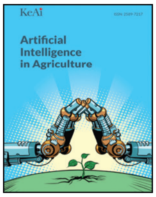
Contents lists available at ScienceDirect

Artificial Intelligence in Agriculture

journal homepage: http://www.keaipublishing.com/en/journals/artificial-intelligence-in-agriculture/



Deep learning models for automatic identification of plant-parasitic nematode

Nabila Husna Shabrina [a,*], Ryukin Aranta Lika [a], Siwi Indarti [b]

[a] Department of Computer Engineering, Universitas Multimedia Nusantara, Jl. Scientia Boulevard, Tangerang 15111, Indonesia
[b] Department of Plant Protection, Faculty of Agriculture, Universitas Gadjah Mada, Jl. Flora, Bulaksumur, Yogyakarta 55581, Indonesia

ARTICLE INFO

Article history:
Received 29 July 2022
Received in revised form 21 October 2022
Accepted 30 December 2022
Available online 4 January 2023

Keywords:
Augmentations
Classifications
Deep learning
Nematode
Optimization

ABSTRACT

Plant-parasitic nematodes cause various diseases that can be fatal to the infected plants. It causes losses to the agricultural industry, such as crop failure and poor crop quality. Developing an accurate nematode classification system is vital for pest identification and control. Deep learning classification techniques can help speed up Nematode identification as it can perform tasks directly from images. In the present study, four state-of-the-art deep learning models (ResNet101v2, CoAtNet-0, Effi- cientNetV2B0, and EfficientNetV2M) were evaluated in plant-parasitic nematode classification from microscopic image. The models were trained using a combination of three different optimizers (Adam, SGD, dan RMSProp) and several data augmentation with image transformations, such as image flip, blurring, noise addition, brightness, and contrast adjustment. The performance of the trained models was varied. Regarding test accuracy, EfficientNetV2B0 and EfficientNetV2M using RMSProp and brightness augmentation give the best result of 97.94% However, the overall performance of EfficientNetV2M was superior, with 98.66% mean class accuracy, 97.99%F1 score, 98.26% average precision, and 97.94% average recall.

1. Introduction

Nematodes are one of the living things with very abundant and diverse species (Abad et al., 2008). From the 26,000 types of nematodes identified, there are over 4100 plant- parasitic nematodes (Jones et al., 2013). Plant-parasitic nematodes are microorganisms that cause extensive damage and substantial crop yield losses (De and Elsen, 2007). They will continue to threaten agricultural production since they are found and infect various plants, including food crops, horticulture, and estate. Estimation of the world's economic crop losses caused by parasitic nematodes may reach 80 billion USD. In Indonesia, nematodes problems increase yearly since the most destructive plant-parasitic nematodes on potato, Globodera rostochiensis, was found (Indarti et al., 2004) and fallowed their distribution become wider in potato growing areas. New species of plant parasitic nematodes were also found in rice, garlic, and potato (Ajri et al., 2021; Indarti et al., 2018; Mutala'liah et al., 2018).

Plant parasitic nematodes are tiny life forms varying in size and shape. Due to their little measure and the assortment of genera, it becomes challenging to distinguish which sorts are shown in a specific test. The current strategy may be a conventional one which is time-consuming and helpless to error. Improvement technique for observation and classification of plant parasitic nematode is essential for pest identification and control. Therefore, the development of automatic image-based methods capable of identifying nematodes quickly and reliably is required.

Implementing deep learning techniques in image-based classification became an alternative for speeding up the nematode's identification process, as it is suitable for detecting discriminative nematodes features and handling large specimens. Various deep learning classification techniques are readily available (e.g., Keras and TensorFlow) and are easy to implement. The deep learning techniques are also proven to recognize pests and diseases, such as in leaves, with high accuracy by providing sufficient data (Li et al., 2021).

Convolutional Neural Networks (CNN) via transfer learning were implemented for multi-crop leaf disease image classification. This proposed research has achieved accuracy for grape and tomato leaf disease classification by 98.40% and 95.71%, respectively (Paymode and Malode, 2022). Data augmentation and extension of deep learning VGG-16 pre-trained network model were applied for the multi-pest classification of Indonesian mango leaves image. The overall accuracy achieved from those training is 73% on the validation dataset and 76% for the testing data (Kusrini et al., 2020). Improved CNN was built for the real-time detection of apple leaf diseases. Using a dataset of 26,377 images of diseased apple leaves, the implementation of INAR-SSD (SSD with Inception module and Rainbow concatenation) gave a detection

* Corresponding author.
E-mail address: nabila.husna@umn.ac.id (N.H. Shabrina).

https://doi.org/10.1016/j.aiia.2022.12.002

performance of 78.80% mAP on the dataset (Jiang et al., 2019). The correlation effect of several optimization techniques on CNN performance for olive disease classification was also observed. It was found that the highest rate in the experiment without data augmentation was 92.59% (Raouhi et al., 2022).

There are several works implemented for nematodes classification using deep learning. Xception models were trained with different initial conditions on the dataset of various stages of nematodes (both juvenile and adult). It is found that the model with pretrained weights on ImageNet performed the best compared to random weight and no weight initialization. However, due to hardware limitations, the authors could not train better but heavier models, and the images in the dataset lack variation (Uhlemann et al., 2020). Another research focused on increasing the availability of nematodes datasets to the public while providing species recognition benchmarks by testing multiple state-of-the-art deep learning models on the dataset. The dataset consists of 2769 nematodes samples classified manually into 19 classes. From this benchmark, it is found that the ResNet family model has the highest performance in terms of accuracy (Li et al., 2021). A new proposed CNN model, defined as NemaNet, was used to identify a new dataset for nematodes soybean crop in Brazil. The NemaNet model reached 96.99% accuracy, while the best fold was 98.03% (Abade et al., 2022).

Previous research does not include many other nematodes species commonly found in Indonesia. This research provides data that contains these species and assesses the performance of a state-of-the-art deep learning model for use in Indonesia. This paper compares four state-of-the-art deep learning models (ResNet101v2, CoAtNet-0, EfficientNetV2B0, and EfficientNetV2M) trained on nematodes dataset species commonly found in Indonesian soil. Data augmentation was employed to determine if image transformation can further increase the accuracy of the tested models. The effect of several optimizer functions was also observed on the performance result. The contributions of the paper are listed as follows:

- To build an additional image dataset for species of nematodes commonly found in Indonesia
- To provide model benchmark comparison for real classification system implementation
- To implement 4 state-of-the-art deep learning techniques (ResNet101v2, CoAtNet-0, EfficientNetV2B0, and EfficientNetV2M) for multi-class classification of plant-parasitic nematodes
- To evaluate the performance of 4 state-of-the-art deep learning models with respect to data augmentation and optimizer function
- To design and develop a web application to classify plant-parasitic nematodes

The structure of this paper is organized as follows. Section 2 presents Materials and Method, starting with the datasets and then describing the implemented deep learning architecture. Section 3 presents the detailed Result and Discussion on the comparison of 4 state-of-the-art deep learning models for multi-class classification of plant-parasitic nematodes. Finally, in Section 4, the paper concludes with a remark on the model comparison and some perspective for future work.

2. Materials and methods

2.1. Proposed research

The schematic in Fig. 1 shows an overview workflow for multi-class plant-parasitic nematodes classification. Initially, plant-parasitic nematodes dataset is collected and classified into several classes. Data preprocessing was applied to the dataset using edge detection, cropping, and converting into grayscale. Image flip, noise addition, image blurring, brightening and contrast are applied in single augmentation to enhance the size of the dataset. The photos are then used as input to train the model using several augmentation and optimizer function combinations. The result was recorded and analyzed based on the metric performance. The best performance is then deployed as a web-based system.

2.2. Nematodes dataset

2.2.1. Data collection

All nematodes were collected from an infected agricultural plant in Indonesia to gather the required pictures. Nematodes were isolated from plant root and rhizosphere soil and then proceeded to the specimen for further observation. First, soil samples were extracted for nematodes presence utilizing the adjusted Whitehead Tray method. Nematodes extractions from root samples were cut into 1 mm pieces; at that point were laid on the channel paper on the nylon screen bolstered by the altered bucket, allowing water to join the root tests. The nematodes swimming out to the water were collected on the tubes to assist both morphological perceptions. The plunging was at room temperature for 24 h. This method alluded to Whitehead Plate Method (Southey, 1986) with adjustment. Before further observation, nematodes were killed, and fixed with cooled Formalin Acetic Acid (FAA) referred to (Southey, 1986). Finally, the nematodes morphological characters of each genus were observed using a light microscope Olympus CX 31 with a magnification of 40–1000. The image was captured with optical connected with the microscope and laptop.

Dataset consists of 957 nematode samples that were classified into 11 classifications. This dataset represents nematodes commonly found in Indonesian soil. Table 1 shows some statistics of nematodes genus distribution in the dataset. As shown in the given table, the genus distribution is diverse. Genus with less than 65 images accounts for 45% of the whole class. This happened due to the limited nematodes distribution area and only a specific host that could develop the nematode. Fig. 2 presents sample nematodes images from the dataset.

2.2.2. Data preprocessing

Obtained samples are then preprocessed with a similar method used for previous work (Lu et al., 2021). Samples were cropped based on edge detection in an attempt to equalize which regions the nematodes specimen exists in the image. This also reduces redundant information by cutting empty spaces. Samples were converted into grayscale images since classification is solely based on morphological features of the samples. All images are then resized to 224 × 224 to fit the input size of all tested models. The data preprocessing steps are shown in Fig. 3.

2.2.3. Data augmentation process

Data augmentation processes often increase image diversity in a dataset by applying image manipulation techniques or synthetically creating new data. This process aims to increase generalization, prevent model overfitting to the training dataset and lead to better performance (Shorten and Khoshgoftaar, 2019). Common manipulation techniques applied in data augmentation processes include image transformation by flipping, rotation, translation, noise addition, blurring, change in brightness, contrast, and other color space transformations. This will increase the discriminative feature of each class that model needs to learn to not overfit. Image manipulation techniques need to be chosen based on the nature of the sample, as some image transformations can result in contextually incorrect data that does not reflect the nature of the real specimen. With the correct augmentation techniques, model performance can be improved, especially on the dataset with imbalanced class distribution. However, the augmentation process increases the time it required to train the model as it increases the size of the dataset. Image transformation techniques also affect model accuracy, as some techniques decrease model accuracy (Shijie et al., 2017).

This research applied augmentation on the fly (online augmentation) to increase data diversity. Image manipulation techniques used are flipping (horizontally and vertically), noise addition, blurring, change in brightness, and change in contrast. Some manipulation

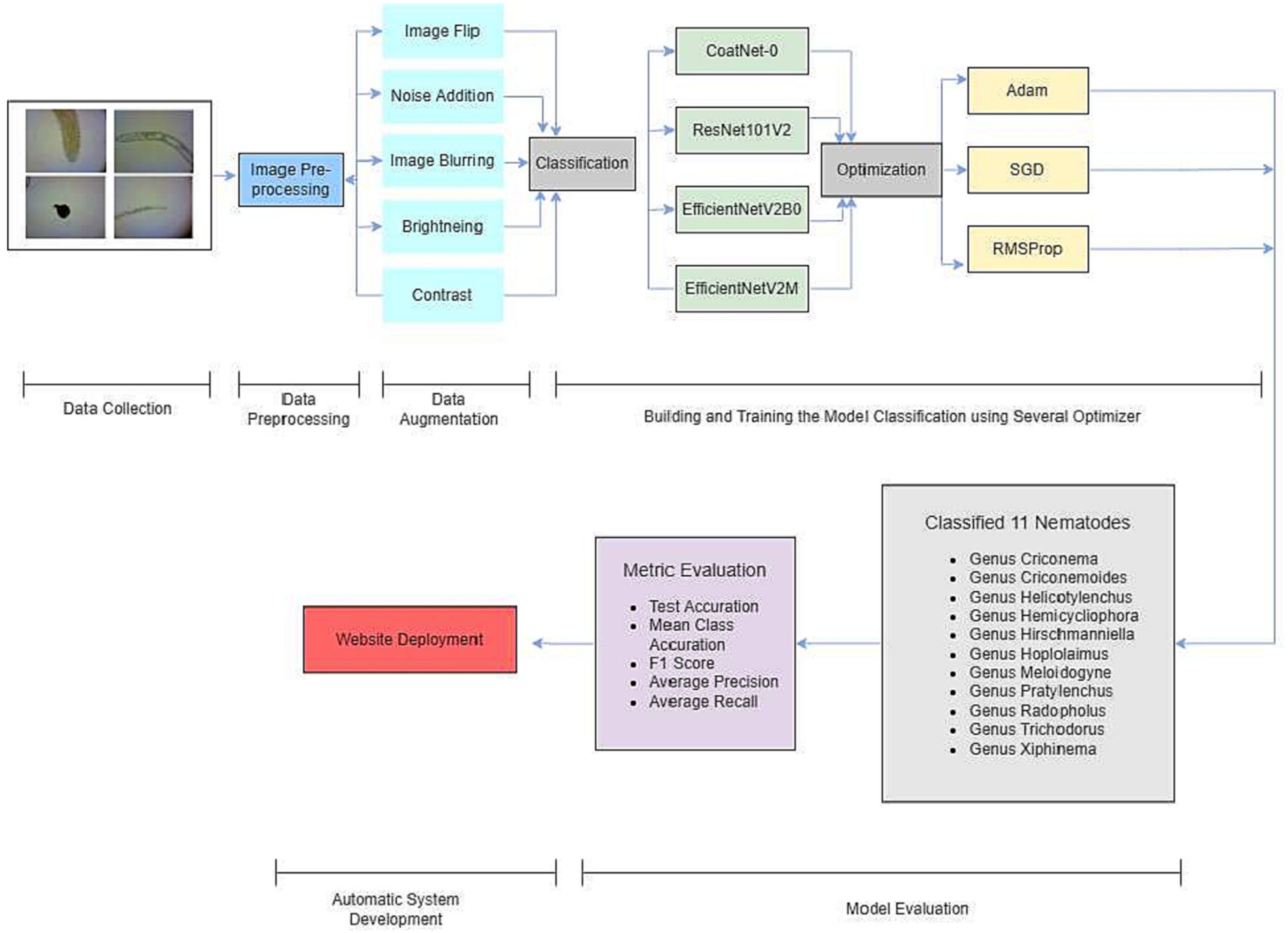


Fig. 1. Research workflow.

techniques such as translation and rotation were not chosen due to the concern that augmentation results could hide important discriminative features of the sample, which can decrease model accuracy. The following list describes the applied image augmentation.

1. Image flip is applied randomly on images in the dataset, and the flip type is chosen randomly as well, whether no flip is applied, the horizontal flip is applied, the vertical flip is applied, or both flips are applied.
2. Noise addition is performed by adding white gaussian noise to the image sample with a mean value of 0 and a standard deviation of 0.15. This is applied randomly with a probability of 50% on each sample.
3. Image blurring is performed by applying a gaussian filter with a kernel size of $\{3 \times 3\}$ and standard deviation value of 1. This is applied randomly with a probability of 50% on each sample.
4. Change in brightness is applied by the increase of brightness with brightness parameter of random value from 0 to 0.3
5. Change in contrast is applied by increase of contrast with contrast parameter of random value from 0 to 3.

Table 1
Nematode genus distribution in the dataset.

Genus	No. of samples
Genus Criconema	3
Genus Criconemoides	103
Genus Helicotylenchus	135
Genus Hemicycliophora	6
Genus Hirschmanniella	130
Genus Hoplolaimus	151
Genus Meloidogyne	211
Genus Pratylenchus	116
Genus Radopholus	12
Genus Trichodorus	30
Genus Xiphinema	60
Total	**957**

The datasets and their augmented variations are then used to train the models. In evaluating the model performance, the dataset is split into a training set, validation set, and test set with a ratio of 80:10:10. To summarize, the dataset consists of 957 nematodes sample with 11 classifications, split into 766, 96, and 95 for training, validation, and test respectively. Note that augmentation processes are performed only on the training set of the dataset.

2.3. Deep learning architecture

Nematodes classifications are performed using several state-of-the-art models in terms of image classification problems, namely ResNetV2, CoAtNet, and EfficientNetV2. Model families are chosen based on their performance on Image Classification on ImageNet Benchmark (Keras, 2021; Stojnic et al., 2022). Based on hardware limitations and the size of datasets used in the experiments, a specific variant of models applied are ResNet101V2, CoAtNet-0, EfficientNetV2B0, and EfficientNetV2M.

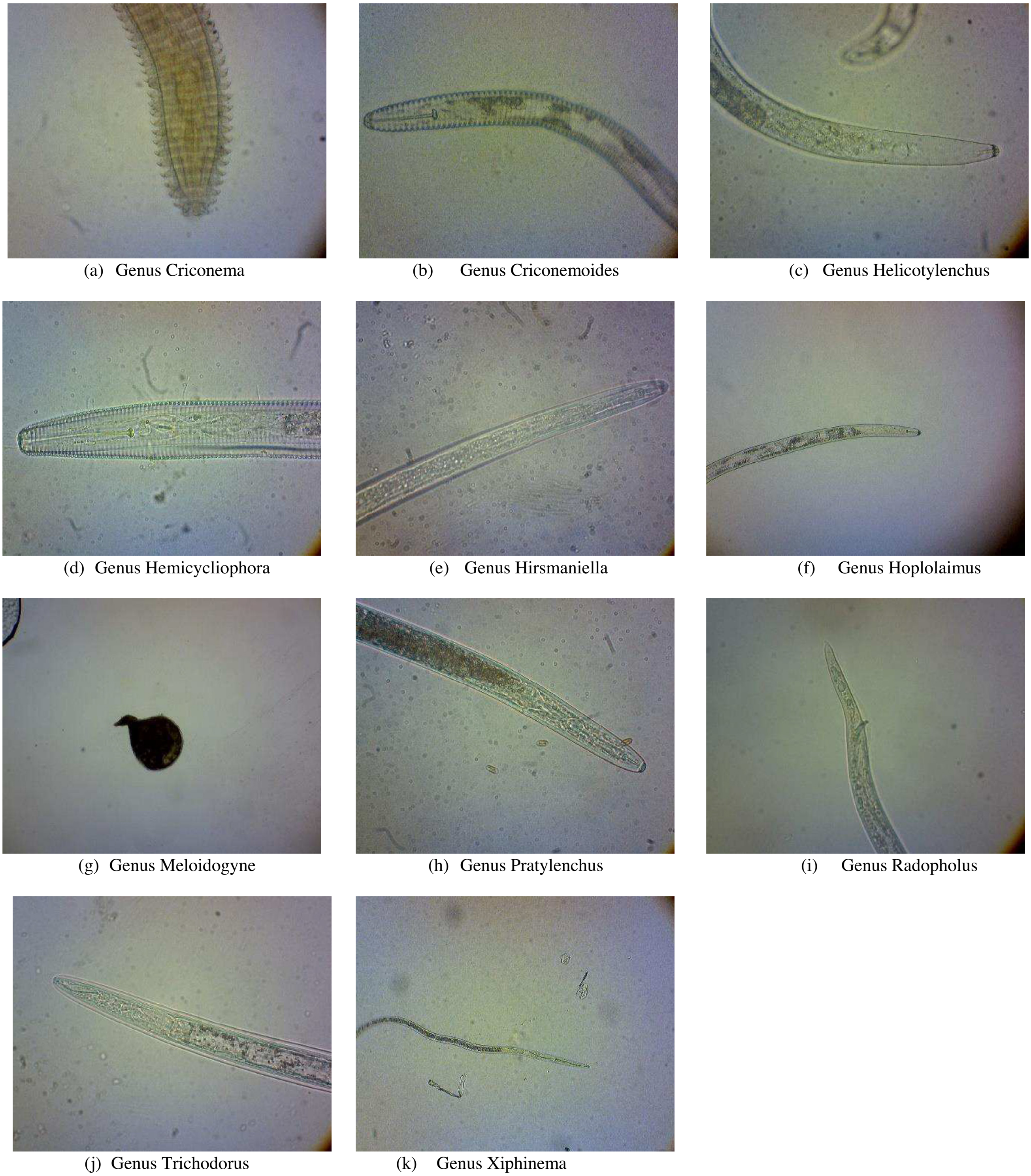

(a) Genus Criconema
(b) Genus Criconemoides
(c) Genus Helicotylenchus
(d) Genus Hemicycliophora
(e) Genus Hirsmaniella
(f) Genus Hoplolaimus
(g) Genus Meloidogyne
(h) Genus Pratylenchus
(i) Genus Radopholus
(j) Genus Trichodorus
(k) Genus Xiphinema

Fig. 2. Sample of plant-parasitic nematodes used in this study

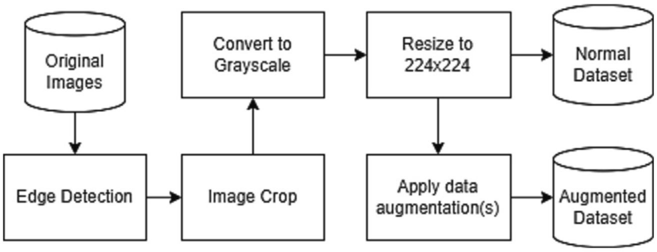


Fig. 3. Data pre-processing flowchart.

2.3.1. Residual networks

The original Residual Networks (ResNet) architecture is a CNN based on VGG architecture that employs residual learning in its building blocks by adding shortcut connections that skips one or more layers. This alleviates the degradation problem that VGG architecture has when a deeper model is used (He et al., 2016a). The ResNetV2 is the improved version of ResNet architecture that improves the speed of data propagation on each residual block by utilizing identity mapping as the skip connections and after-addition activation. This eases

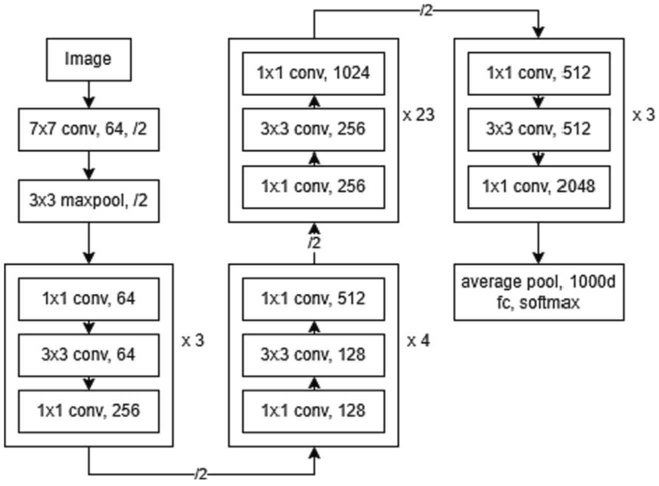


Fig. 4. ResNet101V2 network architecture.

optimization loss compared to the baseline ResNet and improves the regularization of the models. Several variations of the ResNet models, based on the number of layers inside the network, are ResNetV2–50, ResNetV2–101, and ResNetV2–152 2 (He et al., 2016b). The ResNet model network architecture implanted in this paper is presented in Fig. 4.

2.3.2. Convolution and attention networks

The Convolution and Attention Networks (CoAtNet) architecture is a family of hybrid models that combines CNN and Transformer architecture to achieve better generalization and larger capacity. Transformer architectures with a self-attention mechanism have a higher model capacity that can benefit from larger and more diverse datasets, meanwhile, convolutional architectures have a better generalization and faster-converging speed. This architecture consists of 5 stages (1 convolutional layer, 2 MBConv layers, and 2 transformer layers), with different properties depending on the model variant. The main variation of the CoAtNet model consists of 5 basic variations (CoAtNet-0 to CoAtNet-4) and 3 variations with different block parameters (CoAtNet-5 to CoAtNet-7) (Dai et al., 2021). The CoAtNet network architecture is presented in Fig. 5.

2.3.3. EfficientNet

The baseline EfficientNet architecture is based on a new scaling method for increasing model capacity by scaling the dimensions of width, depth, and resolution of the model using a simple compound coefficient (Tan and Le, 2019). Neural Architecture Search (NAS) is then used to design a new baseline model using MBConv blocks and scale it using the compound coefficient to create EfficientNet (Elsken et al., 2019). This new family of models achieve state-of-the-art performance on ImageNet dataset while having a far smaller model and faster converging speed (Tan and Le, 2019). This family of models are then improved even further by employing Fused-MBConv. The specific combination of MBConv and Fused-MBConv used in the network improves training speed and decreases the model size. The new and improved models are then called EfficientNetV2, which with a specific training method, can achieve 5× – 11× faster-converging speed compared to other state-of-the-art models with up to 6× smaller in size (Tan and Le, 2021). EfficientNetV2 implemented in this research was B0 and M version. EficcientNetV2B0 has better trade-off on accuracy and FLOPs, while EfficientNetV2M reduces parameter and FLOPs but run faster in training and inference compared to the V1-B7 version (Tan and Le, 2021). The applied EfficientNetV2B0 and EfficientNetV2M network architecture in this research are presented in Fig. 6 and Fig. 7, respectively.

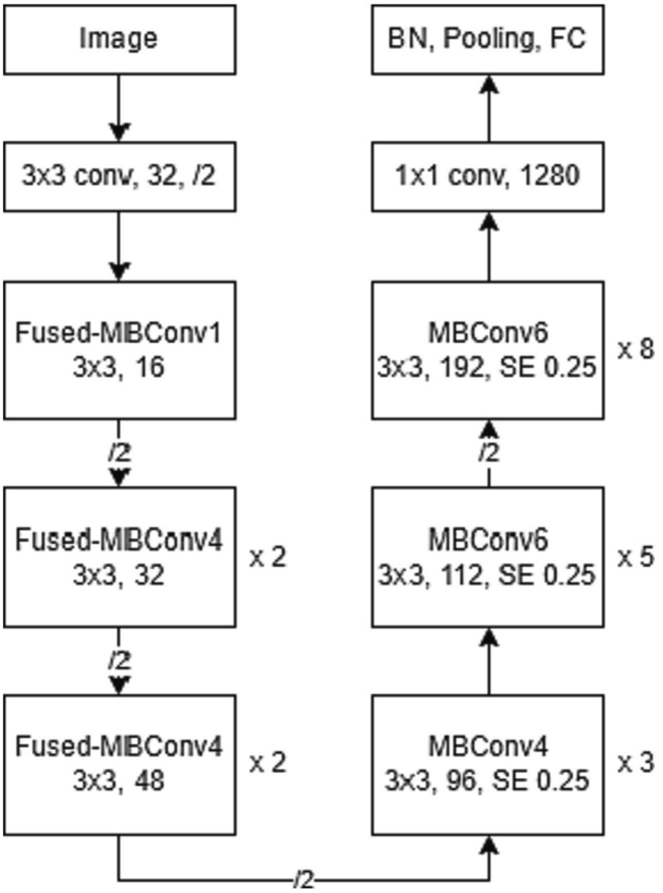


Fig. 6. EfficientNetV2B0 network architecture.

2.3.4. Proposed classification networks

The weights of each model are preserved from the pre-trained model based on ImageNet dataset, with its ImageNet classification head layer removed. Feature extraction vectors of each model are then

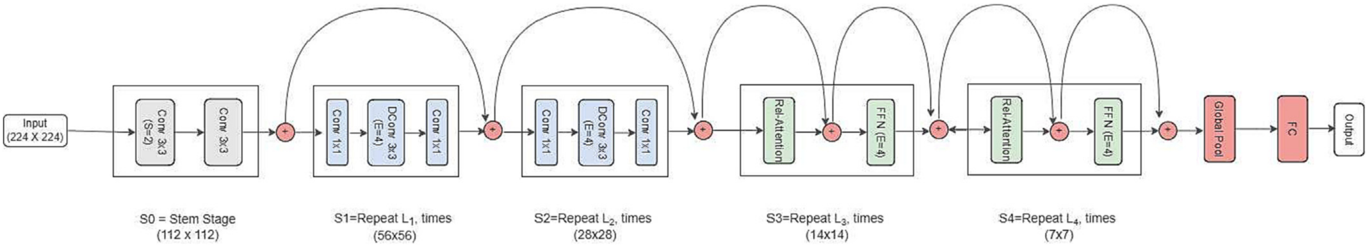


Fig. 5. CoAtNet network architecture.

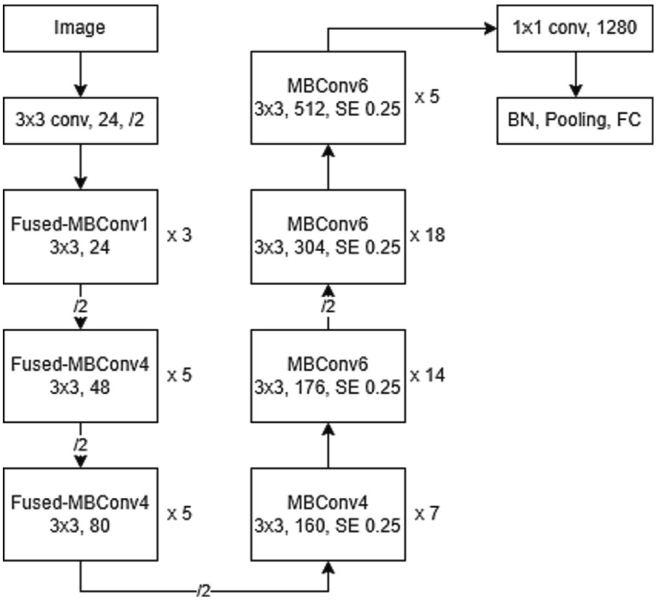


Fig. 7. EfficientNetV2M network architecture.

connected to a dense layer with Soft-Max activation to perform classification, with class count adjusted to the dataset used, which is 11 genus classifications. The proposed classification layer is depicted in Fig. 8.

2.3.5. Optimization techniques

Three different optimizers were used in the training process to determine the best optimizer for each type of model and whether architecture or dataset diversity affects the optimizers' use. This also minimize unfair advantage of some model in case of hyperparameter fits the exact requirement of a specific model-dataset combination that can achieve better performance. Optimizers that were used in the experiments are Adam, SGD, and RMSprop. Adam is chosen based on its favorable performance compared to other optimization method, as it tends to work well in practice (Kingma and Ba, 2014). SGD is chosen to represent gradient descent optimizers and is often used to produce the state-of-the-art results in some deep learning research. SGD also produce better generalization performance (Zhou et al., 2020). RMSprop was chosen as it is suitable for optimizing non-stationary and non-convex problems (Sun et al., 2020).

For consistency, each model will be trained using the same hyperparameter value, including the same activation for dense layer (SoftMax), batch size of 32, and same optimizer parameters (learning rate of 0.001 for Adam and RMSprop, 0.01 for SGD; the momentum of 0 for SGD and RMSprop; a beta value of 0.9–0.999 for Adam), input size of 224x224x3, and sparse cross-entropy loss function for multi-label classification problem. The training epoch was set to 100.

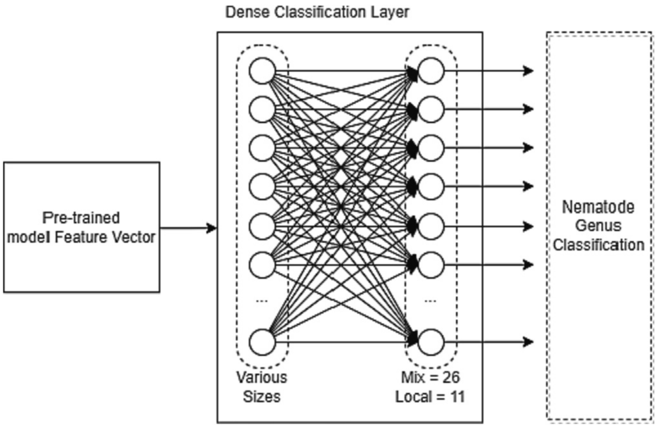

Dense Classification Layer
Pre-trained model Feature Vector
...
...
Various Sizes
Mix = 26
Local = 11
Nematode Genus Classification

Fig. 8. Proposed model classification layer.

2.3.6. Evaluation metrics

Several evaluation metrics were employed in this experiment. Test Accuracy is used to evaluate the average accuracy of the model over all images on the test set. The F1-score metric is a classification evaluation metric based on the harmonic mean of precision, and recall is often used when the class distribution in the dataset is imbalanced. Weighted F1-score metric was used to evaluate the imbalanced datasets. Average precision and recall were recorded for data completeness. The formula for test accuracy, F1 score, average precision and recall are given in Eqs. (1)–(4), respectively (Alsaggaf et al., 2020).

$$Test\ Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1\ score = \frac{2\ x\ Precision\ x\ Recall}{Precision + Recall} \quad (4)$$

where TP – True Positive; FP – False Positive; TN – True Negative; FN – False Negative. Another evaluation metric applied in this experiment is Mean Class Accuracy (Diker et al., 2019; Toğaçar et al., 2021). It is used to evaluate the accuracy of each class that could indicate if the model learns discriminative features of each class compared to other models. The formula is given in Eq. (5)

$$Mean\ Class\ Accuracy = \frac{1}{c}\sum_{i=1}^{c}\frac{1}{n_i}\sum_{j=1}^{n_i} a_j^i \quad (5)$$

where c is the number of classes (11 classes for nematodes genus), n_i is the number of the image in $i - th$ class, a_j^i is the accuracy for image number $j - th$ in $i - th$ class.

2.3.7. System implementation

Code implementation of each model is achieved using Keras (Chollet, 2015) and TensorFlow library that provides fully built models with pretrained weights on ImageNet dataset. Model training and inference will be performed on the Google Colab Notebook (Pro Version), which has a minimum specification of NVIDIA P100 or T4 as GPU, memory up to 25GB, and CPU Xeon Processor @ 2.3GHz based on availability.

3. Result and discussion

3.1. Model performance with augmentation

Fig. 9 present the sample data augmentation of the Genus Trichodorus using image augmentation method. Each model was trained with the specified hyperparameter and optimizers on the datasets augmented with an image transformation technique. The dataset is used to gauge the performance of Nematodes commonly found in Indonesian soil. Model performances on the test dataset are then compared to the model's base performance on the dataset without augmentation.

Table 2 presents ResNet101V2 model performance using augmentation. The "%Test" column is the accuracy of the model classification results on the test dataset. The test dataset represents data that the model has never seen before. The ResNet101v2 model trained with an

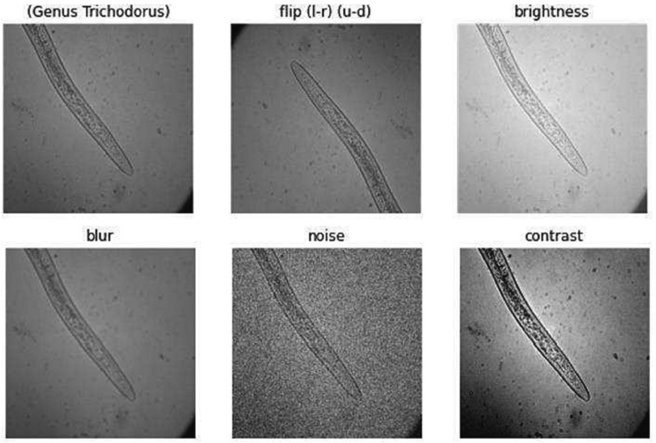


Fig. 9. Sample data of the Genus Trichodorus specimen resulting from image augmentation.

augmentation dataset resulted in varying performance changes. The highest accuracy of 93.81% is obtained when trained without any augmentation, using SGD optimizer, whereas using RMSprop optimizer resulted in the lowest accuracy out of all model combinations, with an accuracy of 18.56%.

Fig. 10a shows the impact of augmentation on ResNet101V2 %Test Accuracy compared to the normal dataset. Using Adam Optimizer, Gaussian Blur Augmentation tends to improve accuracy. The augmentation method implemented using SGD Optimizer leads to decreasing 13.4% to 43.4 in %test accuracy, while applying RMSProp Optimizer increases the %test accuracy from 3.09% to 14.43%.

The CoAtNet-0 model trained with an augmentation dataset resulted in small performance increases and varying performance decreases, as seen in Table 3. The highest accuracy of 96.91% is achieved with the normal dataset using SGD optimizer, while the lowest accuracy of 70.10% is obtained with the contrast augmentation dataset using RMSprop optimizer.

The impact of augmentation on CoAtNet-0 %Test Accuracy is given in Fig. 10b. Using Adam Optimizer, Flip and Gaussian Noise decreases the accuracy while another augmentation tends to improve accuracy by 1% to 5.12%. The augmentation method implemented using SGD Optimizer leads to decreasing 5.15% to 13.4% in %test accuracy. RMSProp Optimizer applied with Brightness and Gaussian Blur increased the %test accuracy by 5.15% and 4.12%, respectively, while other augmentations decreased the %test accuracy.

Table 2
Results obtained from ResNetV2–101 Model.

Optimizer	Data augment	% Test accuracy	Mean class accuracy	F1 score	Average precision	Average recall
Adam	Normal	88.66%	0.7662	0.882	0.8955	0.8866
	Flip	77.32%	0.7715	0.7743	0.8048	0.7732
	Brightness	87.63%	0.9097	0.8758	0.8877	0.8763
	Contrast	81.44%	0.7996	0.8151	0.8294	0.8144
	Gaussian Blur	89.69%	0.8148	0.8948	0.9051	0.8969
	Gaussian Noise	86.60%	0.8697	0.8641	0.8727	0.866
SGD	Normal	93.81%	0.9329	0.9375	0.9398	0.9381
	Flip	69.07%	0.4808	0.6863	0.7065	0.6907
	Brightness	80.41%	0.6813	0.7929	0.7925	0.8041
	Contrast	62.89%	0.5768	0.6311	0.6485	0.6289
	Gaussian Blur	50.52%	0.4774	0.5139	0.5508	0.5052
	Gaussian Noise	68.04%	0.752	0.6848	0.766	0.6804
RMSprop	Normal	18.56%	0.0909	0.0581	0.0344	0.1856
	Flip	21.65%	0.1	0.0771	0.0469	0.2165
	Brightness	32.99%	0.2529	0.2972	0.326	0.3299
	Contrast	22.68%	0.1	0.0846	0.052	0.2268
	Gaussian Blur	24.74%	0.1026	0.1063	0.0677	0.2474
	Gaussian Noise	22.68%	0.087	0.0877	0.0543	0.2268

The EfficientNetV2B0 model trained with augmentation mostly decreased performance, as shown in Table 4. The highest accuracy of 97.94% is obtained with brightness augmentation using RMSprop optimizer, whereas the lowest accuracy of 65.98% is obtained with the contrast augmentation using SGD as the optimizer.

The impact of augmentation on EfficientNetV2B0 %Test Accuracy is presented in Fig. 10c. The increasing test accuracy by 1.03% occurred when the model applied RMSProp with brightness augmentation. Another method leads to a decrease in the %test accuracy from 4.12% to 30.93%.

The EfficientNetV2M model trained with augmentation mostly decreased performance, as seen in Table 5. The highest accuracy of 97.94% is obtained from the model which applied brightness augmentation and RMSprop optimizer. The lowest accuracy of 76.29% is obtained with the noise augmentation dataset using SGD as the optimizer.

The impact of augmentation on EfficientNetV2M %Test Accuracy is presented in Fig. 10d. The result is similar to EfficientNetV2B0 in Fig. 10c but with a lower range. The increasing %test accuracy of 6.19% occurred when the model applied RMSProp with brightness augmentation. Another method tends to give lower results in the %test accuracy, ranging from 1.03% to 21.65%.

Applying augmentation to the dataset resulted in a more varied performances change. Notable performance improvements are on the ResNet101v2 and CoAtNet-0 model with RMSprop optimizer and CoAtNet-0 with Adam optimizer. Nevertheless, it should be noted that ResNet101v2 with RMSprop has terrible base performance compared to other optimizers (18.56% with RMSprop, compared to 88.66% with Adam and 93.81% with SGD).

The best test accuracy is achieved by the EfficientNetV2B0 and EfficientNetV2M model with RMSprop optimizer, trained on the dataset with brightness augmentation. This combination achieved 97.94% accuracy. The CoAtNet family scored on average lower test accuracy difference than other models, indicating higher generalization capability towards augmented data than other family models.

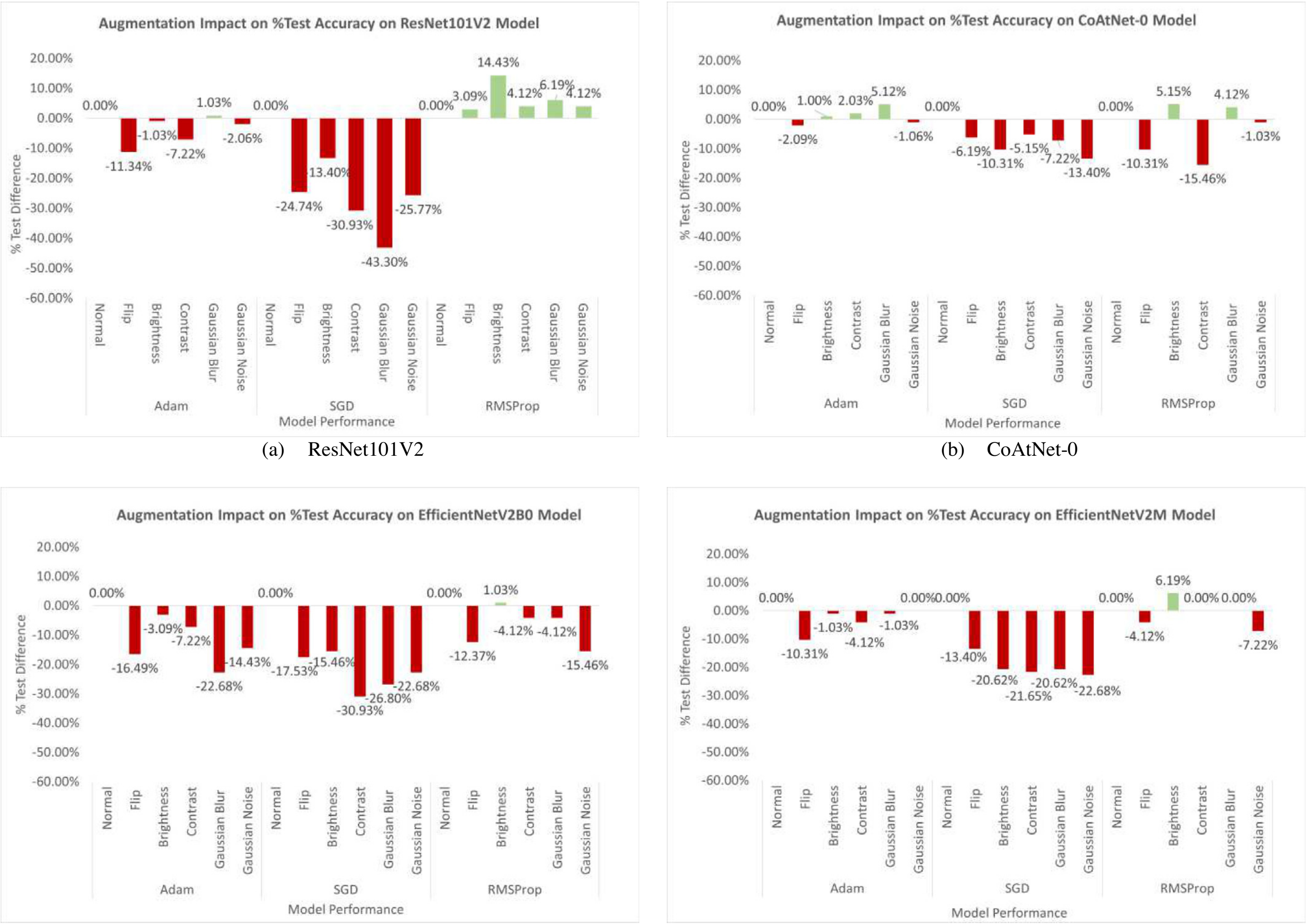
Based on the obtained data, it can be observed that there is a strong relationship between using specific image manipulation technique as an augmentation method and the combination of the model-optimizer that would determine its performance. Choice of augmentation method, optimizers, and parameters are essential in achieving the best result a model can perform.

3.2. Augmentation method impact on model performance

3.2.1. Flip augmentation

From the data gathered, 11 out of 12 models trained on the flip dataset show a decrease in model accuracy. This is presumably due to the relatively small dataset size causing the data to have a similar orientation distribution. All model families have up to ten million parameters. Therefore, the model is not only performing classification based on the discriminatory features of each genus but also based on a similar orientation between each classification. Adding data with different orientations causes the model to be unable to classify data based on orientation and decreases performance. This indicates models overfit the data's environmental aspect, such as data acquisition methods, rather than each genus' discriminative features.

Observation of the performance of ResNet101v2 using Adam optimizer, visualization of the classification layer activation of the Genus Xiphinema class shows the discriminatory orientation of the class. When trained on the dataset without augmentation (normal dataset), the visualization shows the shapes commonly found in the Genus specimen. However, when it was trained using flip augmentation, the layer activation area was not as clear as the normal dataset, as seen in Fig. 11. This shows that according to the model, important features of the genus are aligned in that specific way. In the dataset, the Genus Xiphinema data does have some orientation as described in the activation property of the model with the normal dataset.



(a) ResNet101V2

(b) CoAtNet-0

(c) EfficientNetV2B0

(d) EfficientNetV2M

Fig. 10. Impact of augmentation method on %test accuracy.

Only the ResNet101v2 with RMSprop optimizer got higher accuracy in terms of performance improvement. However, this increase in accuracy can be attributed to the impact of using RMSprop as an optimizer on ResNet101v2 model instead of the augmentation itself because all other results of this model-optimizer combination result in unsatisfactory accuracies.

3.2.2. Brightness augmentation

Applying brightness augmentation resulted in varying performances for each model-optimizer combination. This augmentation caused a small change in accuracy for models with Adam optimizer (±1% change in accuracy) and an increase in the model with RMSprop optimizer (+5.15% to +14.43%). Still, it showed a decrease in models with SGD

Table 3
Results obtained from CoAtNet-0 Model.

Optimizer	Data augment	% Test accuracy	Mean class accuracy	F1 score	Average precision	Average recall
Adam	Normal	85.60%	0.694	0.8576	0.8547	0.866
	Flip	83.51%	0.8072	0.8418	0.8637	0.8351
	Brightness	86.60%	0.8893	0.8639	0.8823	0.866
	Contrast	87.63%	0.7987	0.8798	0.897	0.8763
	Gaussian Blur	90.72%	0.9244	0.9056	0.9147	0.9072
	Gaussian Noise	84.54%	0.8186	0.8455	0.8739	0.8454
SGD	Normal	96.91%	0.9747	0.9691	0.9697	0.9691
	Flip	90.72%	0.8209	0.9007	0.9047	0.9072
	Brightness	86.60%	0.8805	0.8664	0.8793	0.866
	Contrast	91.75%	0.939	0.9162	0.9255	0.9175
	Gaussian Blur	89.69%	0.8924	0.8965	0.9078	0.8969
	Gaussian Noise	83.51%	0.8286	0.8428	0.8808	0.8351
RMSprop	Normal	85.57%	0.7899	0.8562	0.8661	0.8557
	Flip	75.26%	0.8027	0.7533	0.7676	0.7526
	Brightness	90.72%	0.9014	0.9029	0.9158	0.9072
	Contrast	70.10%	0.7625	0.7021	0.7186	0.701
	Gaussian Blur	89.69%	0.8399	0.8889	0.8919	0.8969
	Gaussian Noise	84.54%	0.8663	0.8453	0.8549	0.8454

Table 4
Results obtained from EfficientNetV2B0 Model.

Optimizer	Data augment	% Test accuracy	Mean class accuracy	F1 Score	Average precision	Average recall
Adam	Normal	94.85%	0.9709	0.9486	0.9507	0.9485
	Flip	78.35%	0.7335	0.7727	0.7833	0.7835
	Brightness	91.75%	0.8315	0.9137	0.9201	0.9175
	Contrast	87.63%	0.8212	0.8793	0.8979	0.8763
	Gaussian Blur	72.16%	0.6657	0.7178	0.8044	0.7216
	Gaussian Noise	80.41%	0.7338	0.8006	0.8184	0.8041
SGD	Normal	96.91%	0.9755	0.9691	0.9724	0.9691
	Flip	79.38%	0.8203	0.7941	0.8017	0.7938
	Brightness	81.44%	0.753	0.8114	0.8406	0.8144
	Contrast	65.98%	0.5918	0.6576	0.6946	0.6598
	Gaussian Blur	70.10%	0.6266	0.6983	0.7085	0.701
	Gaussian Noise	74.23%	0.7482	0.7355	0.7512	0.7423
RMSprop	Normal	96.91%	0.9744	0.9689	0.9731	0.9691
	Flip	84.54%	0.8662	0.8421	0.868	0.8454
	Brightness	97.94%	0.8923	0.9747	0.9716	0.9794
	Contrast	92.78%	0.952	0.9272	0.9293	0.9278
	Gaussian Blur	92.78%	0.8102	0.9168	0.9256	0.9278
	Gaussian Noise	81.44%	0.8748	0.8014	0.8763	0.8144

Table 5
Results obtained from EfficientNetV2M Model.

Optimizer	Data augment	% Test accuracy	Mean class accuracy	F1 score	Average precision	Average recall
Adam	Normal	92.78%	0.9397	0.9289	0.9351	0.9278
	Flip	82.47%	0.712	0.8175	0.8164	0.8247
	Brightness	91.75%	0.9247	0.9222	0.932	0.9175
	Contrast	88.66%	0.8012	0.8852	0.9008	0.8866
	Gaussian Blur	91.75%	0.8867	0.9143	0.9249	0.9175
	Gaussian Noise	92.78%	0.9459	0.9275	0.9342	0.9278
SGD	Normal	89.69%	0.8704	0.8953	0.9049	0.8969
	Flip	76.29%	0.6979	0.7613	0.7807	0.7629
	Brightness	69.07%	0.5759	0.6891	0.7164	0.6907
	Contrast	68.04%	0.5661	0.6713	0.7081	0.6804
	Gaussian Blur	69.07%	0.622	0.6764	0.7086	0.6907
	Gaussian Noise	67.01%	0.6278	0.6692	0.7015	0.6701
RMSprop	Normal	91.75%	0.9178	0.9183	0.9309	0.9175
	Flip	87.63%	0.875	0.8668	0.8843	0.8763
	Brightness	97.94%	0.9861	0.9799	0.9826	0.9794
	Contrast	91.75%	0.9107	0.9159	0.9232	0.9175
	Gaussian Blur	91.75%	0.9136	0.9168	0.9217	0.9175
	Gaussian Noise	84.54%	0.8319	0.841	0.8787	0.8454

optimizer (−10.31% to −20.62%). Brightness augmentation tends to improve model performance or decrease less accuracy compared to other augmentation methods. The increase is due to the dataset's high brightness variation between specimens.

3.2.3. Contrast augmentation

Using contrast augmentation on the dataset resulted in decreased model accuracy. Only 2 out of 12 observed data have higher accuracy than their normal data counterpart, namely ResNet101v2 with RMSprop and CoAtNet-0 with Adam optimizer. Based on observations, increasing contrast too much will result in contextually incorrect Nematodes images, i.e., Nematodes images that cannot be obtained from real-world photos. The increasing in contrast will also increase the clarity of some features of specimens that are not discriminatory between Genus. This could cause the model to learn irrelevant features of a Genus.

3.2.4. Gaussian blur augmentation

Image blurring with gaussian blur is expected to help models learn about the general morphological characteristics of each Genus of Nematodes. Augmentation of gaussian blur caused some model-optimizer combinations to have increased accuracy. Based on observations, the process of image blurring in the dataset produces the variations needed by the model to study the features of the target class nematodes. This may be due to the selection of parameter values in the dataset augmentation process, which affects the strength of the blurring and the size of the gaussian filter kernel used.

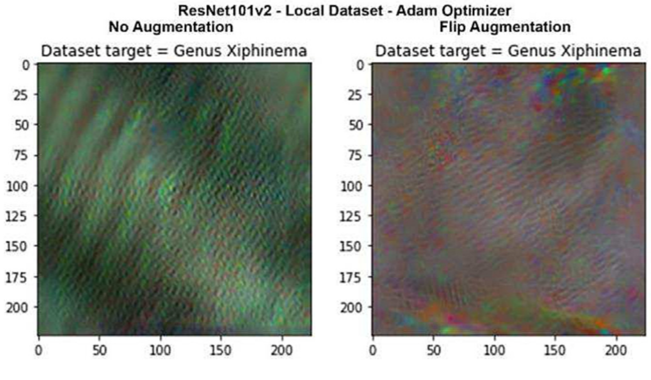


Fig. 11. Comparison of classification layer activation visualization against “Genus Xiphinema” class on Resnet101V2 model without augmentation (left) and with image reversal augmentation (right).

3.2.5. Gaussian noise augmentation

Applying Gaussian noise augmentation to the dataset results in decreased accuracy. An increase in accuracy only occurs on the ResNet101v2 with the RMSprop optimizer. The decrease in accuracy is more significant compared to other types of augmentations. This is presumably due to the addition of noise covers the discriminatory characteristics of each Nematodes class, as well as the selection of parameters from the augmentation process. With the resolution used (224 × 224), the addition of noise is considered ineffective in increasing the model's accuracy.

3.2.6. Optimizer impact on model performance

It should be noted that in this research, each optimizer between models uses the same hyperparameters, a learning rate of 0.001 for Adam and RMSprop, and 0.01 for SGD; momentum value of 0 for SGD and RMSprop; a beta value of 0.9–0.999 for Adam; epsilon value of 1e-07 for Adam and RMSprop; and a rho value of 0.9 for RMSprop. Parameter value selection is based on the general usage of each optimizer, and some of them are the default parameter values from the TensorFlow library. Using different parameter values impacts the model's performance in learning essential features from the dataset. Based on data observation, several relations were obtained between the selection of the optimizer and the model used.

3.2.7. Adam optimizer

The model performances using Adam optimizer resulted in a reasonably high average accuracy for all models. Adam optimizer is considered excellent for achieving satisfactory results without adjusting the optimizer parameters to match the model used.

3.2.8. SGD optimizer

Model performances using SGD optimizer result in varying accuracy. In the CoAtNet model results, the SGD optimizer produces the highest accuracy compared to other optimizers on the dataset without augmentation. However, models with the SGD optimizer significantly decrease accuracy when using augmented datasets, for example, in the ResNet101v2 and EfficientNetv2M models. This is presumably because the parameters used do not match the augmented data. The decrease in performance when using augmentation data shows that SGD optimizer is more sensitive to changes in the dataset. SGD optimizer demands more fine-tuning on its parameters to achieve high performance based on the model and dataset used.

3.2.9. RMSProp optimizer

The use of the RMSprop optimizer produces different accuracy for each model. In the ResNet101v2 model, the results obtained are inferior. However, the EfficientNetV2M model obtained satisfactory results and has one of the highest accuracies in the dataset. Observations show this is caused by the parameter values used. In the original EfficientNetV2 study, state-of-the-art performance was achieved using the RMSprop optimizer but with different parameters (Tan and Le, 2021). While training the ResNet101v2 model, its training and validation loss value fluctuated significantly. Based on research by (Verma et al., 2021), although with different problems, results of the ResNet50 model using a smaller learning rate value of 0.0001 got better accuracy than using a learning rate value of 0.0004. This indicates that the learning rate value used is too high for this model, as in research, ResNet101v2 is trained using a learning rate value of 0.001.

Based on research by (Choi et al., 2019), general optimizers like Adam should not produce worse results than specialized optimizers like SGD and other optimizer variations. Adam optimizer and other adaptive gradient methods have performance comparable to SGD or momentum with proper parameter settings (Choi et al., 2019). The conclusion obtained from optimizer impact on the model performance is that it's better to use an optimizer that is easy to use. Besides, parameter fine-tuning which matches the problem to be solved will produce the best performance.

3.3. The top 5 models

All model performance data are then sorted to determine the best combination of the model, optimizer, and augmentation. Remember that the model's performance obtained from the test may differ in real-world applications. Because the amount of data for each class in the dataset is not balanced, the model with the highest test accuracy does not always have the highest mean class accuracy. A higher mean class accuracy value can be interpreted as a better model's capability to learn all class features than other models.

The top-5 models based on %test accuracy are: EfficientNetV2M using RMSProp and brightness augmentation; EfficientNetV2B0 using RMSProp and brightness augmentation; CoAtNet-0 using SGD without augmentation, EfficientNetV2B0 using SGD without augmentation; and EfficientNeTV2B0 using RMSProp without augmentation. The metric performance for each model is presented in Fig. 12. All models have similar performance in all metrics, except mean class accuracy for EffiencientNetV2B0 using RMSProp and brightness augmentation. However, the mean class accuracy is still acceptable as it is higher than 85%.

It can be concluded that the simultaneous use of various types of augmentation does not always increase the model's performance. Some models achieve high accuracy using augmentation or without augmentation. In this case, augmentation is used more to vary data collection conditions, i.e., microscope illumination level, specimen orientation, or blurry photo. However, in the aspect of Nematodes morphology, the augmentation used did not significantly increase the variation in the morphological characteristics of each class. These augmentations cannot improve the model's ability to generalize classification problems based on the morphology of each class.

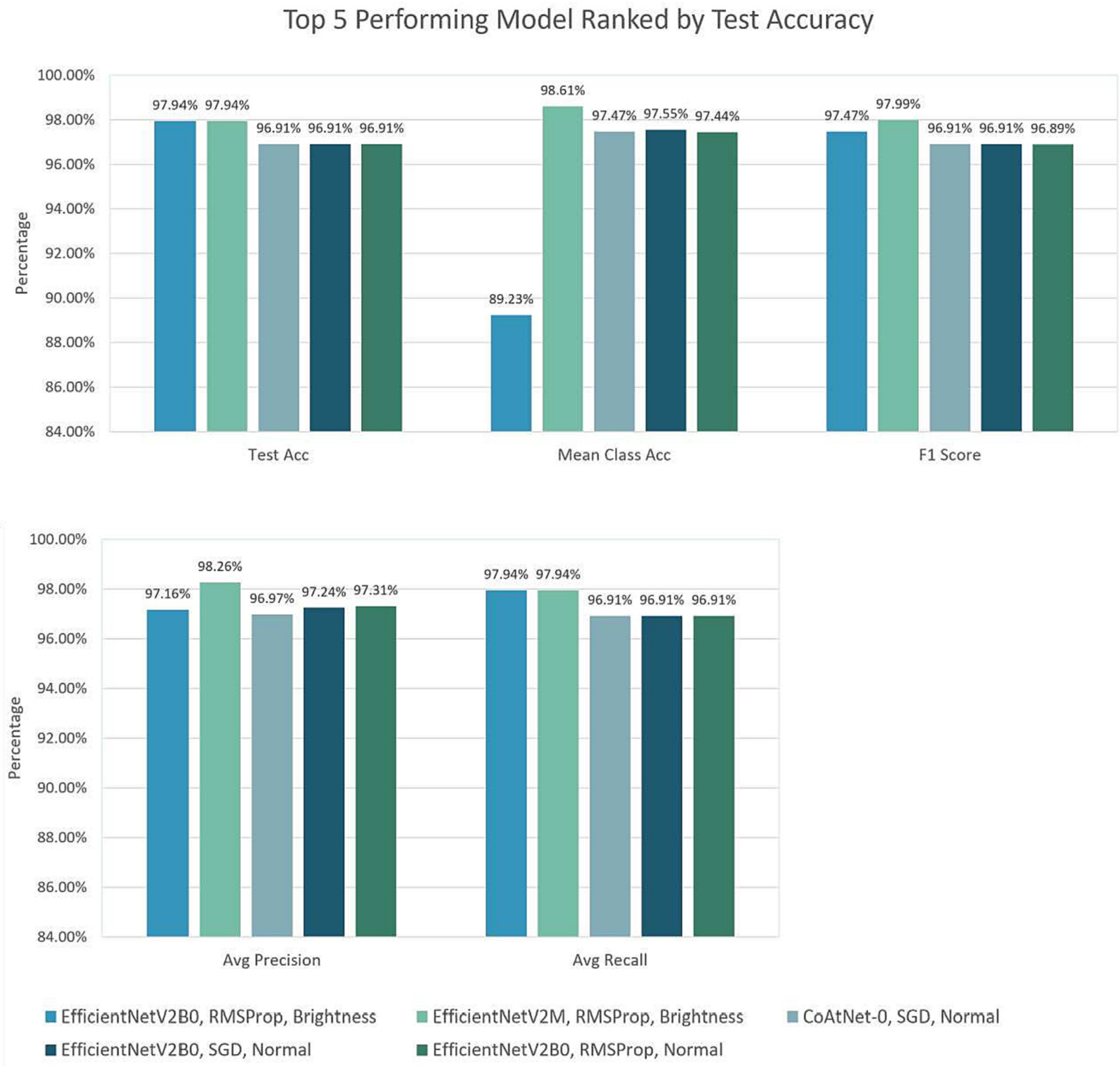
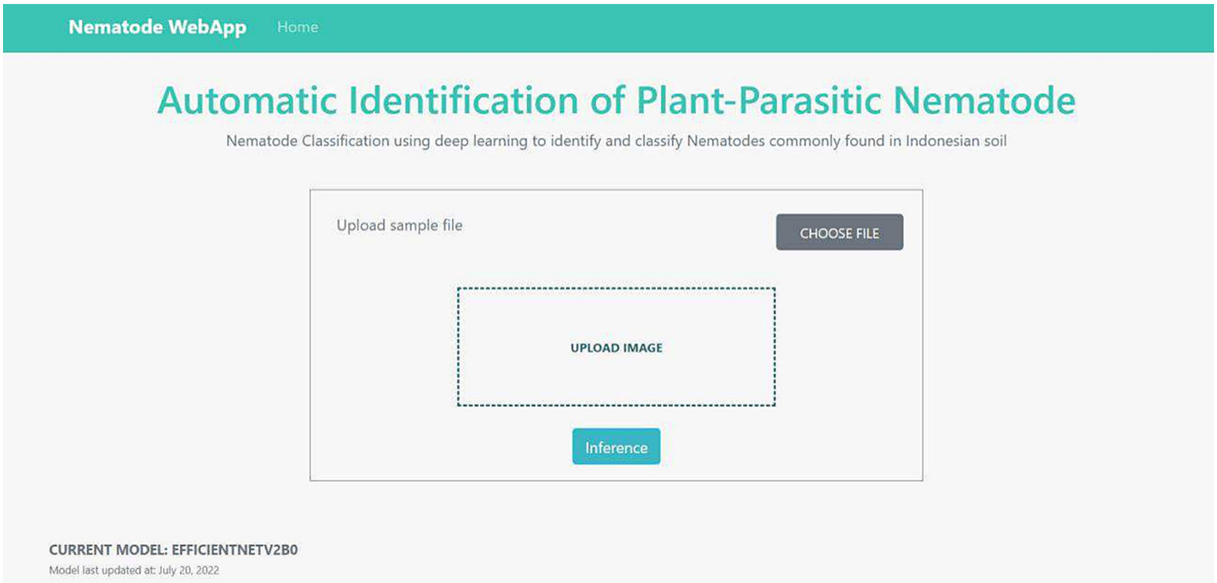


Fig. 12. Metrics comparison from the five-best performing models, ranked based on the accuracy of the test dataset.



Automatic Identification of Plant-Parasitic Nematode

Nematode Classification using deep learning to identify and classify Nematodes commonly found in Indonesian soil

CURRENT MODEL: EFFICIENTNETV2B0

Model last updated at: July 20, 2022

Fig. 13. Home screen of web-based application for automatic identification of plant-parasitic nematodes.

In addition, simultaneous image transformations can hide the important features studied in a class.

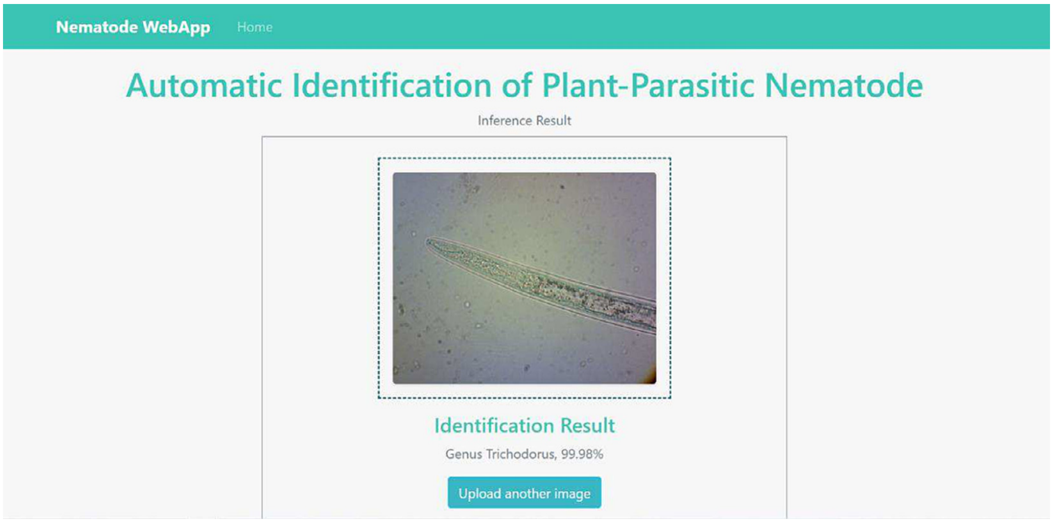
3.4. Model Integration in web applications

To deploy the web app to be accessible to other people, Heroku app, a cloud-based platforms was used. Following the result in Fig. 12, EfficientNetV2, both B0 and M versions have the highest accuracy of 97.94%. Although the models have similar performance, the model size was very distinctive. The EfficientNetV2B0 model is around 25 MB in size while EfficientNetV2B0 is 200 MB. To reduce the operational cost, EfficientNetV2B0 was selected for integration as the web platform. The implementation of the user interface screens is presented in Fig. 13 and Fig. 14. The cloud-based web deployment of the model has resulted in the inference time around 5 s for the classification of the input image and provides a result of pest genus. The web-based for automatic identification of plant-parasitic nematodes can be assessed at https://nematode-classifier.herokuapp.com/.

Using this proposed approach, plant-parasitic nematodes can be identified faster and more reliable. As explained in the previous paragraph, the system only needs five seconds to give the result of identifications with an accuracy reach of up to 97.94%. The system's limitation is that it can only recognize the nematodes in perfect and undamaged conditions, as the dataset only represents those conditions. However, future research will be conducted to meet others' circumstances.

4. Conclusion

A novel plant-parasitic nematodes image dataset was built for species of nematodes commonly found in Indonesia. The feasibility of deep learning models for plant-parasitic classifications, namely



Automatic Identification of Plant-Parasitic Nematode

Inference Result

Identification Result

Genus Trichodorus, 99.98%

Fig. 14. Sample result of web-based application for automatic identification of plant-parasitic nematodes.

ResNet101V2, CoAtNet-0, EfficientNetV2B0 and EfficientNetV2M were explored. Deep learning techniques in image-based classification are promising and advance the automation of the nematode identification process. This work demonstrated the capability of using deep learning models to identify plant-parasitic nematodes commonly found in Indonesian soil. The performance of each of the tested models obtained satisfactory results. The exemplary performance for each model results from combining several optimizer functions (Adam, SGD, and RSMProp) and the augmentation process (Flip, Brightness, Contrast, Gaussian Blur, and Gaussian Noise). Based on the selected model variants (ResNet101v2, CoAtNet0, EfficientNetV2B0, and EfficientNetV2M), the EfficientNetV2B0 and EfficientNetV2M models achieved the best performance on the dataset used, with the highest accuracy of 97.94%. Both of the best models were implemented using RMSProp optimizer and brightness augmentation. The final contribution provides an integrated deep learning model as a web-based application that Indonesia's prospective agriculturists can use directly. Further work is still needed to improve model accuracy and reliability. Future work will focus on the following:

- Further development of the nematodes dataset to cope with unperfect conditions of nematode images and expand to non-parasitic nematodes
- Investigate other deep learning models to improve the performance of multi-class nematodes identification

CRediT authorship contribution statement

Nabila Husna Shabrina: Conceptualization, Methodology, Formal analysis, Visualization, Writing – original draft, Project administration, Funding acquisition. **Ryukin Aranta Lika:** Software, Investigation, Data curation, Writing – original draft. **Siwi Indarti:** Resources, Validation, Writing – original draft.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We would like to thank Rina Maharani, S.P., M.Sc who help us to collect and prepare nematodes dataset. We would also like to acknowledge the support given by Universitas Multimedia Nusantara during this study.

References

Abad, P., Gouzy, J., Aury, J.-M., Castagnone-Sereno, P., Danchin, E.G.J., Deleury, E., Perfus-Barbeoch, L., Anthouard, V., Artiguenave, F., Blok, V.C., Caillaud, M.-C., Coutinho, P.M., Dasilva, C., de Luca, F., Deau, F., Esquibet, M., Flutre, T., Goldstone, J.V., Hamamouch, N., Hewezi, T., Jaillon, O., Jubin, C., Leonetti, P., Magliano, M., Maier, T.R., Markov, G.V., McVeigh, P., Pesole, G., Poulain, J., Robinson-Rechavi, M., Sallet, E., Ségurens, B., Steinbach, D., Tytgat, T., Ugarte, E., van Ghelder, C., Veronico, P., Baum, T.J., Blaxter, M., Bleve-Zacheo, T., Davis, E.L., Ewbank, J.J., Favery, B., Grenier, E., Henrissat, B., Jones, J.T., Laudet, V., Maule, A.G., Quesneville, H., Rosso, M.-N., Schiex, T., Smant, G., Weissenbach, J., Wincker, P., 2008. Genome sequence of the metazoan plant-parasitic nematode Meloidogyne incognita. Nat. Biotechnol. 26, 909–915. https://doi.org/10.1038/nbt.1482.

Abade, A., Porto, L.F., Ferreira, P.A., de Barros Vidal, F., 2022. NemaNet: a convolutional neural network model for identification of soybean nematodes. Biosyst. Eng. 213, 39–62. https://doi.org/10.1016/j.biosystemseng.2021.11.016.

Ajri, M., Indarti, S., Soffan, A., Huu, N.N., 2021. Morphological and phylogenetic characteristics of ditylenchus dipsaci among garlic plants. Jordan J. Biol. Sci. 14, 769–773. https://doi.org/10.54319/jjbs/140418.

Alsaggaf, W., Cömert, Z., Nour, M., Polat, K., Brdesee, H., Toğaçar, M., 2020. Predicting fetal hypoxia using common spatial pattern and machine learning from cardiotocography signals. Appl. Acoust. 167, 107429. https://doi.org/10.1016/j.apacoust.2020.107429.

Choi, D., Shallue, C.J., Nado, Z., Lee, J., Maddison, C.J., Dahl, G.E., 2019. On Empirical Comparisons of Optimizers for Deep Learning. https://doi.org/10.48550/arXiv.1910.05446.

Chollet, F., 2015. Keras: Deep Learning for Humans [WWW Document]. URL https://github.com/keras-team/keras (accessed 2.1.22).

Dai, Z., Liu, H., Le, Q.V., Tan, M., 2021. CoAtNet: Marrying Convolution and Attention for All Data Sizes. https://doi.org/10.48550/arXiv.2106.04803.

De, W.D., Elsen, A., 2007. Challenges in tropical plant nematology. Annu. Rev. Phytopathol. 45, 457–485. https://doi.org/10.1146/annurev.phyto.45.062806.094438.

Diker, A., Comert, Z., Avci, E., Togacar, M., Ergen, B., 2019. A novel application based on spectrogram and convolutional neural network for ECG classification. 2019 1st International Informatics and Software Engineering Conference (UBMYK). IEEE, pp. 1–6 https://doi.org/10.1109/UBMYK48245.2019.8965506.

Elsken, T., Metzen, J.H., Hutter, F., 2019. Neural architecture search: a survey. J. Mach. Learn. Res. 20, 1997–2017. https://doi.org/10.48550/arXiv.1808.05377.

He, K., Zhang, X., Ren, S., Sun, J., 2016a. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp. 770–778 https://doi.org/10.1109/CVPR.2016.90.

He, K., Zhang, X., Ren, S., Sun, J., 2016b. Identity mappings in deep residual networks. Computer Vision – ECCV 2016. Springer International Publishing, pp. 630–645 https://doi.org/10.1007/978-3-319-46493-0_38.

Indarti, S., Bambang, R.T.P., Mulyadi B., Triman, 2004. First record of potato cyst nematode globodera rostochiensis in Indonesia. Australas. Plant Pathol. 33, 325. https://doi.org/10.1071/AP04018.

Indarti, S., Wibowo, A., Subandiyah, S., Ajri, M., 2018. First record: a stem and bulb plant parasitic nematode at garlic area Centre temanggung, central java, Indonesia with species reference to ditylenchus dipsaci. Jurnal Perlindungan Tanaman Indonesia 22, 233. https://doi.org/10.22146/jpti.35321.

Jiang, P., Chen, Y., Liu, B., He, D., Liang, C., 2019. Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks. IEEE Access 7, 59069–59080. https://doi.org/10.1109/ACCESS.2019.2914929.

Jones, J.T., Haegeman, A., Danchin, E.G.J., Gaur, H.S., Helder, J., Jones, M.G.K., Kikuchi, T., Manzanilla-López, R., Palomares-Rius, J.E., Wesemael, W.M.L., Perry, R.N., 2013. Top 10 plant-parasitic nematodes in molecular plant pathology. Mol. Plant Pathol. 14, 946–961. https://doi.org/10.1111/mpp.12057.

Keras, 2021. Keras Applications [WWW Document]. URL https://keras.io/api/applications/ (accessed 2.1.22).

Kingma, D.P., Ba, J., 2014. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arXiv.1412.6980.

Kusrini, K., Suputa, S., Setyanto, A., Agastya, I.M.A., Priantoro, H., Chandramouli, K., Izquierdo, E., 2020. Data augmentation for automated pest classification in mango farms. Comput. Electron. Agric. 179, 105842. https://doi.org/10.1016/j.compag.2020.105842.

Li, L., Zhang, S., Wang, B., 2021. Plant disease detection and classification by deep learning—a review. IEEE Access 9, 56683–56698. https://doi.org/10.1109/ACCESS.2021.3069646.

Lu, X., Wang, Y., Fung, S., Qing, X., 2021. I-nema: A Biological Image Dataset for Nematode Recognition. https://doi.org/10.48550/arXiv.2103.08335.

Mutala'liah, M., Indarti, S., Wibowo, A., 2018. Short communication: the prevalence and species of root-knot nematode which infect on potato seed in central java, Indonesia. Biodivers. J. 20, 11–16. https://doi.org/10.13057/biodiv/d200102.

Paymode, A.S., Malode, V.B., 2022. Transfer learning for multi-crop leaf disease image classification using convolutional neural network vgg. Artif. Intell. Agric. 6, 23–33. https://doi.org/10.1016/j.aiia.2021.12.002.

Raouhi, E.M., Lachgar, M., Hrimech, H., Kartit, A., 2022. Optimization techniques in deep convolutional neuronal networks applied to olive diseases classification. Artif. Intell. Agric. 6, 77–89. https://doi.org/10.1016/j.aiia.2022.06.001.

Shijie, J., Ping, W., Peiyi, J., Siping, H., 2017. Research on data augmentation for image classification based on convolution neural networks. 2017 Chinese Automation Congress (CAC). IEEE, pp. 4165–4170 https://doi.org/10.1109/CAC.2017.8243510.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. J. Big Data 6, 60. https://doi.org/10.1186/s40537-019-0197-0.

Southey, J., 1986. Laboratory Methods for Work with Plant and Soil Nematodes. Her Majesty's Stationary Office, London.

Stojnic, R., Taylor, R., Kardas, M., Elvis Cucurull, G., 2022. Image Classification on Imagenet [WWW Document]. URL https://paperswithcode.com/sota/image-classification-on-imagenet (accessed 2.1.22).

Sun, S., Cao, Z., Zhu, H., Zhao, J., 2020. A survey of optimization methods from a machine learning perspective. IEEE Trans. Cybern. 50, 3668–3681. https://doi.org/10.1109/TCYB.2019.2950779.

Tan, M., Le, Q.V., 2019. Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks. https://doi.org/10.48550/arXiv.1905.11946.

Tan, M., Le, Q.V., 2021. Efficientnetv2: Smaller Models and Faster Training. https://doi.org/10.48550/arXiv.2104.00298.

Toğaçar, M., Ergen, B., Cömert, Z., 2021. Tumor type detection in brain mr images of the deep model developed using hypercolumn technique, attention modules, and residual blocks. Med. Biol. Eng. Comput. 59, 57–70. https://doi.org/10.1007/s11517-020-02290-x.

Uhlemann, J., Cawley, O., Kakouli-Duarte, T., 2020. Nematode identification using artificial neural networks. International Conference on Deep Learning Theory and Applications. SCITEPRESS - Science and Technology Publications, pp. 13–22 https://doi.org/10.5220/0009776600130022.

Verma, P., Tripathi, V., Pant, B., 2021. Comparison of different optimizers implemented on the deep learning architectures for COVID-19 classification. Mater.Today: Proc. 46, 11098–11102. https://doi.org/10.1016/j.matpr.2021.02.244.

Zhou, P., Feng, J., Ma, C., Xiong, C., Hoi, S., E. W, 2020. Towards Theoretically Understanding Why SGD Generalizes better than Adam in Deep Learning. https://doi.org/10.48550/arXiv.2010.05627.