



Available online at <http://scik.org>

Commun. Math. Biol. Neurosci. 2023, 2023:65

<https://doi.org/10.28919/cmbn/7993>

ISSN: 2052-2541

## **A COMPARATIVE ANALYSIS OF CONVOLUTIONAL NEURAL NETWORKS APPROACHES FOR PHYTOPARASITIC NEMATODE IDENTIFICATION**

NABILA HUSNA SHABRINA<sup>1,\*</sup>, SIWI INDARTI<sup>2</sup>, RYUKIN ARANTA LIKA<sup>1</sup>, RINA MAHARANI<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, Universitas Multimedia Nusantara, Tangerang 15111, Indonesia

<sup>2</sup>Department of Plant Protection, Faculty of Agriculture, Universitas Gadjah Mada, Yogyakarta, 55281, Indonesia

Copyright © 2023 the author(s). This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**Abstract:** Phytoparasitic nematode is a microscopic worm that affects the host plants and causes severe losses in the agricultural sector. Accurate and rapid identification of phytoparasitic nematodes is required to determine proper pest control and management. Hence it has become a necessity to automate the phytoparasitic nematode identification procedure. This study conducts a comparative analysis of 15 popular convolutional neural networks models, namely CoAtNet-0, DenseNet121, DenseNet169, DenseNet201, EfficientNetV2B0, EfficientNetV2B3, EfficientNetV2L, EfficientNetV2M, EfficientNetV2S, InceptionResNetV2, InceptionV3, ResNet101v2, ResNet50v2, VGG19, and Xception to deals with phytoparasitic nematode identification from the microscopic image. The results are compared using several evaluation metrics, namely test accuracy, mean class accuracy, F1 score, precision, and recall. The results show that CoAtNet-0 outperformed other models with 98.06% test accuracy, 97.86% mean class accuracy, 0.9803 F1 score, 0.9818 Precision, and 0.9806 Recall.

**Keywords:** convolutional neural networks; classifications; phytoparasitic nematode.

**2020 AMS Subject Classification:** 92B20.

---

\*Corresponding author

E-mail address: [nabila.husna@umn.ac.id](mailto:nabila.husna@umn.ac.id)

Received April 18, 2023

## 1. INTRODUCTION

Phytoparasitic nematode is a harmful nematode that is usually found in plants. Around 4100 identified phytoparasitic nematode species [1], each impacting the host plants differently. Phytoparasitic nematodes cause an enormous crop loss, projected to be around 157 billion USD globally [2]. The current identification process is based on the classical method, where the nematologist performs visual observation from the morphological characteristic of the microscopic nematode image. Other supplementary techniques were based on molecular analysis using fingerprint, DNA sequencing, and protein analysis [3]. Conventional methods are challenging as the morphological characteristic between species is similar. The identification also requires a long process, prone to error, and there is a decrease in the number of nematode experts. Therefore, an alternative method with high accuracy to assist phytoparasitic nematode identification is necessary for proper pest control and management.

The image-based method has been proven to have a notable result in nematode identification processes. The pioneer in integrating the advanced image-based method for nematode identification was first introduced by [4]. The research demonstrated image processing techniques, such as filtering, segmentation, and morphological operations for feature extraction and using the RBF neural network for classification tasks. The ability of a skeleton-based technique for *C. Elegans* nematode detection and separation in population images was demonstrated by [5]. The strategy yield is a vision algorithm with a false rejection rate (FRR) and false acceptance rate (FAR) of 7.9% and 8.4%, respectively. An application for automated identification of nematode's size and shape, called WormSizer, was introduced by [6]. The application implemented several image processing techniques, such as a global thresholding algorithm, image segmentation, and skeletonizing, to measure the nematode's width, length, and volume. An image-based algorithm was accomplished by [7] to acquire a physical attribute of *Meloidogyne* type II species. The proposed algorithm, which applied several image processing methods, such as illumination correction, binarization, and segmentation, achieved an error rate of 15% and 11% for length and width measurement, respectively.

The breakthrough in machine learning and deep learning makes it possible to be implemented in nematode identification. This method is adapted for processing extensive data and recognizing

different and small things in challenging environments, such as microscopic nematode images. A deep convolutional selective encoder architecture was developed [8] to identify and count soybeans cyst nematode (SCN) in clutter-field images. The proposed algorithm is comparable to the expert's result, with SCN in the less cluttered and highly cluttered image resulting in 92% and 95% accuracy, respectively. Research by [9] proposed a new architecture combining DenseNet121 and Inception Blocks for phytonematode identification. The performance was outstanding, with an accuracy reach of 98.99% for the model implemented with the transfer learning method. The feasibility of the Xception model, a Convolutional Neural Networks (CNN)-based method, for classifying entomopathogenic nematodes (EPN) was studied [10]. The performance achieved an average validation accuracy of 69.45% for the adult nematode dataset and 88.28% for the juvenile nematode dataset. A deep learning-based application was developed by [11] for soil nematode identification. A ResNet-101 deployed to a web-based system could correctly identify 60% of the nematode genera, 76% of c-p values, and 76% of feeding types applied in the I-nema dataset [12]. Faster region-based convolutional neural networks were implemented by [13] to detect the non-parasitic and plant-parasitic nematode from the microscopic image. The method resulted in an accuracy reach of up to 87.5%.

Deep learning models were recently implemented for identifying plant-parasitic nematodes with a self-collected dataset of nematodes commonly found in Indonesia [14]. The previous study explored the effect of optimization techniques and augmentation methods on the performance of four different deep-learning models. The result shows that most augmentation methods negatively impacted the model performance. In terms of optimization, it was found that it is better to use an optimizer that is easy to use, but with parameter fine-tuning that is matched to the problem to be solved to produce the best performance.

To further improve the previous research, this study investigates 15 popular and well-known CNN-based methods, namely CoAtNet-0, DenseNet121, DenseNet169, DenseNet201, EfficientNetV2B0, EfficientNetV2B3, EfficientNetV2L, EfficientNetV2M, EfficientNetV2S, InceptionResNetV2, InceptionV3, ResNet101v2, ResNet50v2, VGG19, and Xception. The dataset used in this research is an improved version of the prior study [14], which added images to several classes. The deep learning models were performed via the transfer learning method with the same

optimizer function, namely SGD optimizer. Fine-tuning was applied to match the dataset characteristic and the task which will be solved. The model performance was then compared based on several metric evaluations, such as test accuracy, mean class accuracy, F1 score, precision, and recall, to obtain the best CNN approach for phytoparasitic nematode identification.

## 2. METHODS

**2.1. Research Workflow.** Figure 1 presents the general workflow for this study. Initially, the phytoparasitic nematode dataset was collected in Indonesia's agricultural area and classified by a nematologist expert before further processing. The data pre-processing includes several image pre-processing techniques such as edge detection, cropping, and converting into grayscale. The selected CNN models, namely CoAtNet-0, DenseNet121, DenseNet169, DenseNet201, EfficientNetV2B0, EfficientNetV2B3, EfficientNetV2L, EfficientNetV2M, EfficientNetV2S, InceptionResNetV2, InceptionV3, ResNet101v2, ResNet50v2, VGG19, and Xception, then build and configure using several hyperparameter settings. The pre-processing results are used as the input for training the CNN models. The results are obtained and compared using several evaluation metrics, such as test accuracy, mean class accuracy, F1, precision, and recall, to find the best CNN models.

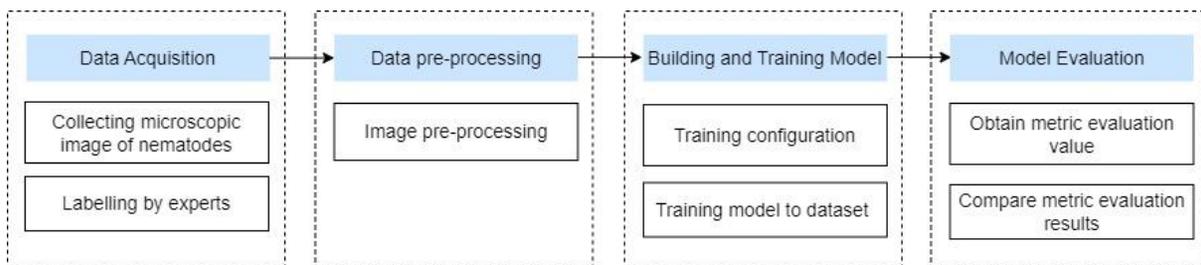


FIGURE 1. Research workflow.

**2.2. Datasets.** The images were captured using an optical system linked to a laptop and a microscope with the samples presented in Figure 2. Initially, a soil sample collection of phytoparasitic nematodes from diseased Indonesian agricultural plants was collected. The nematodes then extracted from the soil using modified Whitehead Tray [15]. The specimen was then prepared for morphological assessment by using an Olympus CX 31 light microscope with a

## PHYTOPARASITIC NEMATODE IDENTIFICATION

magnification range of 40–1000. The picture was captured using an optical system linked to a laptop and a microscope, with the sample results presented in Figure 3. We added fifty-nine new images to the dataset in the prior study [14], resulting in the image distribution classes as presented in Table 1. The datasets consist of 1016 plant-parasitic nematode images divided into 11 classes. As can be seen in Table 1, the total image in each class is considered unbalanced. However, it represents the real-world constraint faced by nematologists as some nematode genera were limited in the agricultural area.

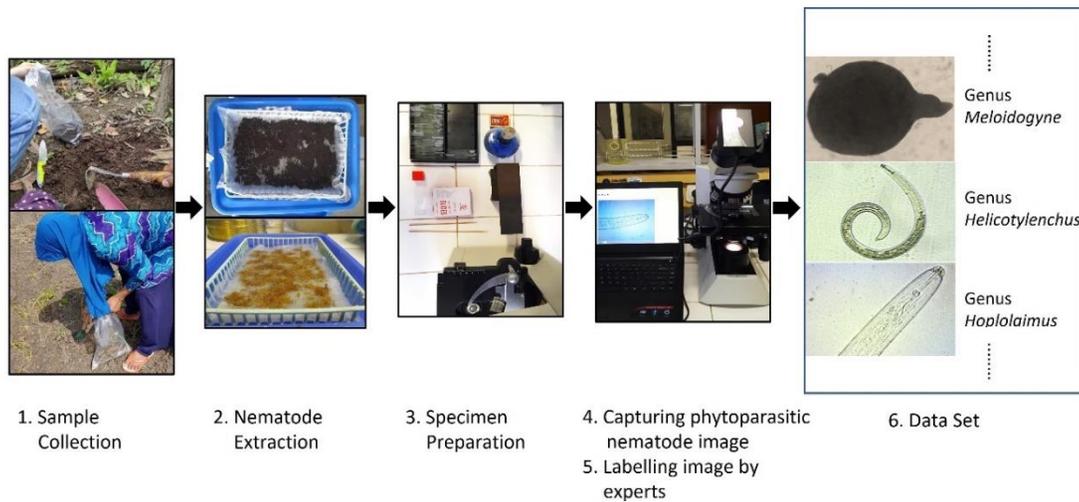


FIGURE 2. Data acquisition workflow.

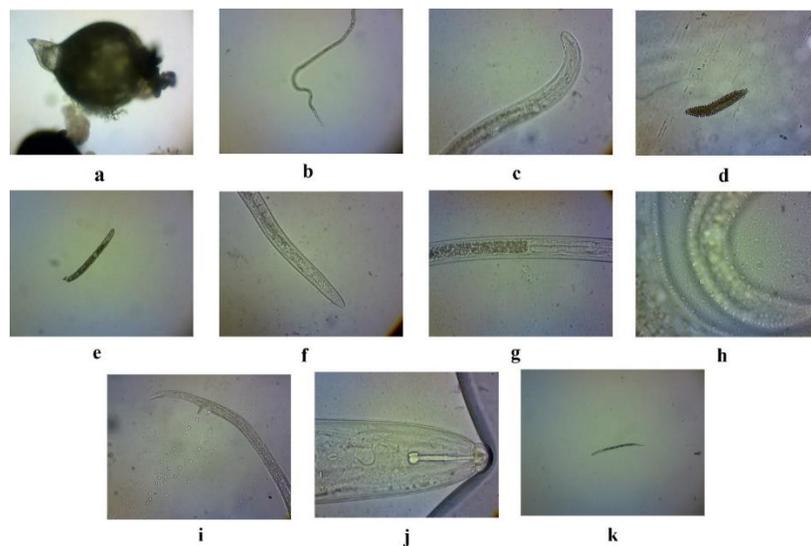


FIGURE 3. Sample of phytoparasitic nematode dataset. a. *Meloidogyne*, b. *Hirschmanniella*, c. *Pratylenchus*, d. *Criconea*, e. *Criconemoides*, f. *Xiphinema*, g. *Trichodorus*, h. *Helicotylenchus*, i. *Radopholus*, j. *Hoplolaimus*, k. *Hemicycliophora*.

TABLE 1. Phytoparasitic nematode dataset distribution.

Genus	No. of samples
Genus Meloidogyne	211
Genus Hirschmanniella	130
Genus Pratylenchus	116
Genus Criconema	4
Genus Criconemoides	103
Genus Xiphinema	85
Genus Trichodorus	44
Genus Helicotylenchus	135
Genus Radopholus	31
Genus Hoplolaimus	151
Genus Hemicycliophora	6
<b>Total</b>	<b>1016</b>

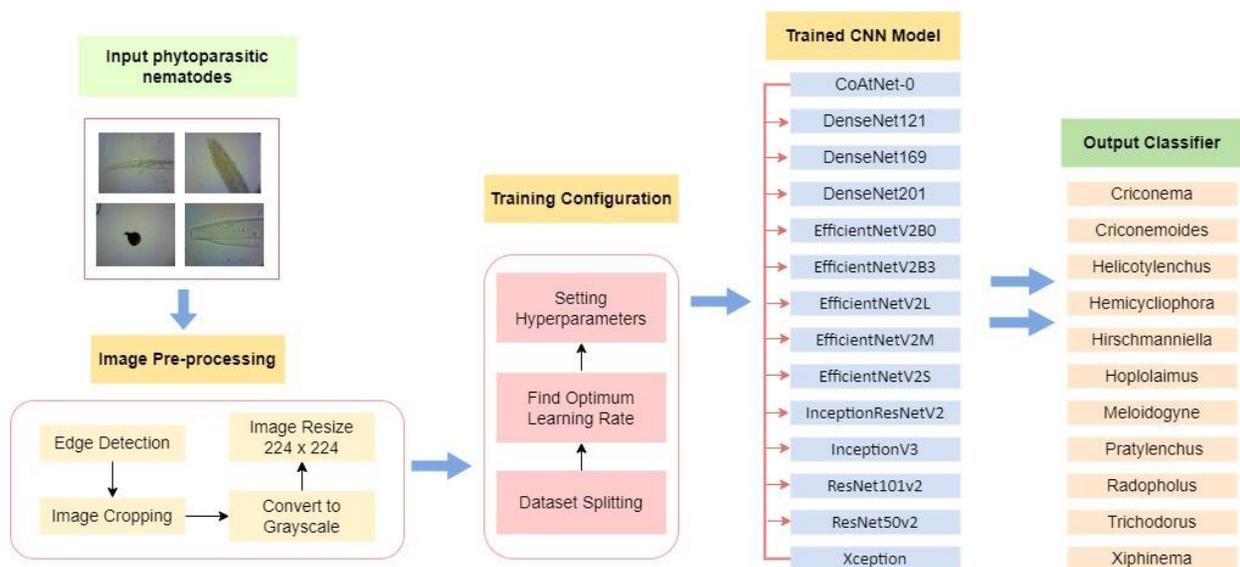


FIGURE 4. CNN implementation workflow.

**2.3. CNN Implementations.** The workflow for training the phytoparasitic nematode using CNN models is presented in Figure 4. The data acquisition result from the procedure in Figure 2 was used as the input image of the CNN architecture. The images were then pre-processed using the same method as the previous works [12–14]. Edge detection was implemented to find the boundary

## PHYTOPARASITIC NEMATODE IDENTIFICATION

of the nematode specimens contained in the image. The detection result was then used as an input for image cropping, a technique to equalize the areas of the image containing the nematode objects by eliminating white space and lessening superfluous information. Grayscale pictures of the samples were then generated because nematode identification relies exclusively on their morphological characteristics. Then, to meet the input size of all models evaluated, all microscopic photos are downsized to 224 x 224. The data was then split into train, validation, and testing data with the composition of 80:10:10, with the total images of 813, 102, and 101 in train, validation, and testing data, respectively.

TABLE 2. Parameters of CNN models

<b>Model</b>	<b>Size</b>	<b>Parameters</b>	<b>Depth</b>	<b>Initial LR</b>
CoAtNet-0	86	25 M	-	0.0021
DenseNet121	27	8.1 M	242	0.001
DenseNet169	49	14.3 M	338	0.003
DenseNet201	71	20.2 M	402	0.003
EfficientNetV2B0	23	7.2 M	-	0.065
EfficientNetV2B3	50	14.5 M	-	0.065
EfficientNetV2L	450	119 M	-	0.049
EfficientNetV2M	203	54.4 M	-	0.134
EfficientNetV2S	78	21.6 M	-	0.029
InceptionResNetV2	208	55.9 M	449	0.013
InceptionV3	84	23.9 M	189	0.009
ResNet101v2	163	44.7 M	205	0.007
ResNet50v2	90	25.6 M	103	0.001
VGG19	549	143.7M	19	0.01
Xception	80	22.9 M	81	0.025

The optimizer will determine how weight updates are carried out during training. This research implements SGD and momentum when training deep convolutional neural networks due to its superior validation and test results. Besides, SGD also has a better generalization to unobserved data [16–17]. However, SGD requires learning rate tuning and is slower to converge than Adam Optimizer. Those issues can be resolved by applying a good starting learning rate (LR). The

selection of an LR during deep neural network training is essential for both quick convergence and minimal error. Learning rate and momentum value significantly affect neural network performances [18]. This study implements LRfinder proposed by [19] to find the initial learning rate value for each model, except for CoAtNet-0, due to memory limitations. The initial learning rate for each model is given in Table 2. Moreover, zero momentum of SGD was also applied when training the CNN model.

Hyperparameters are crucial when training convolutional neural networks since they directly influence the actions of training algorithms and significantly impact the models' performance [20]. For consistency, each model will be trained by utilizing the same hyperparameter value in batch size, activation layer, loss function, and epoch. A batch size of 32 was selected for the CNN models due to its improved generalization performance and memory saver [21]. However, because of memory limitations, the batch size of CoANet-0 was set to 16. For a multi-class classification problem, a sparse cross-entropy loss function was applied. The SoftMax activation layer was implemented for the dense layer. The training epoch was assigned to 100 with a callback early stopping function to end the training process when it reaches a certain condition and to avoid overfitting. LearningRateScheduler callback was also used to take the step decay function as an argument and return the updated learning rates with 0.97 from the initial LR value.

Large datasets are ideal for training convolutional neural network models [22]. However, in this study, the total data is considered small. Transfer learning is frequently used to reduce overfitting caused by smaller datasets [23]. This method transfer knowledge from the model trained using the ImageNet dataset to the new phytoparasitic nematode dataset. This study also avoided training from scratch due to the computational expense of building the architecture. When implementing the transfer learning model, the final fully connected layer was removed, replaced with a layer with 11 output nodes, and then retrained by varying weights to better categorize phytoparasitic nematodes.

This study investigates 14 popular pre-trained deep learning models provided by Keras [24], which are trained using ImageNet [25] weight. The models were selected based on the

computational limitations and image classification performance using the ImageNet benchmark [24–26]. The selected models are CoAtNet-0, DenseNet121, DenseNet169, DenseNet201, EfficientNetV2B0, EfficientNetV2B3, EfficientNetV2L, EfficientNetV2M, EfficientNetV2S, InceptionResNetV2, InceptionV3, ResNet101v2, ResNet50v2, VGG19, and Xception. All models have the same based architecture by applying convolutional architecture. The parameters, an internal value related to model architecture, and the main backbone for each convolutional neural network architecture implemented in this study are summarized in Table 2 and Table 3, respectively. All models were then trained on Google Colab Pro with the specifications of NVIDIA P100 or T4 as GPU, CPU Xeon Processor@ 2.3GHz, and memory up to 25GB based on availability.

TABLE 3. Main characteristics of CNN models

<b>Model</b>	<b>Architecture</b>	<b>Key points</b>
CoAtNet-0	Convolution & transformers	Combination of two different architecture
DenseNet121	Convolution	Deep network with shorter connections between layers
DenseNet169	Convolution	
DenseNet201	Convolution	
EfficientNetV2B0	Convolution	Scaling method and neural architecture search
EfficientNetV2B3	Convolution	
EfficientNetV2L	Convolution	
EfficientNetV2M	Convolution	
EfficientNetV2S	Convolution	
InceptionResNetV2	Convolution	Inception layer with residual connection
InceptionV3	Convolution	Inception layer
ResNet101v2	Convolution	Residual connections
ResNet50v2	Convolution	
VGG19	Convolution	Deep network of small convolution layer
Xception	Convolution	Depth wise separable convolution

**2.3.1 CoAtNet.** The Convolution and Attention Networks (CoAtNet) were proposed by [27] through combining the strength of the convolution and transformer networks. This method has better generalization, a larger capacity, faster convergence, and improved efficiency. CoAtNet-0 consists of three convolutional stages and two transformers (attention) layers, as seen in the architecture presented in Figure 5. Global pooling is implemented in this model as the pooling layer before the final fully connected layer.

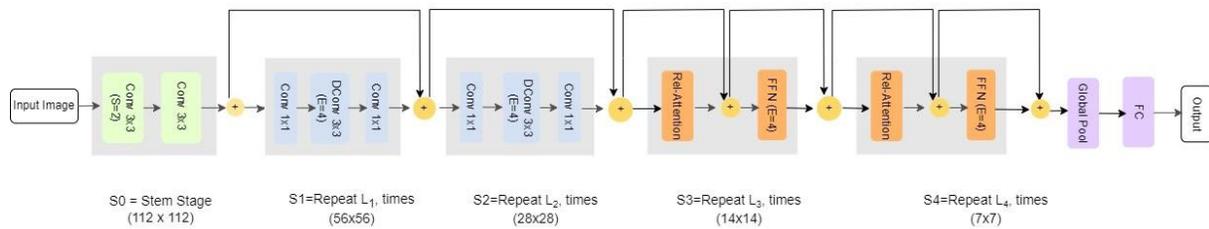


FIGURE 5. CoAtNet-0 architecture.

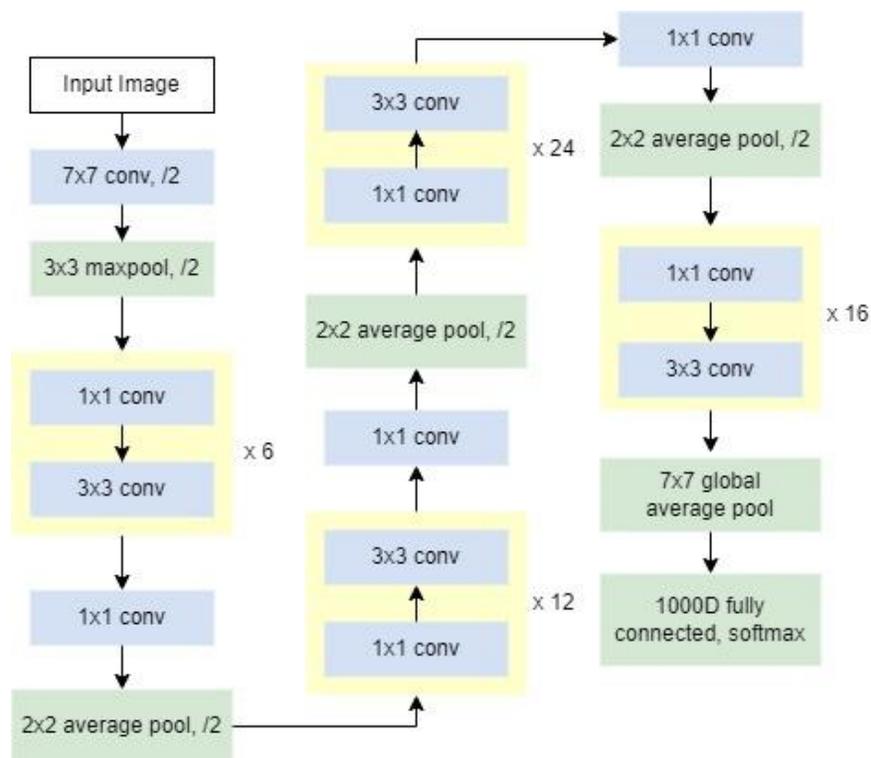


FIGURE 6. DenseNet121 architecture

## PHYTOPARASITIC NEMATODE IDENTIFICATION

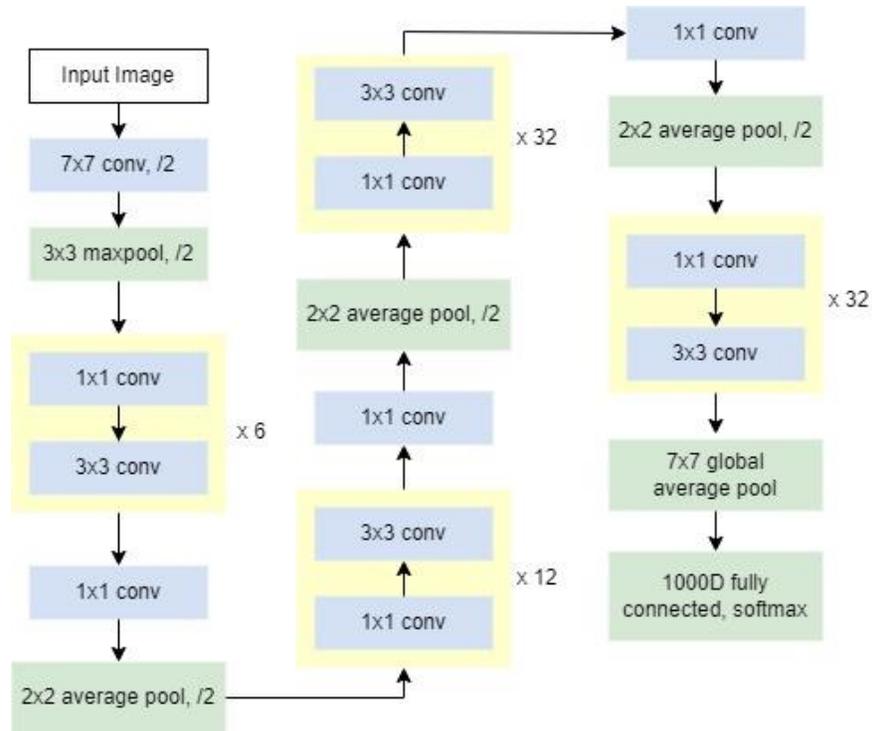


FIGURE 7. DenseNet169 architecture.

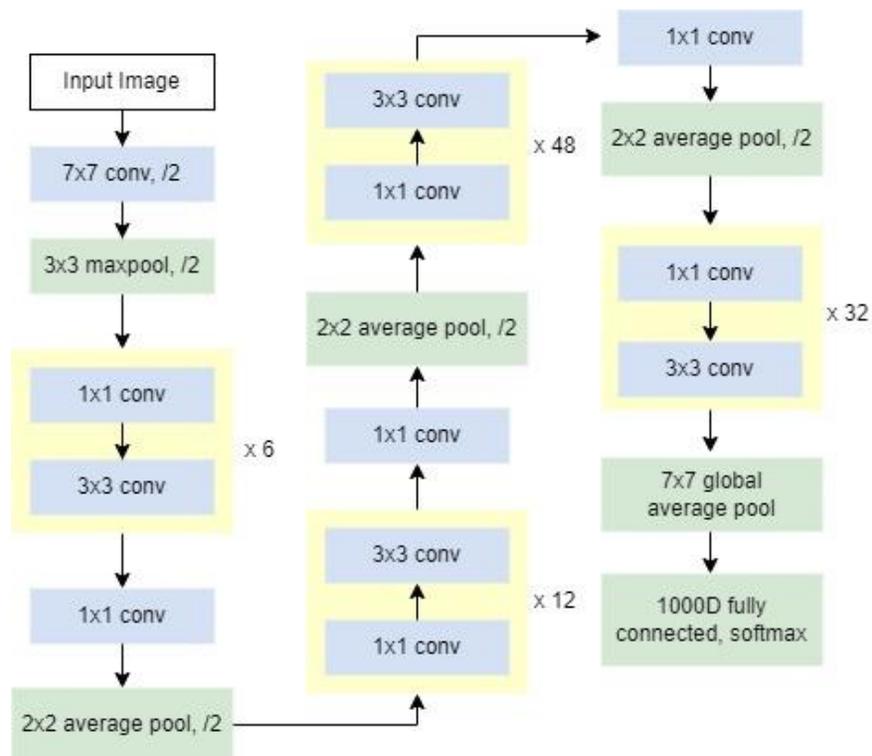


FIGURE 8. DenseNet201 architecture.

**2.3.2 DenseNet.** Dense Convolutional Networks (DenseNet) are based on the connection between each layer in a feed-forward manner [28]. This neural network architecture offers a variety of significant advantages, including eliminating the vanishing-gradient issue, improving feature propagation, promoting feature reuse, and significantly decreasing parameter requirements. The networks are divided into multiple densely connected, namely dense blocks, to facilitate the down-sampling layers that change the feature map. The network configuration in the dense block differentiated the version of this model. The configuration for DenseNet121 is presented in Figure 6, while DenseNet169 and DenseNet201 are presented in Figure 7 and Figure 8, respectively. Those three DenseNet models are implemented in this study.

**2.3.3 EfficientNet.** The EfficientNet model advances the performance compared to other architecture by implementing a smaller model and faster convergence speed. The core of this model uses a new scaling technique with a straightforward compound coefficient to scale the model's breadth, depth, and resolution to increase model capacity [29]. A new baseline model utilizing MBConv blocks is then built using Neural Architecture Search (NAS) and is scaled using the compound coefficient to build EfficientNet [30]. The network's core features of MBConv and Fused-MBConv increase training efficiency while reducing model size.

The newer and enhanced models, namely EfficientNetV2, with a certain training technique, can reach a 5–11 faster convergence rate than the existing cutting-edge models while being up to 6 smaller in size [31]. The EfficientNetV2B has four models starting from B0 to B3 version. The architecture of EfficientNetV2B0 and EfficientNetV2B3 implemented in this study is presented in Figure 9 and Figure 10, respectively. The small version of EfficientNetV2, namely EfficientNetV2S, was also utilized in this study. EfficientNetV2S is scaled up to generate EfficientNetV2M (Medium) and EfficientNetV2L (Large) with a few additional optimizations by limiting image size and adding more layers gradually in a later scale. The architecture for EfficientNetV2L, EfficientNetV2M, and EfficientNetV2S is presented in Figure 11, Figure 12, and Figure 13, respectively.

## PHYTOPARASITIC NEMATODE IDENTIFICATION

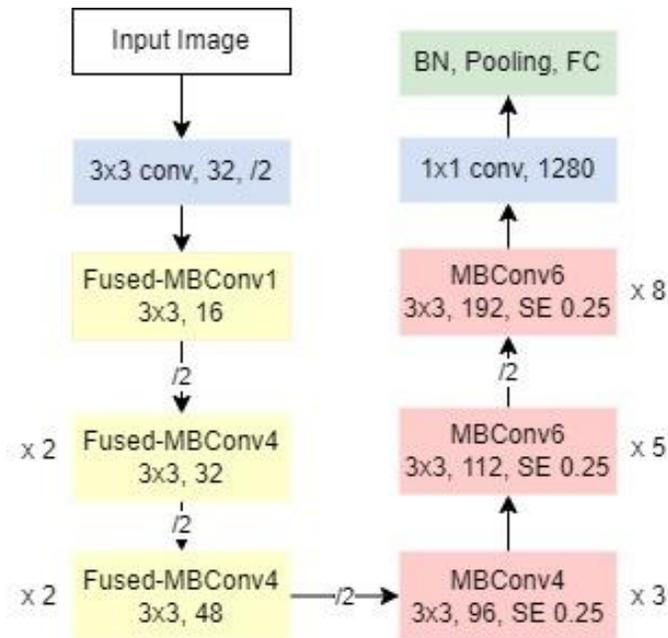


FIGURE 9. EfficientNetV2B0 architecture.

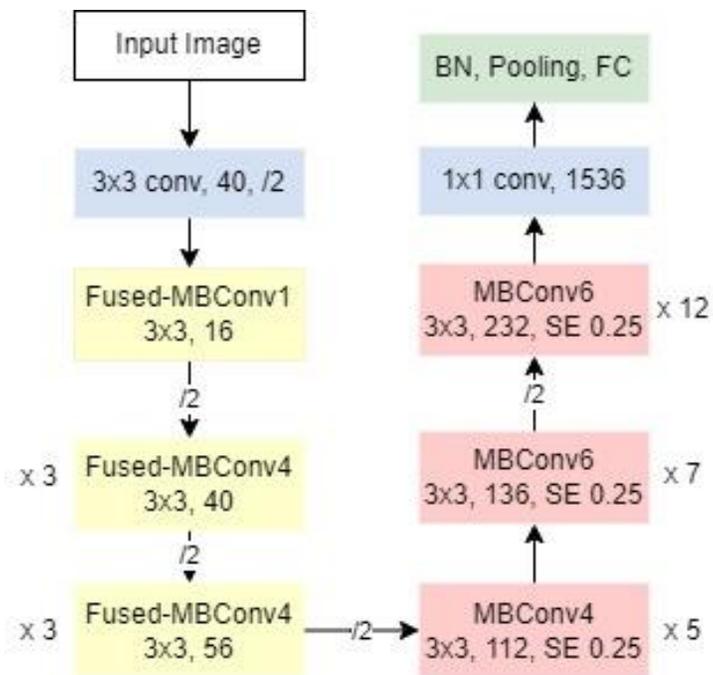


FIGURE 10. EfficientNetV2B3 architecture.

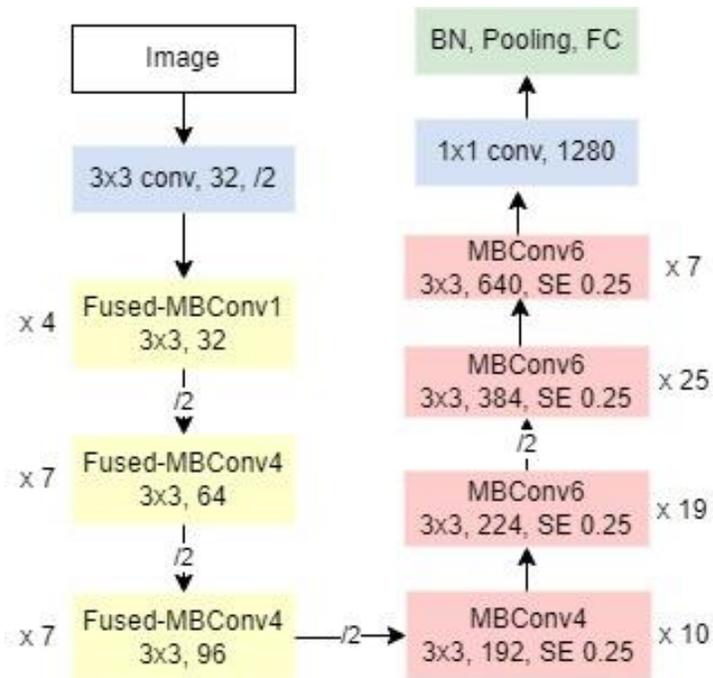


FIGURE 11. EfficientNetV2L architecture.

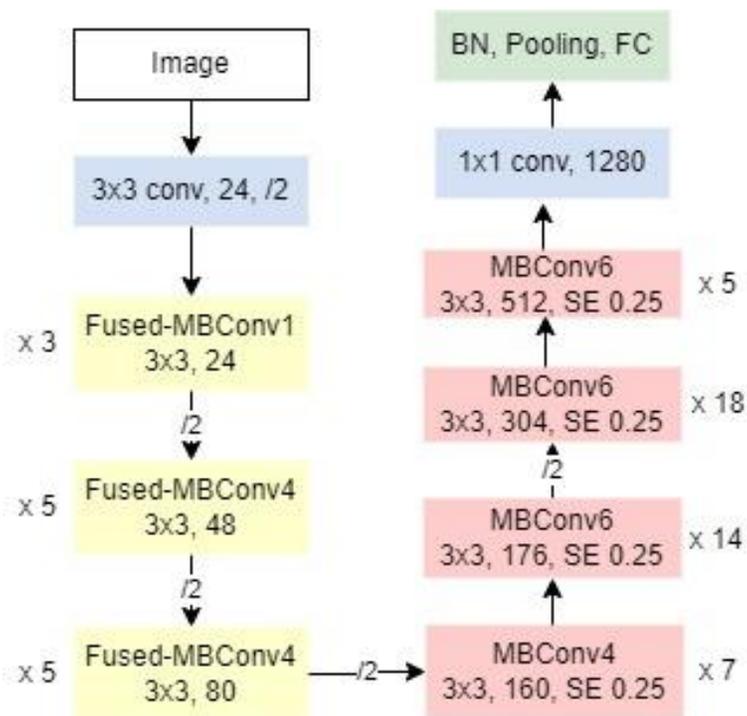


FIGURE 12. EfficientNetV2M architecture.

## PHYTOPARASITIC NEMATODE IDENTIFICATION

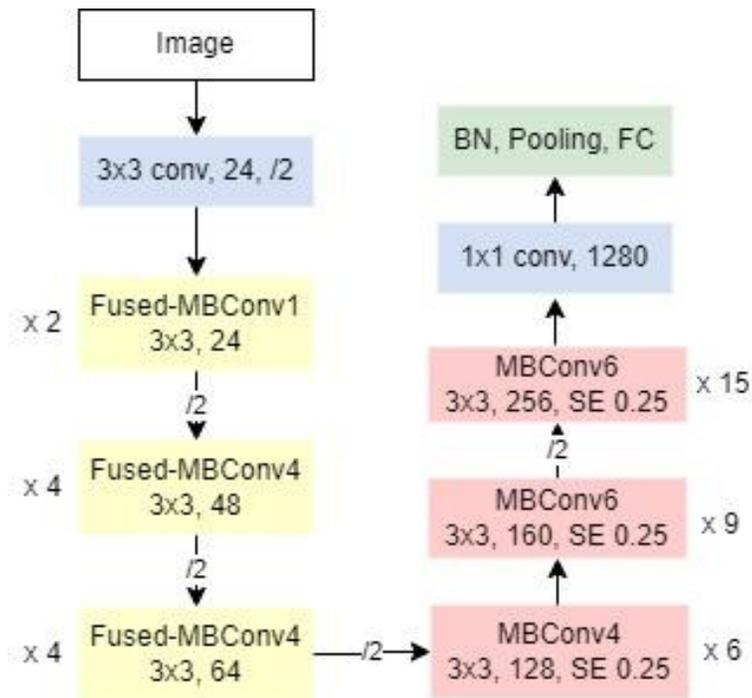


FIGURE 13. EfficientNetV2S architecture.

**2.3.4 Inception.** The scaling up of networks is the core of the inception model. The model uses aggressive regularization and factorized convolutions to improve processing efficiency [32]. In this study, InceptionV3 and InceptionResNetV2 were used. Configuration of InceptionV3, presented in Figure 14, is retained, and improves the network by employing factorized 7x7 convolutions and BatchNorm in the auxiliary classifier. To prevent the network from overfitting, a regularizing component has been added to the loss formula [32]. The InceptionResNetV2 model [33] is a hybrid Inception architecture incorporating elements from ResNet's functionality. This model incorporates residual connections in place of the Inception's filter concatenation stage. This method allows Inception to achieve all its benefits while maintaining its processing efficiency. The InceptionResNetV2 implemented in this research is given in Figure 15.

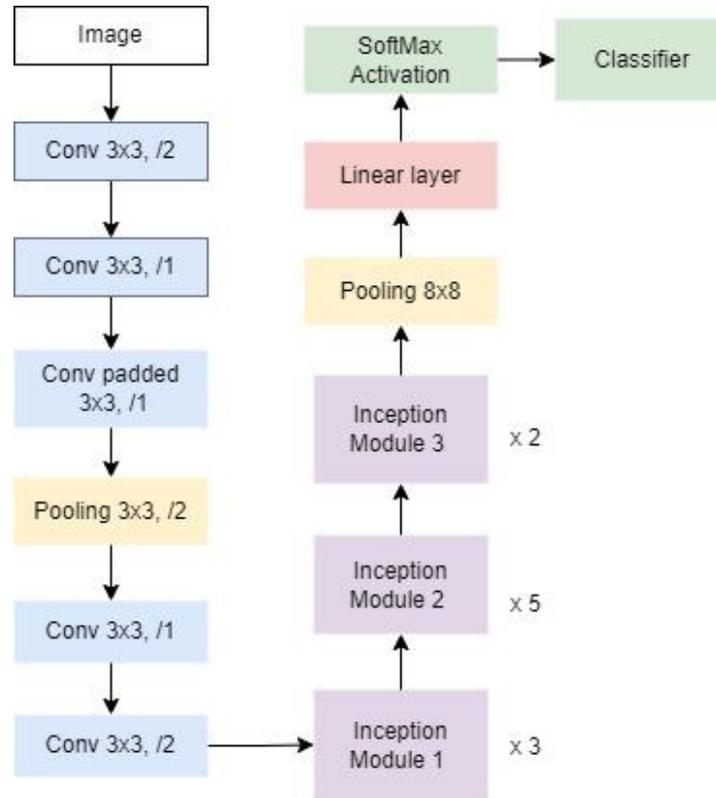


FIGURE 14. InceptionV3 architecture.

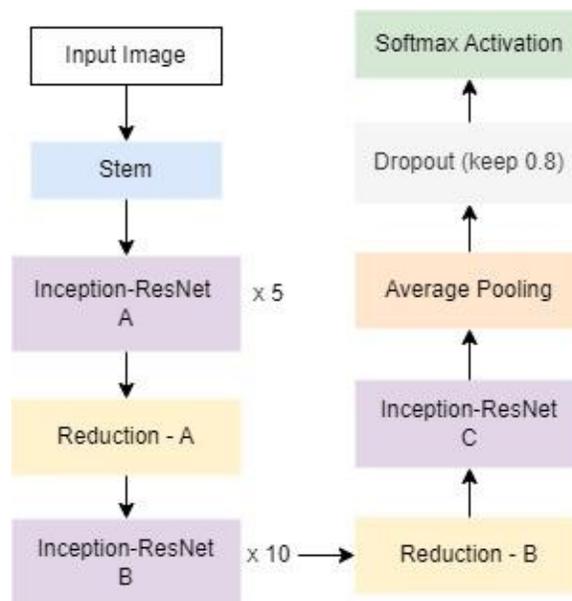


FIGURE 15. InceptionResNetV2 architecture.

## PHYTOPARASITIC NEMATODE IDENTIFICATION

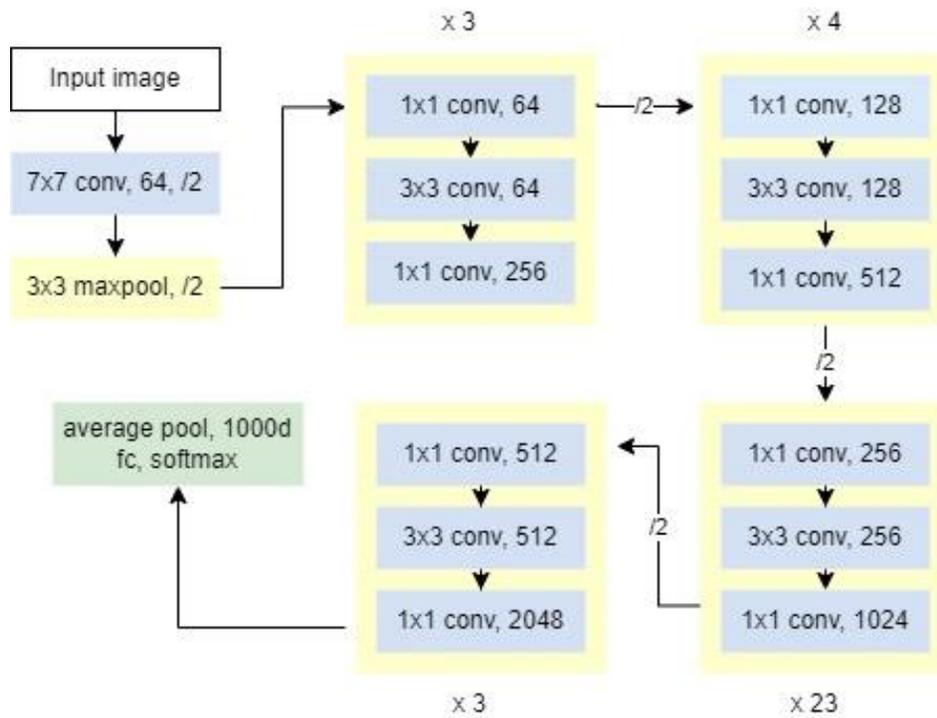


FIGURE 16. ResNet101V2 architecture.

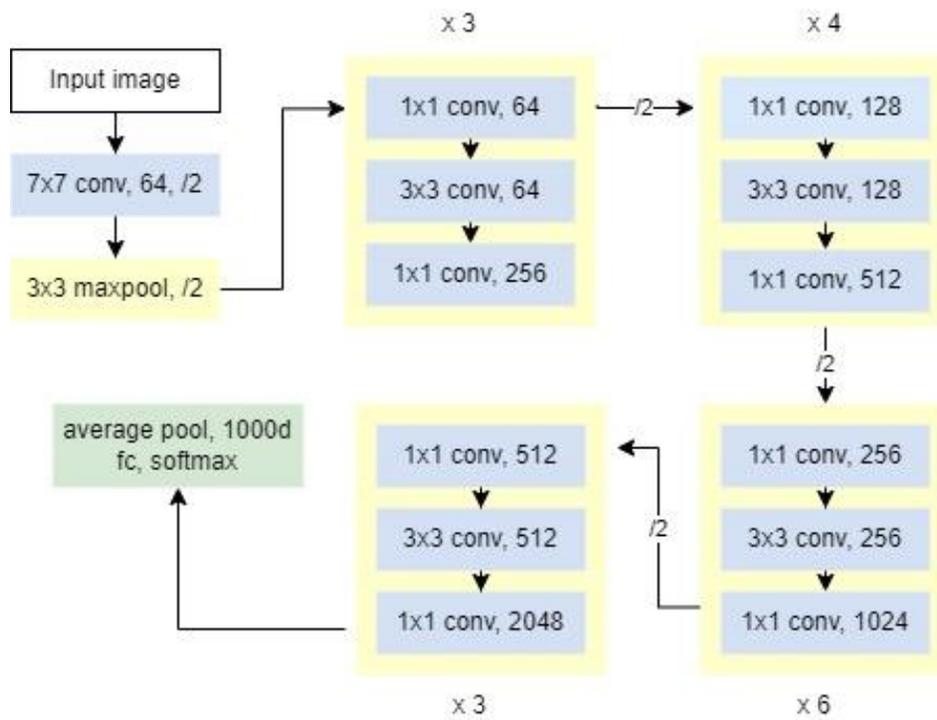


FIGURE 17. ResNet50V2 architecture.

**2.3.5 ResNet.** The Residual Network (ResNet) architecture was built to accommodate the VGG model, which is far deeper and challenging to train [34]. The training procedure can be made simpler with ResNet's residual learning framework. By omitting connections and offering shortcuts, the previous model's problem with disappearing gradients is resolved. The two core ResNet blocks are the identity and convolutional blocks. These blocks can be stacked to produce deep residual networks. By implementing identity mapping on the skip connections, the most recent iteration of this model, known as ResNetV2, enhanced the prior version. Each residual block's data transfer speed can be increased using this method [35]. The variation of version two's model lies in the number of layers. The ResNet101V2 and ResNet50V2 architectures implemented in this research are presented in Figure 16 and Figure 17, respectively.

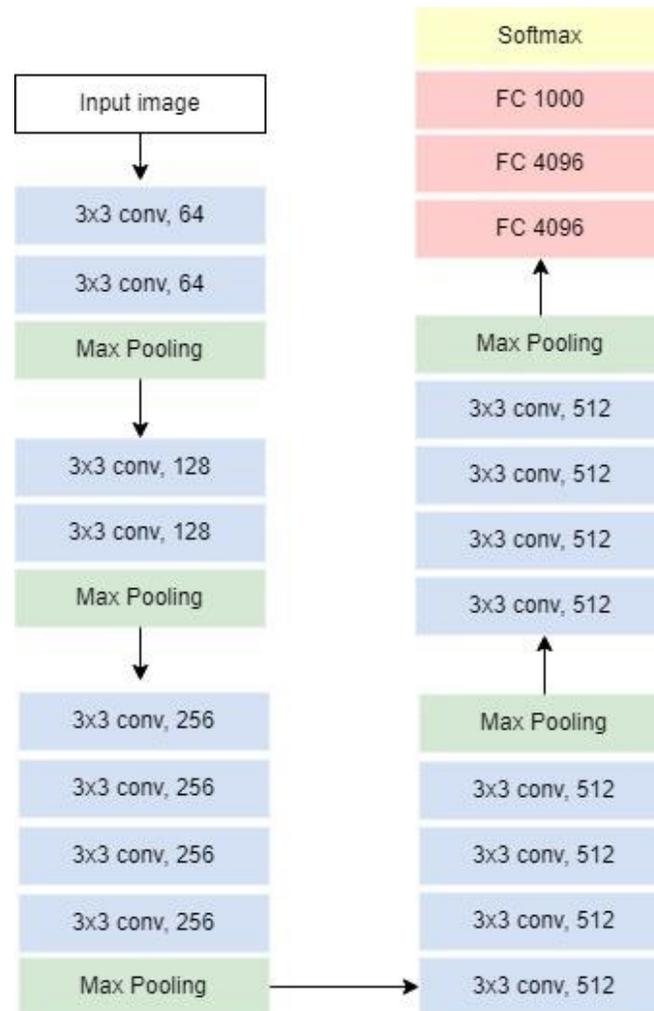


FIGURE 18. VGG19 architecture.



**2.4. Metric Performance Evaluation.** Several metrics were utilized to evaluate the performance of the CNN model: test accuracy, mean class accuracy, Precision, Recall, and F1 score. The test accuracy can be used to evaluate how frequently a machine learning classification algorithm classifies a data point correctly. It can be obtained by calculating the number of items accurately classified out of the total number, which the formula is given in Eqs. (1). Mean class accuracy is used to calculate average accuracy from all classes with the formula presented in Eqs. (2). Precision is achieved by calculating the actual class labels divided by the value obtained by adding all the results from the CNN model, as seen in Eqs. (3). Recall measures the effectiveness of the CNN model by summing True Positive and False Negative across all classes, as given in the formula in Eqs. (4). The F1 score is usually used for the unbalanced dataset by measuring the harmonic mean of Precision and recall with the formula in Eqs. (5).

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

$$\text{Mean Class Accuracy} = \frac{1}{c} \sum_{i=1}^c \frac{1}{n_i} \sum_{j=1}^{n_i} a_j^i \quad (2)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4)$$

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (5)$$

where TP – True Positive; FP – False Positive; TN – True Negative; FN –False Negative; c is the number of nematode genus;  $n_i$  the number of the image in  $i^{th}$  class; and  $a_j^i$  is the accuracy for image number  $j^{th}$  in  $i^{th}$  class.

### 3. RESULTS AND DISCUSSION

This study compared several pre-trained CNN models based on the test accuracy, mean class accuracy, precision, F1 score, precision, and recall, as tabulated in Table 4. The CoAtNet-0 outperformed other CNN pre-trained models with a test accuracy of 98.06%. Both Inception Family and VGG19 followed the test accuracy with the value of 95.15%. The lowest test accuracy was 89.32%, obtained from EfficientNetV2M and ResNet101V2 model. CoAtNet-0 was also

## PHYTOPARASITIC NEMATODE IDENTIFICATION

observed to have the highest mean class accuracy, followed by VGG19, with values of 97.87% and 96.53%, respectively. EfficientNetV2L was the least performed in mean class accuracy, with a value of 82.35%. Based on the Precision metric, a metric which describes how well a model predicts the positive class, CoAtNet-0 obtained the highest result of 98.18% followed with the InceptionV3 with value of 95.57%. The recall metrics measure the percentage of actual positive classes identified by the model. The CoAtNet-0 model resulted in the highest recall with the value of 98.06%, followed with Inception-ResNetV2, Inception-V3 and VGG-19 model with the value of 95.15%.

TABLE 4. Performance metrics result

<b>MODEL</b>	<b>Test Acc. (%)</b>	<b>Mean Class Acc. (%)</b>	<b>F1-score (%)</b>	<b>Precision (%)</b>	<b>Recall (%)</b>
CoAtNet-0	98.06	97.86	98.03	98.18	98.06
DenseNet121	91.26	90.55	91.35	92.23	91.26
DenseNet169	93.20	93.49	93.15	93.45	93.20
DenseNet201	92.23	93.98	92.28	93.04	92.23
EfficientNetV2B0	94.17	93.49	94.13	94.47	94.17
EfficientNetV2B3	92.22	92.42	92.21	92.61	92.23
EfficientNetV2L	91.26	82.35	90.73	91.19	91.26
EfficientNetV2M	89.32	90.35	89.44	91.09	89.32
EfficientNetV2S	94.17	94.77	94.19	94.90	94.17
InceptionResNetV2	95.15	95.76	95.18	95.44	95.15
InceptionV3	95.15	95.34	95.13	95.57	95.15
ResNet101v2	89.32	89.67	89.29	89.77	89.32
ResNet50v2	92.23	91.38	92.44	94.05	92.23
VGG19	95.15	96.53	95.10	95.55	95.15
Xception	94.17	94.09	94.09	94.37	94.17

It is also observed that the training process of EfficientNetV2S, InceptionResNetV2, InceptionV3, ResNet101v2, and VGG19 was stopped before 60 epochs, indicating that those models reached minimum loss faster compared to other models in phytoparasitic nematode dataset. Only three models, namely DenseNet121, EfficientNetV2B3, and EfficientNetV2M, reached up

to 100 epochs during the training process, which shows that those three models required more epochs to reach the optimum result. All models implemented in this study show satisfactory accuracy and loss results during training. However, most of the learning curves show that the model experienced overfitting. This result is presumably due to the small size of the dataset, with insufficient data samples to accurately represent the feature of each nematode genus morphology. Regarding stability and regularization, ResNet101v2 is observed to provide the best result compared to other CNN models. Only ResNet101v2 provides a good fit learning curve for model accuracy, as shown in Figure 20.

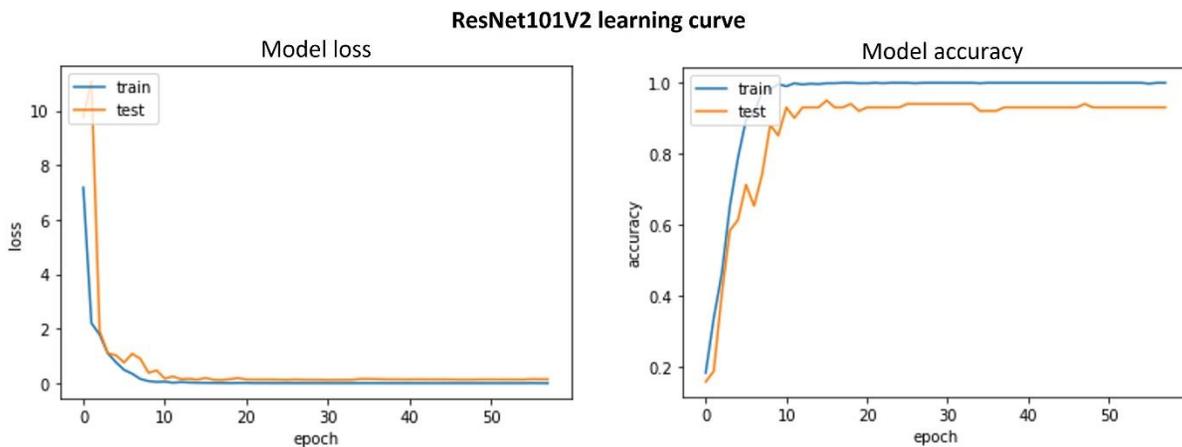


FIGURE 20. ResNet101V2 learning curve.

In conclusion, we recommend utilizing CoAtNet-0 for identifying phytoparasitic nematode in the small dataset due to its outstanding performance (98.06% test accuracy, 97.86% mean class accuracy, 0.9803 F1 score, 0.9818 Precision, and 0.9806 Recall). There is performance improvement from this result compared to the previous study [14]. Note that the previous study with the highest test accuracy of 97.94% resulted from the model implementing data augmentation, which means that the time required to train the model is longer due to the additional pre-processing technique. This study shows that using the right optimizer with fine tuning and proper learning rate can give higher performance.

#### **4. CONCLUSIONS**

This study demonstrated the performance comparison of Convolutional Neural Network approaches to identify phytoparasitic nematodes. The performance from CNN models implemented in this study, namely CoAtNet-0, DenseNet121, DenseNet169, DenseNet201, EfficientNetV2B0, EfficientNetV2B3, EfficientNetV2L, EfficientNetV2M, EfficientNetV2S, InceptionResNetV2, InceptionV3, ResNet101v2, ResNet50v2, VGG19, and Xception showing satisfactory results. CNN models utilized the transfer learning technique with SGD optimizer, and a good initial learning rate can perform well in multi-class phytoparasitic nematode identification. The highest performance was obtained from CoAtNet-0 with test accuracy, mean class accuracy, F1 score, Precision, and Recall of 98.06%, 97.86%, 0.9803, 0.9818, and 0.9806, respectively. The lowest test accuracy was 89.32%, obtained from EfficientNetV2M and ResNet101V2 model. The performance of CNN models can be used to further implement in automate the phytoparasitic nematode identification process.

#### **ACKNOWLEDGEMENT**

We would like to thank you for the support given by Universitas Multimedia Nusantara during this study.

#### **SOURCE OF FUNDING**

This study was funded by Universitas Multimedia Nusantara with Grant Number 062/PI/LPPM-UMN/III/2022.

#### **CONFLICT OF INTERESTS**

The author(s) declare that there is no conflict of interests.

#### **REFERENCES**

- [1] J.T. Jones, A. Haegeman, E.G.J. Danchin, et al. Top 10 plant-parasitic nematodes in molecular plant pathology, *Mol. Plant Pathol.* 14 (2013), 946-961. <https://doi.org/10.1111/mpp.12057>.

- [2] S. Singh, B. Singh, A.P. Singh, Nematodes: a threat to sustainability of agriculture, *Procedia Environ. Sci.* 29 (2015), 215-216. <https://doi.org/10.1016/j.proenv.2015.07.270>.
- [3] M. Bogale, A. Baniya, P. DiGennaro, Nematode identification techniques and recent advances, *Plants*. 9 (2020), 1260. <https://doi.org/10.3390/plants9101260>.
- [4] C.A. Silva, K.M.C. Magalhaes, A.D. Doria Neto, An intelligent system for detection of nematodes in digital images, in: *Proceedings of the International Joint Conference on Neural Networks, 2003.*, IEEE, Portland, OR, USA, 2003: pp. 612–615. <https://doi.org/10.1109/IJCNN.2003.1223431>.
- [5] N.B. Rizvandi, A. Pižurica, F. Rooms, et al. Skeleton analysis of population images for detection of isolated and overlapped nematode *C.Elegans*, in: *16th European Signal Processing Conference (2008)*.
- [6] B.T. Moore, J.M. Jordan, L.R. Baugh, WormSizer: high-throughput analysis of nematode size and shape, *PLoS ONE*. 8 (2013), e57142. <https://doi.org/10.1371/journal.pone.0057142>.
- [7] A. Toribio, L. Vargas, G. Kemper, A. Palomo, An algorithm to extract physical characteristics of nematodes from microscopic images of plant roots, in: *2018 IEEE International Conference on Automation/XXIII Congress of the Chilean Association of Automatic Control (ICA-ACCA)*, IEEE, Concepcion, 2018: pp. 1–5. <https://doi.org/10.1109/ICA-ACCA.2018.8609756>.
- [8] A. Akintayo, G.L. Tylka, A.K. Singh, et al. A deep learning framework to discern and count microscopic nematode eggs, *Sci. Rep.* 8 (2018), 9145. <https://doi.org/10.1038/s41598-018-27272-w>.
- [9] A. Abade, L.F. Porto, P.A. Ferreira, et al. NemaNet: A convolutional neural network model for identification of soybean nematodes, *Biosyst. Eng.* 213 (2022), 39-62. <https://doi.org/10.1016/j.biosystemseng.2021.11.016>.
- [10] J. Uhlemann, O. Cawley, T. Kakouli-Duarte, Nematode identification using artificial neural networks, in: *Proceedings of the 1st International Conference on Deep Learning Theory and Applications*, SciTePress, Lieusaint - Paris, France, 2020: pp. 13-22. <https://doi.org/10.5220/0009776600130022>.
- [11] X. Qing, Y. Wang, X. Lu, H. Li, X. Wang, H. Li, X. Xie, NemaRec: A deep learning-based web application for nematode image identification and ecological indices calculation, *Eur. J. Soil Biol.* 110 (2022), 103408. <https://doi.org/10.1016/j.ejsobi.2022.103408>.
- [12] X. Lu, Y. Wang, S. Fung, et al. I-nema: a biological image dataset for nematode recognition, preprint (2021). <https://doi.org/10.48550/arXiv.2103.08335>.

- [13] N. Angeline, N. Husna Shabrina, S. Indarti, Faster region-based convolutional neural network for plant-parasitic and non-parasitic nematode detection, *Indones. J. Electric. Eng. Computer Sci.* 30 (2023), 316-324. <https://doi.org/10.11591/ijeecs.v30.i1.pp316-324>.
- [14] N.H. Shabrina, R.A. Lika, S. Indarti, Deep learning models for automatic identification of plant-parasitic nematode, *Artif. Intell. Agric.* 7 (2023), 1-12. <https://doi.org/10.1016/j.aiaa.2022.12.002>.
- [15] J. Southey, *Laboratory methods for work with plant and soil nematodes*, Her Majesty's Stationary Office (HMSO), London, 1986.
- [16] P. Zhou, J. Feng, C. Ma, et al. Towards theoretically understanding why SGD generalizes better than ADAM in deep learning, preprint, (2020). <https://doi.org/10.48550/ARXIV.2010.05627>.
- [17] N.S. Keskar, R. Socher, Improving generalization performance by switching from ADAM to SGD, preprint, (2017). <https://doi.org/10.48550/ARXIV.1712.07628>.
- [18] B.C. Chen, A.X. Chen, X. Chai, et al. A statistical test of the effect of learning rate and momentum coefficient of sgd and its interaction on neural network performance, in: 2019 3rd International Conference on Data Science and Business Analytics (ICDSBA), IEEE, Istanbul, Turkey, 2019: pp. 276-281. <https://doi.org/10.1109/ICDSBA48748.2019.00065>.
- [19] [https://github.com/faizanahemad/data-science-utils/blob/master/data\\_science\\_utils/vision/keras/GradCam\\_with\\_misclassified\\_LRFinder.ipynb](https://github.com/faizanahemad/data-science-utils/blob/master/data_science_utils/vision/keras/GradCam_with_misclassified_LRFinder.ipynb), (accessed August 1, 2022).
- [20] J. Wu, X.Y. Chen, H. Zhang, et al. Hyperparameter optimization for machine learning models based on Bayesian optimization, *J. Electronic Sci. Technol.* 17 (2019), 26-40. <https://doi.org/10.11989/JEST.1674-862X.80904120>.
- [21] D. Masters, C. Luschi, Revisiting small batch training for deep neural networks, preprint, (2018). <https://doi.org/10.48550/ARXIV.1804.07612>.
- [22] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, *J. Big Data.* 6 (2019), 60. <https://doi.org/10.1186/s40537-019-0197-0>.
- [23] L. Shao, F. Zhu, X. Li, Transfer learning for visual categorization: a survey, *IEEE Trans. Neural Netw. Learn. Syst.* 26 (2015), 1019-1034. <https://doi.org/10.1109/TNNLS.2014.2330900>.
- [24] Keras, Keras applications. (2021). <https://keras.io/api/applications/> (accessed February 1, 2022).

- [25] J. Deng, W. Dong, R. Socher, et al. ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Miami, FL, 2009: pp. 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>.
- [26] <https://paperswithcode.com/sota/image-classification-on-imagenet> (accessed February 1, 2022).
- [27] Z. Dai, H. Liu, Q. V. Le, & M. Tan, CoAtNet: marrying convolution and attention for all data sizes. (2021). <https://doi.org/10.48550/arXiv.2106.04803>.
- [28] G. Huang, Z. Liu, L. van der Maaten, et al. Densely connected convolutional networks, preprint, (2016). <https://doi.org/10.48550/ARXIV.1608.06993>.
- [29] M. Tan, Q.V. Le, Efficientnet: rethinking model scaling for convolutional neural networks, preprint, (2019). <https://doi.org/10.48550/arXiv.1905.11946>.
- [30] T. Elsken, J. H. Metzen, F. Hutter, Neural architecture search: a survey, *J. Mach. Learn. Res.* 20 (2019), 1-21.
- [31] M. Tan, Q.V. Le, Efficientnetv2: smaller models and faster training, preprint, (2021). <https://doi.org/10.48550/arXiv.2104.00298>.
- [32] C. Szegedy, V. Vanhoucke, S. Ioffe, et al. Rethinking the inception architecture for computer vision, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016: pp. 2818–2826. <https://doi.org/10.1109/CVPR.2016.308>.
- [33] C. Szegedy, S. Ioffe, V. Vanhoucke, et al. Inception-v4, inception-ResNet and the impact of residual connections on learning, preprint, (2016). <https://doi.org/10.48550/ARXIV.1602.07261>.
- [34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Las Vegas, NV, USA, 2016: pp. 770–778. <https://doi.org/10.1109/CVPR.2016.90>.
- [35] K. He, X. Zhang, S. Ren, J. Sun, Identity Mappings in Deep Residual Networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), *Computer Vision – ECCV 2016*, Springer International Publishing, Cham, 2016: pp. 630–645. [https://doi.org/10.1007/978-3-319-46493-0\\_38](https://doi.org/10.1007/978-3-319-46493-0_38).
- [36] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, (2014). <https://doi.org/https://doi.org/10.48550/arXiv.1409.1556>.
- [37] F. Chollet, Xception: deep learning with depthwise separable convolutions, in: 2017 IEEE Conference on

## PHYTOPARASITIC NEMATODE IDENTIFICATION

Computer Vision and Pattern Recognition (CVPR), IEEE, Honolulu, HI, 2017: pp. 1800–1807.

<https://doi.org/10.1109/CVPR.2017.195>.