DOI: 10.15849/IJASCA.240730.03

Int. J. Advance Soft Compu. Appl, Vol. 16, No. 2, July 2024
Print ISSN: 2710-1274, Online ISSN: 2074-8523
Copyright © Al-Zaytoonah University of Jordan (ZUJ)

New Tomato Leaf Disease Classification Method Based on DenseNet121 with Bat Algorithm Hyperparameters Optimization

Syaiful Anam, Cynthia Ayu Dwi Lestari, Fidia Deny Tisna Amijaya, Hagus Tarno, and Zuraidah Fitriah

Mathematics Department, Brawijaya University, Malang, Indonesia
e-mail: syaiful@ub.ac.id
Mathematics Department, Brawijaya University, Malang, Indonesia
e-mail: cynthiaayu@student.ub.ac.id
Mathematics Department, Mulawarman University, Samarinda, Indonesia
e-mail: fidiadta@fmipa.unmul.ac.id
Plant Pests and Diseases Department, Brawijaya University, Malang, Indonesia
e-mail: h_gustarno@ub.ac.id.
Mathematics Department, Brawijaya University, Malang, Indonesia
e-mail: zuraidahfitriah@ub.ac.id

Abstract

Tomatoes are recognized for their nutritional value in disease prevention, yet foliar infections often hinder their production. Monitoring and management of tomato leaf disease need much time, cost, and effort. Artificial Intelligence (AI), specifically Deep Neural Networks offers a promising solution for automating disease detection, such as Convolutional Neural Networks (CNN). CNN has several advantages over the traditional machine learning. Selecting the optimal CNN architecture is crucial for accurate feature extraction from image data. Numerous well-known CNN-based architectures have been proposed and used in numerous studies, including VGGNet and DenseNet. DenseNet121 has been proven to produce high classification accuracy. However, the DenseNet121 performance depends on the selection of optimal hyperparameter. This paper proposes an enhancing DenseNet121 performance through hyperparameter optimization using the Bat Algorithm (BA) for classifying tomato leaf disease. BA is used in this research since it has good performance and some advantages. The proposed method achieves better performance than the original DenseNet121 and has a competitive computational time. It produces an accuracy of 0.9465, macro-average precision of 0.9498, macro-average recall of 0.9465, and macro-average f1-score of 0.9460. This method is prospective to be implemented in devices in the real problem with many various image conditions in the future.

Keywords: *Bat algorithm, Classification method, DenseNet121, Hyperparameter, Tomato leaf disease.*

Received 10 February 2024; Accepted 10 June 2024

1 Introduction

Traditional agricultural techniques are no longer able to supply the world's food needs due to the exponential expansion in population [1,2]. In order to solve these issues, the modern agricultural techniques are necessary to be used to boost agricultural product, such as fruits and vegetables. Humans require the nutrients found in fruits and vegetables in large quantities, and these nutrients can help avoid a number of diseases [3]. Tomato production in Indonesia is rising from 20,000 tons in 1972 to 1.12 million tons in 2022, with an average annual growth rate of 11.60% [4]. Nevertheless, foliar diseases frequently infect tomato plants, which can influence the quality and productivity of the tomato plant. The World Food and Agriculture Organization (FAO) says the plant diseases can result in losses of 20–40% of crop output [5]. Furthermore, the tomato disease can cause lower tomato production and lower tomato quality. Tomato leaf disease is one of the primary factors contributing of productivity loss in tomato growing, which results in considerable losses of 80–90% in the agricultural economy [6].

A method for recognizing and classifying the tomato leaf diseases is needed to keep up the quality and productivity of tomato plants. The manually tomato leaf diseases detection and identification frequently take a lot of time and effort. Furthermore, identifying symptoms of disease on tomato leaves not only necessitates certain knowledge and abilities, but also it needs much time, cost, and effort. For more speedy and precise disease detection on tomato leaves, artificial intelligence (AI) is used to detect diseases on the tomato leaves [7]. It can assist farmers in better and more effective monitoring their crops and in promptly implementing prevention techniques based on the type of disease recognition.

AI-based technological developments for object detection and classification are progressing quickly. AI or machine learning techniques have been used to classify diseases on tomato leaves [8-11]. A traditional machine learning approaches have trouble recognizing key characteristics of different tomato leaf diseases, which leads to low accuracy when utilizing datasets and a large number of classes [7]. A promising alternative for utilizing computer vision-based systems to control and monitor plant disease automatically is deep learning algorithms. One of popular deep learning algorithms types is Convolutional Neural Networks (CNN). It was introduced for processing image data more accurately by increasing the accuracy of predictions [12].

CNN's ability to process visual data hierarchically and adaptively has been proven effective and successfully implemented in various fields. CNN's primary benefit is its ability to identify key features automatically, without human intervention [13]. CNN's capacity to handle image information in a hierarchical and adaptive manner has been successfully implemented in many domains [14-18]. CNN has been used in a number of studies to recognize tomato leaf diseases.

The architecture of CNN affects its performance. CNN architectural design is a very complex and challenging topic to build because there are no predetermined rules for designing layers in a CNN to solve a problem [19]. The right architecture could extract important features from image data in a representative way. Several recent studies have focused on developing CNN architectures that achieve higher classification accuracy. CNN architecture has been widely explored in the agricultural sector for various purposes, such as detecting plant pests and diseases, estimating crop yields, monitoring product quality, recognizing and classifying plant species, and others [20]. Numerous well-known CNN-based architectures have been proposed and used in numerous studies [21-22]. AlexNet, VGG16, GoogleNet, DenseNet121 and ResNet101 architectures have been used to classify tomato leaf disease into 10 classes. Compared to the other architectures, DenseNet-121 results the best accuracy rate and has a smaller model size [23]. The other research shows that DenseNet121 not only requires fewer parameters, but also it has a reasonable computing time to obtain the best classification of 99.75% [24]. DenseNet121 also has been applied successfully in many applications [25-27].

The CNN performance also relies on the number of hyperparameters, such as dropout rate, epoch, optimizer type, momentum rate, and learning rate. All of parameters interact with each other and can be adjusted within the CNN. Generally, hyperparameter selection is done based on a trial-and-error process. CNN hyperparameters has been tuned manually in [28]. In addition, the research by Islam et al. (2023) also manually selected hyperparameters for plant disease classification, but this can require high computational costs and time, giving rise to the need for automatic hyperparameter optimization [29]. Determining CNN hyperparameters is an optimization problem, because the hyperparameters of CNN can be searched using optimization methods.

A metaheuristic optimization algorithm based on swarm intelligence is an alternative for navigating the complexity of CNNs to obtain the best hyperparameters, thereby producing models that are more accurate, efficient, and able to overcome the challenges faced in image analysis and visual data processing. The swarm intelligence algorithm is robust, simple to use, doesn't need derivatives, and can determine the global optimum from a large number of local optimal [30]. The success of using swarm intelligence algorithm to optimize hyperparameters and CNN architecture may vary depending on the dataset and architecture used. The AlexNet hyperparameters optimized by using Particle Swarm Optimization (PSO) has been proven to produce the better performance [31]. The other good swarm intelligence method is Bat Algorithm (BA). BA has resulted the faster convergence rate than Genetic Algorithm (GA) and PSO [32]. BA also has been implemented successfully in many applications [33-37].

Based on the research problems has been discussed, this study proposes a novel approach by using AI for swift and accurate detection of tomato leaf diseases. Specifically, this paper presents a novel tomato leaf disease classification method based on DenseNet121 with BA optimization for hyperparameter tuning to find the best parameter of DenseNet121. This method aims to enhance disease recognition efficiency, empowering farmers with timely insights for proactive disease management and safeguarding tomato crop health and productivity.

2 Related Work

Recent advancements in AI-based technologies for object detection and classification have led to rapid progress in the field. Various machine learning techniques, including SVM [8], Random Forest [9], KNN [10], and Naive Bayes [11] have been used to object classification. However, the traditional machine learning approaches have trouble recognizing key characteristics of different tomato leaf diseases, which leads to low accuracy when utilizing datasets and many classes [7].

An increasingly promising approach for recognizing object, such as plant diseases, is Deep Neural Networks. Convolutional Neural Networks (CNNs) is one of the Deep Neural Networks that has been widely adopted for processing image data more accurately, thereby improving prediction accuracy [12]. CNN's ability to process visual data hierarchically and adaptively has been proven effective and successfully implemented in various fields. CNN's primary benefit is its ability to identify key features automatically, without human intervention [13]. CNN's capacity to handle image information hierarchically and adaptively has been successfully applied in some domains, including pneumonia classification [14], website security [15], stroke prediction [16], SMS spam detection [17], crop identification [18], and others. CNN has been used in several studies to determine tomato leaf diseases.

The architecture of CNN affects its performance. CNN architectural design is a very complex and challenging topic to build because there are no predetermined rules for designing layers in a CNN to solve a problem [19]. The right architecture could extract important features from image data in a representative way. Several recent studies have focused on developing CNN architectures that achieve higher classification accuracy. CNN architecture has been widely explored in the agricultural sector for various purposes, such as detecting plant pests and diseases, estimating crop yields, monitoring product quality, recognizing and classifying plant species, and others [20]. Numerous well-known CNN-based architectures have been proposed and used in numerous studies, including VGGNet [21] and DenseNet [22]. AlexNet, VGG16, GoogleNet, DenseNet121, and ResNet101 architectures have been used to classify 10 disease classes in tomatoes with the result that DenseNet-121 had the best accuracy rate of 99.97% and had a smaller model size compared to other architectures [23]. An empirical comparison of the VGG16, InceptionV4, Resnet50, Resnet101, Resnet152, and DenseNet121 architectures for plant disease identification has been provided in [24]. This research shows that DenseNet121 not only requires fewer parameters, but also it has reasonable computing time to obtain the best classification of 99.75%. DenseNet121 also has been applied successfully in many applications, such as the classification of NTT weaving motif [25], weather image recognition [26], nutrient deficiency classification [27], etc.

The CNN capability relies on some hyperparameters, all of which interact with each other and can be adjusted within the CNN. CNN hyperparameters in the form of dropout rate, epoch, optimizer type, momentum rate, and learning rate for tomato leaf disease classification have been tuned manually [28]. In addition, the research by Islam et al. (2023) also manually selected hyperparameters for plant disease classification in the form of learning rate, optimizer type, and activation function, but this can require high computational costs and time, giving rise to the need for automatic hyperparameter optimization [29]. Determining CNN hyperparameters is an optimization problem because the hyperparameters of CNN can be searched using optimization methods.

A metaheuristic optimization algorithm based on swarm intelligence is an alternative for navigating the complexity of CNNs to obtain the best architecture and hyperparameters, thereby producing models that are more accurate, efficient, and able to overcome the challenges faced in image analysis and visual data processing. The success of using a swarm intelligence algorithm to optimize hyperparameters and CNN architecture may vary depending on the dataset and architecture used. The optimization of the AlexNet architecture using PSO was done by Yamasaki et al. in 2017 [31]. BA is a good swarm intelligence method and BA's convergence rate is faster than GA and PSO [32]. BA also has been proven successfully to be implemented in many applications, such as Artificial Neural Networks training [33], transport network design [34], association rule mining [35], workflow scheduling [36], structural reliability assessment [37], etc.

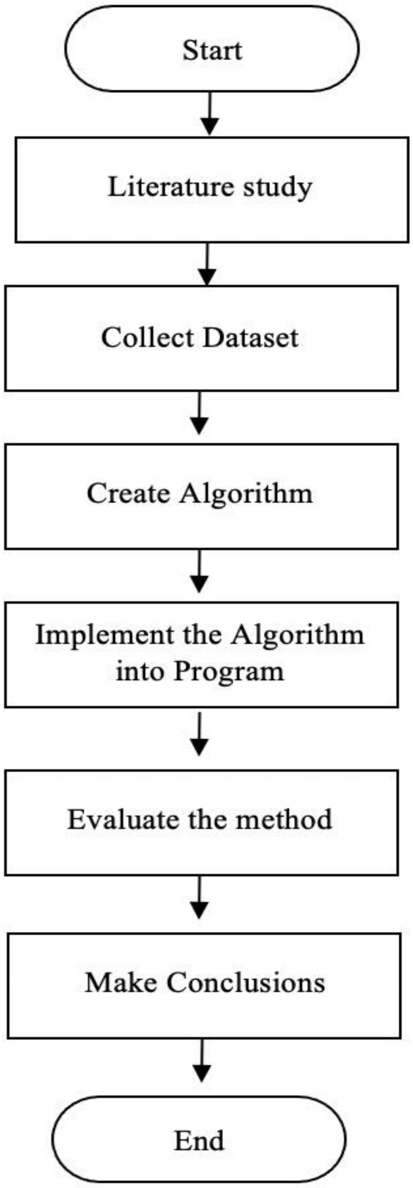

Fig. 1 Research methodology

3 Methodology

This article proposes a new classification method that can accurately identify and classify tomato leaf diseases using DenseNet121 with BA hyperparameter optimization. The goal of this research is to help farmers take appropriate actions to handle the spread of the disease and increase crop yields. The methodology of this research can be seen in Fig 1. This research consists of several stages, including studying literature, collecting datasets, building the algorithms of a method, transforming algorithms into programs, evaluating methods, and making conclusions.

In the first stage, several previous research are studied, such as BA, CNN, tomato disease, and other related articles. The procedure and properties of the BA and CNN methods are studied and then this knowledge is used in building algorithms and programs. The parameters of BA and CNN and DenseNet121 are studied based on previous research. Yang (2014) asserts the parameters value of BA. A_i and r_i can be given the values of 0.5 and 0.5, and value $\alpha = \gamma = 0.9$. Parameters in DenseNet121, such as the number of

Table 1: Parameter range's values

Parameters	Parameter values
The number of neurons in FC layer	128, 256, 512, 1024
The dropout rate	0.1, 0.2, 0.3, 0.4, 0.5
The learning rate	0.1, 0.01, 0.001, 0.0001, 0.00001
The activation function	'relu','sigmoid','tanh'
The optimizer	'adam', 'sgd'

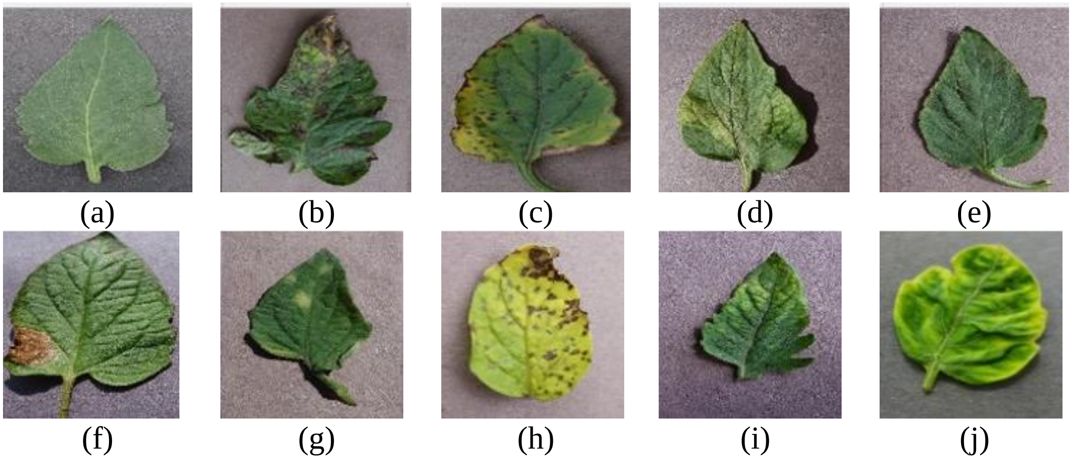

Fig. 2 The image examples of diseases on tomato leaves for each class. (a) Healthy, (b) Bacteria Spot, (c) Early Blight, (d) Septoria Leaf Spot, (e) Late Blight, (f) Leaf Mold, (g) Two-spotted Spider Mite, (h) Mosaic Virus, (i) Target Spot, (j) Yellow Leaf Curl Virus.

neurons in FC layer, the dropout rate, the learning rate, the activation function and the optimizer were studied from previous studies. The range of each parameter was obtained from previous research, as given in Table 1.

The dataset is taken from Kaggle.com [38]. The dataset in this research is divided into 10 classes which are healthy leaf, Bacteria Spot, Early Blight, Septoria Leaf Spot, Late Blight, Leaf Mold, Two-spotted Spider Mite, Mosaic Virus, Target Spot, and Yellow Leaf Curl Virus. Examples of the tomato leaf dataset can be shown in Fig. 2. It can be shown that each disease has different features.

Next, the algorithm of the proposed method is built. The flowchart of the proposed method can be seen in Fig. 3. The first step of the proposed method is that the parameters of BA and dataset are inputted. The second step is doing several pre-processing of CNN such as data resizing, data rescaling, and data augmentation. The data augmentation is done to avoid the overfitting and to make better the model generalization. The operations of data augmentation are rotation, width shift, height shift, shear, zoom, horizontal flip and vertical flip. The range of each operation for data augmentation is 20 for rotation, 0.2 for width shift, 0.2 for height shift, 0.2 for shear, and 0.2 for zoom. The third step is splitting the dataset into training data and testing data. The size of dataset is 10000 and each class has 1000 data images. The ratio of training and testing data are 0.8 and 0.2, respectively.

After the algorithm of proposed method are created and the detail of the proposed method is explained in next section of the proposed method. Therefore, the algorithm is implemented into program by using Python language programming. The last stage is the evaluation of the proposed method. To assess the classification model, it is necessary to calculate the metrics evaluation. The classification model is constructed using

DenseNet121 with BA hyperparameters optimization. As a result, the test data are utilized to evaluate the model. To assess the effectiveness of the classification model, the training and testing data's Accuracy, Recall, Precision, and f_1 Score are computed. The description of each evaluation metric is the following.

a. The accuracy is computed by formulation in (1). A True Positive (*TP*) is a term used to describe the number of positive data that are classified correctly. The number of tuples that are accurately categorized as negative is known as True Negative (*TN*). False Positive (*FP*) represents the number of tuples in the negative class that are classified incorrectly. The number of tuples that fall into the incorrectly negative category is called False Negative (*FN*). Accuracy is the ratio between the number of correct predictions and the number of all tuples.

$$\mathrm{Acc} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}} \tag{1}$$

b. Formulation (2) defines recall. A measure used to calculate the proportion of correctly recognized positive patterns is called *Recall*.

$$\mathrm{Rc} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \tag{2}$$

c. Formulation (3) defines precision. The ratio between all of the tuples in the positive class and the accurately anticipated positive class is used to calculate precision.

$$Prc = \frac{TP}{TP + FP} \tag{3}$$

d. The f_1 *Score* is calculated by using the harmonic mean of recall and precision. The f_1 *Score* formulation can be seen in (4).

$$f_1\ scr = 2 \cdot \frac{Prc \cdot Rc}{Prc + Rc} \tag{4}$$

Since the number of classes is more than 2 classes, this research uses *Macro-Recall*, *Macro-Precision*, *Macro-*f_1 *Score*, and *Accuracy* to evaluate the performance of the classification method. The formulations of *Macro-Recall*, *Macro-Precision, and Macro-*f_1 *Score* are represented by (5), (6), and (7) [39]. *K* represents a number of classes.

$$McrRc = \frac{\sum_{k=1}^{K} Rc_k}{K} \tag{5}$$

$$McrPrc = \frac{\sum_{k=1}^{K} Prc_k}{K} \tag{6}$$

$$Mcrf_1scr = \frac{\sum_{k=1}^{K} f_1\ scr_k}{K} \tag{7}$$

K represents the number of classes. Rc_k, Prc_k and f_1scr_k represents the recall, precision and f_1 score of class k^{th}, respectively. Based on a calculation of the evaluation of the metrics that resulted, furthermore, these are analysed and used for making conclusions

4 The Proposed Method

The part will discuss the developing classification algorithm by using DenseNet121 with BA to classify tomato leaf diseases. A DenseNet121 is a particular kind of CNN that makes use of dense connections between layers. This stage has two main stages which are the

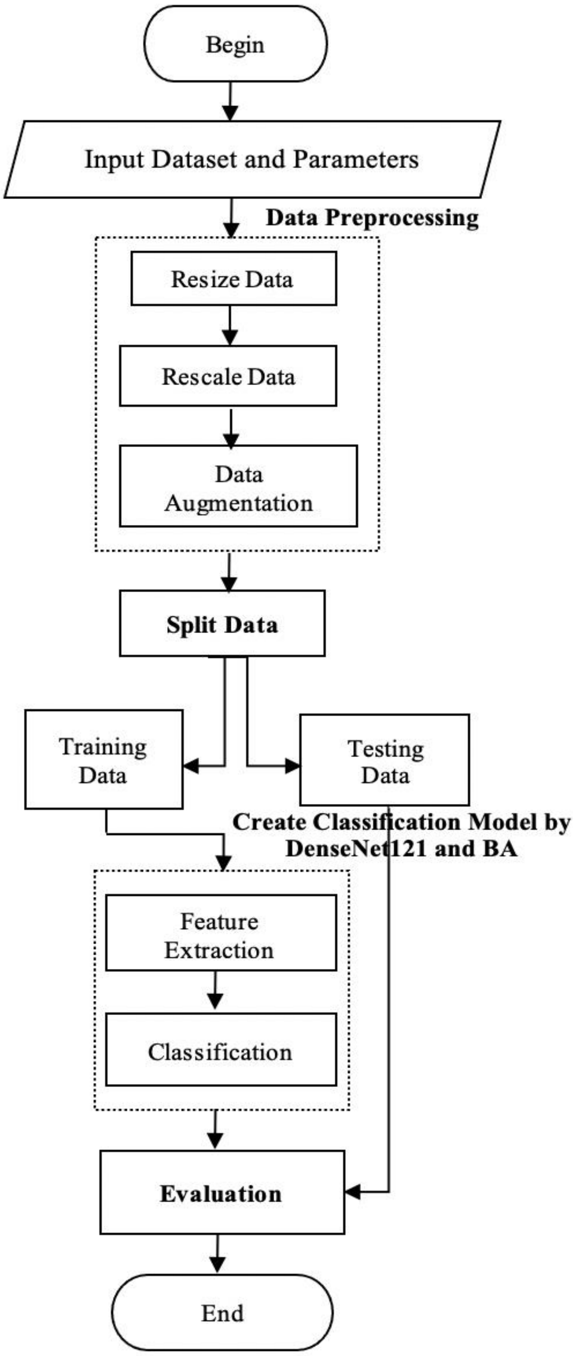

Fig. 3 Flowchart of the proposed method.

feature extraction and classification steps. The feature extraction is done by several operation of DensetNet121 as shown in Fig. 4, while the classification is done on the fully connected-soft max of DenseNet121. DenseNet121 consists of one convolutional layer, one max pooling layer, four dense-block layers, three transition layers, one average pooling layer and one fully connected-SoftMax. Every layer becomes fresh input from the layers above it and sends its maps of features to the layers below.

The kernel size of the initial convolutional layer of DenseNet121 is 7×7. The convolutional layer is CNN's primary component. It has several filters, also known as kernels, the parameters of which must be trained during the training process. Generally, the filters' size is less than that of the original image. Every filter interacts with the picture to produce an activation map. The model of DenseNet121 is composed of 3×3 max-pooling kernels with a stride of two. In the max-pooling operation, a feature map's maximum value is determined for each patch using a pooling process to produce a down-sampled, or pooled, feature map. Dense connections use weights to link each input to each output. Dense connections are made by dense blocks, which link all layers directly together. The number of filters varies within each block, but the feature map dimensions stay

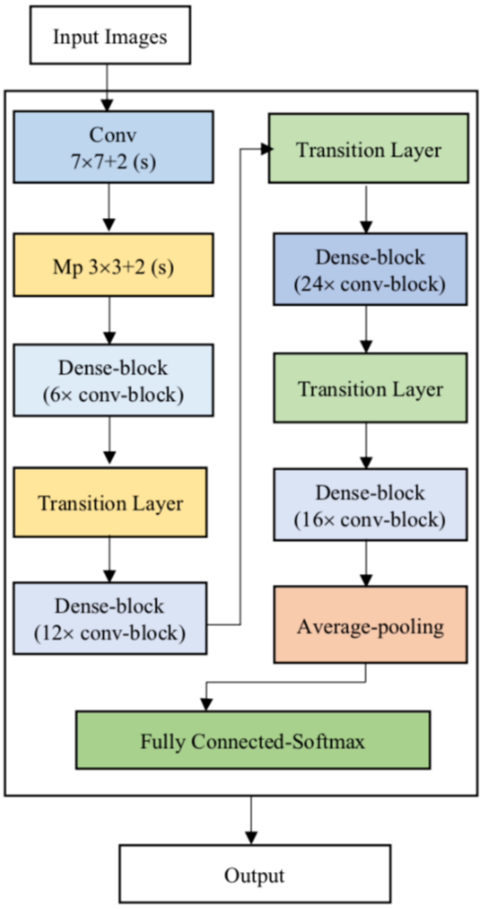

Fig. 4 Architecture of DensetNet121

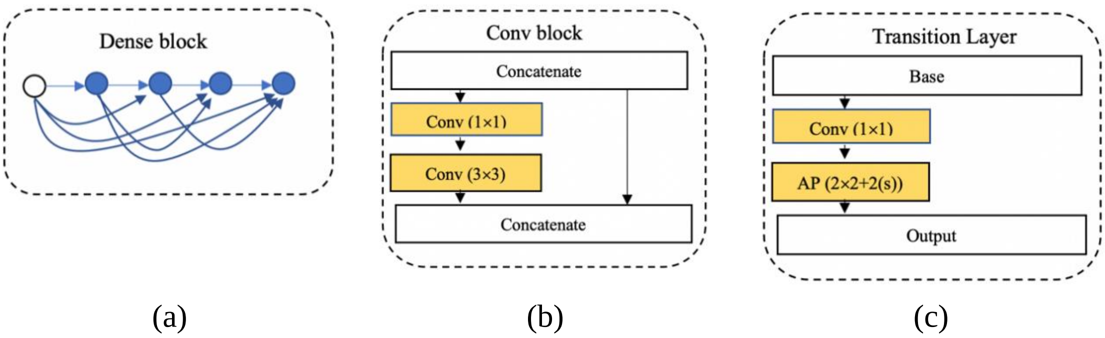

Fig. 5 (a) Dense block. (b) Convolution block. (c) Transition layer

fixed. A module that directly links every layer (with matching feature-map sizes) to every other layer is called a dense block. The dense block architecture can be shown in Fig. 5 (a).

Every dense block is built by some conv block. DenseNet121 has 6, 12, 24, and 16 depths of model of convolutional block. The architectures of the convolutional block and transition layer are shown in Fig. 5(b) and Fig. 5 (c), respectively. Due to the feature map sizes within neighboring dense blocks being the same, the transition layer joins them. The last layer of feature extraction uses the average pooling operation. The last step is to obtain the output by using the SoftMax function [40].

DenseNet121 has some hyperparameters such as a number of neurons in a fully connected layer, dropout rate, learning rate, and activation function. These hyperparameters influence the performance of DenseNet121. For this reason, the new classification model using the DenseNet121 with BA is constructed. The training procedure was used to generate the classification model.

The process for building a classification model using DenseNet121 and BA for classifying tomato leaf disease is shown in the flowchart in Fig. 3. Bat locations in DenseNet121 with BA represent potential candidates for the optimal DenseNet121 hyperparameters. The first phase of the DenseNet121 and BA generates Nb candidates at random for the DenseNet121 hyperparameters, which are defined as $\boldsymbol{x_i^0}, \boldsymbol{i = 1, 2, \ldots, Nb}$. Each parameter range values can be shown in Table 1. Furthermore, DenseNet121 and BA explore and utilize the search space to identify the global optimal point. The fitness function in the DenseNet121 and BA is defined by using accuracy. The following is the full method used by the classification model by using DenseNet121 and BA to classify tomato leaf disease. Algorithm 1 shows the training process to obtain the best hyperparameter of DenseNet121.

Algorithm 1. A new classification model based on DenseNet121 with BA for tomato leaf disease classification.

Input:

- The training dataset ($\boldsymbol{X_{train}}$). $\boldsymbol{X_{train}}$ is a set of images with size of $M \times N$, M and N indicate the height and width of images, respectively. The number of training images dataset is n.
- The BA hyperparameters.

Output:

$\boldsymbol{x}_*$ indicates the optimal DenseNet121 hyperparameter solution.

a. First step is that the bat locations and velocities are set to their initial values, $\boldsymbol{x}_i^0$ and $\boldsymbol{v}_i^0$,$(i = 1, 2, \ldots, Nb)$. Each bat location $\boldsymbol{x}_i$ represents a candidate of DenseNet121 parameters, which are $\boldsymbol{x}_i^0 = \left(x_{i,1}^0, x_{i,2}^0, x_{i,3}^0, x_{i,4}^0\right)$. $x_{i,1}^0$represents the number of neurons in fully connected layer. $x_{i,2}^0$ defines a dropout rate. $x_{i,3}^0$ and $x_{i,4}^0$ represent a learning rate and activation function, respectively.

b. Determine f_{min}, f_{max}, r_i, and A_i, $(i = 1, 2, \ldots, Nb)$. f_{min} and f_{max} represent the minimum frequency and the maximum frequency, respectively. r_i defines a pulse rate and A_i is a loudness.

c. t=0

d. **WHILE** $(t < \ Maximum\ of\ Iteration)$ **DO**

1. **FOR** $j = 1$ to Nb **DO**

   i. The frequency, velocity, and location of bats are updated by calculating (8), (9), and (10). β is a number chosen at random from a uniform distribution [0,1].

$$f_i = f_{min} + (f_{max} - f_{min})\beta, \tag{8}$$

$$\boldsymbol{v}_i^{t+1} = \boldsymbol{v}_i^t + (\boldsymbol{x}_i^t - \boldsymbol{x}_*)f_i, \tag{9}$$

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_i^t + \boldsymbol{v}_i^{t+1}. \tag{10}$$

ii. **if** ($rand < r_i$ **then**

The local solutions are generated by using (11), randomly.

$$\boldsymbol{x}_{new} = \boldsymbol{x}_{old} + \sigma\epsilon_t A^{(t)} \tag{11}$$

ϵ_t represents random number with a normal distribution $N(0,1)$ and $A^{(t)}$ defines the means of bat loudness over time t. While σ is the scale factor. For simplicity, it could be taken $\sigma = 0.01$.

end if.

iii. Use the accuracy to determine fitness values of each bat. It is produced by using a DenseNet121 classification model with hyperparameters $\boldsymbol{x}_i^{t+1}$, training data and validating data.

iv. **if** ($rand > A_i$ **and** $f(\boldsymbol{x}_i) < f(\boldsymbol{x}_*)$), **then**

The existing solution could be updated using the results of steps (i) or (ii).

end if

v. The next step is that increases r_i and reduces A_i by using (12) and (13).

$$A_i^{t+1} = \alpha A_i^t, \tag{12}$$

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)], \tag{13}$$

α is given range $0 < \alpha < 1$ and $\gamma > 0$.

vi. Sorting the bats to find the best possible results ($\boldsymbol{x}_*$).

end for

end while

e. **Output:** *Best* ($\boldsymbol{x}_*$) (the DenseNet121 hyperparameters optimum).

Algorithm 2. Evaluation of a new classification model based on DenseNet121 with BA in training data for classifying tomato leaf disease.

Input:

The training dataset ($\boldsymbol{X}_{train}$).

The best hyperparameters of DenseNet121 resulted in BA which is $\boldsymbol{x}_i^* = (x_{i,1}^*, x_{i,2}^*, x_{i,3}^*, x_{i,4}^*)$

y_{train} (the class label of the training data set)

Output:

Accuracy, Macro Recall, Macro Precision, Macro f_1 Score.

1. Create a classification model by DenseNet121 Model by using training data
2. Utilize the DenseNet121 Model with BA to determine the label prediction y_{pred}.
3. Calculate Accuracy, Macro Recall, Macro Precision and Macro-f_1 Score.

Algorithm 3. Evaluation of a new classification model based on DenseNet121 and BA in testing data for classifying tomato leaf disease

Input:

The testing data ($\mathbf{X}_{\text{test}}$)

DenseNet121 model resulted BA

$y_{testing}$ (the class label of the testing data)

Output:

Accuracy, Macro-Recall, Macro-Precision, Macro- f_1 *Score*.

1. Utilize the DenseNet121 Model with BA to determine the label prediction y_{pred}.
2. Calculate Accuracy, Macro Recall, Macro Precision and Macro f_1 Score

The termination condition in this proposed method is using the maximum number of iterations. The maximum number of iterations used is 10 iterations. After the hyperparameters are found by BA, then they are used in the classification model. The last stage is to evaluate the classification model by using training data and testing data. **Algorithm 2** and **Algorithm 3** are utilized for assessing the model by training data and testing data.

Table 2: Number parameters resulted by DenseNet121 with BA and DenseNet121

Number of parameters	DenseNet121	DensetNet121 with BA
Total params	8,097,354	7,567,434
Trainable params	8,013,706	7,483,786
Non-trainable params	83,648	83,648

5 Results, Analysis and Discussions

This part will display the outcomes of the experiment. The comparison of metrics evaluation produced by the proposed method and the DenseNet121 will be discussed.

Fig. 6 (a) shows the accuracy resulted by DensetNet121 with BA and DenseNet121. It can be shown that the DensetNet121 with BA produces better accuracy than DenseNet121. The accuracy produced by BA and DenseNet121 is 0.9465 and the accuracy produced by DenseNet121 is 0.9175. The accuracy improves significantly after hyperparameter optimization.

While, Fig. 6 (b), Fig. 6 (c), and Fig. 6 (d) display the comparison of the best recall, precision, and f_1 scores between DenseNet121 and DensetNet121 with BA, respectively. It can be shown that the DensetNet121 with BA also produces better recall, precision, and f_1 score than DenseNet12. The recall resulted from BA and DenseNet121 is 0.9465 and the recall resulted from DenseNet121 is 0.9175. The precision produced by BA and DenseNet121 is 0.9498 and the precision produced by DenseNet121 is 0.9229, while the f_1 score resulted by BA and DenseNet121 is 0.9460 and the f_1 score produced by DenseNet121 is 0.9174. For all classification metrics, the BA and DenseNet121 have better performance compared to the DenseNet121.

Fig. 7 shows the computational time resulted by DenseNet121 with BA and DenseNet121. DenseNet121 produces higher computational time than DenseNet121. Table 2 shows the number parameters resulted by DenseNet121 with BA and DenseNet121. It can be shown that the number of total params (trainable and non-trainable parameters) along with the computational time needed to train the model. DenseNet121 contains 8,097,354 parameters, while the

Table 3: Hyperparameter resulted by DensetNet121 with BA

Parameters	Parameter’s value

The best number of neurons in FC layer	512
The best dropout rate	0.3
The best learning rate	0.001
The best activation function	tanh
The best optimizer	ADAM

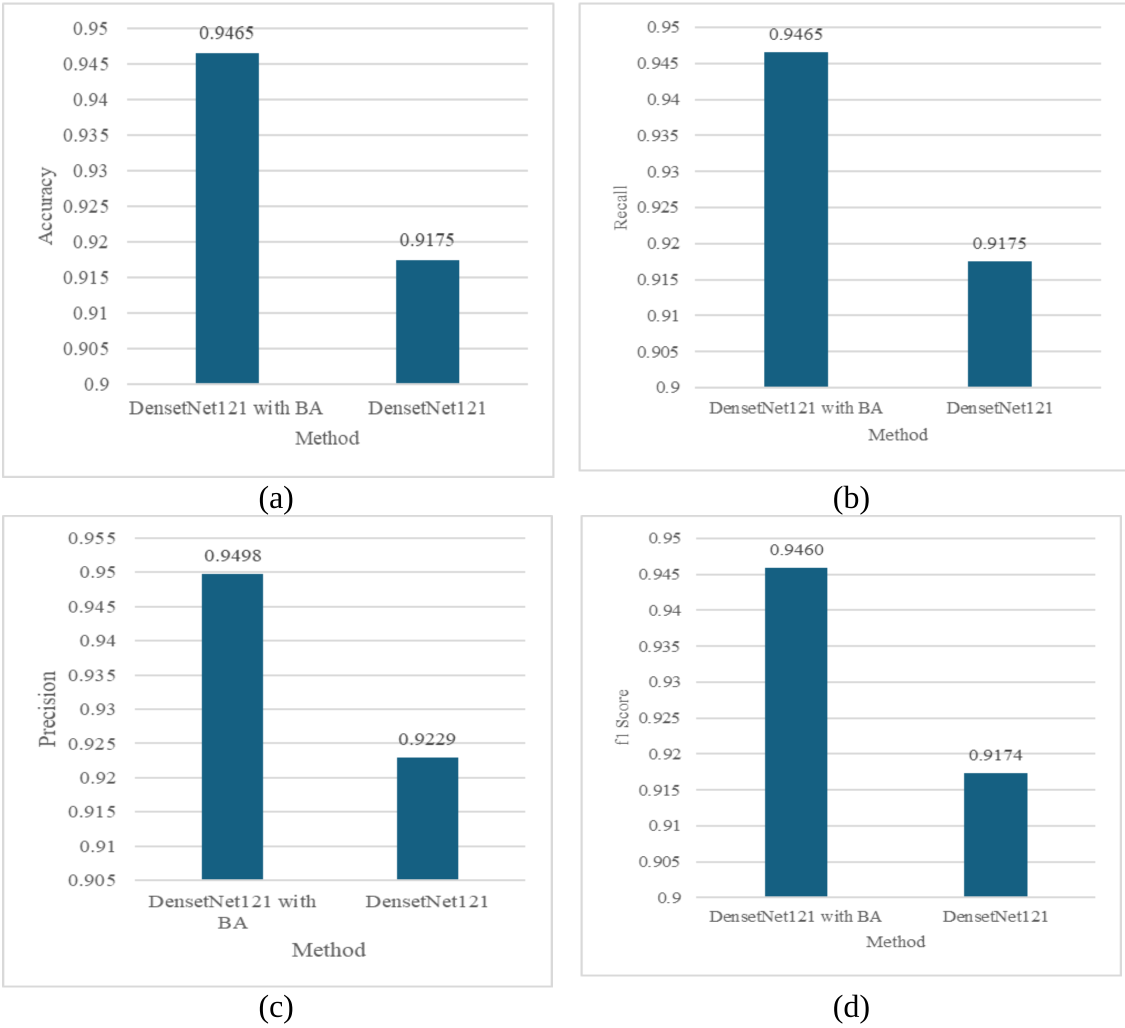

Fig. 6 Comparison of the best (a) accuracy (b) macro-average recall (c) macro-average precision (d) macro-average f_1-score between DenseNet121 and DensetNet121 with BA.

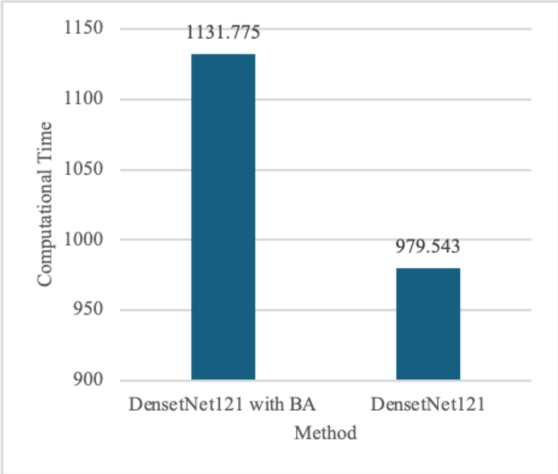

Fig. 7 Comparison of the computational between DenseNet121 and DensetNet121 with BA

number of parameters of DenseNet121 with BA is 7,567,434 parameters. The smaller number of parameters indicates a lighter and more computationally and memory-efficient model. While the proposed architecture uses the population-based optimization, the processing time of the proposed method can compete with the original DenseNet121. The computational time of DenseNet121 with BA and DenseNet121 is not significantly different.

Table 3 shows the hyperparameter resulted by DensetNet121 with BA. The best number of neurons in FC layer is 512, the best dropout rate is 0.3, the best learning rate is 0.001, the best activation function is tanh, and the best optimizer is ADAM.

6 Conclusion

This research proposes a new tomato leaf disease classification method based on DenseNet121 with BA hyperparameters optimization. After evaluating and analyzing the experimental results, it can be asserted that the DenseNet121 with BA performs better than the DenseNet121 for all classification metrics. The accuracy, macro average precision, macro average recall, and macro f1 score resulted by the proposed method are 0.9465, 0.9498, 0.9465, and 0.9460, respectively. The proposed also has competitive computational time compared with DenseNet121. It is caused by the smaller number of parameters. The best hyperparameters resulted by the proposed method are 512 for the neurons in the FC layer, 0.3 for the dropout rate, 0.001 for the learning rate, tanh for the activation function, and ADAM for the optimizer. Finally, this method is prospective to be implemented in devices in the real problem with many various image conditions in the future.

ACKNOWLEDGEMENTS
We express our gratitude to Brawijaya University for supporting this research and giving funding by Hibah Penelitian Unggulan (Excellence Research Grants) 2023 with contract number: 612.26/UN10.C20/ 2023.

References

[1] Rejeb, A., Rejeb, K., Abdollahi, A., Al-Turjman, F., & Treiblmaier, H. (2022). The interplay between the internet of things and agriculture: a bibliometric analysis and research agenda. *Internet of Things*, *19*, 100580.

[2] Gzar, D. A., Mahmood, A. M., & Al-Adilee, M. K. A. Recent trends of smart agricultural systems based on internet of things technology: a survey, *Computers and Electrical Engineering*, *104*(A), 108453.

[3] Republika.com. (2017, July 20). Kurang buah dan sayur, penyakit berbahaya mengancam. https://www.republika.co.id/berita/gaya-hidup/info-sehat/17/07/20/otd8sq328-kurang-buah-dan-sayur-penyakit-berbahaya-mengancam

[4] Knoema. (2021). The production of tomatoes in Indonesia. https://knoema.com/data/indonesia+agriculture-indicators-production+tomatoes

[5] FAO. (2021). Climate change fans spread of pests and threatens plants and crops — new FAO study. https://www.fao.org/news/story/en/item/1402920/icode/

[6] Ouhami, M., Es-Saady, Y., El Hajji, M., Hafiane, A., Canals, R., & Yassa, M. E. (2020). Deep transfer learning models for tomato disease detection. In *Lecture Notes Computer Science* (Vol. 12119, pp. 65–73). Springer.

[7] Thangaraj, R., Anandamurugan, S., Pitchai, P., & Kumar, V. (2022). Artificial intelligence in tomato leaf disease detection: A comprehensive review and discussion. *Journal of Plant Diseases and Protection*, 129(3), 469–488.

[8] Jayanthi, M. G., & Shashikumar, D. R. (2020). Automatic tomato plant leaf disease classification using Multi-Kernel Support Vector Machine. *International Journal of Engineering and Advanced Technology*, 9(5), 560–565.

[9] Annabel, L. S. P., & Muthulakshmi, V. (2019). AI-powered image-based tomato leaf disease detection. In *I-SMAC (IoT in Social, Mobile, Analytics and Cloud). Third International conference on* (pp. 506–511). IEEE.

[10] Nasution, A. S., Alvin, A., Siregar, A. T., & Sinaga, M. S. (2022). KNN algorithm for identification of tomato disease based on image segmentation using enhanced K-Means clustering. *KINETIK: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 4(3), 299-308.

[11] Kirange, D. K. (2020). Machine learning approach towards tomato leaf disease classification. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(1), 490–495.

[12] Tugrul, B., Elfatimi, E., & Eryigit, R. (2022). Convolutional Neural Networks in detection of plant leaf diseases: a review. *Agriculture*, 12(8), 1192. https://doi.org/10.3390/agriculture12081192.

[13] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(53), 53. doi: 10.1186/s40537-021-00444-8.

[14] Yadav, S. S., & Jadhav, S. M. (2019). Deep Convolutional Neural Network based medical image classification for disease diagnosis. *Journal of Big Data*, 6(1), 113. https://doi.org/10.1186/s40537-019-0276-2.

[15] Nedeljkovic, D., & Jakovljevic, Z. (2022). CNN based method for the development of cyber-attacks detection algorithms in industrial control systems. *Computer Security*, 114, 102585.

[16] Islam, M. M., Akter, S., Rokunojjaman, M., Rony, J. H., Amin, A., & Kar, S. (2021). Stroke prediction analysis using machine learning classifiers and feature technique. *International Journal of Electronics and Communications Systems*, 1(2), 57–62.

[17] Huang, T. (2019). A CNN model for SMS spam detection. In *Proc. 4th International Conference on Mechanical, Control and Computer Engineering* (ICMCCE) (pp. 851-85110). IEEE.

[18] Yu, F., Zhang, Q., Xiao, J., Ma, Y., Wang, M., Luan, R., Liu, X., Ping, Y., Nie, Y., Tao, Z., & Zhang, H. (2023). Progress in the application of CNN-based image classification and recognition in whole crop growth cycles. *Remote Sensing*, 15(12), 2988.

[19] Ghosh, A., & Jana, N. D. (2022). Artificial Bee Colony Optimization based Optimal Convolutional Neural Network Architecture design. *In 19th India Council International Conference* (pp. 1–7). IEEE.

[20] Nawaz, M., Nazir, T., Javed, A., Masood, M., Rashid, J., Kim, J., & Hussain, A. (2022). A robust deep learning approach for tomato plant leaf disease localization and classification. *Scientific Reports*, 12(1), 1–18.

[21] Nguyen, T. H., Nguyen, T. N., & Ngo, B. V. (2022). A VGG-19 model with transfer learning and image segmentation for classification of tomato leaf disease. *AgriEngineering*, 4(4), 871–887.

[22] Bakr, M., Abdel-Gaber, S., Nasr, M., & Hazman, M. (2022). Tomato disease detection model based on DenseNet and transfer learning. Appl. *Computer Science*, 18(2), 56–70.

[23] Gehlot, M., & Saini, M. L. (2020). Analysis of different CNN architectures for tomato leaf disease classification. *In Recent Advances and Innovations in Engineering. 5th IEEE International Conference on* (pp. 1-6). IEEE.

[24] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*, 161, 272–279.

[25] Maur, Y. E. H., Santoso, A. J., & Pranowo. (2023). Fine-tuned of DenseNET121 to classify NTT weaving motifs on mobile application. *International Journal on Advanced Science, Engineering and Information Technology*, 13(6), 2156–2163.

[26] Albelwi, S. A. (2022). Deep architecture based on DenseNet-121 model for weather image recognition. *International Journal of Advanced Computer Science and Applications*, 13(10), 559–565.

[27] Rao, B. S., Reddy, R. V. K., Manogna, D., Akhil, K., & Sree, D. D. (2022). Identification of nutrient deficiency in rice leaves using DenseNet-121. *In Edge Computing and Applications. International Conference on* (pp. 1573-1578). IEEE.

[28] Sharma, S., & Vardhan, M. (2023). Hyperparameter tuned hybrid convolutional neural network (H-CNN) for accurate plant disease classification. In *Communication, Circuits, and Systems Communication. International Conference on* (pp. 1–6). IEEE.

[29] Islam, M. R., Siddique, M. A. M. T., Amiruzzaman, M., Abdullah-Al-wadud, M., Al Masud, S. M. R., & Saha, A. K. (2023). An efficient technique for recognizing tomato leaf disease based on the most effective deep CNN hyperparameters. *Annals of Emerging Technologies in Computing*, 7(1), 1–14.

[30] Anam, S., & Fitriah, Z. (2021). Early blight disease segmentation on tomato plant using k-means algorithm with swarm intelligence-based algorithm. *International Journal of Mathematics and Computer Science*, 16(4), 1217–1228.

[31] *Yamasaki, T., Honma, T., & Aizawa, K. (2017). Efficient optimization of convolutional neural networks using particle swarm optimization. In Multimedia Big Data. IEEE Third International Conference on (pp. 70–73). IEEE.*

[32] Yang, X.-S. (2010). A new metaheuristic bat-inspired algorithm. In J. R. Gonzalez et al. (Eds.), Nature inspired cooperative strategies for optimization (NISCO 2010). In *Studies in Computational Intelligence*, (Vol. 284, pp. 65-74). Springer Berlin.

[33] Mohapatra, P. K., Rout, S. K., Bisoy, S. K., Kautis, S., Hamzah, K. M., Jasser, B., & Mohame, A. W. (2022). Application of Bat Algorithm and its modified form trained with ANN in channel equalization. *Symmetry*, 14(10), 2078.

[34] Srivastava, S., & Sahana, S. K. (2019). Application of Bat Algorithm for transport network design problem. *Applied Computational Intelligence and Soft Computing*, 2019, 9864090. https://doi.org/10.1155/2019/9864090

[35] Heraguemi, K. E., Kamel, N., & Drias, H. (2014). Association rule mining based on Bat Algorithm. In L. Pan, G. Păun, M. J. Pérez-Jiménez, & T. Song (Eds.), Bio-Inspired Computing - Theories and Applications. In *Communications in Computer and Information Science* (vol 472). Springer. https://doi.org/10.1007/978-3-662-45049-9_29

[36] Raghavan, S., Sarwesh, P., Marimuthu, C., & Chandrasekaran, K. (2015). Bat algorithm for scheduling workflow applications in cloud. In *Electronic Design, Computer Networks & Automated Verification. International Conference on* (pp. 139-144). IEEE.

[37] Chakri, A., Khelif, R., & Benouaret, M. (2016). Improved Bat Algorithm for structural reliability assessment: Application and challenges. *Multidiscipline Modeling in Materials and Structures*, 12(2), 218-253.

[38] Kaustubh. (n.d.). Tomato Leaf Disease Detection Using CNN. Kaggle. https://www.kaggle.com/kaustubhb999/tomatoleaf

[39] Grandini, M., Bagli, E., & Visani, G. (2020). Metrics for multi-class classification: An overview. Retrieved from http://arxiv.org/abs/2008.05756

[40] Ji, Q., Huang, J., He, W., & Sun, Y. (2019). Optimized deep convolutional neural networks for identification of macular diseases from optical coherence tomography images. *Algorithms*, 12(3), 51.

Notes on contributors

Syaiful Anam received a Doctor of Natural Science and Mathematics degree from Yamaguchi University, Japan in 2015. He also received his Bachelor's Degree in Mathematics from Brawijaya University, Indonesia in 2001 and his Master Degree from Sepuluh Nopember Institute of Technology, Indonesia in 2006. He is currently an assistant professor at Mathematics Department, Brawijaya University, Malang, Indonesia. His research includes data science, computational intelligence, machine learning, digital image processing, and computer vision. He can be contacted at email: syaiful@ub.ac.id.

Cynthia Ayu Dwi Lestari is currently pursuing a master's degree in mathematics with a focus on Artificial Intelligence and Data Science at Brawijaya University, Indonesia. She began her studies in 2022. She also holds a Bachelor's Degree in Mathematics from Brawijaya University, which she completed in 2020. Her current research areas of interest include data mining, machine learning, deep learning, and metaheuristics. She can be contacted by email at: cynthiaayu@student.ub.ac.id.

Fidia Deny Tisna Amijaya holds the Master degree in Mathematics from the Brawijaya University, Malang, Indonesia. He also received his Bachelor Degree in Mathematics from Brawijaya University, Indonesia in 2011 He is an Assistant Professor in Mathematics department, Faculty of Mathematics and Natural Sciences, Mulawarman University, Samarinda, Indonesia. His research interests are in applied mathematics, data mining and computational intelligence. He can be contacted at email: fidiadta@fmipa.unmul.ac.id.

Hagus Tarno is a professor at Department of Plant Pests and Diseases, Faculty of Agriculture, Universitas Brawijaya. His research focuses on insect ecology, insect taxonomy, biodiversity, insect identification, agricultural entomology, integrated pest management, applied entomology, species diversity, pest management, and plant Protection. He can be contacted by email at: h_gustarno@ub.ac.id

Zuraidah Fitriah is a lecturer at the Department of Mathematics, Faculty of Mathematics and Natural Sciences, Brawijaya University. She took a bachelor's and master's degree graduate Mathematics from Brawijaya University. Her research interest focuses on is computational intelligence and data science. She can be contacted by email at: zuraidahfitriah@ub.ac.id