TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 22, No. 6, December 2024, pp. 1414~1421
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v22i6.25519

Tomato leaf disease recognition system using Faster R-CNN

Karel Octavianus Bachri, Bryan Santoso, Duma Kristina Yanti Hutapea, Catherine Olivia Sereati, Lanny W. Pandjaitan
Department of Electrical Engineering, Faculty of Engineering, Atma Jaya Catholic University of Indonesia, Jakarta, Indonesia

Article Info

Article history:

Received Aug 8, 2023
Revised Jul 22, 2024
Accepted Aug 5, 2024

Keywords:

Detectron2
Faster region-based convolutional neural network
Machine learning
Object detection
Tomato leaf disease

ABSTRACT

The objective of this paper is to detect tomato leaf disease using Faster region-based convolutional neural network (R-CNN). The tomato leaf disease recognition system utilizes a dataset consisting of healthy tomato leaves and eight leaf diseases, including early blight, late blight, leaf mold, mosaic virus, septoria, spider mites, yellow leaf curl virus, and leaf miner. The dataset is obtained from various sources, such as Kaggle, Google Images, Bing Images, and Roboflow Universe. Pre-processing techniques, including collage, tile, static crop, and resize, are applied to prepare the dataset for training. Data augmentation methods, such as flipping, 90° rotation, exposure adjustment, and hue modification, are applied to enhance the model's robustness and generalize its performance. Specifically, we implemented Faster R-CNN as part of Detectron2 using its base models and configurations. The results demonstrate that the X101-FPN base model for Faster R-CNN with the default configurations of Detectron2 is efficient and general enough to be applied to defect detection. This approach results in an average precision (AP) detection score of 87.01% for validation results.

Corresponding Author:

Karel Octavianus Bachri
Department of Electrical Engineering, Faculty of Engineering
Atma Jaya Catholic University of Indonesia
Jakarta, Indonesia
Email: karel.bachri@atmajaya.ac.id

1. INTRODUCTION

In tomato cultivation, various diseases caused by pests such as insects, fungi, bacteria, and viruses [1] are often encountered, which can affect the yield and quality of tomatoes. There is also disease on tomato leaves in Kebun Ibu & Co. Figure 1(a) shows the image of healthy leaves, while Figure 1(b) diseased leaves, which affected by leaf miner. Diseases in tomato plants have different characteristics from one another. The characteristic of tomato diseases is shown in Table 1. However, farmers may not be able to identify or misidentify diseases in tomato plants, resulting in plant death. Moreover, misdiagnosis or misidentification of diseases in tomato plants can lead to improper pesticide application, causing damage or even crop failure [2].

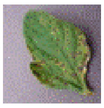
To address this problem, an automated system is required to detect plant diseases. This can be solved if the images of the plant diseases can be captured and detected specifically on tomato leaves using object detection methods. Hence, a model with high accuracy is required, such as Faster region-based convolutional neural network (R-CNN), which provides higher speed and accuracy compared to R-CNN and Fast R-CNN models [3]-[5].

In the research conducted by several researchers [6]-[9], it has been found that models categorized as two-stage detectors achieve higher accuracy compared to one-stage detectors. In plant disease detection systems, high accuracy is crucial. Therefore, this research will utilize a two-stage detector model, such as Faster R-CNN.

(a) (b)

Figure 1. Tomato plant in *Kebun Ibu & Co*: (a) healthy leaves and (b) affected leaves by leaf miner

Table 1. Characteristics of tomato plant diseases [10], [11]

Image	Characteristic/ symptom	Image	Characteristic/ symptom	Image	Characteristic/ symptom	Image	Characteristic/ symptom
 Early blight	Characterized by concentric rings and surrounded by a yellowing zone [12].	 Late blight	Lesions are characterized by being small, dark, and resembling water-soaked spots [13].	 Yellow leaf curl virus	Upward curling of leaf edges (cupping), thickening of leaf veins, yellowing and leaf curling [14].	 Mosaic virus	Presence of mosaic patterns and spots on tomato leaves [15].
 Septoria	Many small round spots of light brown color with white centers [15].	 Leaf mold	Pale green or yellowish patches on the upper surface of leaves [13].	 Spider mites	Yellowish white spots [16].	 Leaf miner	The leaves are veined and serpentine-like patterns [17].

The tomato leaf disease recognition system utilizes the Faster R-CNN with the feature pyramid networks (X101-FPN) model. This model has the highest score for object detection among the available models in the Detectron2 library [18]. Detectron2 is an object detection framework developed by Facebook AI research (FAIR). It is the latest version of the previous object detection framework known as Detectron. Detectron2 is built on top of PyTorch, a deep-learning framework that provides comprehensive functionality for object detection research and development. The pre-trained models available in the Detectron2 model zoo have been trained on the common objects in context (COCO) dataset. Therefore, during training, fine-tuning is performed on the pre-trained models. Detectron2 offers several types of object detection, including bounding-box detection, mask, keypoint detection, instance segmentation, semantic segmentation, and panoptic segmentation. In this research, the type of object detection used is bounding-box detection.

There is active research in this area. For example, using Faster R-CNN with residual networks (res101) based model to detect healthy tomato leaves and four leaf diseases: powdery mildew, blight, leaf mold fungus, and tomato mosaic virus (ToMV), the resulting model achieves mean average precision (mAP) of 98.54% [19]. Also, using an object detection method with Faster R-CNN, region-based fully convolutional networks (R-FCN), and single shot detector (SSD) algorithms merged with the latest feature extractors of deep CNN to detect tomato leaf disease: leaf mold, gray mold, canker, plague, miner, low temperature, powdery mildew, whitefly, and nutritional excess. In this proposed model, R-CNN with visual geometry group 16 (VGG-16) shows better recognition results, which achieves mAP 85.98% [20]. Another example is the use of CNN to detect tomato leaf disease on a dataset consisting of late blight, gray spot, and bacterial canker. This model achieves an accuracy of 89% [21].

Faster RCNN has also been used in tomato disease not using the leaf images, however, there are some distinction from this research, some use the berry instead of the leaves [22]. In other research, the system was built using RetinaNet and focused only on late blight, mosaic virus, tomato septoria leaf spot [23], while other using real-time faster regional convolutional neural network (RTF-RCNN) architecture to identify five disease, which are yellow leaf curl, septoria leaf spot, late blight, bacterial spot, and blight with the total of sample of only 12500 [24]. This paper focuses on developing a prediction model for detecting various types of diseases on tomato leaves, including early blight, late blight, leaf mold, mosaic virus, septoria, spider mites, yellow leaf curl virus, leaf miner, and healthy tomato leaves using combined dataset taken from various open sources.

2. METHOD

This section discusses the method used to achieve the objective. The first part of this section is the method used, which is Faster R-CNN X-101 FPN. The second part of this section is data collection and annotation, which explains the various sources of the dataset and the label corresponds to the data. The third part is data preprocessing and augmentation, which prepares the data to be ready to be processed by the system. The fourth part of this section explains the hyperparameter used in training. The testing phase of this research is explained in the result and discussion.

2.1. Faster R-CNN X-101 FPN

Faster R-CNN is a two-stage detector model that utilizes Fast R-CNN and region proposal network (RPN) as its main architecture. This model is an improvement over the previous model, Fast R-CNN, by replacing the selective search method with RPN. RPN is responsible for generating region proposals that potentially contain objects in an image. RPN generates region proposals by performing regression and classification to determine the location and presence of objects in each proposal [25].

In general, an object detector consists of three components: the backbone, neck, and head. The backbone is responsible for feature extraction from the input image, the neck is used for feature fusion, and the head is used for prediction [26]. The Faster R-CNN X101-FPN models consist of three components, which are:

- Backbone: typically, object detectors use VGG, DenseNet, or residual network (ResNet) as the backbone or underlying network [27]. Faster R-CNN X101-FPN use ResNet as their backbone or base network. ResNet is a type of neural network introduced by Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun in 2015. In addition to ResNet, there are other types of ResNet architectures such as ResNet50, ResNet101, and ResNet152. In 2015, this network successfully achieved first place in the ImageNet large scale visual recognition challenge (ILSVRC) and COCO competitions in terms of image classification, object detection, and image segmentation [28]. The foundation of the ResNet architecture is a convolution neural network (CNN) that consists of convolutional layers, pooling layers, and fully connected layers, with the addition of skip connections in some convolutional layers. The principle behind ResNet is to create a deeper network compared to other conventional networks while simultaneously addressing the vanishing gradient problem [29].

  Specifically, the Faster R-CNN X101-FPN model uses ResNeXt-101-32x8d as its backbone. ResNeXt-101-32x8d is a model trained using the Caffe2 software by researchers at Facebook [18]. The ResNeXt-101 block divides a convolution into multiple parallel convolutions. The difference between the ResNeXt block and the ResNet block is the introduction of a new dimension called cardinality (the size of the transformation set) in the ResNeXt block. The numbers 101, 32, and 8 in the model's name refer to the number of layers (101 layers), cardinality (32), and the number of bottleneck width (8), respectively [30]. A comparison between the ResNet block and the ResNeXt block is shown in Figure 2.

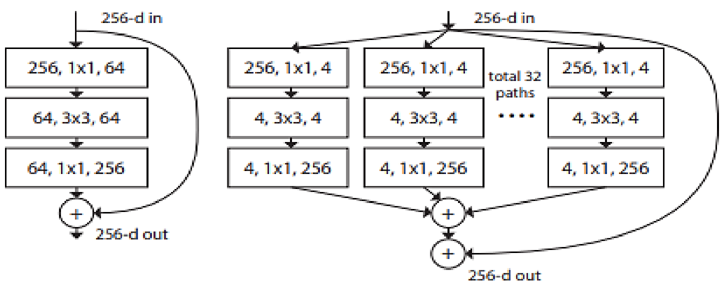


Figure 2. Left: a block of ResNet and right: a block of ResNeXt with cardinality=32 [30]

- Neck: this component is located between the backbone and the head. The Faster R-CNN X101-FPN model uses feature pyramid network (FPN) as their neck. The architecture of FPN that illustrated in Figure 3 consists of three main components: a bottom-up pathway, a top-down pathway, and lateral connections, which connect various levels of features (low-level and high-level features) so that each level in the pyramid can be used to detect objects at different scales. This feature pyramid helps improve detection accuracy by allowing the model to identify both small and large objects [31].
- Head: the head component of the Faster R-CNN X101-FPN model is a combination of Fast R-CNN and RPN. The RPN uses the feature pyramid generated by FPN to find regions that may contain objects.

Figure 3. FPN architecture [31]

2.2. Data collection and annotation

A dataset is a collection of information used to test and analyze a system. In the context of object detection, the required dataset consists of annotated images (with bounding boxes and labels). The dataset comprises healthy tomato leaves and eight leaf diseases: early blight, late blight, leaf mold, mosaic virus, septoria, spider mites, yellow leaf curl virus, and leaf miner. The dataset is obtained from various sources such as Kaggle, Google Images, Bing Images, and Roboflow Universe. Data annotations were done in Roboflow.

2.3. Data pre-processing and augmentation

One of the datasets named Plant Village [10], [11], which can be downloaded through Kaggle, is ineffective when used for object detection cases because each photo contains only one object. Therefore, pre-processing is performed using a photo collage method. Photo collage is one method of combining multiple photos into one image. Thus, the photos in the dataset are merged from four separate photos into a single photo.

In addition, pre-processing is performed using the tile and static crop methods on some images with excessively high resolutions. Since the collected images have varying resolutions, it is necessary to standardize the resolution of each image in the dataset. By standardizing the resolution, the training process will become faster [32]. In this research, the image resolution used for training is set to 512×512.

In computer vision, data augmentation is commonly used to address limited data issues. Its goal is to generate a large, high-quality, and diverse training dataset that will enable the deep learning model to become more robust and generalize well. In addition, data augmentation is one of the crucial stages in dataset creation as it has been proven to yield better accuracy [33].

The data augmentation was performed on 4,132 images, resulting in a total of 10,047 images. The augmentation configuration included flipping the images, applying a 90° rotation, adjusting the exposure, and modifying the hue. After all the processing is completed, the dataset is ready to be used for training the tomato leaf disease recognition system. Generally, datasets can be exported in various ways and formats. One of the formats commonly used is common object in context javascript object notation (COCO JSON). The COCO JSON format was chosen because the Detectron2 architecture supports datasets in COCO JSON format.

2.4. Model training

The object detection system in this research is trained using the Detectron2 framework, which is implemented in Python and executed on Google Colab. In the first step, the download of Detectron2 and other necessary library packages is performed for training the object detection system. The essential library packages that need to be downloaded and are associated with the main functionality in Detectron2 include PyTorch, NumPy, Pycocotools, protocol buffers (Protobuf), Pyyaml, and cv2 (OpenCV).

The hyperparameters are configured during the training process. Hyperparameters are parameters used to configure a machine-learning model and are determined before starting the training process [34]. There are many hyperparameters that can be adjusted when training a Faster R-CNN model. The hyperparameters adjusted in this training are:

a. Images per batch=4, which represents the number of images processed in each batch during the model training [35].
b. Base learning rate=0.001, represents the base learning rate used during the training process.
c. Warm-up iterations=1000, this parameter determines the number of training iterations used for the warm-up stage during model training. This parameter is used to lower the learning rate and reduce the impact of instability that may occur during the initial training phase [35]. In this research, the parameter used for the warm-up stage is set to 1000, which means that from the start of training until the 1000th iteration, the learning rate reduction is gradually applied.
d. Maximum iteration represents the maximum number of iterations performed during model training. In this research, various iterations were used as shown in Figure 4.
e. Region of interest (RoI) batch size per image=128, representing the number of RoI proposals used in each image during the training of the ROI head in the model.

f. Number of classes=10, where 9 classes+1 class is automatically added by Roboflow which represents the class for all images [35].

While all other are left at their default values according to Detectron2's specifications [36].

3. RESULTS AND DISCUSSION

This performance testing involved evaluating several metrics commonly used in the COCO evaluation, such as average precision (AP) dan average recall (AR). The calculation of the AP and AR metric is closely related to the intersection over union (IoU) metric. IoU is an evaluation method used to measure the accuracy of object detection in a dataset.

AP in the COCO evaluation metric is the AP across all categories or classes from the IoU threshold of 0.50 to 0.95 with an interval of 0.05 [37]. Therefore, AP represents the AP calculated at 10 different IoU thresholds (0.50, 0.55, 0.60, ..., 0.95). In addition, it is possible to evaluate it using individual IOU values, with the most used values being 50% and 75%. These are reported as AP50 and AP75, respectively. AP is also calculated across scales at different IoUs (0.50:0.95). These categories are divided into three different sizes: APs for small objects (area<32^2), APm for medium objects (32^2<area<96^2), and APl for large objects (area>96^2).

While the AR is calculated at different maximum detections per image. These categories are divided to be ARmax=1, ARmax=10, and ARmax=100. Like the AP metric, AR is also calculated across scales at different IoUs (0.50:0.95). These categories are divided into three different sizes: ARsmall, ARmedium, and ARlarge. Some types of AP and AR metrics in the COCO evaluation metric are shown in Table 2.

Figure 4 shows the AP against iterations using Faster R-CNN X101-FPN. It can be observed that during the evaluation of the validation data, the model with the best AP is the model obtained during the 37000th iteration with an AP of 87.01. For a more comprehensive view, all COCO evaluation metrics, including AP and average recall, for the model at iteration 37,000 is shown in Table 3. The model performed well in the non-dataset test images. The model can detect the healthy leaves as shown in Figure 5(a), as all the healthy leaves are recognized healthy by the model. Figure 5(b) shows that the model can recognize leaf mold diseased, as all diseased molds are recognized leaf mold with various degree of certainties, while one healthy leaf are recognized healthy, even though without 100% degree of certainty. Figure 5(c) shows that the model can recognized 1 leaf with leaf miner disease among healthy leaves. Figure 5(d) shows that the model can recognize multiple leaf diseases.

Table 2. COCO evaluation metrics [38]

AP	Area	Maximum detection	IoU	AP across scales	Area	Maximu detection	IoU
AP	All	100	0.50:0.95	APs	Small object (area<32^2)	100	0.50:0.95
AP50	All	100	0.50	APm	Medium object (32^2<area<96^2)	100	0.50:0.95
AP75	All	100	0.75	APl	Large object (area>96^2)	100	0.50:0.95
AR	Area	Maximum detection	IoU	AR across scales	Area	Maximum detection	IoU
ARmax=1	All	1	0.50:0.95	ARs	Small object (area<32^2)	100	0.50:0.95
ARmax=10	All	10	0.50:0.95	ARm	Medium object (32^2<area<96^2)	100	0.50:0.95
ARmax=100	All	100	0.50:0.95	ARl	Large object (area>96^2)	100	0.50:0.95

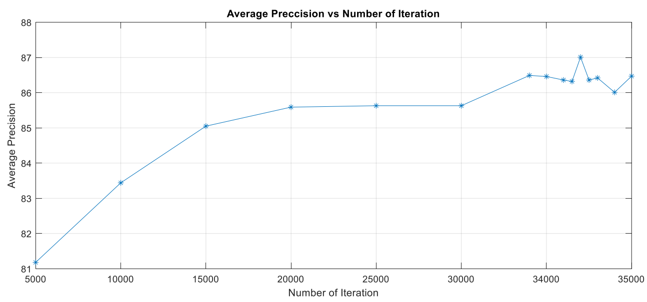


Figure 4. Evaluation results on validation using AP metric against iterations

Table 3. COCO evaluation metrics using Faster R-CNN X101-FPN (iteration 37,000)

IoU	Area	Maxdets	AP	IoU	Area	Maxdets	AR
0.50:0.95	All	100	87.01	0.50:0.95	All	1	59.2
0.50	All	100	95.1	0.50:0.95	All	10	89.8
0.75	All	100	92.95	0.50:0.95	All	100	90.7
0.50:0.95	Small	100	31.25	0.50:0.95	Small	100	58
0.50:0.95	Medium	100	58.17	0.50:0.95	Medium	100	65.9
0.50:0.95	Large	100	89.8	0.50:0.95	Large	100	93.1

(a) (b) (c) (d)

Figure 5. Results on test images: (a) healthy leaves [39], (b) diseased leaves [40], (c) healthy and diseased leaves [41], and (d) diseased leaves in *Kebun Ibu & Co*

4. CONCLUSION

The tomato leaf disease recognition system, based on the Faster R-CNN X101-FPN model architecture using the Detectron2 framework, was trained on a dataset consisting of 10,047 images containing various diseases. The model was trained for 37,000 iterations, resulting in the best AP of 87.01. This system can detect various diseases in tomato leaves, including early blight, late blight, leaf mold, mosaic virus, septoria, spider mites, yellow leaf curl virus, leaf miner, and healthy tomato leaves. Although the model provides satisfactory results, the tomato leaf disease recognition system can be further developed to detect other diseases on tomato leaves as well as symptoms of nutrient deficiency. Additionally, the tomato leaf disease recognition system can also be implemented into mobile applications.

ACKNOWLEDGMENTS

The authors would like to express their sincere gratitude and appreciation to the Institute for Research and Community Development, Atma Jaya Catholic University of Indonesia for providing internal research grant 0269.7/III/LPPM-PM.10.01/03/2023 that supported this activity. This research grant played a crucial role in the successful completion of this research. Additionally, the authors would also like to acknowledge and thank Mr. Arie Hartanto, *Kebun Ibu & Co* for their support and cooperation, which made this research possible.

REFERENCES

[1] H. Nazki, S. Yoon, A. Fuentes, and D. S. Park, "Unsupervised image translation using adversarial networks for improved plant disease recognition," *Computers and Electronics in Agriculture*, vol. 168, p. 105117, Jan. 2020, doi: 10.1016/j.compag.2019.105117.
[2] M. A. Azim, M. K. Islam, M. M. Rahman, and F. Jahan, "An effective feature extraction method for rice leaf disease classification," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 19, no. 2, pp. 463–470, Apr. 2021, doi: 10.12928/TELKOMNIKA.v19i2.16488.
[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.
[4] R. Girshick, "Fast R-CNN," *arXiv preprint arXiv:1504.08083*, 2015.
[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in Neural Information Processing Systems*, pp. 91–99, 2015.
[6] M. Carranza-García, J. Torres-Mateo, P. Lara-Benítez, and J. García-Gutiérrez, "On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data," *Remote Sensing*, vol. 13, no. 1, pp. 1–23, Dec. 2021, doi: 10.3390/rs13010089.
[7] J. Huang *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, IEEE, Jul. 2017, pp. 3296–3305, doi: 10.1109/CVPR.2017.351.
[8] L. A. Varga, S. Koch, and A. Zell, "Comprehensive Analysis of the Object Detection Pipeline on UAVs," *Remote Sensing*, vol. 14, no. 21, p. 5508, Nov. 2022, doi: 10.3390/rs14215508.
[9] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, "Comparative analysis of deep learning image detection algorithms," *Journal of Big Data*, vol. 8, no. 1, p. 66, Dec. 2021, doi: 10.1186/s40537-021-00434-w.

[10] Kaggle, "PlantVillage Dataset," *kaggle.com*. https://www.kaggle.com/datasets/emmarex/plantdisease (accessed Jul. 06, 2023).
[11] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," 2015, [Online]. Available: http://arxiv.org/abs/1511.08060
[12] R. Chaerani and R. E. Voorrips, "Tomato early blight (Alternaria solani): The pathogen, genetics, and breeding for resistance," *Journal of General Plant Pathology*, vol. 72, no. 6, pp. 335–347, Dec. 2006, doi: 10.1007/s10327-006-0299-3.
[13] S. Z. M. Zaki, M. A. Zulkifley, M. M. Stofa, N. A. M. Kamari, and N. A. Mohamed, "Classification of tomato leaf diseases using mobilenet v2," *IAES International Journal of Artificial Intelligence*, vol. 9, no. 2, pp. 290–296, Jun. 2020, doi: 10.11591/ijai.v9.i2.pp290-296.
[14] Z. Yan, A. M. A. Wolters, J. Navas-Castillo, and Y. Bai, "The global dimension of tomato yellow leaf curl disease: Current status and breeding perspectives," *Microorganisms*, vol. 9, no. 4, doi: 10.3390/microorganisms9040740.
[15] C. K. Tettey *et al.*, "Tomato mottle mosaic virus: Characterization, resistance gene effectiveness, and quintuplex RT-PCR detection system," *Journal of Integrative Agriculture*, vol. 21, no. 9, pp. 2641–2651, 2022, doi: 10.1016/j.jia.2022.07.020.
[16] A. T. Nieuwenhuizen, J. Kool, H. K. Suh, and J. Hemming, "Automated spider mite damage detection on tomato leaves in greenhouses," *Acta Horticulturae*, no. 1268, pp. 165–172, Jan. 2020, doi: 10.17660/ActaHortic.2020.1268.21.
[17] S. Mustapha, A. K. Musa, and A. S. Aliyu, "Assessment of leaf miner Liriomyza spp (diptera: Agromyzidae) damage on tomato and the yield output in Ilorin, Southern Guinea Savannah, Nigeria," *Agrosearch*, vol. 20, no. 2, pp. 1–9, Oct. 2021, doi: 10.4314/agrosh.v20i2.1.
[18] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2 model zoo," *github.com*, 2019. https://github.com/facebookresearch/detectron2 (accessed Jul. 06, 2023).
[19] Y. Zhang, C. Song, and D. Zhang, "Deep Learning-Based Object Detection Improvement for Tomato Disease," *IEEE Access*, vol. 8, pp. 56607–56614, 2020, doi: 10.1109/ACCESS.2020.2982456.
[20] A. Fuentes, S. Yoon, S. C. Kim, and D. S. Park, "A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition," *Sensors (Switzerland)*, vol. 17, no. 9, p. 2022, Sep. 2017, doi: 10.3390/s17092022.
[21] S. Adhikari, E. KC, L. Balkumari, B. Shrestha, and B. Baiju, "Tomato Plant Diseases Detection System Using Image Processing," *Researchgate.Net*, vol. I, pp. 81–86, 2018.
[22] Q. Wang and F. Qi, "Tomato diseases recognition based on faster RCNN," in *Proceedings - 10th International Conference on Information Technology in Medicine and Education, ITME 2019*, IEEE, Aug. 2019, pp. 772–776, doi: 10.1109/ITME.2019.00176.
[23] I. Kumar, A. Bisht, and J. Rawat, "Identification of tomato plant disease using faster R-CNN and RetinaNet," *Convergence of Cloud Computing, AI, and Agricultural Science*, pp. 306–327, 2023, doi: 10.4018/979-8-3693-0200-2.ch016.
[24] M. Alruwaili, M. H. Siddiqi, A. Khan, M. Azad, A. Khan, and S. Alanazi, "RTF-RCNN: An Architecture for Real-Time Tomato Plant Leaf Diseases Detection in Video Streaming Using Faster-RCNN," *Bioengineering*, vol. 9, no. 10, p. 565, Oct. 2022, doi: 10.3390/bioengineering9100565.
[25] R. Budiarsa, R. Wardoyo, and A. Musdholifah, "Face recognition for occluded face with mask region convolutional neural network and fully convolutional network: a literature review," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 5, pp. 5662–5673, Oct. 2023, doi: 10.11591/ijece.v13i5.pp5662-5673.
[26] S. Bouraya and A. Belangour, "Deep Learning based Neck Models for Object Detection: A Review and a Benchmarking Study," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, pp. 161–167, 2021, doi: 10.14569/IJACSA.2021.0121119.
[27] A. Rahim, A. Maqbool, and T. Rana, "Monitoring social distancing under various low light conditions with deep learning and a single motionless time of flight camera," *PLoS ONE*, vol. 16, p. e0247440, Feb. 2021, doi: 10.1371/journal.pone.0247440.
[28] C. R. de Lima, S. G. Khan, S. H. Shah, and L. Ferri, "Mask region-based CNNs for cervical cancer progression diagnosis on pap smear examinations," *Heliyon*, vol. 9, no. 11, Nov. 2023, doi: 10.1016/j.heliyon.2023.e21388.
[29] I. Valova, C. Harris, N. Gueorguieva, and T. Mai, "In-Between Layers Modular Residual Neural Network for the Classification of Images," *Procedia Computer Science*, vol. 185, pp. 223–230, 2021, doi: 10.1016/j.procs.2021.05.023.
[30] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, IEEE, Jul. 2017, pp. 5987–5995, doi: 10.1109/CVPR.2017.634.
[31] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, IEEE, Jul. 2017, pp. 936–944, doi: 10.1109/CVPR.2017.106.
[32] S. Saponara and A. Elhanashi, "Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance," in *Lecture Notes in Electrical Engineering*, 2022, pp. 10–17, doi: 10.1007/978-3-030-95498-7_2.
[33] C. N. Vasconcelos and B. N. Vasconcelos, "Experiments using deep learning for dermoscopy image analysis," *Pattern Recognition Letters*, vol. 139, pp. 95–103, Nov. 2020, doi: 10.1016/j.patrec.2017.11.005.
[34] V. C. S. R. Rayavarapu and A. Mahapatro, "NLOS identification and mitigation in UWB positioning with bagging-based ensembled classifiers," *Annales des Telecommunications/Annals of Telecommunications*, vol. 77, no. 5–6, pp. 267–280, Jun. 2022, doi: 10.1007/s12243-021-00884-6.
[35] A. Suljovic, S. Cakic, T. Popovic, and S. Sandi, "Detection of Plant Diseases Using Leaf Images and Machine Learning," *2022 21st International Symposium INFOTEH-JAHORINA, INFOTEH 2022 - Proceedings*, 2022, doi: 10.1109/INFOTEH53737.2022.9751245.
[36] API Documentation, "detectron2.config," *detectron2.readthedocs.io*. https://detectron2.readthedocs.io/en/latest/modules/config.html (accessed Jul. 05, 2023).
[37] A. Wenda, I. Permana, Yusmar, and N. N. Kurniawati, "Identification of paddy leaf diseases based on texture analysis of blobs and color segmentation," *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 4, pp. 2018–2026, Aug. 2020, doi: 10.12928/TELKOMNIKA.V18I4.14614.
[38] COCO, "COCO Evaluation," *cocodataset.org*. https://cocodataset.org/#detection-eval (accessed Jul. 06, 2023).
[39] S. N. U. of Science and Technology, "json2txt Dataset," *universe.roboflow.com*. https://universe.roboflow.com/seoul-national-university-of-science-and-technology-dgcyc/json2txt-wj58i (accessed Jul. 06, 2023).
[40] Risingsun, "tomato_disease Dataset," *universe.roboflow.com*. https://universe.roboflow.com/risingsun-b9rl6/tomato_disease-bc3ye (accessed Sept. 03, 2023).
[41] V. N. University, "dataset_split Dataset," *universe.roboflow.com*. (accessed Jul. 06, 2023).

BIOGRAPHIES OF AUTHORS

Karel Octavianus Bachri is a member of Atma Jaya Catholic University of Indonesia. He obtained his Doctorate in Intelligent Instrumentation in 2019. His research interests are intelligent instrumentation and system on chip (SoC). He was also involved in several artificial intelligence projects to design cognitive instrumentation systems. Currently, he is working on intelligent monitoring and control systems for agricultural sustainability. He can be contacted at email: karel.bachri@atmajaya.ac.id.

Bryan Santoso is a student at the Electrical Engineering, Atma Jaya Catholic University of Indonesia. He has a strong interest in machine learning, programmable logic controller (PLC), and internet of things (IoT). He can be contacted at email: bryan.201904520015@student.atmajaya.ac.id.

Duma Kristina Yanti Hutapea is currently a faculty member at Atma Jaya Catholic University of Indonesia in Jakarta. Her expertise lies in neuro signal processing, biomedical engineering, and telecommunication. She completed her bachelor's degree in electrical engineering from the University of Indonesia in 2001. She then pursued further studies at Universiti Teknologi PETRONAS in Malaysia, earning a Master of Science (M.Sc.) degree in 2010 and a Doctorate in 2018. During her doctoral studies in Malaysia, she was assigned as a research scientist at Universiti Teknologi PETRONAS for a period of 3 years, from 2015 to 2018. After completing her doctoral studies and her role as a research scientist, she joined the faculty of the Master's Program in Electrical Engineering at Atma Jaya Catholic University. She can be contacted at email: duma.kristina@atmajaya.ac.id.

Catherine Olivia Sereati is a lecturer and researcher at the Atma Jaya Catholic University of Indonesia. Catherine's interest subjects of research are Electronic Instrumentation Systems and System on Chip (SoC). She was also involved in several research projects to design cognitive instrumentation systems. Some of them are building a software cognitive interpretation of ship movements, for Indonesian marine security purposes, and cognitive electro cardiograph (ECG) design. Currently, her research project is focusing on designing the architecture of Cognitive processors. She can be contacted at email: catherine.olivia@atmajaya.ac.id.

Lanny W. Pandjaitan is a lecturer at the Faculty of Engineering, Atma Jaya Catholic University of Indonesia. She completed her Bachelor's degree at ITB Faculty of Industrial Technology, Electrical Engineering in 1973, ITB Master's Program, STEI, Electrical Engineering, in 1994, and Doctoral Program at ITB, STEI, Electrical Engineering, in 2000. She is one of the Professors of Engineering Electrical at Atma Jaya Catholic University of Indonesia. Her area of expertise is intelligent systems based electronic instrumentation. She can be contacted at email: mm.lanny@atmajaya.ac.id.