

# Bell Pepper Leaf Disease Classification Using Fine-Tuned Transfer Learning

Yuris Alkhalifi<sup>a,\*</sup>, Agus Subekti<sup>b,c</sup>

<sup>a</sup> Universitas Bina Sarana Informatika  
Jl. Kramat Raya No.98, Kwitang, Senen  
Jakarta Indonesia

<sup>b</sup> Universitas Nusa Mandiri  
Jl. Kramat Raya No.18  
Jakarta, Indonesia

<sup>c</sup> Research Center for Telecommunications, BRIN  
Jl. Sangkuriang 21/154D  
Bandung, Indonesia

## Abstract

Leaf diseases of plants are common worldwide. Using image processing, farmers could spot diseases in pepper plants more rapidly and get advice from plant disease experts. In this paper, researchers developed a Transfer Learning classification model for bell pepper leaf disease, with the Transfer Learning model trained on images of healthy and diseased bell pepper leaves. Classification of healthy and diseased bell pepper leaves has been carried out, and fine-tuned Transfer Learning has been applied using several pre-trained CNN models. To achieve the best outcome, four pre-trained models, including MobileNet, VGG16, ResNetV250, and DenseNet121, and three Fully Connected (FC) layer architectures were tested. The Fully Connected (FC) layer with four Transfer Learning architectures achieved the best accuracy value of 99.33% on DenseNet121 architecture with one layer and Cohen's Kappa value of 0.9865.

**Keywords:** CNN, Transfer Learning, fine tuning, bell pepper.

## I. INTRODUCTION

Bell Pepper (*Capsicum Annum*. L) belongs to the chili family, similar to beans, corn, and pumpkin. Bell pepper is America's oldest cultivated plant [1]. Bell pepper is a widely grown, nutritious, and delicious vegetable plant with high antioxidants that can reduce the risk of certain human diseases [2]. This plant also has significant economic value to farmers worldwide [3]. Then the farmers have more opportunities to sell their goods locally and internationally due to rising consumer demand [4].

However, similar to other plants, there are many obstacles and challenges, such as bacteria, that are often encountered in controlling these plants. The Bacterium *Xanthomonas campestris* pv. *vesicatoria* are the most common cause of leaf diseases [5], [6]. This bacterium creates bacterial spots in the bell pepper leaves [7]. To prevent a decline in the quantity and quality of bell pepper yield, disease control must be implemented [8]. Effective prevention in plant control involves early diagnosis using identification technologies based on image processing, such as Convolutional Neural Networks (CNN) [9]–[12].

By using image processing, it is beneficial for farmers to detect early symptoms that appear on plants

automatically, thereby reducing large-scale monitoring work [13]. Image processing has led to the discovery of considerably more subtle shortcut characteristics, including high-frequency patterns that are virtually imperceptible to the human eye, and shortcut chances that are relatively straightforward to spot [14]. By relying on automation technology, farmers can also be helped by making decisions regarding productivity and quality, which is currently known as smart farming[15].

Several kinds of research regarding the detection of diseases in leaves use image processing [8], [16], [17] have offered several recommendations for the categorization of bell pepper leaf disease. The datasets from the studies consist of 20 images of leaves, including bell pepper, potatoes, and corn leaves. The number of classes on each leaf varies. The bell pepper leaf disease has two classes: the bacterial class and the healthy class. Each image on this dataset has a resolution of 256×256 pixels [18].

The accuracy of classifying the healthy bell pepper leaf and the bell pepper leaf with bacterial spots using the CNN architecture suggested by the authors in [16] was 96.78%. In another research from [17], the best accuracy for the proposed method was 94.35%. Moreover, in the research from [8], the best accuracy for the proposed method was 96.88%. According to that research, there is still a drawback. The drawback of this research is that it does not use Transfer Learning [19]–[21]. Transfer Learning (TL) allows us to develop a deep CNN more cost-effectively than basic CNN [22].

This research will implement a CNN Transfer Learning-based classification approach for the bell

\* Corresponding Author.

Email: yuris.yak@bsi.ac.id

Received: March 30, 2023 ; Revised: June 16, 2023

Accepted: August 8, 2023 ; Published: August 31, 2023

Open access under CC-BY-NC-SA

© 2023 BRIN

pepper leaf disease with the same dataset. The pre-trained model will be retrained by unfreezing the model and by using fine-tuning [23]–[25]. Fine-tuning makes the parameters can adapt to the images [26]. Three FC layers and four pre-trained models namely MobileNet [27], VGG16 [28], ResNetV250 [29], and DenseNet121 [30] make up the proposed model. Compared to earlier similar research, our approach delivers classification results with improved accuracy. The model's findings are incorporated into a website-based tool, allowing the program to classify new images of pepper leaves immediately. To the best of our knowledge, fine-tuned TL for bell pepper leaf disease classification has not been performed previously.

## II. METHODOLOGY

Figure 1 shows the method performed in our work. It comprises four main steps: data collection, data preprocessing, experiment scenario, and evaluation.

### A. Data Collection

The secondary data used in this work was acquired from the PlantVillage Dataset [12]. Images of various leaves, including tomato, potato, and pepper, may be found in this dataset. However, as this research focused on recognizing images of bell pepper leaves, only images of bell pepper leaves were captured for this research. Table 1 displays the dataset that was utilized.

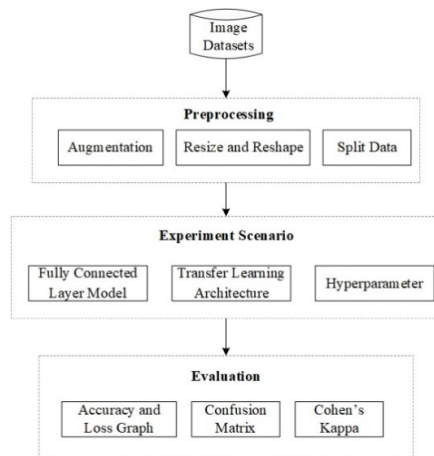




Figure 1. Research methods.

TABLE 1  
IMAGE DATASET

Class Name	Image Example	Number of Images
Bacterial		997 Images
Healthy		1478 Images

### B. Data Preprocessing

Preprocessing is the practice of cleaning, modifying, and rearranging raw data before processing and analysis, also known as data preparation [31]. It includes three steps: image augmentation, image resizing, and dataset split.

#### 1) Image Augmentation

Augmentation can improve the classification accuracy of deep learning algorithms by enhancing existing data rather than acquiring new data [32]. The training data image is used in the augmentation process to produce new images. Preventing overfitting is one of the objectives of augmentation [33], [34]. Each class of 1000 photos will have 2000 images added, with vertical flip = True, horizontal flip = True, zoom range = 0.2, rotation range = 360, width shift range = 0.2, and height change range = 0.2 augmentation techniques used. Now the total images have increased to 4475 images. Table 2 is an illustration of the augmentation outcomes.

#### 2) Image Resize

Image scaling is done to speed up the computer training process. Moreover, it will not demand a lot of storage space [34], [35]. The image dimensions will be changed from 256×256 pixels to 32×32 pixels. Table 3 illustrates the image resizing for the bell pepper leaf.

#### 3) Dataset Split

The data will be conducted in an 8:1:1 data-sharing environment, with 80% for training data, 10% for validation data, and 10% for test data, with 3579 images of training data, 448 images of validation, and 448 images of testing data, as shown in Table 4.

TABLE 2  
EXAMPLE OF IMAGE AUGMENTATION RESULTS








Original Image	
	
<i>rotation range</i>	<i>width shift range</i>
	
<i>height shift range</i>	<i>zoom range</i>
	
<i>horizontal flip</i>	<i>vertical flip</i>
	

TABLE 3  
IMAGE RESIZING RESULT

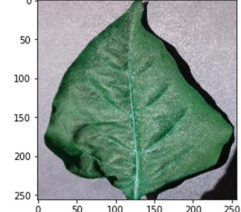
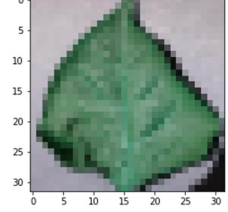
Original Image	Resizing Image
Original Dimensions : (256, 256, 3)	Resized Dimensions : (32, 32, 3)
	

TABLE 4  
DATASET SPLIT

Training Data (80%)	Validation Data (10%)	Testing Data (10%)
3579 Images	448 Images	448 Images
Total number of Images: 4.475		

### C. Experiment

The CNN approach with fine-tuned TL will be used in the experiments to categorize the image of the bell pepper leaves. Figure 2 shows a description of the experiments.

#### 1) Fully Connected Layer Model

This research uses an image with dimensions of  $32 \times 32$  pixels, enabling an iterative process on the image with a Fully Connected (FC) layer. Then, three scenarios are explored on this FC layer: FC with one layer, FC with two layers, and FC with three layers. The Densenet121 architecture is employed for feature learning. Tables 5-7 represent the three FC layer model scenarios.

#### 2) Transfer Learning Architecture

The three FC layer models mentioned earlier will be compared with accuracy levels on the DenseNet121 architecture. Once the best FC model is selected from the three FC layer models, it will be employed as an FC layer in a comparison of four TL architectures, namely

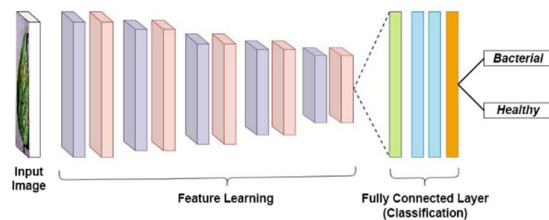


Figure 2. Experiment overview.

TABLE 5  
FC 1 LAYER

#	Parameter	Weight / Loss	Activation / Optimizer
Fully Connected Layer			
1	Global Average Pooling 2D	-	-
Layer 1			
2	Dense	1024	ReLU
3	Batch Normalization	-	-
4	Dropout	0.3	-
Classifier			
5	Dense	2	Sigmoid
6	Loss	binary_crossentropy	RMSprop

TABLE 6  
FC 2 LAYERS

#	Parameter	Weight / Loss	Activation / Optimizer
Fully Connected Layer			
1	Global Average Pooling 2D	-	-
Layer 1			
2	Dense	1024	ReLU
3	Batch Normalization	-	-
4	Dropout	0.3	-
Layer 2			
5	Dense	512	Relu
6	Batch Normalization	-	-
7	Dropout	0.3	-
Classifier			
8	Dense	2	Sigmoid
9	Loss	binary_crossentropy	RMSprop

TABLE 7  
FC 3 LAYERS

#	Parameter	Weight / Loss	Activation / Optimizer
Fully Connected Layer			
1	Global Average Pooling 2D	-	-
Layer 1			
2	Dense	1024	ReLU
3	Batch Normalization	-	-
4	Dropout	0.3	-
Layer 2			
5	Dense	512	ReLU
6	Batch Normalization	-	-
7	Dropout	0.3	-
Layer 3			
8	Dense	256	Relu
9	Batch Normalization	-	-
10	Dropout	0.3	-
Classifier			
11	Dense	2	Sigmoid
12	Loss	binary_crossentropy	RMSprop

MobileNet, ResNet50V2, VGG16, and DenseNet121, to determine the highest accuracy value.

#### 3) Changing Hyperparameters

This step will involve hyperparameter modifications in the form of dropouts and optimizers from the top Transfer Learning architecture models. It is common practice to utilize dropouts to avoid overfitting and accelerate the learning process [36], [37]. Optimizers are algorithms that adjust the weights and biases in the neural network by reducing the distance between the network output and the target [38]. This stage determines whether it is the best model developed with the highest accuracy value at the Transfer Learning architecture stage.

### D. Evaluation

A performance matrix will assess the model with the best accuracy and optimality. Accuracy and loss graphs, confusion matrices, image classification of training data, and Cohen's Kappa are all included in the performance matrix.

## III. RESULTS AND DISCUSSION

The results of the research provide a summary of the model that was developed and put to the test using the research methods covered in the previous section. Also, the outcomes will be discussed and evaluated based on

the photos, tables, and graphs acquired. The sentences that follow will talk about a particular understanding.

## A. Experiment Scenario

### 1) Fully Connected Layer Model

In this experiment scenario, DenseNet121 was used as a Transfer Learning-based model for feature extraction purposes. It then connected to the FC layer with the layer number that was built previously, and the results obtained can be seen in Table 8.

Table 8 shows that the accuracy rates for FC 1 layer, 2 layers, and 3 layers are 99.33%, 98.88%, and 98.66% respectively, with the running time for the second layer being 26 minutes and 5 seconds and for the third layer being 25 minutes and 7 seconds. With an accuracy of 99.33% and a running time of 26 minutes, the FC 1 layer has the highest accuracy rate. Table 9 shows the four TL topologies' results with the FC 1 layer.

### 2) Transfer Learning Architecture

The highest accuracy model is the FC 1 layer model. Then, with adaptable Transfer Learning-based models, FC layer is utilized in the architecture. The performance of four Transfer Learning architectures is compared using a Transfer Learning-based model that is varied. Table 9 displays the findings for each Transfer Learning architecture linked to the FC 1 layer.

According to Table 9, the MobileNet + FC 1 Layer architecture has an accuracy rate of 94.64% with an iteration time of 14 minutes and 31 seconds, the ResNet50V2 + FC 1 Layer architecture has an accuracy rate of 98.88% with an iteration time of 1 hour and 19 minutes and 41 seconds. The VGG16 + FC 1 Layer architecture has an accuracy rate of 95.98% with an iteration time of 2 hours 1 minute and 24 seconds. As a result, the DenseNet121 + FC 1 Layer model has the highest accuracy rate, with a 99.33% accuracy rate and an iteration time of 26 minutes.

## B. Change Hyperparameter

### 1) Change Dropout

Figure 3 shows that if the original model (DenseNet121) developed in the previous stage without a dropout, the accuracy is 98.66%. If the dropout weight to 0.2, the accuracy is 98.88%. The use of dropouts in the design is suitable by creating an accuracy rate of 99.33%.

TABLE 8  
COMPARISON OF FC LAYER MODELS

#	Model	Accuracy	Time
1	FC 1 Layer	99.33%	00:26:00
2	FC 2 Layers	98.88%	00:26:05
3	FC 3 Layers	98.66%	00:25:07

TABLE 9  
TL ARCHITECTURE COMPARISON

#	Model	Accuracy	Time
1	MobileNet + FC 1 Layer	94.64%	00:14:31
2	Resnet50V2 + FC 1 Layer	98.88%	01:19:41
3	VGG16 + FC 1 Layer	95.98%	02:01:24
4	DenseNet121 + FC 1 Layer	99.33%	00:26:00

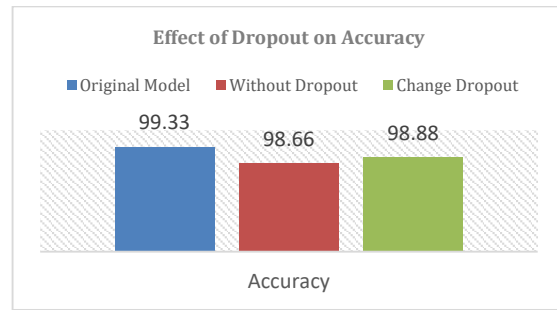


Figure 3. Effect of dropout on accuracy (DenseNet121).

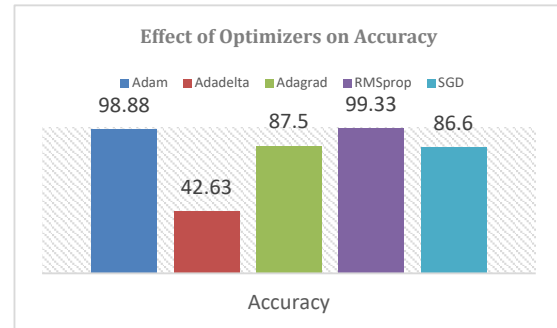


Figure 4. Effect of optimizers on accuracy.

### 2) Changing Optimizers

Figure 4 shows that the DenseNet121 using Adam optimizers produces an accuracy of 98.88%. Adadelta optimizers get the lowest results from other optimizers by getting an accuracy of 42.63%. Adagrad optimizers get an accuracy rate of 87.5%. Root means square propagation (RMSprop) gets an accuracy rate of 99.33%, and stochastic gradient descent (SGD) optimizers get an accuracy rate of 86.6%. As a result of these findings, it was appropriate to use the RMSprop optimizers in the previous architecture.

## C. Evaluation

The assessment step will use the DenseNet121 architecture with FC 1 layer because it is the best architecture. A comparison of the accuracy levels will be conducted between the prior research and the current research. This comparison aims to see the performance results from the testing phase that has been carried out.

### 1) Accuracy and Loss Graph

The graph of accuracy in testing the DenseNet121 + FC 1 Layer architectural model can be seen in Figure 5. As seen in Figure 5, the graph representing the training data is represented by a blue line, while the graph representing the validation data is represented by an orange line. Up until the last epoch, or the 20<sup>th</sup> epoch, the best accuracy on the training data increased, reaching an accuracy of 1,000 or 100%. The accuracy grows until the 13<sup>th</sup> epoch, which is to attain an accuracy of 0.9978 or 99.78%, for the best accuracy on the validation data graph.

Figure 6 depicts the loss graph. A blue line in Figure 6 represents the graph on the training data, and an orange line represents the graph on the validation data. The 20<sup>th</sup>

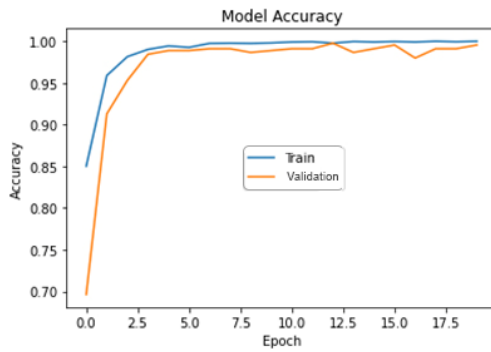


Figure 5. Accuracy graph.

epoch of the training data graph contains the best loss value, which has a value of  $4.9838 \times 10^{-5}$ , or 0.000049838. The 13<sup>th</sup> epoch has the best loss value on the validation data graph, with a value of 0.0076. At this stage, the system generates the file (\*.hdf5) that will be used for system implementation. This file is formed based on the validation data loss.

## 2) Confusion Matrix

The performance of the DenseNet121 + FC 1 Layer architecture model is then evaluated using a confusion matrix table that depicts the expected and actual classes. Table 10 contains the confusion matrix.

The image has been successfully tested by producing classification, as shown by the confusion matrix in Table 10. In the True Positive (TP) portion, which included the Bacterial class, 212 images were properly classified, but there were 3 images in the False Negative (FN) area where the Bacterial class was incorrectly identified as the Healthy class. There are 0 images in the False Positive (FP) section, which is the healthy class, which is anticipated to be bacterial class, whereas 233 images in the True Negative (TN) portion, which is the healthy class, are classified properly.

The performance is shown in Table 11 as accuracy, recall, precision, error rate, and f1-score, which can be viewed from the confusion matrix table.

TABLE 10  
CONFUSION MATRIX

	Pred. Bacterial	Pred. Healthy
True Bacterial	212	3
True Healthy	0	233

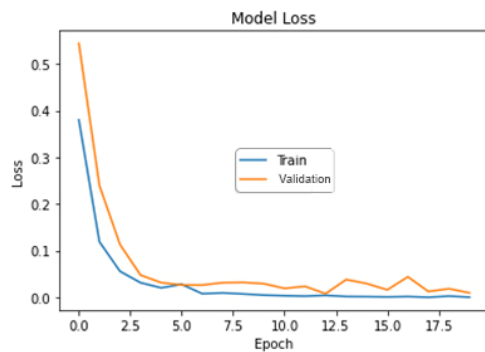


Figure 6. Loss graph.

TABLE 11  
PERFORMANCE

#	Performance	Accuracy
1	Accuracy	99.33%
2	Precision	100%
3	Recall	98.60%
4	Error Rate	0.67%
5	f1-score	99.29%

TABLE 12  
TABLE FORMAT

		Rater 1		Row Marginals
		Pred. Bacterial	Pred. Healthy	
Rater 2	True Bacterial	212	3	215
	True Healthy	0	233	233
Column Marginals		212	266	488

## 3) Cohen's Kappa

To calculate Cohen's Kappa, refer to Table 12. The formula of Cohen's Kappa may be performed as in (1).

$$k = \frac{Pr(a) - Pr(e)}{1 - Pr(e)} \quad (1)$$

$Pr(a)$  represents the observed agreement, and  $Pr(e)$  represents the change agreement.  $Pr(a)$  formula can be seen in (2) and  $Pr(e)$  formula can be seen in (3).

$$Pr(a) = \frac{TP + TN}{n} \quad (2)$$

$$Pr(e) = \frac{\left(\frac{cm^1 \times rm^1}{n}\right) + \left(\frac{cm^2 \times rm^2}{n}\right)}{n} \quad (3)$$

The calculation for  $Pr(a)$  can be seen in (4)-(6), and the calculation for  $Pr(e)$  can be seen in (7)-(9).

$$Pr(a) = \frac{(212 + 233)}{488} \quad (4)$$

$$Pr(a) = \frac{455}{488} \quad (5)$$

$$Pr(a) = 0.9333 \quad (6)$$

$$Pr(e) = \frac{\left(\frac{212 \times 215}{488}\right) + \left(\frac{266 \times 233}{488}\right)}{488} \quad (7)$$

$$Pr(e) = \frac{224.48}{488} \quad (8)$$

$$Pr(e) = 0.5011 \quad (9)$$

The calculation of Cohen's Kappa can be seen in (10)-(12).

$$k = \frac{99.33 - 0.5011}{1 - 0.5011} \quad (10)$$

$$k = \frac{0.4922}{0.4989} \quad (11)$$

$$k = 0.9865 \quad (12)$$

From (10)-(12), know that the DenseNet121 + FC 1 Layer model has a Kappa value of 0.9865. The strength of agreement was achieved with a very good label.

## 4) Image Classification Test

The DenseNet121 + FC 1 Layer architecture will classify 32 images from random test data. The image classifications can be seen in Figure 7.





Figure 7. Image classifications on test data.

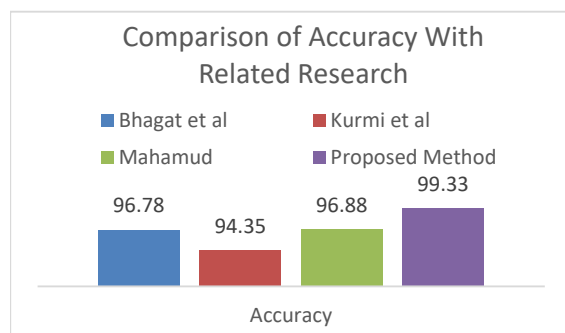


Figure 8. Comparison of accuracy with related research.

**Classifications:** Incorrect image classifications are shown on the red label, whereas accurate image classifications are shown on the green label. As shown in Figure 7, the outcomes of 32 images are expected to exhibit a report in a green label, in other words, a total of 32 images collected from training data, were classified correctly, because the level of accuracy is relatively high with an accuracy of 99.33%.

#### 5) Comparison of Accuracy with Related Research

The suggested approach in this research is the DenseNet121 + FC 1 Layer design. To evaluate the research innovation, the accuracy of this proposed method is compared to that of earlier studies. Figure 8 illustrates a comparison of precision with relevant research.

Figure 8 shows the research was conducted by Bhagat *et al.* [16] had an accuracy rate of 96.78%, Mahamud *et al.* [8] had an accuracy rate of 96.88%, and Kurmi *et al.* [17] had an accuracy rate of 94.35%, according to a comparison of accuracy with comparable research in Figure 8. The classification accuracy rate for the final image in this research is 99.33%, achieved by the proposed DenseNet121 + FC 1 Layer. The use of fine-tuned Transfer Learning, in which the model has already been trained and fitted to the new model, gives the suggested method the highest grade. In other words, this research did better than other studies in terms of accuracy, which led to innovative research.

## IV. CONCLUSION

The DenseNet121 + FC 1 Layer design is the best TL architecture in this research, with a dense layer of 1024 hidden units, ReLu activation, batch normalization, and dropout of 0.3. The TL DenseNet121 architecture with careful tuning proved successful in classifying images of

bell pepper leaves, getting an accuracy score of 99.33% and a Cohen's Kappa value of 0.9865.

## DECLARATIONS

### Conflict of Interest

The authors have declared that no competing interests exist.

### CRediT Authorship Contribution

Yuris Alkhalifi: Investigation, Data Curation, Writing - Original Draft, Visualization, Writing - Review and Editing; Agus Subekti: Conceptualization, Methodology, Investigation, and Supervision.

### Funding

The authors received no financial support for the publication of this article.

## REFERENCES

- [1] L. M. Anaya-Esparza, Z. V. de la Mora, O. Vázquez-Paulino, F. Ascencio, and A. Villarruel-López, "Bell peppers (*capsicum annum* L.) losses and wastes: source for food and pharmaceutical applications," *Molecules*, vol. 26, no. 17, Sep. 2021, Art no. 5341, doi: 10.3390/MOLECULES26175341.
- [2] Y. González-García *et al.*, "Effect of three nanoparticles (Se, Si and Cu) on the bioactive compounds of bell pepper fruits under saline stress," *Plants*, vol. 10, no. 2, 2021, Art no. 217, doi: 10.3390/plants10020217.
- [3] C. Krasnow and C. Ziv, "Non-chemical approaches to control postharvest gray mold disease in bell peppers," *Agronomy*, vol. 12, no. 1, Jan. 2022, Art no. 216, doi: 10.3390/agronomy12010216.
- [4] H. P. Susetyo, "Organisme pengganggu tanaman (opt) pada tanaman paprika dan teknik pengendalian," *Holtukultura.com*. <http://hortikultura.pertanian.go.id/?p=2068> (accessed Jan. 15, 2021).
- [5] M. Chandrasekaran, M. Paramasivan, and S. C. Chun, "Bacillus subtilis CBR05 induces vitamin B6 biosynthesis in tomato through the de novo pathway in contributing disease resistance against *Xanthomonas campestris* pv. vesicatoria," *Sci. Rep.*, vol. 9, 2019, Art no. 6495, doi: 10.1038/s41598-019-41888-6.
- [6] E. O. Hassan and M. A. Zyton, "Management of bacterial spot of pepper caused by *Xanthomonas campestris* pv. vesicatoria," *American J. Biosci. Bioeng.*, vol. 5, no. 1, pp. 41-49, 2017, doi: 10.11648/j.bio.20170501.17.
- [7] J. Hausner, N. Hartmann, M. Jordan, and D. Büttner, "The predicted lytic transglycosylase HpaH from *Xanthomonas campestris* pv. vesicatoria associates with the type III secretion system and promotes effector protein translocation," *Infect. Immun.*, vol. 85, no. 2, 2017, Art no. e00788-16, doi: 10.1128/IAI.00788-16.
- [8] F. Mahamud *et al.*, "Bell pepper leaf disease classification using convolutional neural network," *Int. Conf. Intell. Comput. Optimization 2022. Lecture Notes Netw. Syst.*, vol. 569, pp. 75-86, 2023, doi: 10.1007/978-3-031-19958-5\_8.
- [9] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-net: efficient channel attention for deep convolutional neural networks," in *2020 IEEE/The Computer Vision Foundation Conf. Computer Vision and Pattern Recognition*, Jun. 2020, pp. 11531-11539, doi: 10.1109/CVPR42600.2020.01155.
- [10] Y.-D. Zhang, S. C. Satapathy, D. S. Guttery, J. M. Górriz, and S.-H. Wang, "Improved breast cancer classification through combining graph convolutional network and convolutional neural network," *Inf. Process. Manag.*, vol. 58, no. 2, 2021, Art no. 102439, doi: <https://doi.org/10.1016/j.ipm.2020.102439>.
- [11] A. Subeesh *et al.*, "Deep convolutional neural network models for weed detection in polyhouse grown bell peppers," *Artif. Intell. Agriculture*, vol. 6, pp. 47-54, 2022, doi: <https://doi.org/10.1016/j.aiia.2022.01.002>.
- [12] P. K. Chaitanya and K. Yasudha, "Image based plant disease detection using convolution neural networks algorithm," *Int. J. Innovative Sci. Res. Technol.*, vol. 5, no. 5, pp. 331-334, 2020, Accessed: Aug. 22, 2023. [Online]. Available:

- <https://www.ijisrt.com/image-based-plant-disease-detection-using-convolution-neural-networks-algorithm>.
- [13] V. Singh and A. K. Misra, "Detection of plant leaf diseases using image segmentation and soft computing techniques," *Inf. Process. Agriculture*, vol. 4, no. 1, pp. 41–49, 2017, doi: <https://doi.org/10.1016/j.inpa.2016.10.005>.
  - [14] R. Geirhos *et al.*, "Shortcut learning in deep neural networks," *Nat. Mach. Intell.*, vol. 2, no. 11, pp. 665–673, Nov. 2020, doi: [10.1038/s42256-020-00257-z](https://doi.org/10.1038/s42256-020-00257-z).
  - [15] M. Jhuria, A. Kumar, and R. Borse, "Image processing for smart farming: detection of disease and fruit grading," *2013 IEEE 2nd Int. Conf. Image Inf. Process.*, 2013, pp. 521–526.
  - [16] M. Bhagat, D. Kumar, R. Mahmood, B. Pati, and M. Kumar, "Bell pepper leaf disease classification using CNN," *2nd Int. Conf. Data, Engineering and Applications*, 2020, doi: [10.1109/IDEA49133.2020.9170728](https://doi.org/10.1109/IDEA49133.2020.9170728).
  - [17] Y. Kurmi, S. Gangwar, D. Agrawal, S. Kumar, and H. S. Srivastava, "Leaf image analysis-based crop diseases classification," *Signal Image Video Process.*, vol. 15, pp. 589–597, 2021, doi: [10.1007/s11760-020-01780-7](https://doi.org/10.1007/s11760-020-01780-7).
  - [18] *PlantVillage dataset/Kaggle2*, T. O. Emmanuel, 2019. [Online]. Available: <https://www.kaggle.com/emmanrex/plantdisease> (accessed Jan. 15, 2021).
  - [19] T. Rahman *et al.*, "Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray," *Appl. Sci.*, vol. 10, no. 9, May 2020, Art no. 3233, doi: [10.3390/app10093233](https://doi.org/10.3390/app10093233).
  - [20] M. A. H. Akhand, S. Roy, N. Siddique, M. A. S. Kamal, and T. Shimamura, "Facial emotion recognition using transfer learning in the deep CNN," *Electron.*, vol. 10, no. 9, May 2021, Art no. 1036, doi: [10.3390/electronics10091036](https://doi.org/10.3390/electronics10091036).
  - [21] I. Deep Mastan and S. Raman, "DeepCFL: deep contextual features learning from a single image," in *Proc. 2021 IEEE Winter Conf. Appl. Comput. Vision*, 2021, pp. 2896–2905. doi: [10.1109/WACV48630.2021.00294](https://doi.org/10.1109/WACV48630.2021.00294).
  - [22] I. Z. Mukti and D. Biswas, "Transfer learning based plant diseases detection using resnet50," in *2019 4th Int. Conf. Electrical Information and Communication Technology*, Dec. 2019. doi: [10.1109/EICT48899.2019.9068805](https://doi.org/10.1109/EICT48899.2019.9068805).
  - [23] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Comput. Electron. Agric.*, vol. 161, pp. 272–279, 2019, doi: <https://doi.org/10.1016/j.compag.2018.03.032>.
  - [24] R. A. Welikala *et al.*, "Fine-tuning deep learning architectures for early detection of oral cancer," in *2020 Int. Symp. Mathematical and Computational Oncology. Lecture Notes Comp. Sci.*, vol. 12508, pp. 25–31, 2020, doi: [10.1007/978-3-030-64511-3\\_3/FIGURES/3](https://doi.org/10.1007/978-3-030-64511-3_3/FIGURES/3).
  - [25] S. Aggarwal, S. Gupta, A. Alhudhaif, D. Koundal, R. Gupta, and K. Polat, "Automated covid-19 detection in chest X-ray images using fine-tuned deep learning architectures," *Expert Syst.*, vol. 39, no. 3, Mar. 2022, Art no. e12749, doi: [10.1111/exsy.12749](https://doi.org/10.1111/exsy.12749).
  - [26] A. B. Mutiara, "Implementasi deep learning: matlab dan python-keras-tensorflow," Depok, 2020. Accessed: Jul. 21, 2021. [Online]. Available: <https://mooc.aptikom.or.id/mod/resource/view.php?id=1095>
  - [27] A. G. Howard *et al.*, "MobileNets: efficient convolutional neural networks for mobile vision applications," Apr. 2017. Accessed: Aug. 22, 2023. [Online]. Available: <https://arxiv.org/abs/1704.04861v1>.
  - [28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learning Representations 2015*, 2015. Accessed: Aug. 22, 2023. [Online]. Available: <https://arxiv.org/abs/1409.1556v6>.
  - [29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2016, pp. 770–778, doi: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
  - [30] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. 30th IEEE Conf. Computer Vision and Pattern Recognition 2017*, 2017, pp. 2261–2269, doi: [10.1109/CVPR.2017.243](https://doi.org/10.1109/CVPR.2017.243).
  - [31] C. El Morr, M. Jammal, H. Ali-Hassan, and W. El-Hallak, "Data preprocessing," in *Machine Learning for Practical Decision Making: A Multidisciplinary Perspective with Applications from Healthcare, Engineering and Business Analytics*, C. El Morr, M. Jammal, H. Ali-Hassan, and W. El-Hallak, Eds., Cham: Springer International Publishing, 2022, pp. 117–163, doi: [10.1007/978-3-031-16990-8\\_4](https://doi.org/10.1007/978-3-031-16990-8_4).
  - [32] T. Rahman *et al.*, "Exploring the effect of image enhancement techniques on covid-19 detection using chest X-ray images," *Comput. Biol. Med.*, vol. 132, 2021, Art no. 104319, doi: <https://doi.org/10.1016/j.combiomed.2021.104319>.
  - [33] X. Wu, S. Lv, L. Zang, J. Han, and S. Hu, "Conditional bert contextual augmentation," in *Int. Conf. Computational Science 2019. Lecture Notes Comput. Sci.*, 2019, vol. 11539, pp. 84–95. doi: [10.1007/978-3-030-22747-0\\_7/TABLES/4](https://doi.org/10.1007/978-3-030-22747-0_7/TABLES/4).
  - [34] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, Dec. 2019, Art no. 60, doi: [10.1186/s40537-019-0197-0](https://doi.org/10.1186/s40537-019-0197-0).
  - [35] D. M. Hibban and W. F. Al Maki, "Classification of ornamental betta fish using convolutional neural network method and grabcut segmentation," in *2021 Int. Conf. Data Science and Its Applications*, 2021, pp. 102–109, doi: [10.1109/ICoDSA53588.2021.9617213](https://doi.org/10.1109/ICoDSA53588.2021.9617213).
  - [36] C. Ha, V.-D. Tran, L. Ngo Van, and K. Than, "Eliminating overfitting of probabilistic topic models on short and noisy text: the role of dropout," *Int. J. Approximate Reasoning*, vol. 112, pp. 85–104, 2019, doi: <https://doi.org/10.1016/j.ijar.2019.05.010>.
  - [37] X. Ying, "An overview of overfitting and its solutions," *J. Phys. Conf. Ser.*, vol. 1168, no. 2, Feb. 2019, Art no. 022022, doi: [10.1088/1742-6596/1168/2/022022](https://doi.org/10.1088/1742-6596/1168/2/022022).
  - [38] A. T. Putra, K. Usman, and S. Saidah, "Webinar student presence system based on regional convolutional neural network using face recognition," *Jurnal Teknik Informatika*, vol. 2, no. 2, pp. 109–118, Mar. 2021, doi: [10.20884/1.jutif.2021.2.2.82](https://doi.org/10.20884/1.jutif.2021.2.2.82).