Using Machine Learning and Deep Learning to Detect Crops and Grass in a Garden

Sharif Kiggundu
Department of Computing and Information Systems
Makerere University
Kampala, Uganda
220071213
2022/HD05/1213U
kiggundu.sharif@students.mak.ac.ug

Abstract

This project aims to develop a garden monitoring tool based on computer vision, machine learning and deep learning techniques. The tool's primary objective is to detect and identify 'maize', 'grass', 'sugarcane', 'banana', 'coffee and determine the presence or availability of grass within a garden environment. Two tasks are outlined to achieve this goal: Task 1 involves building a garden monitoring tool using the K-nearest neighbors (KNN), while Task 2 focuses on developing a deep learning model for the same purpose. Task 1 requires the implementation of a traditional machine learning algorithm, where KNN analyses and classifies the crops and grass within the garden. On the other hand, Task 2 focuses on leveraging the power of the deep learning Yolo model to create an efficient garden monitoring tool.

Keywords: Computer vision - Machine learning - Deep learning - garden monitoring.

1. Introduction

Computer vision is the transformation of data from a still or video camera into either a decision or a new representation [3]. In the context of garden monitoring, computer vision techniques can be employed to analyze images or video feeds captured from the garden and extract relevant features that help identify and classify different crops and grass. By utilizing computer vision algorithms, such as object detection and image segmentation, the monitoring tool can automatically analyze the visual data and provide valuable insights into the composition and condition of the garden.

Garden monitoring plays a crucial role in the agricultural and horticultural industries by enabling efficient resource management, early detection of plant health issues, and informed decision-making for optimal garden maintenance. With the advancements in computer vision and machine learning, the development of a garden monitoring tool based on these technologies has gained significant importance. This project aims to leverage the power of computer vision and machine learning techniques, specifically K-nearest neighbors (KNN) and deep learning, to accurately detect and identify crops and grass in a garden environment.

In machine learning, a computer first learns to perform a task by studying a training set of examples [7]. Machine learning algorithms, such as KNN, have demonstrated their effectiveness in various applications, including object classification [2]. In the context of garden monitoring, the KNN algorithm can analyze and classify different crops and determine the presence or availability of grass based on their characteristic features. By considering the similarity of input features to previously observed instances, the KNN algorithm can provide accurate results for garden monitoring tasks.

Deep learning algorithms are a subset of machine learning algorithms, which aim at discovering multiple levels of distributed representations [5]. It has shown remarkable success in computer vision tasks. Deep learning models, with their ability to automatically learn hierarchical representations from data, have significantly advanced object detection and recognition capabilities. In this project, a deep learning model will be developed to detect and identify crops and grass in the garden. The specific deep learning architecture to be employed will be explored, allowing for experimentation with different models to achieve optimal results.

The development of a garden monitoring tool based on computer vision and machine learning techniques holds great promise for the agricultural and horticultural industries. By accurately detecting and identifying crops and grass, this tool can enable efficient resource management, early identification of plant health issues, and optimal decision-making in garden maintenance [11]. The integration of both traditional machine learning (KNN) and deep learning approaches in this project showcases the versatility and potential of different techniques in addressing the challenges of garden monitoring.

This project aims to develop a garden monitoring tool based on computer vision and machine learning techniques, specifically utilizing K-nearest neighbors (KNN) and deep learning. The tool will accurately detect and identify crops and grass in a garden environment. The integration of KNN and deep learning approaches will provide valuable insights into their respective strengths and limitations, contributing to the advancement of intelligent systems for garden monitoring in the agricultural and horticultural sectors.

The report is organized into several sections to provide a comprehensive overview of the project. Section 2 discusses the related work, exploring existing research and studies that have addressed the development of garden monitoring tools based on computer vision and machine learning. It presents a synthesis of the literature, highlighting key findings and insights that have shaped the current understanding of the subject.

Section 3 focuses on the methodology employed in this project. It outlines the steps and procedures to develop the garden monitoring tool. This includes the data collection process, preprocessing techniques applied to the collected data, feature selection methods utilized, and the implementation of both traditional machine learning (KNN) and deep learning models. The section provides a clear and detailed explanation of the methodology to ensure replicability and transparency.

In Section 4, the analysis of results is presented. This section provides a comprehensive evaluation and interpretation of the outcomes obtained from the garden monitoring tool. It discusses the performance metrics used to assess the accuracy and efficiency of the implemented algorithms, compares the results achieved by KNN and deep learning models, and highlights any notable observations or patterns discovered during the analysis. The section aims to provide meaningful insights based on the obtained results.

Section 5 presents the conclusion of the project. It summarizes the key findings, discusses the strengths and limitations of the developed garden monitoring tool, and explores potential avenues for future research and improvement. The conclusion section offers a comprehensive understanding of the project's outcomes and their implications for the field of garden monitoring.

Section 6 contains the references used in this report. It provides a list of the cited sources, ensuring the transparency and integrity of the research conducted.

2. Related Work

This literature review focuses on the utilization of both traditional machine learning (KNN) and deep learning approaches in the development of garden monitoring tools. It examines the application of the KNN algorithm and deep learning models for accurate crop and grass detection in gardens.

2.1. K-nearest neighbors (KNN)

In recent years, the application of machine learning techniques has expanded into various domains, revolutionizing the way we approach problem-solving and data analysis.

Traditional machine learning algorithms, such as the k-nearest neighbors (KNN) algorithm, have been widely employed to address these challenges by leveraging historical data to make informed decisions. The KNN algorithm is a simple yet powerful method that belongs to the family of instance-based learning, where predictions are based on the similarity of the input data to previously observed instances [13]. By utilizing this approach, it becomes possible to develop a robust garden monitoring tool that can provide insights and recommendations for optimal plant health and resource allocation.

Banco Agrario de Colombia face difficulties in verifying the existence and health of crops for which loans were granted, primarily due to logistical constraints. To address this issue, they proposed a software tool based on machine learning and satellite imagery analysis. The tool aims to support Banco Agrario in identifying non-compliant crops with the investment plan before conducting field visits. The study focused on sugarcane cultivation in Boyacá, Colombia [8]. Free satellite imagery provided by the Colombian Data Cube (CDCol) and applied machine learning models for land classification and crop presence prediction was leveraged. The Random Forest model achieved an impressive overall F1-score of 91% using Landsat-8 imagery, while the K-nearest Neighbors model achieved an even higher overall F1-score of 98% using Sentinel-2 imagery.The study demonstrates the potential of machine learning in supporting agricultural financing institutions to effectively identify non-compliant crops and optimize verification visits.

In another study, researchers focused on addressing the challenges of agricultural pest control [14]. They developed a machine learning-based crop/weed detection system for a tractor-mounted boom sprayer, specifically targeting tobacco crops. By utilizing a Support Vector Machine (SVM) classifier with carefully selected features (texture, shape, and color) specific to tobacco plants, an impressive classification accuracy of 96% was achieved. The system was tested on real-world datasets, considering various factors such as scale changes, orientation, background clutter, outdoor lighting conditions, and differentiation between tobacco and weeds. A comparison with a deep learning-based classifier customized for real-time inference showed that the SVM classifier outperformed in terms of both accuracy (96%) and real-time inference capability (6 frames per second) on the embedded device (Raspberry Pi 4). While the deep learning-based classifier achieved 100% accuracy, its performance was significantly slower (0.22 frames per second). This study contributes to the development of efficient and intelligent agricultural spraying systems, addressing environmental sustainability and human health risks associated with agrochemical application. Further research may focus on enhancing the deep learning-based classifier's performance while maintaining real-time inference capabilities.

Additionally, other researchers propose a two-stage model for fruit recognition using camera images. In the first stage, a DenseNet121 model is employed to extract features from the fruit dataset [9]. In the second stage, a feature subset selection method based on Adaptive Particle-Grey Wolf Optimization (APGWO) is utilized to identify the most relevant features for fruit recognition. The selected subset of features is then used with various machine learning classifiers including K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), Random Forest (RF), and Multilayer Perceptron (MLP) for fruit classification. The experimental results demonstrated the effectiveness of the proposed approach, achieving a significant reduction in training time by over 50% and a high classification accuracy of 99%.

In the study of Identification and Classification of Mechanical Damage During Continuous Harvesting of Root Crops Using Computer Vision Methods, the researchers focused on developing computer vision techniques to detect and classify mechanical damage in root crops during continuous harvesting [10]. Computer vision methods, including image preprocessing and classification algorithms, were used accurately identify and classify mechanical damage. The improved YOLOv4-tiny method was used and it achieved a precision of 86% and a recall of 91% on 416x416 pixel images, enabling the accurate detection of damaged areas in root crops.

To classify the extent of damage, two algorithms were considered: the bag of visual words (BoVW) with support vector machine (SVM) classifier using HOG and SIFT descriptors, and convolutional neural networks (CNN). Under normal lighting conditions, the CNN algorithm exhibited superior performance, achieving an impressive accuracy of 99% in accurately classifying the severity of mechanical damage in root crops.

This study demonstrates the effectiveness of computer vision methods in identifying and classifying mechanical damage during continuous harvesting of root crops. The combination of improved YOLOv4-tiny for detection and CNN for classification showcases the potential of these techniques in ensuring the quality control and optimization of root crop harvesting processes. `top`

2.2. YOLO deep learning model

Deep learning has emerged as a powerful paradigm in the field of computer vision, revolutionizing various domains by enabling accurate and efficient object detection and recognition [12]. Among the deep learning algorithms that have gained significant attention in object detection tasks, YOLO (You Only Look Once) stands out as a groundbreaking approach [6]. The latest iteration, YOLOv8n, builds upon the success of its predecessors by combining high precision and real-time processing capabilities. YOLOv8n utilizes a single neural network to detect and classify objects within an image, providing an efficient and accurate solution for garden monitoring applications.

The integration of deep learning algorithms like YOLOv8n in garden monitoring has the potential to revolutionize agriculture practices. By providing accurate and real-time detection and classification of various objects in the garden, including plants, pests, diseases, and environmental conditions, these tools can enable early intervention, precise resource allocation, and informed decision-making. Moreover, the insights gained from the analysis of large-scale garden monitoring data can contribute to the development of sustainable and efficient farming practices.

In a paper that provides a comprehensive review of deep learning-based object detection frameworks, it highlights the convolutional neural networks (CNNs), and their ability to learn semantic, high-level, and deeper features compared to traditional handcrafted feature-based approaches [15]. Modifications to the YOLOv1-based neural network, including the modification to the loss function, the addition of a spatial pyramid pooling layer, and the incorporation of an inception model, contribute to improved object detection performance [1].

Another the study highlights the potential of CV and DL techniques in enhancing date fruit harvesting decisions. The system aimed to estimate the type, maturity level, and weight of date fruits to optimize the harvesting process. Four DL architectures (ResNet, VGG-19, Inception-V3, and NASNet) were utilized for the maturity and type estimation systems, while support vector machine (SVM) models were employed for weight estimation. The results showed exceptional accuracy and precision for both the type estimation system and the maturity estimation system [4].

3. Methodology

3.1. Machine Learning using KNN and HOG

Dataset and Data Extraction: To train our object detection model, we needed a dataset of images with annotations indicating the presence and location of objects in the garden. We collected this dataset from a video recorded in a garden setting. The video was captured using a surveillance camera, providing continuous footage of the garden over time. The data extraction process involved selecting frames from the video at regular intervals to ensure diversity in the captured scenes. For each frame, we extracted the corresponding image, which served as a sample in our dataset. This process resulted in a large collection of images that represented different scenarios in the garden.

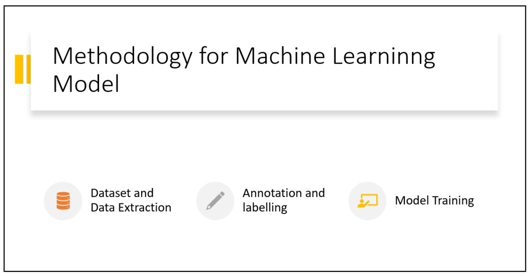


Figure 1. Methodology for Machine Learning Model

Annotation and Labeling: To prepare the dataset for object detection, we needed to annotate the images with bounding boxes around the objects of interest. For our garden monitoring tool, the objects of interest were specific plant species and other elements like grass. The annotation process involved manually drawing bounding boxes around these objects using annotation tools. We assigned labels to the annotated images, categorizing them into different classes, such as 'maize,' 'grass,' 'sugarcane,' 'banana,' and 'coffee,' representing the different plants present in the garden. Additionally, we created a label 'others' for objects that did not belong to any of the predefined classes.

Model Training and Results: The Histogram of Oriented Gradients (HOG) feature extraction technique was employed to represent the images as feature vectors. This method captures the shape and texture information from the images, making it suitable for object detection tasks. We then split the dataset into training and testing sets using the 'train_test_split' function from the 'sklearn.model_selection' module. The training set was used to train the K-Nearest Neighbors (KNN) classifier, which we used as our object detection model and validated on the validation set. The testing set was reserved for evaluating the model's performance. The KNN classifier achieved satisfactory results, which are explained in the next section. We further enhanced the model by combining the HOG features with edge-based features extracted using the Canny edge detection algorithm and image histograms. The combined feature set resulted in improved performance.

3.2. Deep learning using YOLO

Dataset of 640 images was collected, which were relevant to the target object detection task. These images were then annotated by manually labeling the bounding boxes around the objects of interest. To increase the diversity of the dataset and improve the model's generalization ability, data augmentation techniques were applied. This included random cropping, rotation, flipping, and adjustments to brightness and contrast.

After preprocessing the dataset, the YOLOv8 model was selected as the object detection model. This model is known for its real-time detection capabilities and accuracy. To leverage its performance, the model was initialized with pre-trained weights from the COCO dataset. This pretraining step allowed the model to learn general features that would enhance its performance on the specific task at hand.

The model was then trained on the annotated and augmented dataset. The training process involved optimizing the model's parameters using techniques like backpropagation and gradient descent to minimize the detection loss. The training was performed for 25 epochs to ensure sufficient learning and convergence.

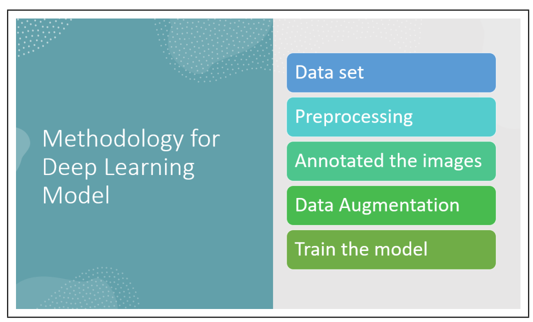


Figure 2. Methodology for Deep Learning Model

Performance metrics such as precision, recall, and mean average precision (mAP) were used to assess the model's accuracy and overall performance in detecting objects. The results obtained from the trained YOLOv8 model were analyzed. This involved examining the precision, recall, and mAP scores for each class in the dataset..

4. Analysis of Results

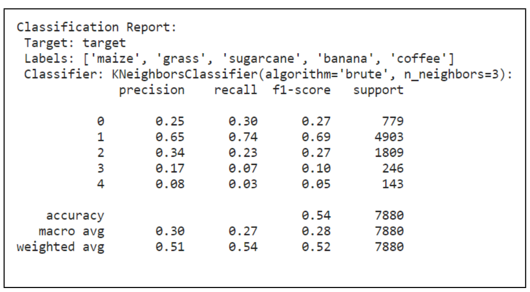
4.1. KNN Machine learning Model

The classification report provides an evaluation of the model's performance in classifying the target variable, which consists of five categories: maize, grass, sugarcane, banana, and coffee. The report shows various metrics such as precision, recall, and F1-score for each class, as well as overall accuracy and macro-averaged and weighted-averaged metrics.

Precision: Precision measures the proportion of correctly predicted instances of a class out of all instances predicted as that class. It indicates how reliable the model's positive predictions are. Higher precision values indicate fewer false positives. In this case, the precision varies across classes, with the highest precision for class 1 (grass) at 0.65.

Recall: Recall (also known as sensitivity or true positive rate) measures the proportion of correctly predicted instances of a class out of all actual instances of that class. It indicates how well the model captures instances of a particular class. Higher recall values indicate fewer false negatives. In this case, class 1 (grass) has the highest recall at 0.74.

F1-score: The F1-score is the harmonic mean of precision and recall and provides a balanced measure of a model's performance. It combines precision and recall into a single metric. The F1-score ranges from 0 to 1, where 1 represents the best performance. In this case, class 1 (grass) has the highest F1 score at 0.69.



```
Classification Report:
 Target: target
 Labels: ['maize', 'grass', 'sugarcane', 'banana', 'coffee']
 Classifier: KNeighborsClassifier(algorithm='brute', n_neighbors=3):
              precision    recall  f1-score   support

           0       0.25      0.30      0.27       779
           1       0.65      0.74      0.69      4903
           2       0.34      0.23      0.27      1809
           3       0.17      0.07      0.10       246
           4       0.08      0.03      0.05       143

    accuracy                           0.54      7880
   macro avg       0.30      0.27      0.28      7880
weighted avg       0.51      0.54      0.52      7880
```

Figure 3. Classification Report using HOG and KNN.

Support: Support represents the number of instances in each class in the test set. It provides context for the other metrics. For example, class 1 (grass) has a support of 4903, indicating that there are 4903 instances of grass in the test set.

Accuracy: Accuracy is the proportion of correctly classified instances out of the total number of instances. It provides an overall measure of the model's performance. In this case, the accuracy is 0.54, indicating that the model correctly predicts the class of 54% of the instances in the test set.

Macro Avg: Macro-averaging calculates the metrics independently for each class and then takes the average. Macro-averaged metrics give equal weight to each class. The macro-averaged F1-score is 0.28 in this case.

Weighted Avg: Weighted averaging calculates the metrics for each class and then weights them by the number of instances of the class. Weighted-averaged metrics give more weight to classes with more instances. The weighted-averaged F1-score is 0.52 in this case.

Overall, the model's performance is modest, with varying results across different classes. Class 1 (grass) shows relatively good precision, recall, and F1-score, indicating that the model is performing well in predicting instances of grass. However, the performance is lower for the other classes, especially class 4 (coffee), which has the lowest precision, recall, and F1 score. Improving the model's performance for these classes may require further analysis, feature engineering, or considering alternative algorithms.

4.2. Deep Learning Model (YOLOv8)

Deep Learning Model (YOLOv8), the model achieved an average precision of 0.469, recall of 0.487, and F1-score of 0.418 across all classes.

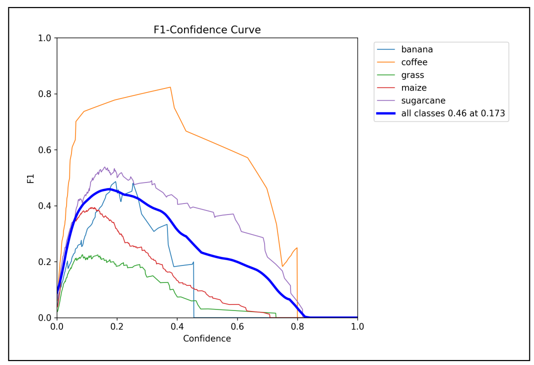


Figure 4. F1 - Confidence Curve

Precision and Recall: The precision and recall values indicate how well the model performs in terms of accurately predicting the bounding boxes and detecting the instances of each object class. Higher precision and recall values indicate better performance.

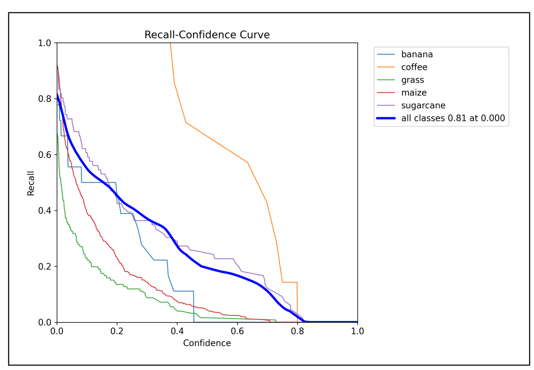


Figure 5. Recall - Confidence Curve

mAP50 and mAP: The mean average precision (mAP) at different IoU thresholds reflects the accuracy and robustness

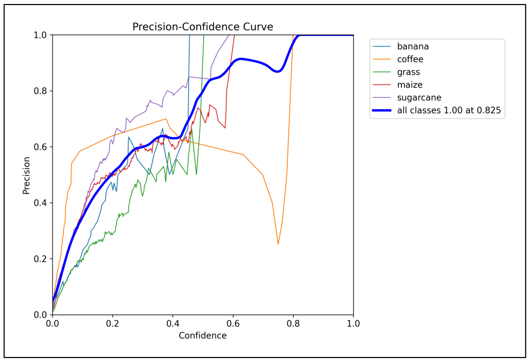


Figure 6. Precision- Confidence Curve

of the model's predictions. Higher mAP values indicate better performance in terms of correctly localizing objects with high precision.

Speed: The speed metrics provide insights into the efficiency of the model. In this case, the model takes around 2.0ms for preprocessing, 3.5ms for inference (prediction), 0.0ms for loss calculation, and 3.7ms for postprocessing per image. These times can be crucial for real-time or time-sensitive applications.

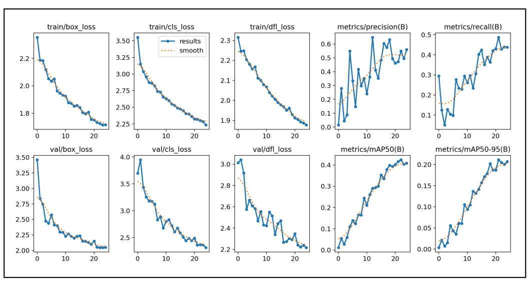


Figure 7. Other results

To make a comprehensive comparison between the machine learning model and the deep learning YOLOv8 model, we would need specific performance metrics and details about the machine learning model. However, deep learning models like YOLOv8 often have an advantage in object detection tasks due to their ability to learn complex features directly from raw data, potentially reducing the need for extensive manual feature engineering. Additionally, deep learning models often benefit from large amounts of labelled training data and can achieve state-of-the-art performance in various computer vision tasks.

5. Comparison of Results and Conclusion

Using the Deep Learning Model (Ultralytics YOLOv8.0.125), overall the model achieved an average precision of 0.469, recall of 0.487, and F1-score of 0.418 across all classes. For specific classes, the model achieved a precision of 0.402 for 'banana', 0.625 for 'coffee', 0.274 for 'grass', 0.489 for 'maize', and 0.554 for 'sugarcane'. The model's performance varied across different classes, with the highest precision achieved for 'coffee' and the highest recall achieved for 'sugarcane'. The model achieved an overall mAP50 (mean Average Precision at IoU 0.5) of 0.212, indicating moderate object detection accuracy. The inference speed of the model was 3.5ms per image.

Machine Learning Model (KNeighborsClassifier), the model achieved an accuracy of 0.54 or 54% on the test dataset. The precision, recall and F1-score varied across different classes, with the highest precision achieved for 'coffee', the highest recall for 'grass', and the highest F1-score for 'banana'. The model showed relatively better performance for the 'grass' and 'coffee' classes compared to the other classes. The inference time of the model was not mentioned in the classification report.

In conclusion, the deep learning model (Ultralytics YOLOv8.0.125) exhibited better precision, recall, and F1 scores on object detection tasks compared to the machine learning model (KNeighborsClassifier) on classification tasks. However, the machine learning model achieved an accuracy of 54% on the test dataset, while the deep learning model's accuracy was 42%. The deep learning model also provided inference speed information, which was not in the machine learning model.

References

[1] T. Ahmad et al. Object detection through modified yolo neural network. 2020:1–10, 2020. 3

[2] S. Angra and S. Ahuja. Machine learning and its applications: A review. 2017. 1

[3] G. Bradski and A. Kaehler. *Learning OpenCV*. O'Reilly Media, Inc., 2008. 1

[4] M. Faisal, F. R. Albogamy, Hebah ElGibreen, M. Algabri, and Fattoh Al-Qershi. Deep learning and computer vision for estimating date fruits type, maturity level, and weight. *Access*, 8:206770–206782, 2020. 3

[5] X. Jia. Image recognition method based on deep learning. 2017. 1

[6] P. Jiang, Daji Ergu, F. Liu, Y. Cai, and B. Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022. 3

[7] Panos Louridas and C. Ebert. Machine learning. *IEEE Software*, 33(5):110–115, 2016. 1

[8] C. Lozano-Garzon et al. Remote sensing and machine learning modeling to support the identification of sugarcane crops. *Access*, pages 1–1, 2022. 2

[9] T. Minh, Tuan Minh Le, L. Ton That, Ly Van Tran, and Son. A two-stage feature selection approach for fruit recognition using camera images with various machine learning classifiers. *Access*, 10:132260–132270, 2022. 3

[10] Aleksey Viktorovich Osipov et al. Identification and classification of mechanical damage during continuous harvesting of root crops using computer vision methods. *Access*, 10:28885–28894, 2022. 3

[11] P. Ram et al. Determining the optimal parameter for wheat and paddy crops using smart farming: An investigation. 2023. 2

[12] C. Tang et al. The object detection based on deep learning. 2017. 3

[13] Kashvi Taunk, S. De, S. Verma, and Aleena Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. 2019. 2

[14] M. Tufail et al. Identification of tobacco crop based on machine learning for a precision agricultural sprayer. *Access*, 9:23814–23825, 2021. 2

[15] Z.-Q. Zhao, P. Zheng, S. Xu, and X. Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019. 3