Detecting Pests and Diseases in Plants Using Efficient Network

Timur Haryo Mahissanular[a1], Mardhiya Hayaty[a2]

[a]Computer Science, Universitas Amikom Yogyakarta
Yogyakarta, Indonesia
[1]timur.m@students.amikom.ac.id
[2]mardhiya_hayati@amikom.ac.id (Corresponding author)

Abstract

The agricultural sector in Indonesia is still faced with low agrarian production caused by pests and diseases. Therefore, agricultural land that is still vulnerable to pests but can detect the development of pest attacks must be designed. This study uses the PlantVillage dataset. The dataset will go through the preprocessing stage for dimension adjustment, and then the result will be used for building the network. The results are evaluated using a confusion matrix and showed that the convolutional neural network performs well in image processing and obtains architectural optimization in its field. The method we propose is an Efficient Network by selecting the correct input size. Implementing an Efficient Network in the convolutional neural network architecture increases its F1-score to 93%, indicating that Efficient Network has a higher F1-Score than the baseline convolution neural network. Implementing this network architecture can quickly increase the CNN baseline to a more varied target resource while maintaining the efficiency of the resulting model.

Keywords: *Image Recognition, Convolutional Neural Network, Efficient Network, Pre-trained Model, Sequential*

1. Introduction

The agricultural sector in Indonesia contributes significantly to the development of the Indonesian economy. The contribution of the agricultural sector to the Gross Domestic Product (GDP), according to price, prevailed in 2020 by 13.70%, an increase of 0.99% compared to the previous year[1]. However, the agricultural sector is still vulnerable to diseases and pests. If this is not addressed immediately, agricultural production becomes less than optimal. For this reason, the development approach in the agricultural sector needs to change from a conventional approach to a digital system.

In infield practice, it is difficult to distinguish the diseases and pests that attack plants. The science of studying every disease and pest is a must-have thing. But this is also not easy to obtain. Information, including plant diseases and pests, can be more accessible in this advanced era. Image classification technology has been widely used in various fields, including agriculture. Most image classifications use the Convolutional Neural Network algorithm.

There are breakthroughs in image labeling, object detection, and image classification [2], [3]. Neural network systems have shown performance breakthroughs in the area of object detection and image classification [4]–[6], specifically in the Convolutional Neural Network (CNN). This research focuses on identifying the best neural networks for optimizing CNN. CNN is a classification algorithm that uses deep neural networks in processing [7], [8]. This algorithm is quite effective but can improve performance for accuracy and efficiency [9].

Convolutional Neural Network algorithm has been done extensively in the image classification process. This algorithm works through layers with different roles and calculations [10]. These neural network layers must be created according to input data [11]. With varying neural network layers, CNN implementation takes time to find the appropriate neural network composition for the case at hand. In addition, the main obstacle to this algorithm is the lack of ability to process images

that are too large [4]. This is mainly due to memory constraints, hardware types, and relatively small training data.

Optimization can be done on Convolutional Neural Network to get a better model and cheaper resources. One way to optimize is to transfer learning or utilize pre-trained models, as was done in research [6], which produced a very significant performance compared to CNN without transfer learning. The learning transfer method can also reduce the problem of overfitting the resulting model. In addition, other pre-trained models can be used for transfer learning, which uses the Efficient Network. This architecture can be applied to improve transfer learnings and is more efficient than most of its previous architectures, such as Residual Network (for example, ResNet50), GoogLeNet (for example, InceptionV3), and VGG (for example, VGG16 or VGG19) [12]. In addition, the use of EfficientNet can save time and computing power. So, this gives a higher accuracy value than other pre-trained models. This high accuracy level uses a practical scale at depth, length, and width [13].

Pre-trained models from the Efficient Network are divided into eight models from B0 to B7, where the higher model will require more parameters and high-resolution input [9]. The image data collected has a relatively low resolution of 256 x 256. Based on the research results conducted [9], the optimization carried out in this study will use EfficientNet-B0.

2. Research Method

Many researchers have researched image classification using CNN. This research leads to optimizing the CNN algorithm by using the pre-trained model, the Efficient Network. This method is expected to be used to support agricultural production. Furthermore, machine learning technology can accelerate the analysis of diseases or pests that attack plants.

Research [14] shows the automatic detection of diseases in plants using pure CNN and tomato plant images. The accuracy achieved was 93.26% with epoch 30. Other studies were also conducted [15] using more general plant disease data. The resulting model has an accuracy of 88.7%. Both experiments that have been carried out require a longer training time to get accuracy above 85%. The researcher also mentioned that the number of data used was still limited.

Another approach is through deep learning models to improve accuracy in multiclass classification has been carried out in previous research [16]. The proposed CNN was formed in the VGG classifier and AlexNetOWTBn architecture. The dataset used has 87,848 images of healthy and infected plant leaves. The pictures used in this study have 58 classes, including healthy plants. The resulting model has undergone several stages, such as preprocessing, which converts the size image input to 256x256. In addition, the study only divided data into 2, namely training data and testing data, with a ratio of 4:1. The resulting accuracy reaches 99.53%. The performance of the resulting model can be poorly charged because the researcher explains that the data used for testing is data duplication of training data.

In other studies, there was the optimization of CNN using GPipe. GPipe increasingly encourages validation accuracy that utilizes ImageNet with up to 84.3% accuracy using the 557M parameter [17]. It is so large that it can only be processed using a parallelism library by creating special channels by distributing the network and spreading each section to different accelerators. Such models are built for ImageNet, and the latest research shows that better ImageNet models perform better in other dataset learning transfers [18]. Although the accuracy is high, researchers explain that their research has taken high-memory hardware sources.

Research has been carried out by utilizing the Efficient Network. The Efficient Network showed the highest performance in research [19]. Optimization of the CNN algorithm has also been done in agriculture. In a study [20], CNN image classification experiments were conducted to optimize EfficientNet B4 and B5. Data used is from original image data and augmented image data. The accuracy results obtained for the original image reached 99.91% and for augmented data by 99.97%.

Among the studies conducted above, there is an improvement in performance and efficiency in image classification and segmentation. But in real cases, the images obtained will be a little complicated to be extracted and model. Therefore, efficient computing and faster training data are

needed in optimizing existing algorithms. This study proposes implementing the Efficient Network to classify plant pests and diseases compared to the Convolutional Neural Network in general.

2.1 Convolutional Neural Network

Convolutional Neural Network performs well, especially in image processing, and has received architectural optimization in this field. This algorithm is similar to a neural network, but the main difference lies in the hidden layer. Neurons in this hidden layer are only connected to a subset of neurons in the previous layer. These networks are interconnected so that they can learn their functions implicitly. Network architecture leads to a feature extraction hierarchy. Filters produced from the first layer, also called trained filters, can be visualized as a collection of color matrices. The second layer can be visualized as a collection of shapes, the next layer filter might study the object part, and the last filter may identify the object.

2.2 Efficient Network

The development of CNN architecture depends on available resources, and adjustments occur to attain improved performance when its resources also increase. The new basic architecture called the Efficient Network is designed and enhanced to make compound adjustments between CNN's basic architecture and the resources used [21]. A new basic architecture called EfficientNetB0 was initially designed and scaled using a composite scaling scheme to produce the EfficientNet family. There are eight types of Efficient Network models, namely EfficientNet-B0 to EfficientNet-B7. The traditional way to scale a model is to increase the width or depth of the CNN or the resolution of the input image, which has been done arbitrarily. Scaling networks systematically improve the model's performance to balance all the combined architectural width, depth, and image resolution coefficients. Figure 1 shows the scaling networks in Efficient Network.

Table 1. Default Input Size of Images in Efficient Network Model

Model	Input Size
EfficientNet-B0	224 x 224
EfficientNet-B1	240 x 240
EfficientNet-B2	260 x 260
EfficientNet-B3	300 x 300
EfficientNet-B4	380 x 380
EfficientNet-B5	456 x 456
EfficientNet-B6	528 x 528
EfficientNet-B7	600 x 600
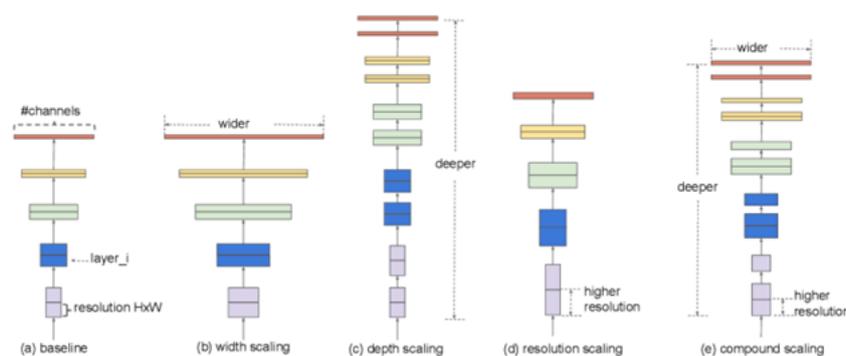
Figure 1. Scaling in Efficient Network

2.3 Imbalanced Data

Various researchers with various approaches have researched imbalanced data. [22]–[24] addresses the problem of imbalanced data at the algorithm level. The problem of imbalanced classes in machine learning explains classification tasks where data and related classes are not represented equally. So the ability of classification algorithms is disrupted by this imbalance [25].

Imbalanced data has a ratio used to see data distribution in one class with another. This ratio can be found by determining the minority class (N-) divided by the majority class (N+). If the majority class is greater than the minority class or the division results are less than one, then the data is classified as imbalanced.

$$IR = \frac{N_-}{N_+}(1)$$ (1)

2.4 Dataset

In this work, the PlantVillage dataset had 15 classes and 20,638 images of 3 types of plants, namely pumpkins, potatoes, and tomatoes. Ideally, the dataset is divided into 3; those are training data, validation data, and testing data [26]. Figure 2 shows 15 samples of each class in this dataset. This dataset has a color image and an exact resolution of 256 x 256.

The distribution of the dataset has a ratio of 8:1:1 for training, validation, and testing data. However, the ratio between the highest and last classes is 0.05, so the dataset is imbalanced.
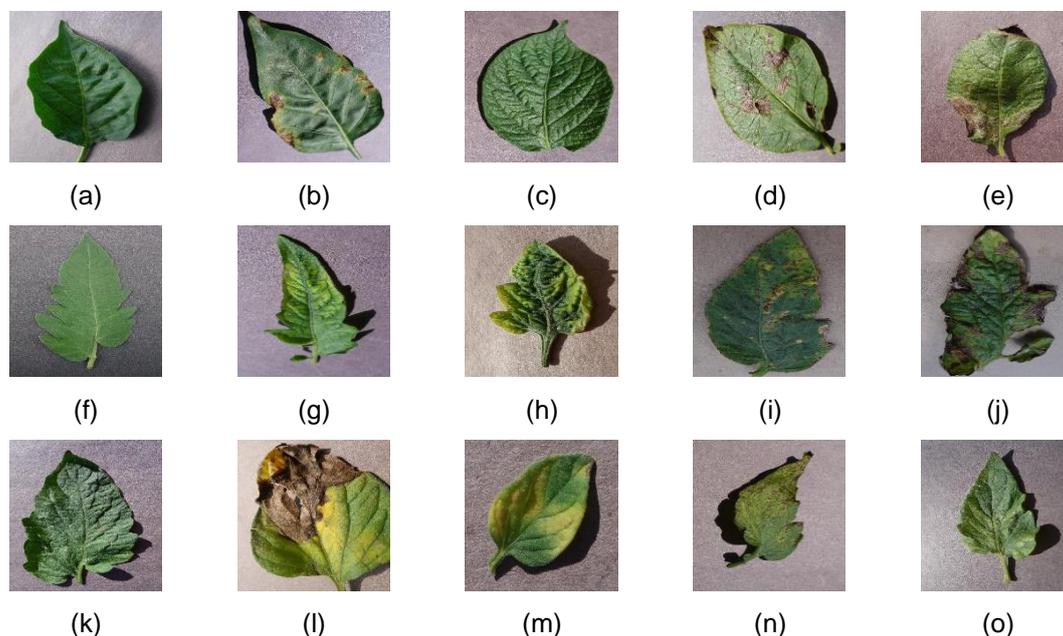
Figure 2. Samples plant disease: (a) Pepper ball healthy, (b) Pepper ball bacterial spot, (c) Potato healthy, (d) early Potato blight, (e) Potato late blight, (f) Tomato healthy, (g) Tomato mosaic virus, (h) Tomato yellow leaf curl virus, (i) Tomato bacterial spot, (j) Tomato early blight, (k) Tomato target spot, (l) Tomato late blight, (m) Tomato leaf mold, (n) Tomato septoria leaf spot, (o) Tomato spider mites two-spotted spider mite

2.5 Preprocessing

Preprocessing is done through 2 stages for the PlantVillage dataset before splitting, which will be processed. The first stage is scaling the picture. The image will be adjusted to the resolution to equate the existing dataset with the efficient net-B0 input size. The results of this image scaling will change the image's dimensions from 256 x256 to 224 x 224, corresponding to the EfficientNet-B0 model size input. Balancing all network dimensions against available resources will improve overall performance [27].

This study uses ImageDataGen from hard-to-apply dataset augmentation. Preprocessing in this study applied augmentation, which would enhance the variation of the image so that the resulting accuracy was also upgraded [28]. At this stage, the model will get richer and save overhead on memory when training is being carried out.
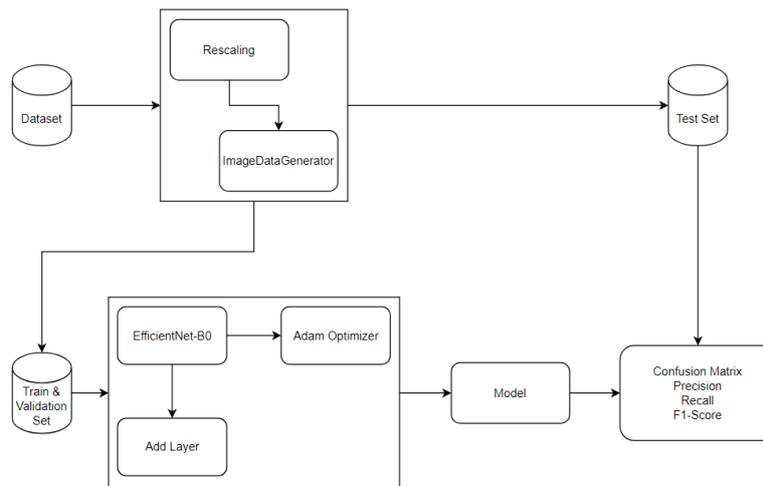
Figure 3. Block diagram of proposed work

2.6 Proposed CNN

This research uses EfficientNet-B0 for the learning transfer process. In addition, a Convolutional 2D layer applies additional filters in the training process. This extra filter can be used to find out the edges of the output. The following additional layer is Max Pooling 2D to decrease the dimension of the model's feature map without losing essential information from the matrix [29].

Flatten layers are added before fully-connected layers to convert the output from previous layers into vectors with single extended features. This vector will produce the final classification matching the existing classes' numbers. A Dense layer is added sequentially for fully-connected layers and uses the RELU activation function. The added layer output is 1 Dense layer with 15 output units according to the class number of the current data class. The added layer output has a SOFTMAX activation function. The schematic representation of the proposed EfficientNet B0 model is shown in Figure 4.

Figure 4. Schematic representation of Proposed EfficientNet

Details about the layers formed and the order of the proposed model, the output shapes of every layer, the number of parameters in each layer, and the total parameters on the model are presented in Table 2.

Table 2. Layers type and parameters used in the proposed model

Layer (type)	Output Shape	Param #
EfficientNetB0	(None, 7, 7 1280)	4.049.571
conv2d	(None, 5, 5, 16)	184.336
max_pooling2d	(None, 2, 2, 16)	0
flatten	(None, 64)	0
dense	(None, 64)	4160
dense_1	(None, 15)	975

Total Parameters: 4.239.042
Trainable Parameters: 189.471
Non-trainable Parameters: 4.049.571

2.7 Evaluation

A dataset containing image samples not applied during training has validated the model's performance. The confusion matrix of the produced model has also been extracted. Therefore, every class in the dataset has also calculated its precision, recall, and F1 score. F1-score will be a comparison between CNN's baseline performance and EfficientNet-B0.

3. Result and Discussion

Experiments were carried out on local computers using the Cuda GPU. This work uses two experiment scenarios; first, using CNN architecture, and the second operating the EfficientNet-B0 model. This experiment was carried out by running ten epochs. Each epoch has a batch of 32. A total of 2 CNN architectures have been created and used for the same treatment at the preprocessing stage.

3.1 Experiments, results, and analysis

This study shows that applying the EfficientNet-B0 model in CNN architecture improves its evaluation metric, the F1 score. Table 3 compares the F1 scores for each class. The results indicate that the Efficient Network model had a higher F1 score in almost every category than the CNN model.

Table 3. F1-score percentage result from CNN and EfficienNet-B0 model

Name	CNN	EfficientNet-B0
Pepperbell Bacterial Spot	74%	96%
Pepperbell Healthy	92%	98%
Potato Early Blight	89%	99%
Potato Late Blight	76%	93%
Potato Healthy	53%	87%
Tomato Bacterial Spot	91%	95%
Tomato Early Blight	63%	79%
Tomato Late Blight	74%	91%
Tomato Leaf Mold	73%	90%
Tomato Septoria Leaf Spot	70%	90%
Tomato Spidermites Two Spotted Spidermite	74%	93%
Tomato Target Spot	75%	87%
Tomato Tomato YellowLeaf Curl Virus	94%	98%
Tomato Tomato Mosaic Virus	74%	89%
Tomato Healthy	96%	96%

In the case of Tomato Healthy Class, the resulting scores between baseline CNN and EfficientNet were the same. Both architectures used sequential models, so the output model from the previous layer was transferred and processed in the next layer. Additionally, the visualization for Tomato Healthy appears to be relatively flat. This means that the number of parameters extracted to make the model, both in the case of the baseline CNN and EfficientNet-B0, remained relatively the same [30].

To compare the output models further, this study used a confusion matrix to evaluate the results of the predictions for both proposed architectures. Table 4 compares the average True Positive Rate (TPR), the average False Positive Rate(FPR), and the average False Negative Rate (FNR) between the baseline CNN and EfficientNet-B0 models.

Table 4. TPR, FPR, and FNR based on the confusion matrix

Architecture	Average TPR	Average FPR	Average FNR
Baseline CNN	0.738	0.017	0.262
EfficientNet-B0	0.923	0.005	0.082

Using a confusion matrix, the evaluation took the True Positive Rate, False Positive Rate, and False Negative Rate, as indicated by the label above. The True Positive Rate shows the prediction ratio using data not included in the training process, that is, the testing dataset. The False Positive Rate presents prediction results that should be wrong but were falsely determined to be correct by the machine. Finally, the False Negative Rate refers to those instances in which the device detects a prediction result as an incorrect value even though it is, in fact, correct.

The architecture EfficientNet-B0 can reduce the prediction error rate compared to CNN. For the False Positive Rate, the decrease was not very significant compared to the False Negative Rate. This is because an imbalanced dataset was used during the training process. This imbalance greatly influenced the training process since it caused the machine to have difficulties getting the correct data sample and the corresponding class.

4. Conclusion

This study teaches how plant diseases and pests can be appropriately detected through image processing despite having low image resolution. This study proposes a simple but effective method of applying the Efficient Network to the Convolutional Neural Network to cultivate these results. Using this network architecture can quickly increase CNN's baseline towards more variable resource targets while maintaining the efficiency of the resulting model. The results of this study also prove that the application of EfficientNet-B0 can improve the model's efficiency from the CNN baseline and produce an F1-score of 93%.

References

[1] S. S. T. Agung Setyo Wibowo, "Indikator Pertanian 2020", 2021. [Online]. Available: https://www.bps.go.id/.

[2] L. J. Li, H. Su, Y. Lim, and L. Fei-Fei, "Objects as attributes for scene classification" *Lecture Notes in Computer Science*, vol. 6553 LNCS, no. PART 1, pp. 57–69, 2012, doi: 10.1007/978-3-642-35749-7_5.

[3] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A Review on Deep Learning Techniques Applied to Semantic Segmentation" pp. 1–23, 2017, [Online]. Available: http://arxiv.org/abs/1704.06857.

[4] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. S. Kruthiventi, and R. V. Babu, "A taxonomy of deep convolutional neural nets for computer vision" *Frontiers in Robotics and AI*, vol. 2, no. JAN, pp. 1–13, 2016, doi: 10.3389/frobt.2015.00036.

[5] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs" in *3rd International Conference Learning Representative ICLR 2015* [Online]. Available: file:///D:/COLLAGE/6 Semester/Skripsi/ref/Nov-w1/Object Detectors Emerge in Deep Scene CNNs.pdf.

[6] W. Li *et al.*, "Classification of High-Spatial-Resolution Remote Sensing Scenes Method Using Transfer Learning and Deep Convolutional Neural Network" *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2020, doi: 10.1109/JSTARS.2020.2988477.

[7] D. K. Das *et al.*, "Computer aided tool for automatic detection and delineation of nucleus from oral histopathology images for OSCC screening" *Applied Soft Computing*, Vol.83, 2019, doi: 10.1016/j.asoc.2019.105642.

[8] C. Dong, C. C. Loy, and X. Tang, "Accelerating the Super - Resolution Convolutional Neural Network SR for Different Upscaling Factors" *Eccv*, 2016.

[9] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks" in *36th International Conference on Machine Learning, ICML 2019*, 2019, vol.

2019-June, pp. 10691–10700.

[10] Y. Toda and F. Okura, "How convolutional neural networks diagnose plant disease" *Plant Phenomics*, vol. 2019, 2019, doi: 10.34133/2019/9237136.

[11] N. Sharma, V. Jain, and A. Mishra, "An Analysis of Convolutional Neural Networks for Image Classification" *Procedia Computer Science*, vol. 132, no. Iccids, pp. 377–384, 2018, doi: 10.1016/j.procs.2018.05.198.

[12] M. Tan and Q. V Le, "EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling" *The latest from Google Research*, 2019.

[13] G. Marques, D. Agarwal, and I. de la Torre Díez, "Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network" *Applied Soft Computing*, vol. 96, p. 106691, November 2020, doi: 10.1016/j.asoc.2020.106691.

[14] P. Tm, A. Pranathi, K. Saiashritha, N. B. Chittaragi, and S. G. Koolagudi, "Tomato Leaf Disease Detection Using Convolutional Neural Networks" in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 2–4, 2018, doi: 10.1109/IC3.2018.8530532.

[15] M. Francis and C. Deisy, "Disease Detection and Classification in Agricultural Plants Using Convolutional Neural Networks - A Visual Understanding" in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pp. 1063–1068, 2019, doi: 10.1109/SPIN.2019.8711701.

[16] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis" *Computers and Electronics in Agriculture*, vol. 145, no. September 2017, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[17] Y. Huang *et al.*, "GPipe: Efficient training of giant neural networks using pipeline parallelism" *NIPS'19: Proceedings of the 33rd International Conference on Neural Information Processing Systems*, [Online]. Available: https://dl.acm.org/doi/pdf/10.5555/3454287.3454297.

[18] S. Kornblith, J. Shlens, and Q. V Le, "Do Better ImageNet Models Transfer Better?" in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2661–2671, 2019, doi: 10.1109/CVPR.2019.00277.

[19] J. Wang, Q. Liu, H. Xie, Z. Yang, and H. Zhou, "Boosted efficientnet: Detection of lymph node metastases in breast cancer using convolutional neural networks" *Cancers (Basel)*, vol. 13, no. 4, pp. 1–14, February 2021, doi: 10.3390/cancers13040661.

[20] Ü. Atila, M. Uçar, K. Akyol, and E. Uçar, "Plant leaf disease classification using EfficientNet deep learning model" *Ecological Informatics*, vol. 61, p. 101182, March 2021, doi: 10.1016/j.ecoinf.2020.101182.

[21] B. Baheti, S. Innani, S. Gajre, and S. Talbar, "Eff-UNet: A novel architecture for semantic segmentation in unstructured environment" in *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, vol. 2020-June, pp. 1473–1481, 2020, doi: 10.1109/CVPRW50498.2020.00187.

[22] P. Chujai, K. Chomboon, P. Teerarassamee, N. Kerdprasop, and K. Kerdprasop, "Ensemble Learning For Imbalanced Data Classification Problem" in *International Conference on Industrial Application Engineering*, January 2015, pp. 449–456, 2015, doi: 10.12792/iciae2015.079.

[23] H. J. Xing and W. T. Liu, "Robust AdaBoost based ensemble of one-class support vector machines" *Information Fusion*, vol. 55, no. July 2019, pp. 45–58, 2020, doi: 10.1016/j.inffus.2019.08.002.

[24] B. Krawczyk, A. Cano, and M. Wozniak, "Selecting local ensembles for multi-class imbalanced data classification" in *International Joint Conference on Neural Networks (IJCNN)*, vol. 2018-July, 2018, doi: 10.1109/IJCNN.2018.8489572.

[25] G. Douzas, F. Bacao, and F. Last, "Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE", *Information Sciences*, vol. 465, pp. 1–20, 2018, doi: 10.1016/j.ins.2018.06.056.

[26] G. Geetharamani and A. P. J., "Identification of plant leaf diseases using a nine-layer deep convolutional neural network" *Computers & Electrical Engineering*, vol. 76, pp. 323–338, 2019, doi: 10.1016/j.compeleceng.2019.04.011.

[27] G. Rostami, M. Hamid, and H. Jalaeikhoo, "Impact of the BCR-ABL1 fusion transcripts on different responses to Imatinib and disease recurrence in Iranian patients with Chronic Myeloid Leukemia" *Gene*, vol. 627, no. 1, pp. 202–206, 2017, doi: 10.1016/j.gene.2017.06.018.

[28] A. A. J. E. V. Priyangka and I. M. S. Kumara, "Classification Of Rice Plant Diseases Using the Convolutional Neural Network Method" *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, vol. 12, no. 2, p. 123, 2021, doi: 10.24843/lkjiti.2021.v12.i02.p06.

[29] I. M. Mika Parwita and D. Siahaan, "Classification of Mobile Application Reviews using Word Embedding and Convolutional Neural Network" *Lontar Komputer: Jurnal Ilmiah Teknologi Informasi*, vol. 10, no. 1, p. 1, 2019, doi: 10.24843/lkjiti.2019.v10.i01.p01.

[30] T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang, "Error-driven incremental learning in deep convolutional neural network for large-scale image classification" *MM '14: Proceedings of the 22nd ACM international conference on Multimedia*, pp. 177–186, 3 November 2014, doi: 10.1145/2647868.2654926.