotic Arm

Figure 1 shows the overall architecture of SiliAI which includes a 6DOF robotic arm with 2 finger angular gripper. The system also includes a 720p camera on the front for image recognition which will be fed to the AI model. Lastly, the platform will be the place where the chilis will be sorted and the sorting bins are present to hold and store the sorted chili base on its characteristics.

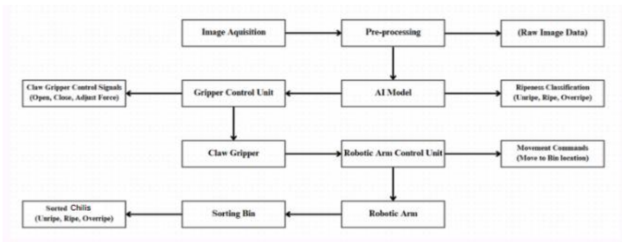

Fig. 2. Block Diagram of the Integrated Robotic Model

Figure 2 illustrates an automated chili sorting system with two main components: the camera system and the robotic arm control system. Using the YOLOv8 object detection approach, the camera captures images of chilis on a conveyor belt and sends them to the YOLOv8 model, which distinguishes between unripe (green), ripe (red), and overripe (rotten) chilis. The results and control signals are processed in a Python program connected to an Arduino Uno, which acts as the central processing unit. The Arduino communicates with the robotic arm and gripper for precise control. Based on the detected ripeness, the program directs the robotic arm to sort the chilis into the appropriate bins: right for unripe, left for ripe, and upper left for overripe. This system combines image processing, decision-making, and robotic control to automate chili sorting by ripeness.

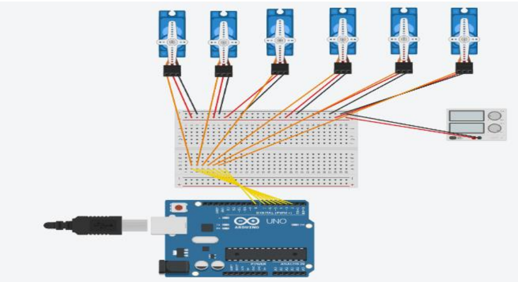

Fig. 3. Servo Motor and Wiring Connections

Figure 3 shows the servo connections of the SiliAi, using Tinkercad to illustrate the Arduino wiring. The setup starts with connecting the servo motors to the positive and negative ports of the breadboard, while the signal inputs are inserted into the Arduino. A 5V 5A power supply energizes the six servo motors. The Arduino is connected to the servo motors' signal inputs via digital pins that can use pulse width modulation (PWM). These six servo motors act as actuators, allowing the robot to move based on electrical signals from the Arduino.

B. Project Development

B.1 Hardware Development

The hardware setup includes an Arduino Uno, Ansen 1080p Webcam, Sensor Shield for Arduino, MG996R Metal Gear Servo Motor, 6DOF metallic robotic arm, and a 2-finger angular gripper. The construction materials include a wooden platform and screws, and three plastic bins are used for storing green, red, and rotten chilis. The robotic arm and gripper feature a pitch mechanism that allows the arm to move up and down from 180 to 90 degrees.

Due to the moving infrastructure's multiple degrees of freedom (DOF), it can apply force in various dimensions. The system also includes a yaw mechanism for side-to-side movement (180 degrees). The camera is positioned 6.5 cm from the platform's edge and at a height of 28 cm, ensuring optimal vision and control. This setup leverages the robotic arm's versatility and the camera's strategic placement for precise sorting and handling of chilis.

B.2 Software Development

- *Arduino IDE*

The researchers utilized Arduino IDE (Integrated Development Environment) to code and calibrate the servo motors by writing and uploading sketches (programs) that specify the angles and movements required for the servo motors' operation. Within the Arduino IDE, it enables the researchers to write code in a simplified version of C/C++, utilizing built-in libraries such as the Servo library, which provides functions to easily control the servo motors. By defining the servo motor objects and specifying the desired angles (typically between 0 to 180 degrees), the code instructs the servo motors to move to the set positions. During the calibration process, researchers can fine-tune these angles to ensure precise movements and optimal functionality. The code can be uploaded to the Arduino board, allowing real-time adjustments and testing of the servo motors' responses, ensuring accurate and reliable operation of the robotic arm and gripper.

- *Roboflow & YOLOV8*

Roboflow and YOLOv8 were used to develop deep learning-based computer vision applications, enabling advanced tasks such as detection, segmentation, posture estimation, tracking, and ripeness classification. These tools streamlined the entire process from data labeling to model training, significantly enhancing the precision and efficiency of developing computer vision solutions. By leveraging computer vision and artificial intelligence, these applications can automatically interpret and analyze visual data, allowing systems to identify objects in images, distinguish between different classes of items, estimate the pose of subjects, and track movement over time.

- *Python*

The researchers used the Arduino Uno microcontroller, interfacing it with an AI system using Python. Python scripts developed in VS Code send commands to the Arduino Uno over a serial connection, controlling the robotic arm and gripper based on AI algorithms or user input. This setup allows the AI system to evaluate data, make decisions, and instruct the Arduino Uno to perform specific tasks. The integration combines the Arduino Uno's real-time responsiveness with Python's computational power for sophisticated robotic control and automation. Feedback loops with sensors on the robotic arm provide real-time data to the Python scripts, enabling dynamic adjustments and precise operations.

III. Results and Discussions

The figure above illustrates the YOLOv8 model's proficiency in object and color detection. It categorizes objects by "rottenness" (e.g., Rotten 0.7, Rotten 0.8) to assess their deterioration. Additionally, it accurately classifies chili peppers as "red" or "green" with confidence scores (e.g., Red 0.9, Green 0.9),

demonstrating effective color attribute detection. Overall, these results highlight the model's robust performance in object detection and attribute classification, underscoring its reliability and precision across diverse datasets.

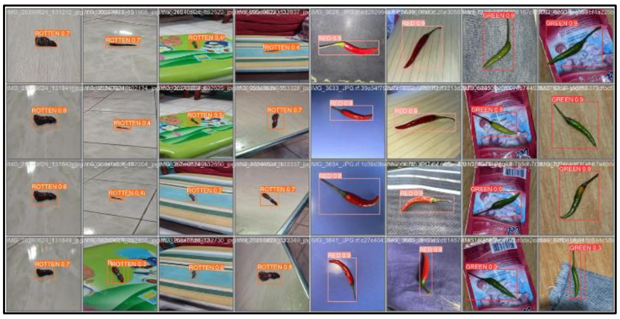

Fig. 4. Data-Captured Images

Fig. 5 provides a non-normalized confusion matrix detailing actual prediction counts: 21 correct predictions for the GREEN class, 21 correct and 3 misclassified as background and 2 as ROTTEN for the RED class, and 17 correct with 8 misclassified as background for the ROTTEN class. Background class misclassifications include 1 each as RED and ROTTEN. Figure 6 complements this with a normalized confusion matrix: perfect prediction (precision 1.00) for GREEN, a precision of 0.95 for RED with misclassifications as background (0.23) and ROTTEN (0.15), a precision of 0.94 for ROTTEN with frequent confusion with background (0.62), and background misclassifications of 0.05 and 0.06 as RED and ROTTEN, respectively.

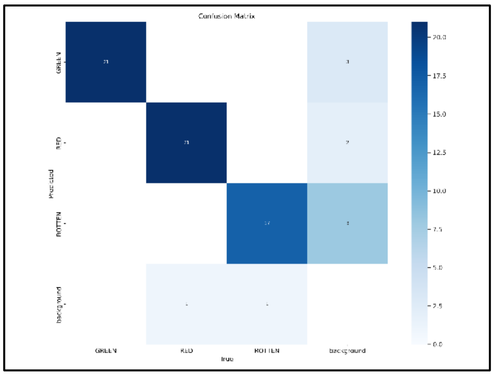

Fig. 5. Non-Normalized Confusion Matrix

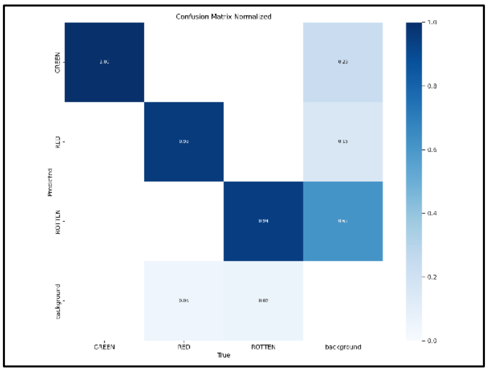

Fig. 6. Normalized Confusion Matrix

Fig. 7 and Fig. 8 collectively present a comprehensive analysis of the classification model's performance and training metrics. Fig. 7 highlights the model's effectiveness across GREEN, RED, and ROTTEN classes, showcasing peak F1 scores with GREEN achieving 0.93, perfect precision (1.00) observed at a confidence threshold of 0.837, and high precision values for all classes (GREEN: 0.995, RED: 0.970, ROTTEN: 0.913), resulting in a macro-average precision of 0.959 at an IoU threshold of 0.5. The Recall-Confidence Curve reinforces strong performance with a mean average precision (mAP) of 0.959, emphasizing the model's accuracy and recall capabilities. In Fig. 8, The blue lines represent the outcomes of the training metrics, while the orange dots indicate the smoothness of the validation metrics. The visualization of the train-valid metrics graph demonstrates a positive trend where the orange dots closely track the blue lines, indicating effective performance of the training data on the provided examples.

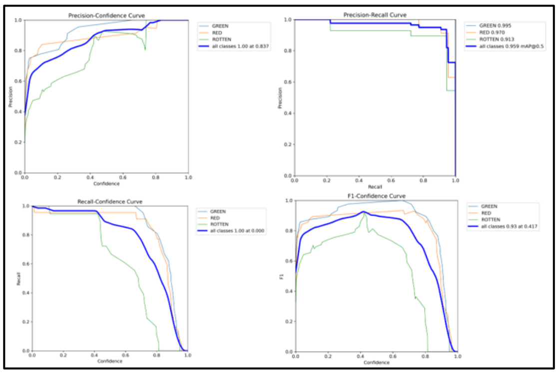

Figure 7. Split Ratio Performance Metrics Curve

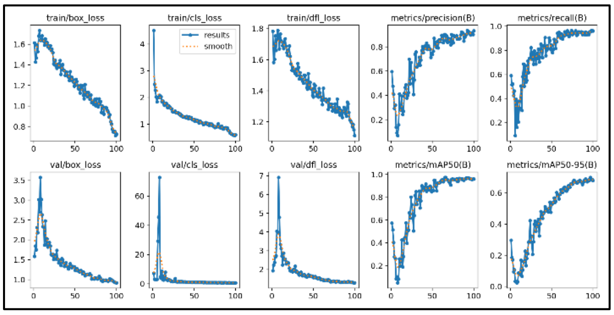

Fig. 8. Split Ratio Metrics Graph

Fig. 9 shows the visualizations of a summary of the dataset, which includes three categories: "GREEN," "RED," and "ROTTEN." Each category shows a similar count of instances, indicating a balanced dataset. This balanced distribution, depicted in the bar chart on the top-left, ensures that the model can learn equally from each category, reducing bias.

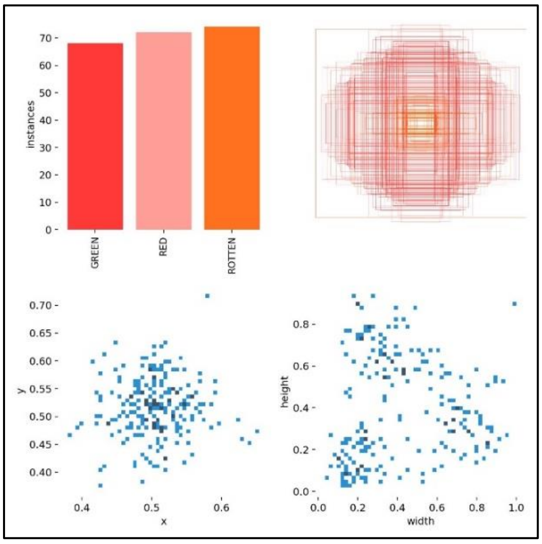

Fig. 9. Distribution of Chilis in the dataset

The top-right image is a heatmap or density plot, illustrating the distribution of data points across some features, with color intensity indicating density. This helps to understand the spread and concentration of data points within the feature space.

Scatter plots in the bottom-left and bottom-right images show the relationship between pairs of features, such as 'x' vs 'y' and 'width' vs 'height.' These plots indicate that data points are spread across the feature space with some clustering visible in certain areas, suggesting patterns or correlations within the dataset.

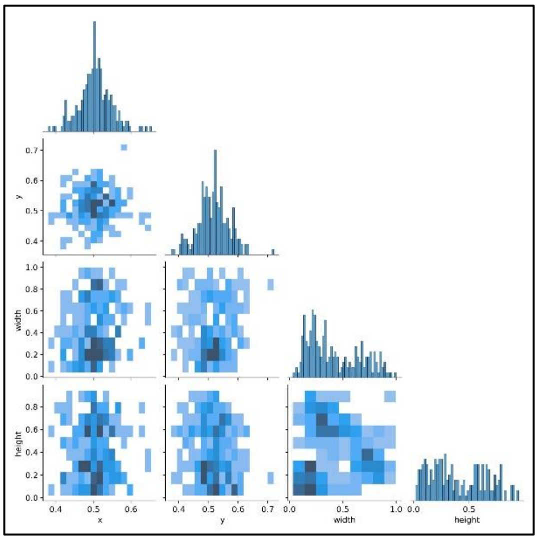

Fig.10. Label Correlogram

Fig. 10 presents a scatterplot matrix illustrating the dataset comprehensively. Diagonal histograms show feature distributions, while off-diagonal scatterplots reveal pairwise relationships with darker regions indicating denser data points. These visualizations are pivotal for understanding dataset structure and guiding decisions in preprocessing and model training. Analyzing these relationships helps identify crucial features and interactions, thereby enhancing model development effectiveness.

```
Ultralytics YOLOv8.0.196 🚀 Python-3.10.12 torch-2.3.0+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 1
                   all         61         61      0.915      0.956      0.959      0.699
                 GREEN         61         21      0.969          1      0.995      0.832
                   RED         61         22      0.884      0.955       0.97       0.81
                ROTTEN         61         18      0.891      0.912      0.913      0.456
Speed: 0.3ms preprocess, 7.0ms inference, 0.0ms loss, 2.7ms postprocess per image
```

Fig. 11. Performance Metrics of SiliAI

Fig. 11 outlines the performance metrics of a YOLOv8 object detection model implemented using Python 3.10.12 and PyTorch 2.3.0 with CUDA support, running on a Tesla T4 GPU with 15,102 MiB of memory. The model architecture comprises 168 layers, totaling 111,267,745 parameters, and operates without gradients during inference. It carries a computational load of 28.4 GFLOPs. The table summarizes detection performance for three specific classes: GREEN, RED, and ROTTEN, along with aggregated performance across all classes.

The YOLOv8 model processed 61 images, detecting 21 GREEN, 22 RED, and 18 ROTTEN instances of chilies. It achieved high precision and recall for GREEN chilies (0.969 precision, perfect recall), with a mean Average Precision (mAP) of 0.995 at IoU 0.5. RED chilies were identified with 0.884 precision and 0.955 recall, resulting in an mAP of 0.97 at IoU 0.5. ROTTEN chilies showed 0.891 precision and 0.912 recall, with an mAP of 0.913 at IoU 0.5. Overall, the model maintained a precision of 0.915 and recall of 0.956, achieving an mAP of 0.95 at IoU 0.5. Speed metrics included 0.3 ms preprocessing, 7.0 ms inference, and 2.7 ms postprocessing per image, processing at 1.65 images per second. Initially, an automatic optimizer selection process was conducted, overriding the manually set learning rate (lr=0.01) and momentum (momentum=0.937). The model automatically selected the AdamW optimizer, a variant of Adam that incorporates weight decay to enhance generalization. The specific parameters for AdamW included a learning rate (lr) of 0.001429 and a momentum of 0.9.

The optimizer used includes parameter groups with varied weight decays: 57 parameters with no decay, 64 parameters with a decay rate of 0.0005, and 63 parameters where biases have no decay applied. The image also notes that training and validation images are consistently sized at 800 pixels, ensuring uniformity throughout these processes to potentially enhance model performance and efficiency. AdamW is employed with these settings to benefit from adaptive learning rates and effective weight decay management, aiding in mitigating overfitting and improving generalization to unseen data.

IV. Conclusion and Recommendations

This research developed and implemented an AI-driven robotic arm system, SiliAI, for sorting chili peppers into categories of red, green, and rotten. By integrating YOLOv8 for object detection with a 6DOF robotic arm and 2-finger angular gripper, the system significantly enhances sorting efficiency and accuracy. The performance metrics, including high precision, recall, and mean average precision (mAP), validate the system's effectiveness in accurately classifying chili peppers based on ripeness. The image processing results demonstrated the model's effectiveness in detecting objects based on color attributes and "rottenness." While it excelled in classifying green and red classes, it faced challenges with rotten and background differentiation, as shown in the confusion matrix. Precision-recall and F1-confidence curves underscored its accuracy in distinguishing between ripe, unripe, and rotten chilis, particularly in predicting the green class. The proposed system automates the labor-intensive sorting process, reduces post-harvest losses, improves the quality of produce, and enhances operational efficiency. By addressing the specific challenges associated with sorting chili peppers, this research contributes to agricultural automation and aligns with the United Nations Sustainable Development Goals (SDGs) by promoting sustainable agricultural practices and ensuring food security.

References

[1] A. A. BSc PhD, "Chili Peppers 101: Nutrition Facts and health Effects," Healthline, May 13, 2019. https://www.healthline.com/nutrition/foods/chili-peppers

[2] C. Maraveas, "Incorporating artificial intelligence technology in smart greenhouses: Current state of the art," Applied Sciences, vol. 13, no. 1, p. 14, Dec. 2022, doi: 10.3390/app13010014.

[3] B. O. Olorunfemi, N. I. Nwulu, O. A. Adebo, and K. A. Kavadias, "Advancements in Machine Visions for fruit Sorting and Grading: A bibliometric analysis, systematic review, and future Research Directions," Journal of Agriculture and Food Research, vol. 16, p. 101154, Jun. 2024, doi: 10.1016/j.jafr.2024.101154.

[4] M. Daheriya, D. K. Verma, E. K. Kuruba, K. C. Mahajan, and N. U. Joshi, "Advances in Multi-Fruit and Vegetable Grading: A Comprehensive review," International Journal of Environment and Climate Change, vol. 13, no. 9, pp. 1239–1245, Jul. 2023, doi: 10.9734/ijecc/2023/v13i92351.

[5] "Siling Labuyo - Arca del Gusto - Slow Food Foundation," Slow Food Foundation, Dec. 09, 2019. https://www.fondazioneslowfood.com/en/ark-of-taste-slow-food/siling-labuyo-2/

[6] Z. Zhou et al., "Advancement in artificial intelligence for on-farm fruit sorting and transportation," Frontiers in Plant Science, vol. 14, Apr. 2023, doi: 10.3389/fpls.2023.1082860.

[7] A. Dzedzickis, J. Subačiūtė-Žemaitienė, E. Šutinys, U. Samukaitė-Bubnienė, and V. Bučinskas, "Advanced Applications of Industrial Robotics: New trends and possibilities," Applied Sciences, vol. 12, no. 1, p. 135, Dec. 2021, doi: 10.3390/app12010135.

[8] Riliwan Adekola Adebayo, Igberaese Clinton Festus-Ikhuoria, Oladiran Kayode Olajiga, and Constance Obiuto Nwankwo, "Robotics in Manufacturing: A Review of Advances in Automation and Workforce Implications," researchgate.net, Mar. 2024. https://www.researchgate.net/publication/379439076_Robotics_in_Manufacturing_A_Review_of_Advances_in_Automation_and_Workforce_Implications

[9] B. Sundararaman, S. Jagdev, and N. Khatri, "Transformative Role of Artificial Intelligence in Advancing Sustainable Tomato (Solanum lycopersicum) Disease Management for Global Food Security: A Comprehensive Review," Sustainability, vol. 15, no. 15, p. 11681, Jul. 2023, doi: 10.3390/su151511681.

[10] R. Roy and A. Srivastava, "ROLE OF ARTIFICIAL INTELLIGENCE (AI) IN ENHANCING OPERATIONAL EFFICIENCY IN MANUFACTURING MEDICAL DEVICES," ˜ theœ Journal of Multidisciplinary Research, pp. 35–40, Apr. 2024, doi: 10.37022/tjmdr.v4i1.580.

[11] M. U. Masood and M. Haghshenas-Jaryani, "A study on the feasibility of robotic harvesting for chile pepper," Robotics, vol. 10, no. 3, p. 94, Jul. 2021, doi: 10.3390/robotics10030094.

[12] M. Z. Abdullah, A. S. Fathinul-Syahir, and B. M. N. Mohd-Azemi, "Automated inspection system for colour and shape grading of starfruit (Averrhoa carambola L.) using machine vision sensor," Transactions of the Institute of Measurement and Control, vol. 27, no. 2, pp. 65–87, doi: 10.1191/0142331205tm138oa.

[13] M. Javaid, A. Haleem, I. H. Khan, and R. Suman, "Understanding the potential applications of Artificial Intelligence in Agriculture Sector," Advanced Agrochem, vol. 2, no. 1, pp. 15–30, Mar. 2023, doi: 10.1016/j.aac.2022.10.001.

[14] J. Bhattacharyya, "Step by step guide to object detection using Roboflow," Analytics India Magazine, Dec. 02, 2020. https://analyticsindiamag.com/step-by-step-guide-to-object-detection-using-roboflow/