Revolutionizing Agriculture: IoT-Enhanced YOLO Object Detection for Automated Disease Identification

Khondokar Riaz Mahmud[1], Md Mahabubar Rahman[2], Md Hridwanur Rahman[3], Abu Ekhtiar Tawhid[4], and A.K.M. Muzahidul Islam[5]

United International Unversity, Dhaka

Abstract. In modern agriculture, the integration of cutting-edge technologies has become paramount in addressing the challenges posed by growing food demand and limited resources. This paper presents an innovative approach that combines Internet of Things (IoT) devices with the YOLO (You Only Look Once) object detection algorithm to revolutionize disease detection in agriculture. By harnessing IoT devices for real-time field monitoring and leveraging the robust capabilities of YOLO, our model automates disease identification, alleviating the manual effort required by farmers. This integrated solution offers a streamlined and accurate method to detect diseases promptly, ensuring timely interventions to minimize crop losses. The paper details the methodology behind this integration, including IoT device deployment, data preprocessing using Roboflow, and YOLO model training. The results demonstrate the efficacy of the approach, with quantitative metrics showcasing the model's accuracy and efficiency compared to conventional manual methods. This novel amalgamation of IoT and YOLO holds the potential to reshape the landscape of modern agriculture, fostering sustainable practices and securing global food production.

Keywords: Agriculture · YOLO Object Detection · Disease Identification.

1 Introduction

Agriculture, as the cornerstone of human civilization, has significantly evolved with technological advancements. To address the pressing challenges of a growing global population and limited arable land, innovative solutions are crucial to optimize crop production and ensure food security. This paper introduces a groundbreaking paradigm in agriculture, combining Internet of Things (IoT) devices with the precision of the YOLO (You Only Look Once) object detection algorithm for automated disease detection. Traditional farming practices often rely on manual labor and visual inspection for identifying crop diseases, leading to time-consuming and error-prone processes. Moreover, delayed disease identification can result in significant crop losses, threatening livelihoods and food supply chains. In response, the integration of IoT and advanced machine learning techniques presents a transformative solution. IoT devices, such as sensors and cameras, have become pervasive in agriculture, enabling real-time data collection of crucial variables like soil moisture, temperature, and weather conditions. Leveraging this data with the YOLO algorithm's object detection capabilities allows for rapid and accurate disease identification. The paper outlines a comprehensive methodology for integrating IoT devices and YOLO for automated disease detection. This includes IoT device deployment for field monitoring, data preprocessing using Roboflow, and YOLO model training with annotated datasets. The results demonstrate the effectiveness of this approach in reshaping disease management practices in agriculture.

Subsequent sections delve deeper into the technical intricacies of the proposed approach, detailing the implementation methodology, YOLO model performance results, and a comprehensive discussion on the implications, advantages, and potential challenges. This research empowers farmers with a tool that reduces manual effort and enhances disease detection accuracy, contributing to the sustainability and resilience of global food systems.

2 Literature Review

It has taken a lot of effort to create smart farming solutions using IoT technology in the agricultural industry. [9]. But the primary focus of this study revolves around computer vision with deep learning models and combining that with iot for more precision agriculture. Use of computer vision techniques and machine learning usage in agriculture now has become more popular.[1] The Convolutional Neural Network (CNN) has emerged as the leading classifier for image recognition, demonstrating remarkable capabilities in image processing and classification [4]. Early applications of deep learning in plant image recognition utilized CNN-based models to classify various plant species and crop diseases [3]. For instance, Mohanty et al. [7] achieved a 99.35% accuracy in recognizing 26 crop diseases and 14 crop species. Ma et al. [6] identified four cucumber diseases with 93.4% accuracy using a deep CNN, while Kawasaki et al. [5] developed a CNN-based system with 94.9% accuracy for identifying cucumber leaf disease. The potential of computer vision techniques for plant disease detection was further explored by creating a unique dataset called PlantDoc [8]. The dataset, containing 3,451 data points from 12 plant species and 14 disease categories, significantly improved accuracy by up to 29% in classifying plant diseases using three trained models. Another novel approach proposed by [10] combined an improved version of the YOLOv3 algorithm with spatial pyramid pooling to detect small agricultural pests. This method addressed issues of low recognition accuracy caused by variable posture and scale of crop pests, resulting in an average identification accuracy of 88.07% in real-world conditions.

Looking ahead, the adoption of advanced detection models, such as object segmentation networks, is expected to play a vital role in the identification of plant diseases and infestations, further enhancing the field of computer vision-based plant pathology[2].

3 Dataset

The dataset consists of 2451 images, each with 3150 annotations, covering six classes of rice diseases: Brown Spot, Leaf Blight, Leaf Scald, Leaf Blast, Narrow Brown Spot, and Healthy. This substantial dataset is crucial for training accurate and robust machine learning models for rice disease detection. The annotations, including bounding boxes and class labels, provide ground truth information for supervised learning. Each disease class is described, and potential issues like class imbalance are mentioned. Techniques such as data augmentation and preprocessing are recommended to improve the model's performance. The dataset is split into training, validation, and testing subsets for model evaluation. Object detection models like YOLO and Roboflow 3 are suggested as effective choices. Evaluation metrics like mean average precision (mAP) and intersection over union (IoU) are essential for assessing model performance. Ethical considerations, including obtaining permissions for dataset usage and respecting privacy and intellectual property rights, should be addressed when using real-world rice field images. Overall, this dataset provides a valuable resource for advancing rice disease detection in agriculture.

4 YOLOv8: An Overview

YOLOv8 is a cutting-edge advancement in object detection algorithms, particularly renowned for its object detection, image classification, and instance segmentation capabilities. Developed by Ultralytics, the same team responsible for the influential YOLOv5 model, YOLOv8 introduces a host of architectural enhancements and developer-friendly features that surpass its predecessors. Being in an active development phase, YOLOv8 constantly incorporates new features and improvements based on community feedback. Ultralytics commits to providing long-term support for their models, ensuring a continuous evolution towards excellence.

4.1 The Evolution of YOLO into YOLOv8

The YOLO series has garnered widespread recognition in the computer vision realm. Known for its potent accuracy and compact model size, YOLO models are accessible for single-GPU training, making them suitable for deployment across a spectrum of applications, from edge devices to cloud platforms. YOLO's journey began in 2015 with Joseph Redmond's first version. Over time, YOLOv8's author, Glenn Jocher at Ultralytics, transitioned the YOLOv3 repository to PyTorch, a deep learning framework from Facebook. This transition laid the groundwork for YOLOv5's emergence, which quickly gained prominence due to its Pythonic structure, enabling flexibility, and easy sharing of improvements across the community. Parallelly, YOLO spawned various models like Scaled-YOLOv4, YOLOR, YOLOv7, YOLOX, and YOLOv6, each contributing novel techniques to enhance accuracy and efficiency. Ultralytics' diligent research led to YOLOv8's inception, which was officially launched on January 10th, 2023, as the latest pinnacle of YOLO models.

4.2 Advantages of YOLOv8

Here are key reasons to consider incorporating YOLOv8 into your computer vision projects: High Accuracy: YOLOv8 boasts impressive accuracy, as demonstrated by its results on both the COCO benchmark and the Roboflow 100 benchmark. It achieves top-tier mean average precision (mAP) values while maintaining competitive inference speeds. Developer Convenience: YOLOv8 offers a suite of developer-centric features, including an intuitive command-line interface (CLI) and a well-structured Python package. These enhancements streamline the training process and coding experience.

Strong Community Support: The extensive community around YOLO and the growing support for YOLOv8 mean you'll have ample resources and assistance available when working with the model. Key Architectural Updates in YOLOv8 While there isn't a published paper on YOLOv8 yet, insights gathered from the repository shed light on the model's key updates:

Anchor-Free Detection: YOLOv8 adopts an anchor-free approach, directly predicting object centers instead of offsets from predefined anchor boxes. This simplifies predictions and improves post-processing steps like Non-Maximum Suppression.

New Convolutions: YOLOv8 introduces changes in convolutional layers, adopting 3x3 convolutions and revising the main building block. These architectural adjustments contribute to improved performance. Mosaic Augmentation: YOLOv8 refines the training routine, incorporating mosaic augmentation. This technique stitches together multiple images, exposing the model to diverse scenarios and enhancing its ability to detect objects effectively.

4.3 Accuracy of YOLOv8

On the COCO benchmark, YOLOv8 achieves remarkable accuracy, particularly the medium-sized model (YOLOv8m) with a 50.2% mAP. Furthermore, YOLOv8 surpasses its predecessors on the Roboflow 100 benchmark, demonstrating superior performance across various task-specific domains.

In conclusion, YOLOv8 represents a significant leap forward in object detection capabilities. With its robust accuracy, developer-friendly features, and strong community support, YOLOv8 emerges as a compelling choice for advancing computer vision projects.

5 Methodology

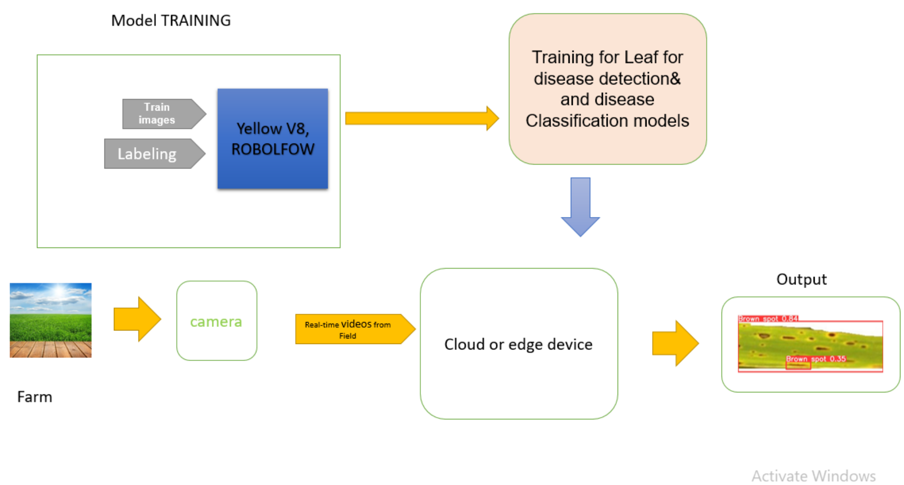

Fig. 1. General overview of the System

Step 1: Dataset Collection Gather a dataset containing 2451 images of rice plants exhibiting various conditions, including the 6 classes of rice diseases (Brown spot, Leaf Blight, Leaf Scald, Leaf blast, Narrow brown spot, and healthy).

Step 2: Pre-processing 1 (Initial Dataset Pre-processing) Perform initial data preprocessing on the raw images. This may involve resizing the images to a consistent resolution, normalizing pixel values, and possibly removing any irrelevant or low-quality images.

Step 3: Dataset Labeling Manually label the dataset by annotating each image with bounding boxes and class labels corresponding to the location and type of rice disease present.

Step 4: Pre-processing 2 (Labeling Pre-processing) After labeling the dataset, consider performing additional preprocessing to ensure the accuracy and quality of annotations. This might include reviewing and correcting annotations, as well as removing any duplicated or inconsistent annotations.

Step 5: Image Upload to Roboflow and YOLOv8 Format Preparation Upload your labeled dataset to the Roboflow platform. Convert the annotations into the YOLOv8 format, which involves providing normalized coordinates of the bounding boxes along with class indices.

Step 6: Pre-processing 3 (Roboflow Pre-processing) Utilize Roboflow's tools to apply further preprocessing if necessary. This could involve augmenting the dataset with transformations like rotation, flipping, and adjusting brightness/contrast to increase the diversity of the training data.

Step 7: Start Coding Using Google Colab Set up a Google Colab environment. Import necessary libraries and dependencies required for working with YOLOv8 and training deep learning models.

Step 8: Pre-processing 4 (Colab Environment Setup) In Colab, configure your environment by installing required packages, cloning the YOLOv8 repository (if not already done), and organizing your project structure.

Step 9: Image Data Code Writing in Roboflow Write code in Colab to retrieve image data directly from Roboflow's cloud storage or API. This ensures you're working with the latest labeled data.

Step 10: Pre-processing 5 (Data Pre-processing in Colab) Further preprocess the data within Colab as needed. This could involve data shuffling, splitting into training and validation sets, and performing any final data transformations.

Step 11: Training Using YOLOv8 Implement the training process using YOLOv8 architecture in Colab. This involves setting up the model architecture, defining loss functions, optimizing hyperparameters, and initiating the training loop.

Step 12: Testing the Trained Model Evaluate the trained YOLOv8 model's performance on a separate testing dataset. Calculate metrics such as mean average precision (mAP) to assess the model's accuracy in detecting rice diseases.

Step 13: Output - Infected Parts Detected Use the trained YOLOv8 model to analyze images and identify the locations of rice diseases. The model's output will include predicted bounding boxes around the infected parts, along with the corresponding disease class labels.

By following these steps, you'll be able to preprocess, label, train, and evaluate a YOLOv8 model for detecting rice diseases using the Roboflow platform and Google Colab. The outcome will help identify and manage diseased parts of rice crops more effectively, contributing to better agricultural practices.

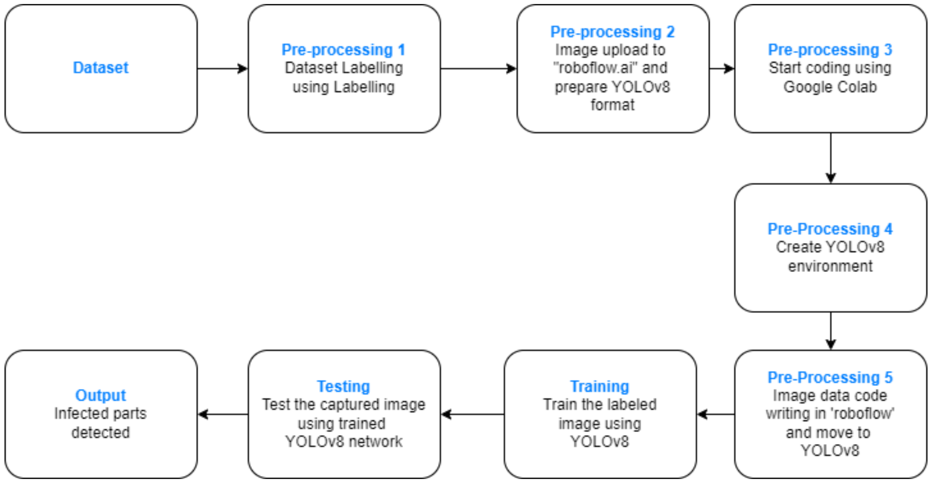

Fig. 2. Block diagram of training and testing the proposed YOLOv8 model

6 Result

We trained the Robofolow 3 object detection model and yolov8 model and compared their results. One of the key indicators of the YOLOv8 and Robofolow 3 object detection model's performance is its accuracy. This can be measured using metrics such as Mean Average Precision (mAP)). These metrics provide insights into how well the model can detect and localize diseases in rice plants. mAP: The mAP score reflects the average precision of the model across different disease classes. A higher mAP indicates better overall performance. It's crucial to assess both the mAP for individual classes and the mean mAP across all classes to understand where the model excels and where it may need improvement. Robofolow 3 object detection had map of 36.6% and the yolov8 had 31.1%.

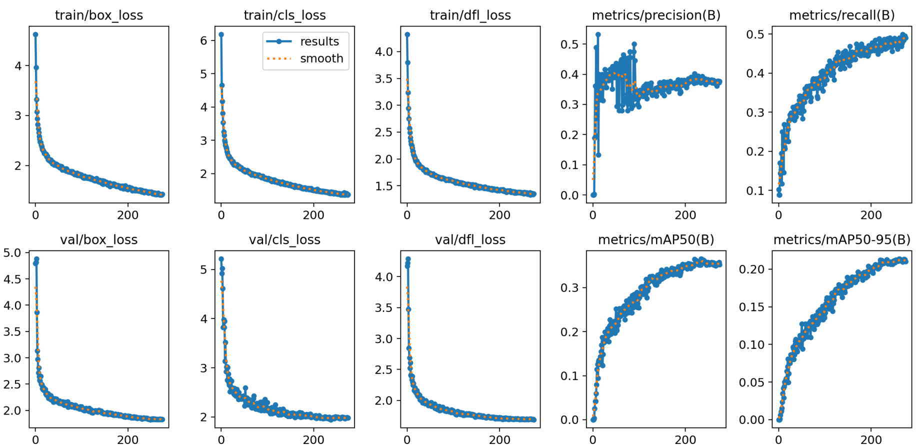

Fig. 3. Results of Roboflow model

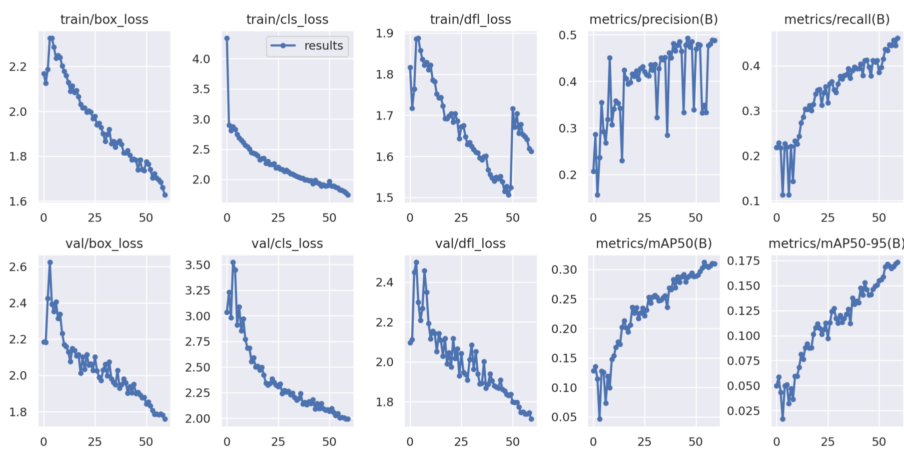

Fig. 4. Results of Yolov8 model

A high score of 36.6% was attained by the RoboFlow 3 object detection model, surpassing YOLOv8's mark of 31.1%. While the RoboFlow model might appear to be the apparent winner after 300 epochs of training, it underwent substantial training. But when we look at YOLOv8 Small, which consumed less energy and trained for only 60 epochs, it produced very identical outcomes. This demonstrates YOLOv8 Small's remarkable capabilities despite its shortened training period and simplified design including less parameters.

7 Conclusions and future perspective

Agriculture's use of YOLOv8 and Roboflow 3 introduces a revolutionary method of object detection for better crop disease detection. This study highlights the value of ongoing improvement and growth in addition to demonstrating impressive results. By giving farmers timely information about crop health and threats, the proposed algorithm can be expanded to produce user-friendly mobile applications that ultimately promote sustainable agriculture and food security.

References

1. Bansal, J.C.: Computer Vision and Machine Learning in Agriculture, Volume 3. Springer Nature (2023)
2. Fuentes, A., Lee, J., Lee, Y., Yoon, S., Park, D.S.: Anomaly detection of plant diseases and insects using convolutional neural networks. In: Proceedings of the International Society for Ecological Modelling Global Conference
3. Grinblat, G.L., Uzal, L.C., Larese, M.G., Granitto, P.M.: Deep learning for plant identification using vein morphological patterns. Computers and electronics in agriculture **127**, 418–424 (2016)
4. Kamilaris, A., Prenafeta-Boldú, F.X.: Deep learning in agriculture: A survey. Computers and electronics in agriculture **147**, 70–90 (2018)
5. Kawasaki, Y., Uga, H., Kagiwada, S., Iyatomi, H.: Basic study of automated diagnosis of viral plant diseases using convolutional neural networks. In: Advances in Visual Computing: 11th International Symposium, ISVC 2015, Las Vegas, NV, USA, December 14-16, 2015, Proceedings, Part II 11. pp. 638–645. Springer
6. Ma, J., Du, K., Zheng, F., Zhang, L., Gong, Z., Sun, Z.: A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. Computers and electronics in agriculture **154**, 18–24 (2018)
7. Mohanty, S.P., Hughes, D.P., Salathé, M.: Using deep learning for image-based plant disease detection. Frontiers in plant science **7**, 1419
8. Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., Batra, N.: Plantdoc: A dataset for visual plant disease detection. In: Proceedings of the 7th ACM IKDD CoDS and 25th COMAD, pp. 249–253 (2020)
9. Stočes, M., Vaněk, J., Masner, J., Pavlík, J.: Internet of things (iot) in agriculture-selected aspects. Agris on-line Papers in Economics and Informatics **8**(1), 83–88
10. Zhang, S., Huang, W., Zhang, C.: Three-channel convolutional neural networks for vegetable leaf disease recognition. Cognitive Systems Research **53**, 31–41 (2019)