International Journal of Multidisciplinary Studies and Innovative Technologies

e-ISSN: 2602-4888
dergipark.org.tr/en/pub/ijmsit
Research Article

2024, 8 (2), 144-150
DOI: 10.36287/ijmsit.8.2.15
Received: December 1, 2024; Accepted: December 18, 2024

Potato Leaf Disease Detection Using Faster R-CNN and YOLO Models

Sara Medojević[1*]

[1*]*Faculty of Applied Sciences, University of Donja Gorica , Podgorica, Montenegro (sara.medojevic2@gmail.com) (ORCID: 0009-0002-9995-9416)*

Abstract – Potato is one of the most important food crops globally in terms of total food production, significantly impacting the global economy. Infected potato plants show visible symptoms on their leaves, which drastically simplifies the process of early detection, disease prevention, and minimizing the risk to uninfected plants. Smart farming and new advanced technologies incorporate different tools for real-time monitoring and analysis. Most of the models used for potato leaf disease detection are based on Deep Learning architectures, most commonly on Convolutional Neural Network (CNN) architecture, which is suitable for computer vision and image recognition. This paper depicts and compares the performances of the YOLOv11 Object Detection (Fast) model, YOLOv11s model, and Faster R-CNN X101-FPN model. These models were trained on a dataset developed for object detection in Roboflow. This dataset consists of 1200 images and 1500 annotations. A single object was labeled as one of the six classes: Pest, Bacteria, Fungi, Healthy, Phytophthora, and Nematode. Performance metrics show that these models achieve reputable results without excessive training time, making them suitable for real-time monitoring systems. YOLOv11 Object Detection (Fast), YOLOv11s, and Faster R-CNN X101-FPN achieved mAP50 scores of 95.1%, 97.6%, and 92.62%, respectively.

Keywords *– YOLO models, Faster R-CNN model, Roboflow, object detection, potato leaf disesase*

Citation: Medojević, S. (2024). Potato Leaf Disease Detection Using Faster R-CNN and YOLO Models. International Journal of Multidisciplinary Studies and Innovative Technologies, 8(2): 144-150.

I. INTRODUCTION

Potato is one of the most important food crop globally in terms of total food prodaction, standing alongside rice, wheat and corn [1]. The latest FAOSTAT data indicate that potato production surpassed 376 million metric tons globally in 2022 [2]. China, India, Ukraine, and Russia are the primary regions where potato markets and production are widely established. Potato production has significant impact on global economy. The potato market size is estimated at 115.74 billion dolars in 2024, and is expected to reach 137.46 billion dolars by 2029 [3]. Potato production is hampered by various potato diseases which contribute to the yield loss. Phytophthora infestans is the most widespread potato disease, accounting for up to 10 billion dolars in yield losses and management costs [4]. Early detection of diseases and damage play a crucial role in harvesting the maximal potato capasity.

The potato plant has a complex structure, with both underground and aboveground components as well as external and internal structures. Infected potato plants display visual symptoms that can be identified by inspecting the leaves. This greatly aids in prevention, early disease detection, and in stopping the further spread of infection, there by protecting healthy plants.

Categorization of potato leaf diseases can be based on the pathogens that cause them. These pathogens include bacteria, fungi, viruses, mycoplasma, nematodes, and adverse environmental conditions [5]. This paper focuses on the following categories of potato diseases: bacteria, fungi, Phytophthora, nematodes, and pests.

Diseases caused by bacteria primarily impact tubers and stems, with changes in the leaves being a byproduct of the bacterial infection. Bacterial infections prevent water absorption and impair the plant's ability to extract nutrients from the soil. One of the most common bacterial pathogens is Ralstonia solanacearum, a soil-borne bacterium that infects plants through the roots. Symptoms of bacterial infection include rapid wilting and curling of leaves, which can lead to the collapse of entire plants [6],[7].

Diseases caused by fungi can result from a broad group of organisms. Depending on the specific pathogen, potato leaves may exhibit various symptoms. One of the most common fungal diseases is Early Blight, caused by Alternaria solani. Symptoms of this infection include small, dark brown to black spots that appear in circular patterns. Additionally, leaves with slightly sunken spots may have yellow tissue surrounding the affected areas [6].

Phytophthora disease is caused by the oomycete plant pathogen Phytophthora infestans. One of the most common types is Late Blight. Symptoms of this infection include dark gray to brown water-soaked spots on leaf tissue, often surrounded by white, mold-like growth around the edges. Certain lesions may enlarge and develop into necrotic patches [4],[6].

Nematode infections can be divided into two main categories: root-knot nematodes and cyst nematodes. One common symptom is chlorosis, which is caused by reduced chlorophyll content due to nutrient deficiencies, appearing as yellowing of the leaves [6],[8].

Potato pests can be divided into three categories: sucking pests, tuber and root damaging pests, and foliage feeders or defoliating pests. Symptoms of pest damage include distorted leaves with holes and/or leaves dotted with a silver coloration [6].

Healthy leaves appear as uniformly green leaves with no discoloration and a perfect leaf shape without any imperfections [6].

The development of advanced technology and the Internet of Things (IoT) has significantly transformed agriculture and improved sustainable agricultural practices. To maximize crop yields and improve resource management, traditional farming methods with limitations such as reliance on human labor, simple tools and machinery, and basic observation have needed to be replaced with smart farming methods. Smart farming incorporates IoT, Global Positioning Systems (GPS), sensors, robotics, drones, precision equipment, actuators, and data analytics for real-time monitoring of crops, soil, water, nutrients, and microclimate. These measurements help maintain soil quality, reduce soil degradation, conserve water resources, improve land biodiversity, and ensure a natural and healthy environment. Inspecting potato leaves for visible signs of infection aids in early identification, disease prevention, and minimizing the risk to uninfected plants, supporting sustainable agricultural practices. Real-time monitoring contributes to plant protection, product quality, fertilization, and disease detection. Leveraging available data and predictive models enables informed decision-making [9].

Artificial Intelligence (AI) is a term that encompasses a wide range of fields and techniques, some of which may overlap [10]. AI imitates human intelligence, with the ability to learn, recognize patterns, adapt, and create models based on previously acquired knowledge and data [11]. Deep learning is a subfield of machine learning that uses algorithms to analyze multi-layered representations of data, enabling the modeling of complex relationships within that data [12]. Various Deep Learning (DL) models are built upon Convolutional Neural Networks (CNN), with notable examples including the Region-Based Convolutional Neural Network (R-CNN), Mask Region-Based Convolutional Neural Network (Mask R-CNN), AlexNet, ResNet, Single Shot Multibox Detector (SSD), and YOLO (You Only Look Once) [13-15]. Such models have a wide range of applications in agriculture and can be used for detecting potato leaf diseases.

One of the first papers on this topic was published by Islam, Dinh, Wahid, and Bhowmik at the IEEE 30th Canadian Conference on Electrical and Computer Engineering in 2017 [16]. Significant advancements in this field have been made since 2020, marked by the publication of numerous scientific papers. The study by Ashikuzzaman, Roy, Lamon, and Abedin [17], compares the performances of nine Deep CNN models: Inception V3, VGG16, VGG19, InceptionResNetV2, NasNetMobile, NasNetLarge, ResNet50V2, ResNet101V2, and DenseNet201, with the DenseNet201 model achieving the highest validation accuracy of 96%. The main objective of the study by Zarrouk, Yandouzi, Grari, Bourhaleb, Rahmoune, and Hachami [18], focuses on detecting late blight disease using the following models: Faster-RCNN (RS50), Faster-RCNN (VGG19), Faster-RCNN (VGG16), YOLOv8, YOLOv7, and YOLOv6. The best-performing models are Faster-RCNN (RS50) with a precision of 93.92%, recall of 94.01%, and mAP of 95.32%, and Faster-RCNN (VGG16) with a precision of 91.96%, recall of 91.47%, and mAP of 93.22%. The paper by Kothari, Mishra, Gharat, Pandey, Gharat, and Thakur [19], focuses on comparing the performances of four models: CNN, GoogleNet, ResNet50, and VGG16. All of the models have classification accuracy around 97%. Papers written on this topic continue to improve. The majority of papers on this topic focus on image classification, with fewer studies addressing object detection and segmentation.

II. Materials and Method

A. An Overview of the Dataset

The data used in this paper were obtained from the dataset [20], originally developed by multiple teams from Multimedia Nusantara University and Gadjah Mada University. Images were collected from several potato farms, primarily located in Central Java, and in an uncontrolled environment. This method of data collection provided a wide range of image lighting, sharpness, angles, backgrounds, the number of leaves in an image, and types of diseases. The dataset consists of 3076 images, which were divided into seven classes: nematode, fungi, bacteria, pest, virus, Phytophthora, and healthy. The images are in JPEG color format with a resolution of 1500 x 1500 pixels.

In order to develop a new dataset, the data needed to be downloaded from the original dataset, cleaned, augmented, labeled, and annotated. The process of cleaning the data included selecting relevant data to be part of the new dataset, removing any anomalies that could negatively impact the training of models, and isolating relevant objects in the images to bypass excessive noise that can occur when there is an excessive number of objects in a single image. The original dataset was unbalanced. The Nematode class consists of 68 images, while the Fungi class consists of 748 images. To develop a balanced dataset, data needed to be augmented. The process of augmentation included data manipulation techniques such as rotation, manipulating the background of images, cropping, and blurring parts of images. The Roboflow platform was used to create the dataset. Preprocessing of the data included enabling the Auto-Orient option and resizing. All images were resized to 640x640 pixels and were prepared for annotation. Because this dataset is created for object detection, the annotation process involved drawing bounding boxes around objects and labeling them accordingly. Examples of infected and healthy leaves can be seen in Fig. 1.

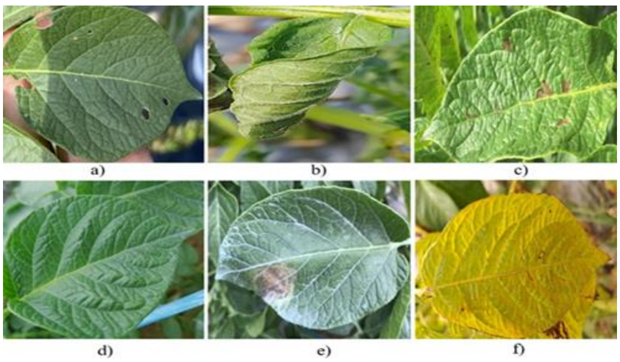

Fig. 1. Samples of leaves that fall into one of the six classes: a) Pest, b) Bacteria, c) Fungi, d) Healthy, e) Phytophthora, f) Nematode

A single object could be labeled as one of the six classes: Pest, Bacteria, Fungi, Healthy, Phytophthora, and Nematode. This process can be simplified by using Roboflow's new Auto Label feature.

This new dataset [21], consists of 1200 images and 1500 annotations. The dataset was divided into three sets, in the proportion of 70/20/10, that are used for training (840 images), validation (224 images), and testing (116 images). After finishing the dataset, dataset was exported in YOLOv11 format with TXT annotations and YAML configurations suitable for YOLOv11 model and COCO format with JSON annotations sutible for Efficient Det Pytorch and Detectron 2 models.

B. Selection of Deep Learning Models, Methods and Tools

A subset of Machine Learning, called Deep Learning, consists of deep neural networks. These neural networks are complex, multilayered structures made of interconnected nodes [22]. Deep neural network architecture has one input layer, hundreds or thousands of hidden layers, and one output layer. These layers enable the extraction of intricate features from the data.

The most commonly implemented deep learning architecture used for computer vision and image recognition tasks is Convolutional Neural Networks (CNN) [23]. A Convolutional Neural Network takes input data represented as a tensor. For an input image, three-dimensional tensors are commonly used, characterized by the image's height, width, and the number of channel layers. The number of channel layers corresponds to color channels (R, G, B) and is typically three for RGB images [24]. Convolutional Neural Network architecture consists of:

1) Convolution Layer,
2) Pooling Layer,
3) Fully-Connected Layer.

The Convolutional Layer is used to extract specific features by applying both linear and nonlinear operations, specifically convolutional operations and activation functions. Convolution is a particular linear operation specialized for feature extraction, where a kernel is applied across the input tensor. Both the input tensor and kernel are matrices. To calculate the value of a certain element, the element-wise product between the kernel and corresponding elements of the tensor needs to be summarized. The matrix that contains values calculated in this way is called a feature map. Stride is the distance between two consecutive positions of the kernel on the tensor. The process of building a feature map, where stride = 0, is shown in Fig. 2.

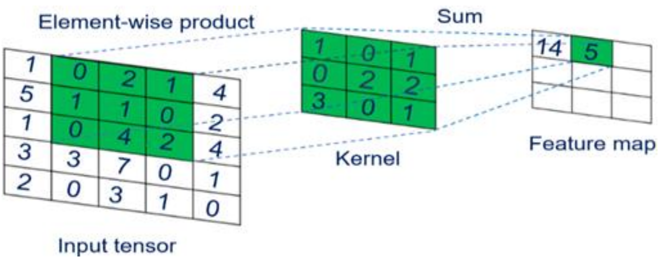

Fig. 2. Process of building the feature map

The dimension of the feature map depends on the dimensions of the tensor and the kernel. One way to increase the dimension of the feature map is by using zero-padding. Zero-padding can be: valid padding, same padding, and full padding.

An activation function is applied after each convolutional layer. During the training of a model based on convolutional neural networks, the activation function introduces nonlinearity. This nonlinearity captures complex relationships among features within an image, enabling the model to identify hidden patterns and intricate connections between characteristics that linear operations alone would not be able to capture. Two of the most t commonly used activation functions are the Rectified Linear Unit (ReLU) and the Sigmoid function.

The Pooling Layer is a layer used for reducing the dimensions of the feature map. This layer is important because it decreases the number of learnable parameters, as well as the amount of data processed within the layer. By doing so, it reduces the memory requirements during training and helps mitigate the issue of overfitting. Two main types of pooling are: max pooling and average pooling.

Fully-Connected Layer is the final layer in a Convolutional Neural Network. As the name suggests, all nodes in this layer are fully connected to the nodes in the previous layer [24-26]. The process of transforming input data in a Convolutional Neural Network is shown in the Fig. 3.

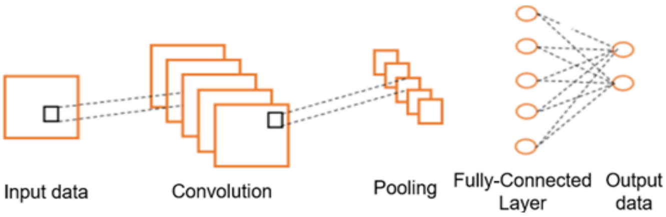

Fig. 3 Transforming input data in a Convolutional Neural Network

Convolutional neural networks (CNNs) can be used for classification, object detection, and image segmentation. This paper focuses on object detection.

Object detection involves locating an object within an image by marking the located object with a rectangular bounding box and classifying it into one of the predefined classes. Object detection methods can be divided into two groups: single-stage object detectors and two-stage object detectors. Single-Stage Object Detectors eliminate the need for a separate region of interest (RoI) extraction process, directly classifying and marking objects. Examples of single-stage object detectors include YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector). Two-Stage Object Detectors are network models that detect objects in two phases. In the first phase, regions of interest are identified, and in the second phase, the objects within these regions are classified [27]. The first part of the architecture is called the backbone. This part is responsible for extracting features from the input data. The extracted features are then passed to the second part, known as the neck. In the neck, the features from the backbone are aggregated and adjusted before being forwarded to the head for further processing. The head is the final part of the architecture, where the prediction is made [28]. The process of object detection in images can be represented by the architecture shown in Fig. 4.

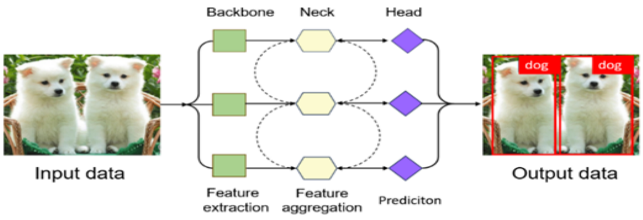

Fig. 4 Architecture of an object detection model

The Ultralytics YOLO model (You Only Look Once) is based on Convolutional Neural Networks. YOLO is a popular model for object detection and image segmentation. Although there are 11 versions of this model, this paper utilizes its latest version [29]. When using YOLOv11, it's possible to choose from several variants: YOLOv11n, YOLOv11s, YOLOv11m, YOLOv11l, and YOLOv11x. While YOLOv11x delivers high precision, its extended training time and large number of parameters demand substantial GPU resources, resulting in slower performance. In this paper, YOLOv11s was selected for its balanced precision, faster processing speed, and reduced parameter count, making it well-suited for rapid predictions. The YOLO model leverages the Ultralytics libraries, which are designed to work in Python. These libraries allow for easy configuration of training parameters, such as the number of epochs, image size, and task type (such as detection, segmentation, or classification). During model training, it is essential to set the number of epochs. An epoch represents one complete pass through the entire dataset. To ensure the model performs at a satisfactory level, it should be trained over a sufficiently large number of epochs.

The metrics used in this study are: recall, precision, mAP@0.50, and mAP@0.50-0.95. Recall is a metric used to calculate the rate of true positive instances. Precision is a metric used to calculate the model's ability to make positive predictions for attributes that are actually positive. The higher the precision, the more skilled the model is at identifying true positives and avoiding false positives. Mean Average Precision (mAP@0.50) represents the average precision calculated at an Intersection over Union (IoU) threshold of 0.50. It measures the model's accuracy while considering only "easier" detections. Mean Average Precision (mAP@0.50-0.95) is the average precision calculated across various IoU thresholds, ranging from 0.50 to 0.95. This metric provides a comprehensive view of model performance across different levels of detection difficulty [30].

The YOLOv11 Object Detection (Fast) model, developed by Roboflow, utilizes the COCO dataset as a checkpoint. This model offers faster training times, though with slightly lower accuracy compared to its counterpart, the Accurate model [31].

Region-based Convolutional Neural Network (R-CNN) is a Deep Learning framework used for object detection. R-CNN uses a Region Proposal Network (RPN) to suggest regions that potentially contain objects within an image. These regions are generated without annotated data. The algorithm employs a method called selective search, which is an approach that balances the number of proposals while maintaining high object recall, ensuring efficient object detection. The proposed regions are then processed by a CNN, which extracts features, and a binary Support Vector Machine (SVM), which helps identify objects in the regions. A bounding box regressor is used for refining the location and size of the bounding box to closely match the actual object, while the classifier predicts the category of each object.

R-CNN faces several limitations, including the rigid and non-learnable Selective Search algorithm, which can generate poor region proposals for object detection. Because of real-time applications and significantly increases disk memory usage, new variations of R-CNN have been introduced: Fast R-CNN, Faster R-CNN, Mask R-CNN, and Cascade R-CNN. Faster R-CNN improves the original R-CNN by integrating a Region Proposal Network (RPN), which generates region proposals directly from CNN feature maps, removing the need for selective search. It also shares convolutional features between the RPN and the detection network, which reduces computation time. As a result, Faster R-CNN achieves real-time processing speeds of approximately 0.1 seconds per image [32]. An iteration refers to a single update step during training. Detectron2 offers the tools and framework needed for developing and training a Faster R-CNN model [33].

III. RESULTS

All of the models were developed in Google Colab using GPU T4. **YOLOv11 Object Detection (Fast) model** was trained for 300 epochs. Fig. 5 captures changes in mAP50 and mAP50:95 throughout 300 epochs.

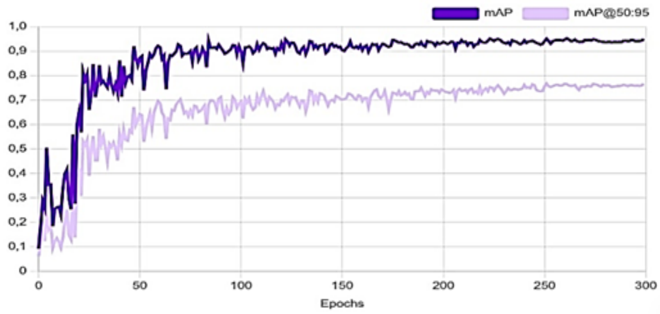

Fig. 5 Changes in mAP50 and mAP50:95 throughout 300 epochs for the YOLOv11 Object Detection (Fast) model

After 300 epochs, the model has an mAP50 of 95.1% and an mAP50:95 of 76.7%. This model has a precision of 96.4% and a recall of 90.2%. Fig. 6 shows how this model detects different classes.

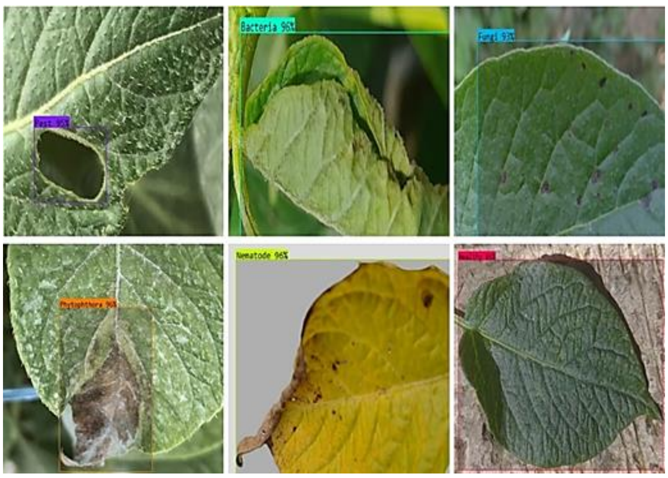

Fig. 6 Different results of the YOLOv11 Object Detection (Fast) model

YOLOv11s model was trained for 300 epochs.

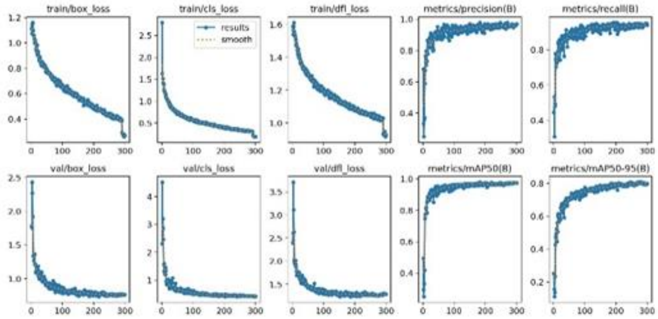

Fig. 7 Yolov11s key performance metrics, train and validation losses during model training process

Throughout the model training process, it is important to monitor training and validation losses as well as other performance metrics. These changes are represented in Fig. 7. This model has a precision of 96.1%, recall of 93.1%, mAP50 of 97.6%, and mAP50-95 of 80.6%. In-depth performance metrics are shown in Table 1.

Table 1. Performance metrics of the YOLOv11s model by class

Class	Precision	Recall	mAP50	mAP50-95
All	96.1%	93.1%	97.6%	80.6%
Bacteria	93.1%	91.8%	95%	77.1%
Fungi	97.9%	97.6%	98.7%	80%
Healthy	100%	91.1%	99.5%	95.3%
Nematode	90.4%	97.7%	97.9%	92.6%
Pest	96.8%	82.8%	94.9%	66.8%
Phytophthora	98.2%	97.7%	99.3%	71.8%

A normalized confusion matrix is one of the tools used in evaluating the performance of the model by comparing true and predicted detections. The normalized confusion matrix for this model has a prominent main diagonal with fewer values in the row and column used for showing true and predicted detections of the background. The normalized confusion matrix is represented in Fig. 8.

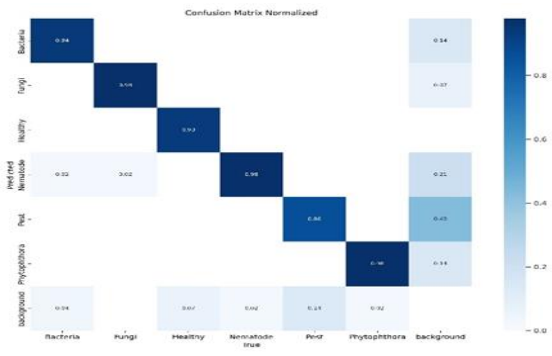

Fig. 8 Normalized confusion matrix of the YOLOv11s model

A graphical representation illustrating the variation in a model's F1 score across different thresholds is known as an F1-Confidence Curve, shown in Fig. 9. This graph shows that F1 score is 0.94 for all the classes when the threshold is set to 0.471.

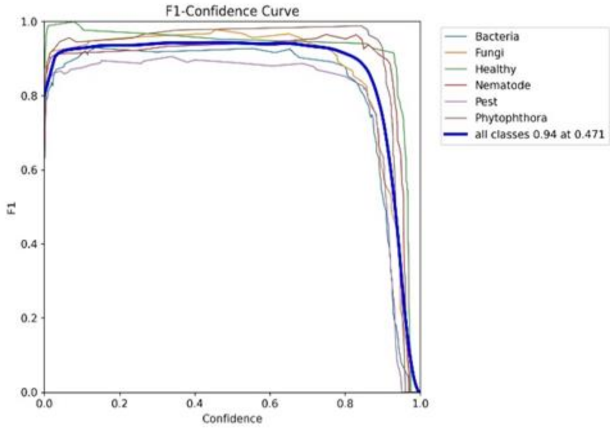

Fig. 9 F1-Confidence Curve of the YOLOv11s model

After developing the model, the results can be seen in Fig. 10.

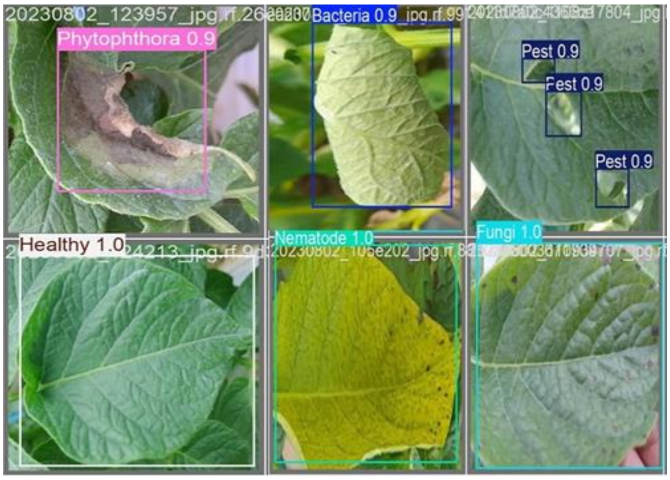

Fig. 10 Different results of the YOLOv11s model

Detectron2 Faster R-CNN X101-FPN model was trained for 2000 iterations, with a batch size of 4 images and a base learning rate of 0.001. After training, this model achieved an AP50 of 92.62%, APs of 12.08%, APm of 58.97%, and APl of 70.83%. The AP for classes: Bacteria, Fungi, Healthy, Nematode, Pest, and Phytophthora are 69.39%, 70.68%, 82.08%, 85.04%, 50.90%, and 63.81%, respectively. The results of this model are shown in Fig. 11.

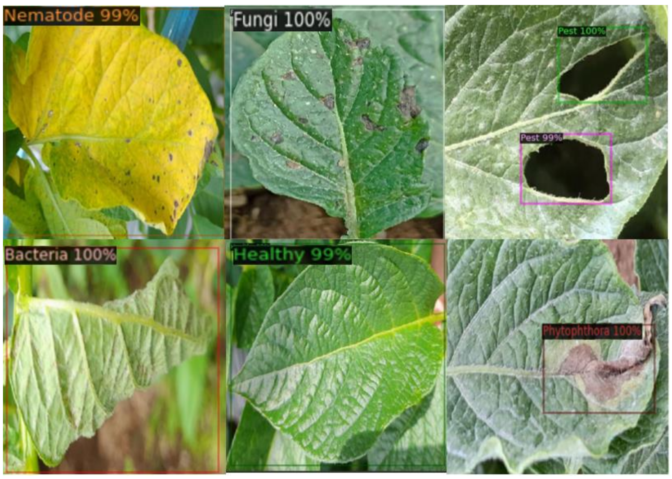

Fig. 11 Different results of the Faster RCNN X101-FPN model

Table 2 depicts mAP50, the number of epochs/iterations, and the time spent while training the previously shown three models.

Table 2. Performance metrics of the YOLOv11s model by class

Model	mAP50	Epochs/Iterations	Training time
YOLOv11 Object Detection (Fast)	95.1%	300 Epoches	1 hour
YOLOv11s	97.6%	300 Epoches	1.87 hours
Faster R CNN X101-FPN	92.62%	2000 Iteration	1.58 hours

IV.DISCUSSION

The YOLOv11 Object Detection (Fast) model prioritizes processing speed, which is often used to achieve real-time performance. Fig. 5 shows that the model achieves reputable results at the 200th epoch, after which the mAP is slightly refined. YOLOv11s has the longest training time but the best performance metrics out of these three models. The class Healthy has the best performance, while the class Pest has the worst performance out of all six classes. The reason for this is that different pests can cause numerous varying size holes on a single leaf. Depending on the background, some of these holes can be hard to detect. When a leaf is drastically damaged, the model may struggle with drawing bounding boxes because it is hard to identify where one damage ends and another pest damage begins. The confusion matrix showed that the model's most frequent errors are misidentifying background elements as objects or failing to detect actual objects, considering them part of the background. This was expected because the dataset contains numerous images with multiple leaves, some with more or less blurred backgrounds, making it difficult for the model to differentiate between leaves and background. The Faster R-CNN X101-FPN model, although showing good AP, has the longest training time and the worst AP out of these three models. Just like the other models, this model also struggles to identify the Pest class, which has the worst AP out of all six classes. The best performance is seen in the Nematode and Healthy classes. The AP values indicate that the model performs best at recognizing large objects and worst at recognizing small objects, which is a common trend for most models. Reason for this is that larger object occupy more pixels and contain more identifiable features, making it easier for the model to detect and classify them accurately, while smaller object have fewer distinctive features and often merge with the background.

V.CONCLUSION

All three models demonstrate strong performance and efficient training times. The YOLOv11s model achieves the best metric results, although it has a slightly longer training time compared to the other two models. The YOLOv11 Object Detection (Fast) model demonstrated strong performance metrics while having the shortest training time. These models can be deployed within surveillance systems to enable real-time monitoring and predictive analytics. Future improvements could include developing a larger dataset and utilizing more powerful GPUs capable of supporting models that achieve higher precision while maintaining rapid prediction speeds.

Statement of Research and Publication Ethics

The author declares that this study complies with Research and Publication Ethics

REFERENCES

[1] Y. P. S. Bajaj, *Potato*, vol 3. Springer Science & Business Media, 2013.
[2] (2024) Food and Agriculture Organization of the united Nations. [Online]. Available: https://www.fao.org/faostat/en/#search/potato
[3] (2024) Potato Market Size- ndustry Report on Share, Growth Trends & Forecasts Analysis (2024-2029) on Mordor Intelligence [Online]. Available:https://www.mordorintelligence.com/industry-reports/potato-market
[4] S. M. Dong, S.Q. Zhou,“ Potato late blight caused by Phytophthora infestans: From molecular interactions to integrated management strategies,” Journal of Integrative Agricultur, vol. 21, pp. 3456--3466, Dec. 2022.
[5] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
[6] W. J.Hooker., *Compendium of potato diseases. International Potato Center*, International Potato Center, 1981.
[7] A. Charkowski, K. Sharma, M. L. Parker, G. A. Secor and J. Elphinstone, “Bacterial Diseases of Potato” The potato crop: its agricultural, nutritional and social contribution to humankind, pp. 351--388, Dec. 2019.
[8] M. Sun, S. Chen and J. E. Kurle, “Interactive effects of soybean cyst nematode, arbuscular-mycorrhizal fungi, and soil pH on chlorophyll content and plant growth of soybean“ Phytobiomes Journal., vol. 6, pp. 95--105, Jan. 2022.
[9] M. Dhanaraju, P. Chenniappan, Poongodi, K. Ramalingam, S. Pazhanivelan, R. Kaliaperumal, “Smart farming: Internet of Things (IoT)-based sustainable agriculture” ,Agriculture, vol. 12, pp. 1745 , Sep. 2022.
[10] (2024) Subsets of Artificial Intelligence on Free Learning Platform for Better Future. [Online]. Available: https://tinyurl.com/2azettr3
[11] C. R. Arias, “An introduction to artificial” AI, Faith, and the Future: An Interdisciplinary Approach., pp. 12, Jun. 2022.
[12] L. Deng, Y. Dong, “Deep learning: methods and applications.” Foundations and trends® in signal processing., vol. 7, pp. 197--387 , 2014.
[13] (2024) R-CNN – Region-Based Convolutional Neural Networks on GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/r-cnn-region-based-cnns/
[14] (2024) Different types of CNN models on Opengenus. [Online]. Available: https://iq.opengenus.org/different-types-of-cnn-models/
[15] (2024) Ultralytics YOLO11. [Online]. Available: https://docs.ultralytics.com/models/yolo11/
[16] M. Islam, A. Dinh, K. Wahid, P. Bhowmik, Detection of potato diseases using image segmentation and multiclass support vector machine., 2017 IEEE 30th canadian conference on electrical and computer engineering (CCECE), 2017 .
[17] M. Ashikuzzaman, K. Roy, A. Lamon, S. Abedin, Potato Leaf Disease Detection By Deep Learning: A Comparative Study., 2024 6th International Conference on Electrical Engineering and Information \& Communication Technology (ICEEICT), 2024 .
[18] Y. Zarrouk, M. Yandouzi, M. Grari, M. Bourhaleb, M. Rahmoune, K. Hachami., “Revolutionizing Potato Late Blight Surveillance: uav-driven Object Detection Innovations, Journal of Theoretical and Applied Information Technology., vol. 102, Apr. 2024.
[19] D. Kothari, H. Mishra, M. Gharat, V. Pandey, M. Gharat, R. Thakur, “Potato leaf disease detection using deep learning” Int. J. Eng. Res. Technol, vol. 1, pp. 569–571, Nov. 2022.
[20] (2023) Potato Leaf Disease Dataset in Uncontrolled Environment on mendeley data. [Online]. Available: https://data.mendeley.com/datasets/ptz377bwb8/1
[21] (2024) Potato_leaf_disease Computer Vision Project dataset on Roboflow. [Online]. Available: https://universe.roboflow.com/potato-leaf-diseases/potato_leaf_disease
[22] (2024) What is deep learning? on IBM. [Online]. Available: https://www.ibm.com/topics/deep-learning
[23] (2024) Difference between Shallow and Deep Neural Networkson on GeeksforGeeks [Online]. Available: https://www.geeksforgeeks.org/difference-between-shallow-and-deep-neural-networks/
[24] (2024) Convolutional Neural Network (CNN) on TensorFlow [Online]. Available: https://www.tensorflow.org/tutorials/images/cnn

[25] R. Yamashita, M. Nishio, R. K. G. Do, K. Togashi, “Convolutional neural networks: an overview and application in radiology” Insights into imaging, vol. 9, pp. 611--629, Jun. 2018.

[26] (2024) What are convolutional neural networks? on IBM. [Online]. Available:https://www.ibm.com/topics/convolutional-neural-networks

[27] (2019) Object Detection: Architectures, Models, and Use Cases on The Random Walk Blog. [Online]. Available: https://randomwalk.ai/ blog/object-detection-architectures-models-and-use-cases/

[28] (2024) What is object detection? on IBM. [Online]. Available: https://www.ibm.com/topics/object-detection

[29] G. Jocher, J. Qiu (2024), Ultralytics YOLO11, version = 11.0.0, year = {2024}, [Online].Available:https://github.com/ultralytics/ ultralytics

[30] H. Vedoveli (2013) Metrics Matter: A Deep Dive into Object Detection Evaluation.[Online].Available:https://medium.com/@henriquevedoveli/metrics-matter-a-deep-dive-into-object-detection-evaluationef01385ec62

[31] J. Solawetz, P. Guerrie.(2022). What to Think About When Choosing Model Sizes. Roboflow Blog: https://blog.roboflow.com/ computer-vision-model-tradeoff/

[32] (2024) R-CNN – Region-Based Convolutional Neural NetworksNetworkson on GeeksforGeeks [Online]. Available: https://www.geeksforgeeks.org/r-cnn-region-based-cnns/

[33] (2019) Y. Wu, A. Kirillov, F. Massa, et al. Detectron2.[Online]. Available: https://github.com/facebookresearch/detectron2