IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 10 December 2023, accepted 29 December 2023, date of publication 8 January 2024,
date of current version 16 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3351003

RESEARCH ARTICLE

Pest Identification Based on Fusion of Self-Attention With ResNet

SK MAHMUDUL HASSAN[1] **AND ARNAB KUMAR MAJI**[2], **(Senior Member, IEEE)**
[1]School of Computer Science and Engineering, VIT-AP University, Amravati 522237, India
[2]Department of Information Technology, School of Technology, NEHU, Shillong 93022, India

Corresponding author: SK Mahmudul Hassan (hassanmahmudul89@gmail.com)

ABSTRACT Pest identification is a challenging task in the agricultural sector, as accurate and timely detection of pests is essential for effective pest control and crop protection. Conventional approaches to pest detection, such as entomological knowledge and manual examination, take a lot of time and are prone to human mistakes. The advent of Deep Learning (DL) techniques has revolutionized the field of computer vision, enabling automated and efficient pest recognition systems.In this research, we compared the effectiveness of many deep learning models and suggested an enhanced approach for more effective feature extraction. In the proposed approach, we have incorporated two parallel attention mechanisms in ResNet architectures and it has a significant improvement in performance. Experimental result shows that the performance accuracy obtained in ResNet50-SA, ResNet101-SA, and ResNet152-SA is 99.80%, 88.48% and 96.68%, respectively. The performance of ResNet50-SA outperforms the other state of art deep learning by a large margin. The result shows that ResNet with self-attention (SA) has a better ability to extract features and focus on the important features which increases the performance.

INDEX TERMS Pest identification, deep learning, residual network, self attention.

I. INTRODUCTION

An integral component of the nation's economy is the agricultural sector. Pest damage yields have a significant impact on their productivity and quality. According to the FAO (Food and Agriculture Organisation), pests are one of the most important problems in agricultural production since they cause 20-40% of global crop losses. The identification and management of pests in agricultural systems are crucial for ensuring crop health and maximizing yields. Traditional approaches of pest identification often rely on visual inspection by human experts, which is inefficient, time-consuming, and subject to errors. Constant observation is seen to be one of the major challenges in agriculture. With the advancements in computer vision and machine learning (ML) techniques, there has been a growing interest in leveraging these technologies to automate the process of pest identification [1]. The traditional approach in ML consists of feature representation from the image and classifiers to categorize the images. The hand-crafted feature extraction technique includes Grey Level Co-Occurance Matrix (GLCM), Scale-Invariant Feature Transform (SIFT), Speeded Up Roboust Features (SURF), etc. The mainly used classifier are K-Nearest Neighbour (KNN), Support Vector Machine (SVM), Random Forest, Decision Tree etc.

Wen et al. [2] identified six different orchard insects, with the help of local feature extraction. Six different classifiers with cross-fold validation is used for classification and achieved maximum accuracy of 89.5% using Nearest Mean Classifier (NMC) and 88.4% using Support Vector Machine (SVM). Wang et al. [3] extracted several orders of geometrical features in automated insect identification and for classification, they have used Artificial Neural Network (ANN) and SVM and attain accuracy rate of 93% using ANN. Xiao et al. [4] identified four important vegetable pests using SIFT based feature descriptor and SVM classifier and recorded an average accuracy of 91.56%. Shape and moment invariant features used by Yaakob and Jain [5] to identify insects. Determining the optimal set of features is the key challenge in hand-crafted based approach. With the advancement in machine learning techniques, particularly deep learning techniques, it overcomes the challenges as it

The associate editor coordinating the review of this manuscript and approving it for publication was Utku Kose.

extracts the necessary and important features automatically. In classification, attention mechanism in deep learning will help to analyse the pixels in better way and improve the feature learning in pest recognition. In this paper, we have used self-attention (SA) in deep learning models for better extraction of features. The paper's primary contribution can be summed up as follows:

1) Nine different categories of pest with real field-conditioned images are collected. To expand the number of images in the dataset and strengthen the model's resilience, a number of data augmentation approaches are employed.
2) One improved ResNet model for identification of pest is proposed. In the original ResNet architecture, a parallel attention mechanism is integrated for better extraction of features.
3) A number of cutting-edge deep-learning models are used to compare the performance of the proposed model. Results indicate that the proposed model performed more accurately.

The remaining portion of this manuscript is structured as follows: Section II presents the literature on the identification of pests. Section III describes the materials and methods utilized in this study, Section IV presents about the dataset and analysis of result, and lastly, Section V culminates the paper with the concluding remarks.

II. LITERATURE REVIEW

Recently, deep-learning based approaches are used frequently in plant identification, plant disease detection, weed classification, as well as pest identification. Research on the identification of pests using deep learning is an emerging topic, and in recent times, several methods have been proposed.

Wang et al. [6] introduced deep learning architectures such as LeNet and AlexNet to classify various crop pests. They have analyzed the performances with different convolutional kernel, different filter size and achieved an accuracy rate of 90%. They created their own pests dataset for their work, which includes 30000 images and 82 distinct types of pests.

To detect the various pests, Liu et al. [7] suggested an 8-layer DL architecture based on AlexNet. To localize the pest region, they have adopted a contrast region-based methodology in their strategy. To achieve an accuracy of 95.1%, the author also optimized the parameter, which includes batch size, convolutional number, convolutional stride, dropout, and loss function.

In order to classify 24 different pest classes, Xia et al. [8] suggested a better network architecture based on the VGG-19 model. In their method, the authors employed the Region Proposal Network (RPN) to remove the irrelevant backdrop and retrieve the precise position of the pest from the feature map. In terms of mAp and training time, their suggested model performs better than state-of-the-art models like Single Shot Multibox Detector (SSD) and Fast Region-based Convolutional Neural Network (RCNN).

A fine-tuned Googlenet architecture was suggested by Yanfen Li et al. [9] to recognize various pests in natural scenes. In this study, the author gathered images from the internet and used certain captured photos to train the neural network. The model performance is validated using several data augmentation strategies as well as k-fold cross-validation. In contrast, the suggested model provides performance accuracy that is 6.22% greater than cutting-edge deep learning models.

To identify 24 distinct pests, Jiao et al. [10] presented one feature fusion module called the Anchor-free region convolutional neural network (AF-RCNN) model. The suggested method outperforms Faster R-CNN by 15.3% and YOLO detector by 39.4%.

Peng et al. [11] proposed an improved DenseNet based architecture named as MADN to classify the pests. MADN model aims to enhance feature extraction. They have used DenseNet121 as a base model and introduced Selective kernel unit (MADN-SK), the Representative batch normalization (MADNRBN) module, and the ACON activation function (MADN-ACON) into the DenseNet. On HQIP102 data set author recorded 5.17% higher accuracy than DenseNet121.

Wang et al. [12] implemented VggA, Vgg16, Inception V3, and ResNet50 in the identification of pests in crops and designed a lightweight CNN model named as CPAFNet. This model was used to classify 20 different insect species, and it had a 92.63% accuracy rate.

Thenmozhi and Reddy [13] classified different pests using a number of pre-trained deep learning architectures, including AlexNet, ResNet, GoogLeNet, and VGGNet. There has been a proposal for a single CNN model that consists of six convolutional layers, five maximum pooling layers, one fully connected layer, and one output layer. The performances were assessed using three distinct datasets, NBAIR, Xie1, and Xie2, and the accuracy rates were 96.75%, 96.47%, and 95.77%, respectively.

Cheng et al. [14] used a deep residual network to identify 10 different categories of crop pests in a complex background. In comparison with SVM and back propagation, their model performs better and achieved an accuracy rate of 98.67%.

Khanramaki et al. [15] ensembled different deep-learning models to identify three common citrus pests. With an accuracy rate of 99.04%, the model surpassed some other deep learning models when assessed using 10-fold cross-validation.

Guo et al. [16] proposed multi-scale local context features and the self-attention mechanism to identify Chinese agricultural diseases and pest. The original BiLSTM-CRF model is enhanced by fusion of multi-scale local context features extraction using CNN with different kernel sizes.

Zhang et al. [17] proposed one modified dilated residual network to identify stored grain pests. In their approach, to improve the vision of the convolution, a dilated convolution is used with residual connection. 5-fold cross-validation is used to evaluate the performance and they recorded an average accuracy of 96.72%.

TABLE 1. Performance comparison with other dataset.

Paper	Method	Dataset	Class	Accuracy(%)
Zhao et al. [23]	ResNet50 with PCSA	Internet source image	10	98.17
Tetila et al. [24]	Inception V3 ResNet50 VGG16, VGG19 Xception	UAV captured image	13	93.82
Khanramaki et al. [15]	AlexNet, VGG16, ResNet50 InceptionResNetV2	1774 captured image	3	99.04
Cheng et al. [14]	ResNet101	550 collected image	10	98.67
Thenmozhi et al. [13]	AlexNet, VGG GoogleNet ResNet Proposed CNN	NBAIR dataset	40	97.47
Wang et al. [12]	CPAFNet	CPAF dataset	20	92.26
Ayan et al. [25]	GAEnsemble	D0 dataset	40	98.81
Prasath B. et al. [26]	ResNet50, VGG16, Weight Optimized deep neural network	IP102 dataset	40	96
Denan Xia et al. [8]	VGG19 with RPN	Field Image	24	89.22(mAP)
Hongxing Peng et al. [11]	Densly connected CNN	HQIP102 dataset	102	75.28
Yanfen Li et al. [9]	Fine-tuned GoogLeNet	Internet source image	10	96.67
Lin Jiao et al. [10]	AF-RCNN	Captured image	24	56.4(aAP)
Xuchao Guo et al. [16]	Multi sale self-attention CNN	AgCNER dataset	11	94.15
Y. A. Nanehkaran et al. [27]	CNN Model	Collected image from Internet	13	91.33
Yingying Zhang et al. [17]	Modified dilated residual network	Collected from lab and internet	6	96.72
Junde Chen et al. [28]	Es-MbNet	Collected plant disease image	9	99.37
Qiang Dai et al. [19]	Fusion of residual and dense connection with self-attention	Xie1 dataset	24	-
Ching-Ju Chen et al. [20]	YOLO v3 and LSTM	Collected field image	1	90
Junde Chen et al. [22]	MAM-IncNet	Camellia oleifera leaf images	5	95.87

Zhang and Liu [18] identified orange diseases and pests by utilizing a combination of self-attention with DenseNet architecture. In order to improve the ability to extract the lesion features, position self-attention and channel self-attention are employed in this research. This model is able to identify six kinds of orange illnesses and pests with an accuracy of 96.90%.

Dai et al. [19] proposed a low-resolution pest identification based on a fusion of quadra-attention with residual and dense connection. To increase the dataset size in their work, the authors utilized a generative adversarial network (GAN), which significantly improved the model's performance. The use of both residual and dense connection retains more information from previous layer and makes the model trainable with a deeper layer by reducing the number of parameters.

Chen et al. [20] proposed one smart pest identification system based on Artificial Intelligence and Internet of Things (AIoT) and classified 24 different categories of pests. In this work, the author used YoLo V3 for recognition of pests and also used LSTM to predict the occurrence of pests by gathering weather information from the environment.

Li et al. [21] implemented ResNeXt-50 model to classify several pests and achieved an accuracy rate of 86.95%. The performance of the designed model with various combinations of transfer-learning, data augmentation, and learning rate strategies was compared by the author. They have demonstrated that performance is improved by combining transfer learning with fine-tuning and cutmix.

Chen et al. [22] identified pests and diseases in Camellia oleifera plant. In this work, the author proposed a deep-learning model called MAM-IncNet, where the author used an optimized inception layer in SSD and VGG16 framework as a feature extractor. Five different types of pests were identified and achieved an accuracy rate of 95.87%. Table 1 shows the summarization of the existing works.

III. MATERIALS AND METHODOLOGY

A. SYSTEM OVERVIEW

In this work, we have identified different pests that affect the production of crops. At first, we collected the images from

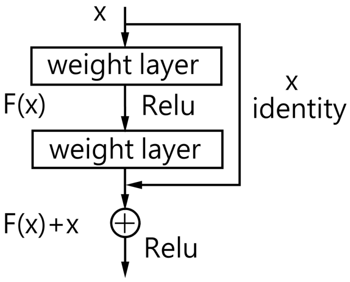


FIGURE 1. **Residual block.**

the internet and divided the dataset randomly into training and validation sets. Several data augmentation techniques are employed to enhance the size of the dataset. To extract the robust features, we have introduced a self-attention mechanism in ResNet architectures to identify different pests. Using the Adam optimizer and a learning rate of 0.001, the models are trained and evaluated. A brief explanation of the proposed model is explained in the subsequent sections.

B. DEEP LEARNING ARCHITECTURES

The Deep Learning (DL) architectures that are used in this work are discussed in this section. Here, we investigate the performances of different ResNet architectures e.g. ResNet50, ResNet101, ResNet152, ResNet50V2, ResNet101V2, and ResNet152V2. We have also proposed an attention-based residual model and implemented the ResNet models with attention.

1) RESIDUAL NETWORK

Residual architecture is similar to CNN, having convolution, pooling, activation, and fully connected layers. Only the identity connection between the layers makes ResNet unique. Increasing the depth of deep learning models results in a degradation in performance. He et al. [29] first introduced the term residual network in their paper and won 1st prize with an error rate of 3.57% in the ILSVRC 2015 classification competition. The key idea behind ResNet is the use of residual blocks, which are composed of shortcut connections or "skip connections." These connections allow the network to skip one or more layers during forward propagation, enabling the direct flow of information from one layer to a later layer. The block diagram of the residual network is shown in Figure 1.

Assuming x represents the input to the residual network, $F(x)$ represents the output obtained by processing two weight layers with x. The output of the residual network is $H(x)$ and is obtained by adding x with $F(x)$. The final generated is given as follows:

$$F(x) = w_{n+1} ReLU(w_n x)$$
$$H(x) = F(x) + x$$

Here w_n, w_{n+1} represents the weight of the two layers.

2) ResNet ARCHITECTURES

Comprising 48 convolutional layers, one max-pooling layer, and one global average pooling layer, ResNet50 is a 50-layer architecture. Similar to ResNet34, the primary layer contains one convolution layer using 7 × 7 convolution filter and one pooling layer. Instead of a 2-layer bottleneck block, ResNet50 uses a 3-layer bottleneck block, which increases the performance accuracy as compared with Resnet34 architecture. In ResNet50 architecture, it uses 16 residual blocks. The number of floating point operations (FLOPs) of this architecture is 3.8×10^9. ResNet50V2 is the improved version of ResNet50 architecture. For improved feature extraction, just the propagation formulation of the connections between blocks is changed. After the addition layer in the residual block, the last ReLU activation function is removed in ResNetV2 to add the second non-linearity as an identity mapping. Figure 3 compares the flow diagram of residual connections in ResNetV1 and ResNetV2 architecture. Figure 2 displays the ResNet50 architecture block diagram. The layer structure of ResNet101 and ResNet152 is similar to ResNet50, except the architecture has 33 and 50 residual blocks, respectively. These residual blocks increase the depth and parameters of the models. The number of FLOPs used is 7.6×10^9 and 11.3×10^9 in ResNet101 and ResNet152 architecture, respectively.

C. SELF-ATTENTION

Based on the job at hand, a neural network can selectively focus on various portions of the input data through a mechanism called self-attention. In the context of convolutional neural networks (CNNs), self-attention can be used to enhance the feature maps generated by the convolutional layers. In recent times, self-attention has gained much popularity, and has been widely used in various domains such as machine translation [30], [31], language processing [32] speech recognition [33], plant disease detection [34] etc. The basic idea is to calculate a set of attention weights for each feature map, indicating the importance of each spatial location in the feature map for the final prediction. These attention weights can then be used to weight the feature maps before passing them on to the next layer in the network.

The use of self-attention in CNNs has been shown to improve performance on a wide range of image classification and segmentation tasks by allowing the network to focus on the most relevant parts of the input data. However, it can also be computationally expensive, especially for large input images, and may require careful tuning of hyper-parameters to achieve optimal performance.

D. RESNET WITH SELF-ATTENTION

ResNet architecture with self-attention layer is shown in Figure 4. Here, the architecture of the state-of-art ResNet model is termed as BaseNet, which contains the ResNet models along with global average pooling without a

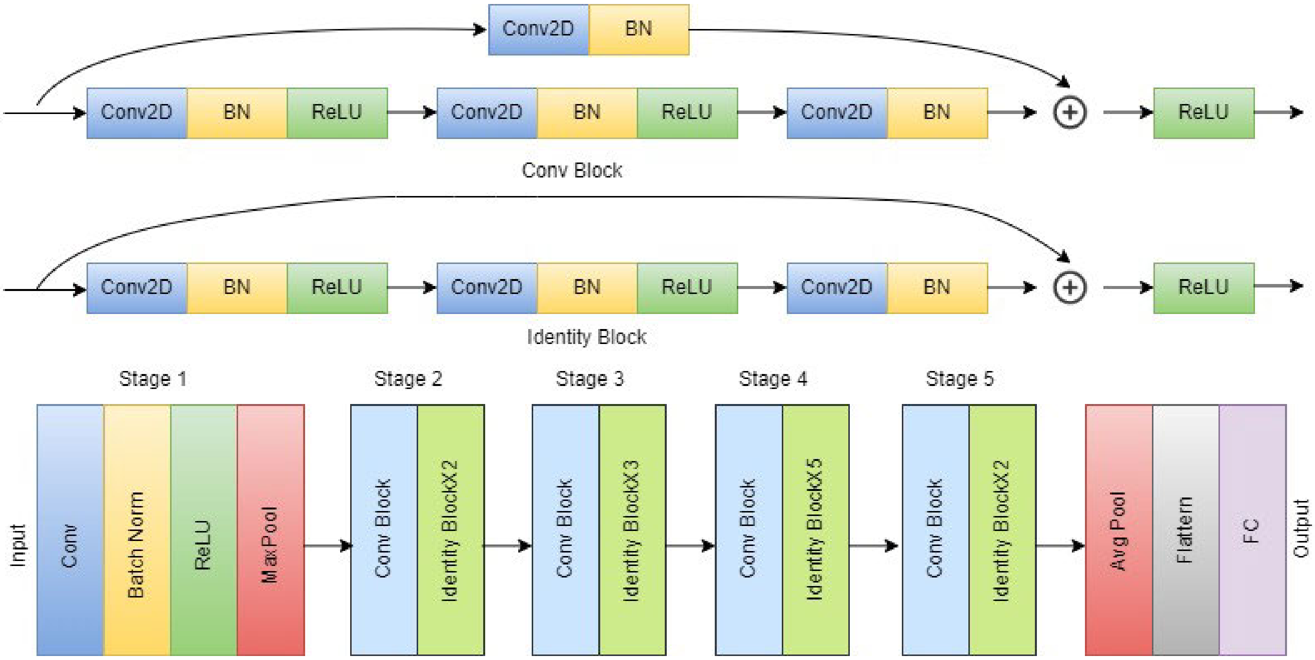


FIGURE 2. ResNet-50 architecture.

TABLE 2. Layer details of the proposed self-attention in ResNet.

Layer name	Kernel Size	Output Size
Input Layer	-	224 × 224
Conv Layer	$7 \times 7, 64$	112 × 112
Pooling Layer	3 × 3	56 × 56
Stacked of Conv layer1	$\begin{bmatrix} 1 \times 1, & 64 \\ 3 \times 3, & 64 \\ 1 \times 1, & 256 \end{bmatrix} \times 3$	56 × 56
Stacked of Conv layer2	$\begin{bmatrix} 1 \times 1, & 128 \\ 3 \times 3, & 128 \\ 1 \times 1, & 512 \end{bmatrix} \times 4$	28 × 28
Stacked of Conv layer3	$\begin{bmatrix} 1 \times 1, & 256 \\ 3 \times 3, & 256 \\ 1 \times 1, & 1024 \end{bmatrix} \times 6$	14 × 14
Stacked of Conv layer4	$\begin{bmatrix} 1 \times 1, & 512 \\ 3 \times 3, & 512 \\ 1 \times 1, & 2048 \end{bmatrix} \times 3$	7 × 7
Avg Pooling	-	1024
Attention layer1	-	1024
Attention layer 2	-	1024
Concatenation	-	3072
Dense layer	-	9

classification layer. The BaseNet is used to extract the feature vector from the CNN models. Next, we create two parallel attention branches using two Dense layers. The first attention branch calculates attention probabilities, while the second attention branch applies these probabilities to the feature vector. Element-wise multiplication between the attention probabilities and the feature vector is performed. Finally, we concatenate the attention vectors with the feature vector. Figure 5 shows the attention module used in this work. Table 2 shows the layer details of the proposed ResNet50 with Self-Attention.

IV. RESULTS

A. EXPERIMENTAL SETUP

In this section, we have primarily discussed the tools and the hardware used in the experiment. The programming language used for the DL model is Python with Python 3.6 version. Deep learning tools such as Keras, Tensorflow and OpenCV are used. To implement the DL algorithm, we have used Google Colab Pro platform with GPU.

B. DATASET DESCRIPTION

In this paper, we have used open-access crop pests datasets [35] and also collected some images from the internet. The pest dataset consists of 3150 images of 9 different crop pests namely aphids, armyworm, bollworm, beetle, grasshopper, mites, sawfly, mosquito, stem borer. Table 3 shows the detail description of the dataset. Sample images from the dataset are displayed in Figures 6 and 7. To make the dataset uniform the images are resized to 224 × 224 size. Different data augmentation techniques are used to increase the dataset size. Data augmentation effectively increases the diversity among the data and reduces data imbalance issues. In this work, flipping, contrast enhancement, rotation, brightness enhancement, and saturation are applied to increase the dataset size. Figure 8 shows some of the enhanced images. Table3 shows the detailed description of the dataset used along with scientific names.

C. PERFORMANCE EVALUATION

We have assessed the model's performance in this experiment using categorical cross-entropy loss and accuracy. To assess the model's performance, we have additionally taken into

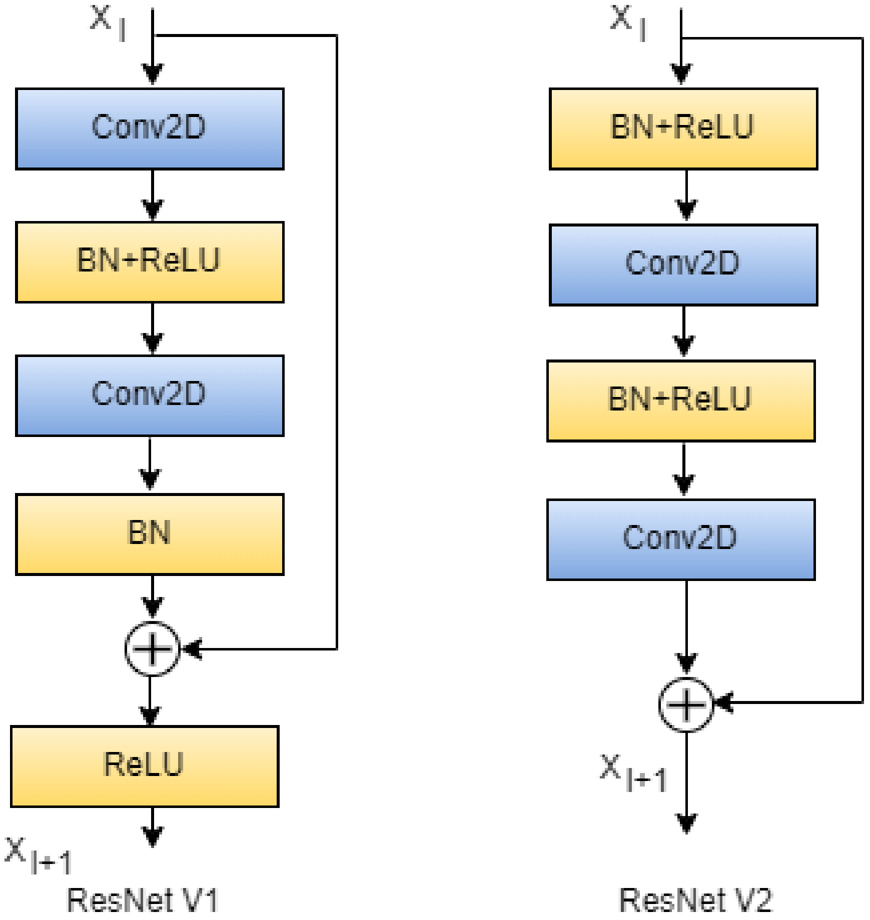


FIGURE 3. Architecture of ResNetV1 and ResNetV2.

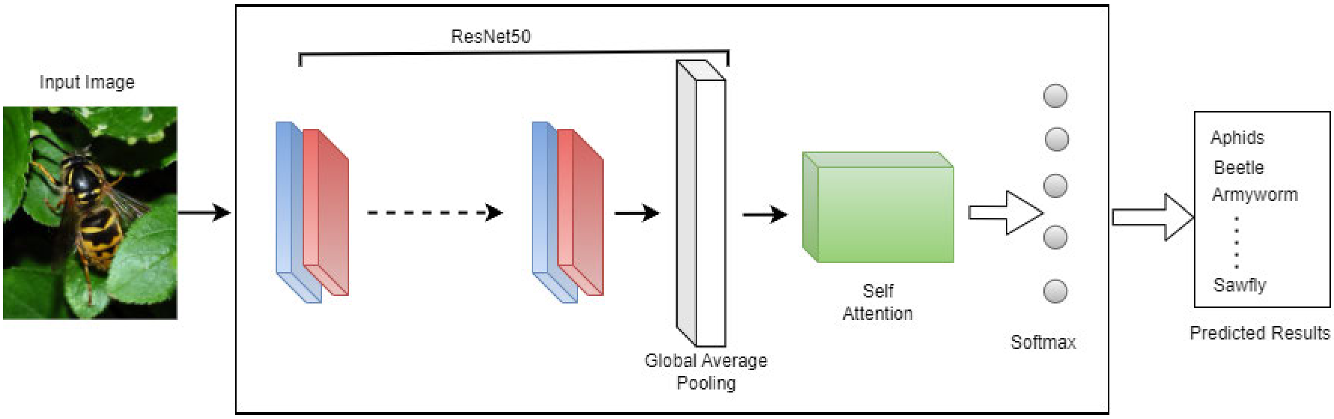


FIGURE 4. Overall architecture of ResNet with self-attention.

account specificity, f1-score, accuracy, and recall. The performance matrices are expressed as- *Accuracy:* It is the statistical measure of how the classifier classifies the images and is expressed by the following equation as shown.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \tag{1}$$

Precision: It is defined as the number of positive predictions made correctly as true positive.

$$Precision = \frac{T_p}{T_p + F_p} \tag{2}$$

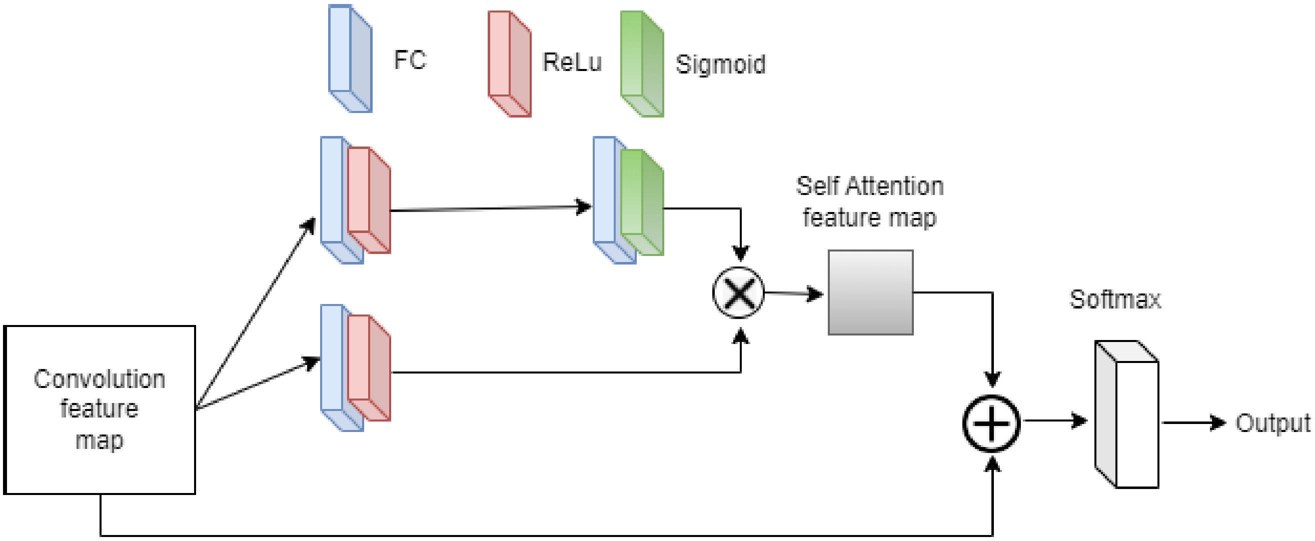


FIGURE 5. Block diagram of self attention network. In this diagram ⊕ represents the concatenation operation and ⊗ represents the multiplication operation.

TABLE 3. Dataset description.

Pest Name	Scientific Name	Class	Original image	Augmented Image
Aphids	Aphidoidea	c1	350	500
Armyworm	Spodoptera Frugiperda	c2	350	500
Beetle	Coleoptera	c3	350	500
Bollworm	Pectinophora Gossypiella	c4	350	500
Grasshopper	Gomphocerinae	c5	350	500
Mites	Acariformes	c6	350	500
Mosquito	Culicidae	c7	350	500
Sawfly	Symphyta	c8	350	500
Stem borer	Scirpophaga Incertulas	c9	350	500

(a) Aphids
(b) Beetle
(c) Mites

FIGURE 6. Sample images of pests.

TABLE 4. Hyperparameter used.

Optimizer	Batch size	Learning rate	Weight decay	Epoch
Adam	32	0.001	0.0001	50

Recall: It is also called sensitivity, defined as the correctly predicted positive instance over the total positive instance.

$$Recall = \frac{T_p}{T_p + F_n} \tag{3}$$

Specificity: It is defined as the correctly predicted negative instance to the total number of negative instances.

$$Specificity = \frac{T_n}{T_n + F_p} \tag{4}$$

f1-score: It is defined as the harmonic mean of precision and recall.

$$f1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

(a) Grasshopper (b) Armyworm (c) Sawfly

FIGURE 7. Sample images of pests.

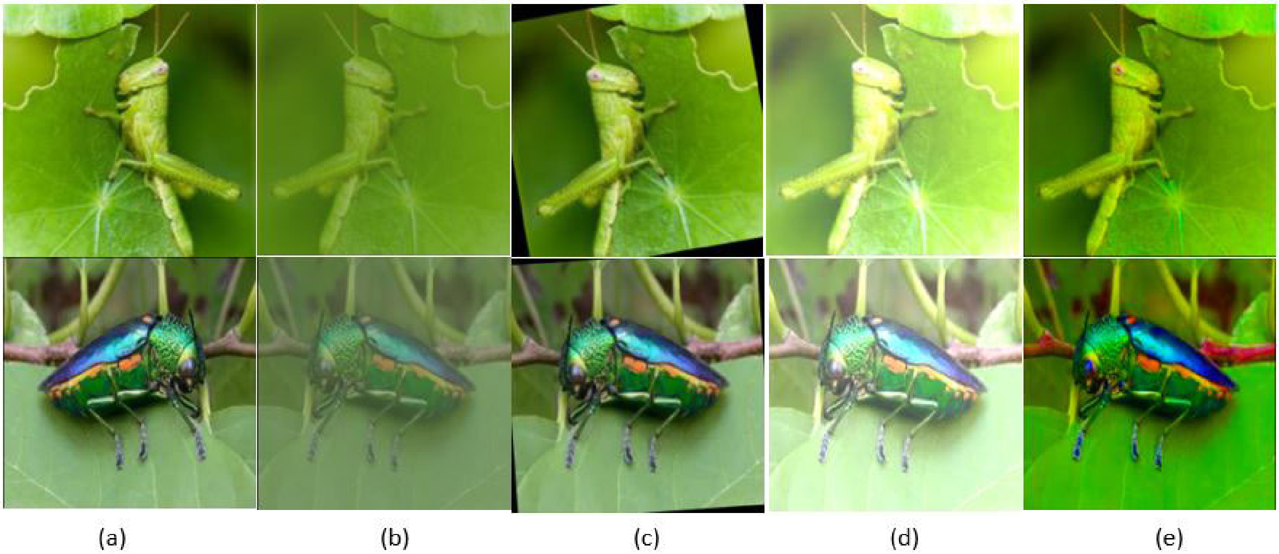


FIGURE 8. Augmented images processed by (a) Flipping (b) Contrast adjustment (c) Rotation (d) Changing Brightness (e) Saturation.

TABLE 5. Performance comparison of different ResNet models.

Model	Training Acc	Training loss	Validation Acc	Validation loss	Epoch
ResNet50	0.3457	1.9407	0.2832	2.0480	50
ResNet50V2	0.9961	0.0219	0.9578	0.1335	50
ResNet101	0.3301	1.9435	0.2531	2.0328	50
ResNet101V2	0.9980	0.0197	0.9623	0.1248	50
ResNet152	0.3438	1.9330	0.2452	2.0318	50
ResNet152V2	0.9961	0.0303	0.9598	0.1345	50
ResNet50 with self-attention	1.0000	0.0022	0.9980	0.0107	50
ResNet101 with self-attention	0.9707	0.0961	0.8848	0.5982	50
ResNet152 with self-attention	0.9871	0.0396	0.9668	0.1620	50

Execution time: Time required to complete one epoch while training the model.

In this case, true positive is denoted by T_p, true negative by T_n, false positive by F_p, and false negative by F_n.

D. RESULTS

The dataset is randomly splitted into an 80% training and a 20% validation set in order to assess the model's performance. The deep learning models with varying batch sizes, learning rates, and weight decay have been fine-tuned to obtain the optimum result. Table 4 displays the hyper-parameter that was utilized in the model's training. The performances of several ResNet models in terms of training accuracy, training loss, validation accuracy, and validation loss are displayed in Table 5. Figure 12- Figure 20 shows the accuracy and loss of the implemented ResNet models. From

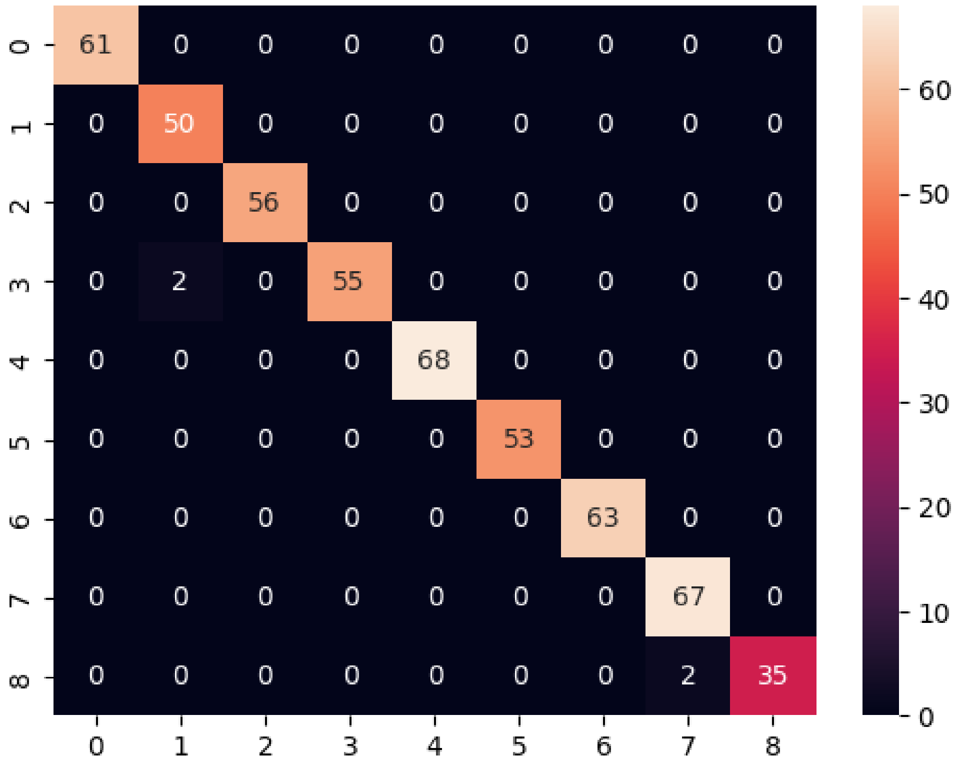


FIGURE 9. Confusion matrix of ResNet50-SA.

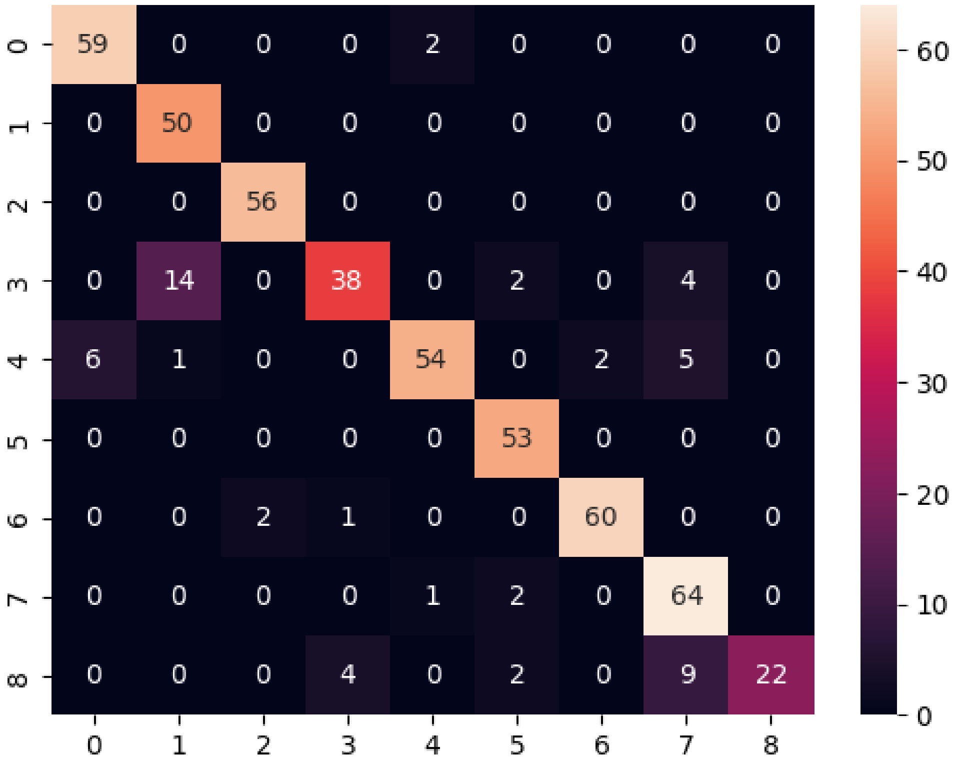
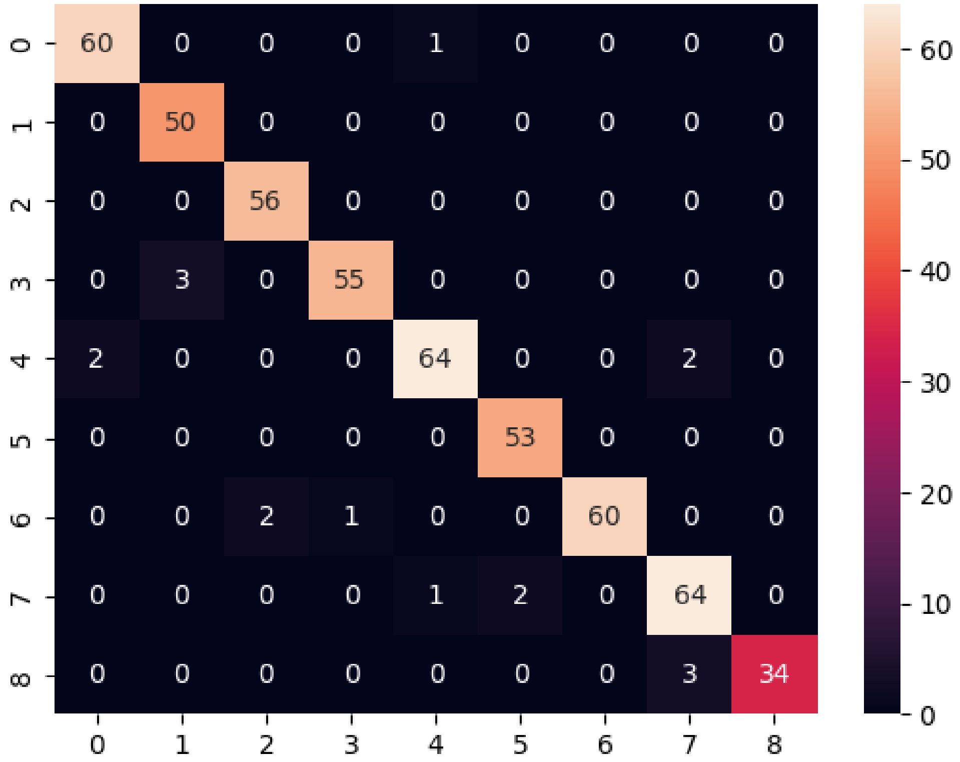
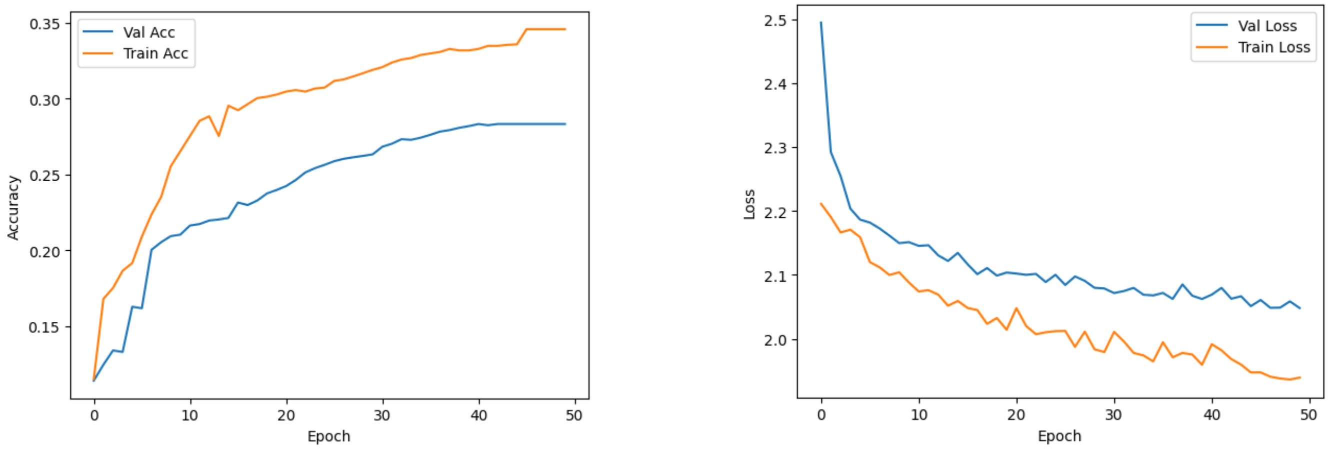


FIGURE 10. Confusion matrix of ResNet101-SA.



	0	1	2	3	4	5	6	7	8
0	60	0	0	0	1	0	0	0	0
1	0	50	0	0	0	0	0	0	0
2	0	0	56	0	0	0	0	0	0
3	0	3	0	55	0	0	0	0	0
4	2	0	0	0	64	0	0	2	0
5	0	0	0	0	0	53	0	0	0
6	0	0	2	1	0	0	60	0	0
7	0	0	0	0	1	2	0	64	0
8	0	0	0	0	0	0	0	3	34

FIGURE 11. Confusion matrix of ResNet152-SA.



(a) Accuracy

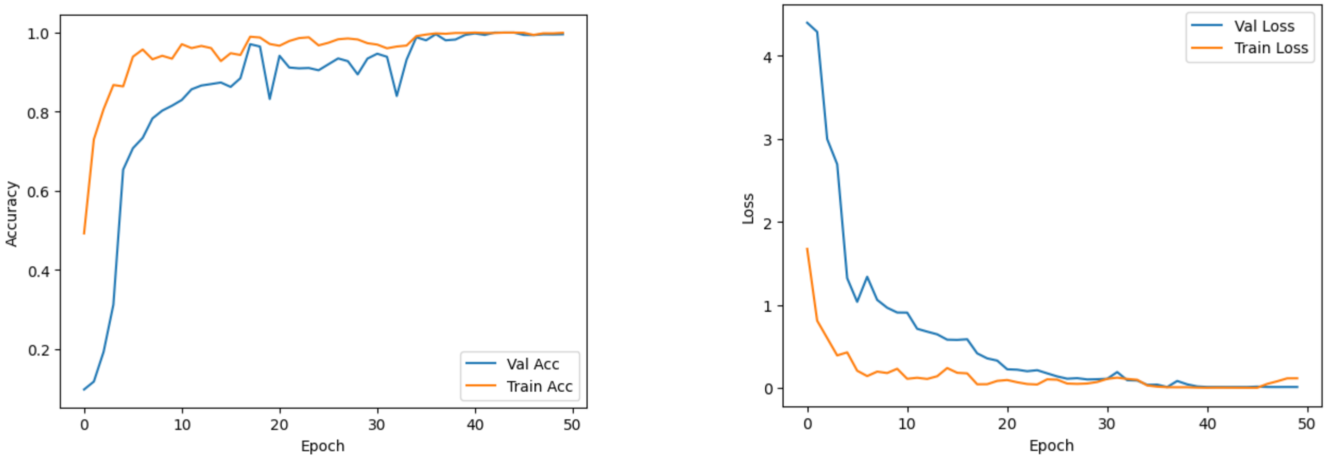
(b) Loss

FIGURE 12. Performance results on ResNet50 model.

the figures, it can be seen that ResNet with self-attention gives higher performances than state-of-the-art ResNet models.

Precision, recall, f1-score, and specificity have all been assessed as additional metrics for the DL models. Table 6 displays the performance result in terms of precision, recall, specificity, and f1-score. Table 6 demonstrates that while all of the models' training times are nearly identical, their performances vary greatly.

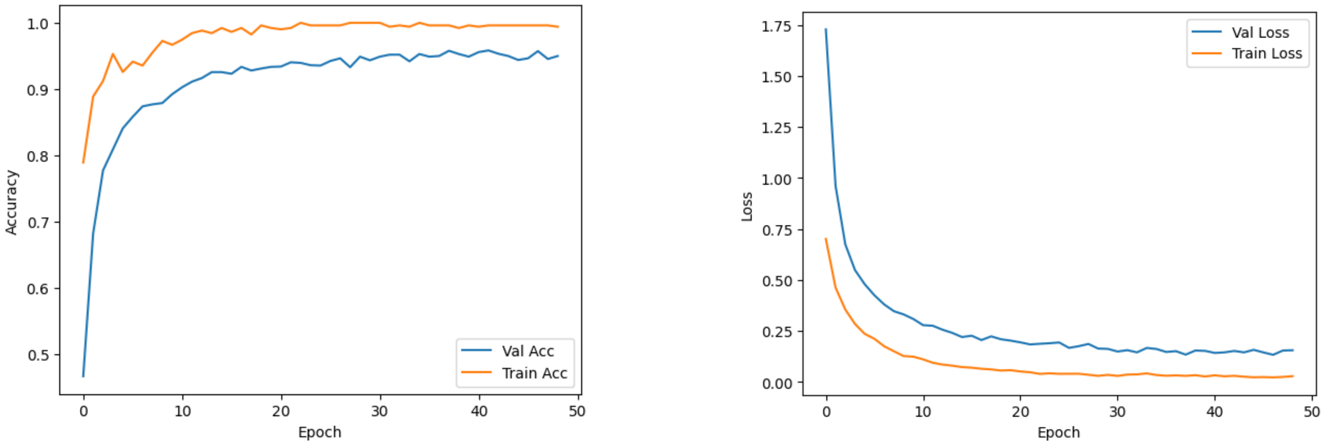
Additionally, we further verify the performance of the proposed approach for the identification of each pest class. Table 7 shows the experimental result of ResNet50-SA on each pest class. Table 7 shows that the proposed ResNet50-SA approach can accurately identify the pest images with higher performance accuracy. To demonstrate the performance completely, we have drawn the confusion matrix of the proposed models. Figure9-Figure11 shows the confusion matrix of ResNet50-SA, ResNet101-SA and ResNet152-SA,



(a) Accuracy

(b) Loss

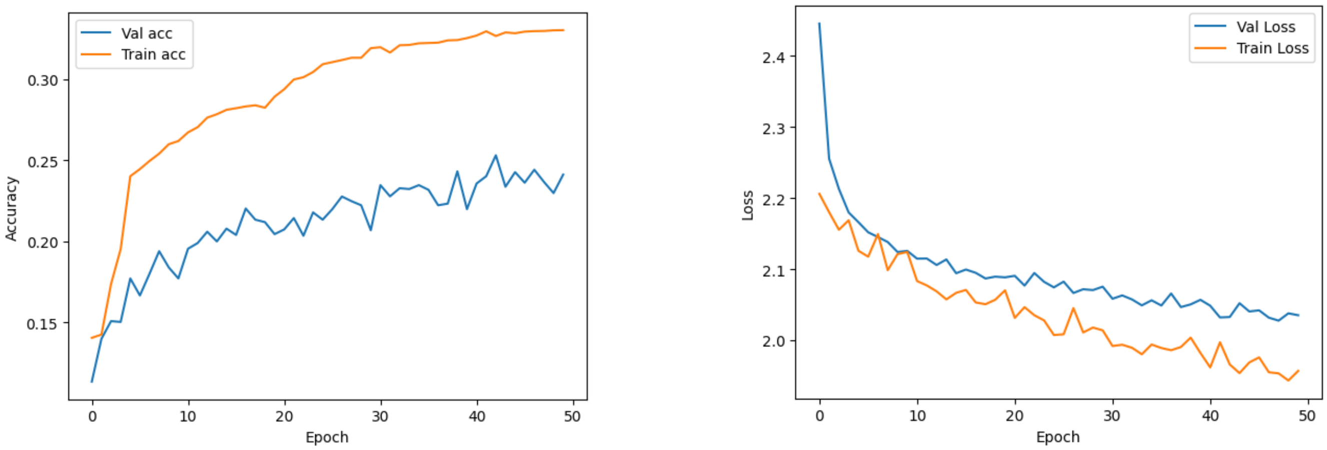
FIGURE 13. Performance results on ResNet50-SA model.



(a) Accuracy

(b) Loss

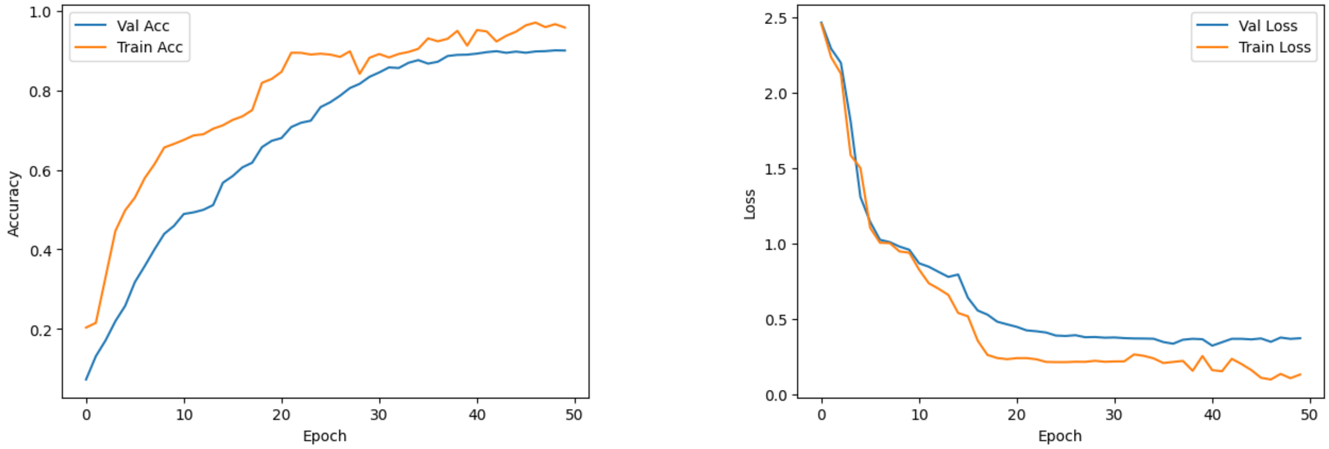
FIGURE 14. Performance results on ResNet50V2 model.



(a) Accuracy

(b) Loss

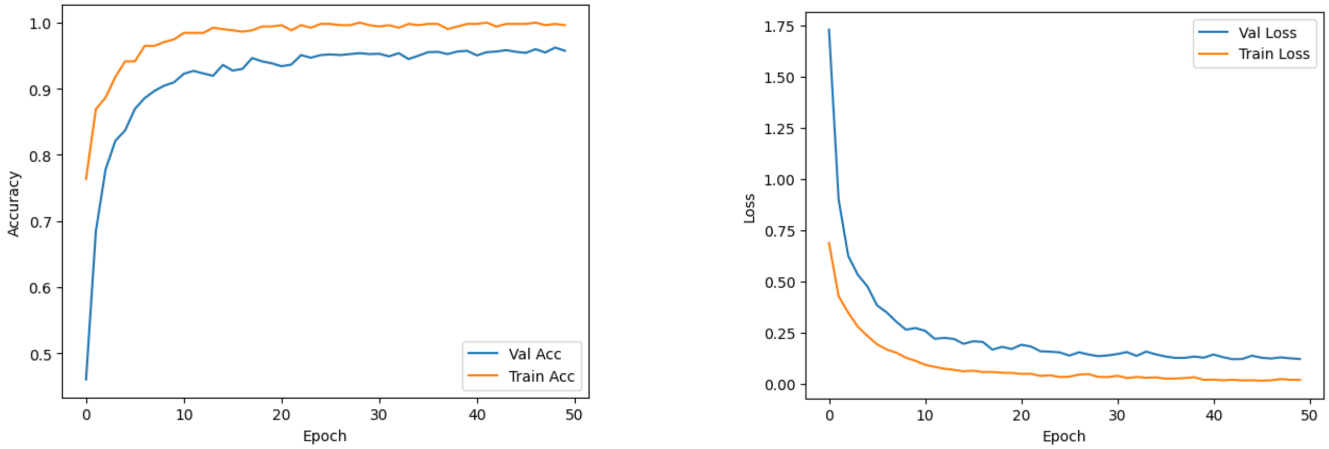
FIGURE 15. Performance results on ResNet101 model.



(a) Accuracy

(b) Loss

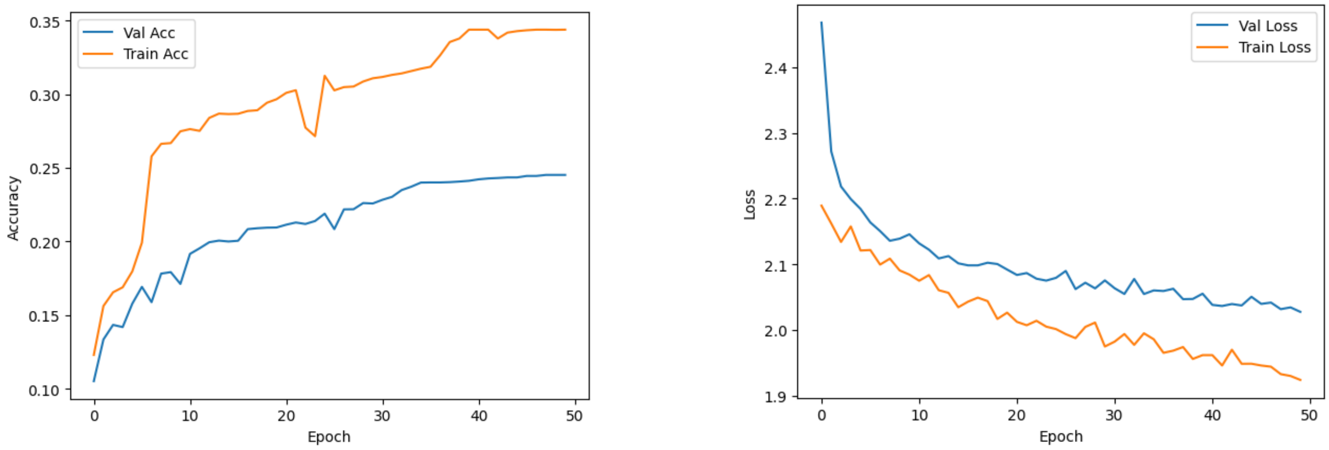
FIGURE 16. Performance results on ResNet101-SA model.



(a) Accuracy

(b) Loss

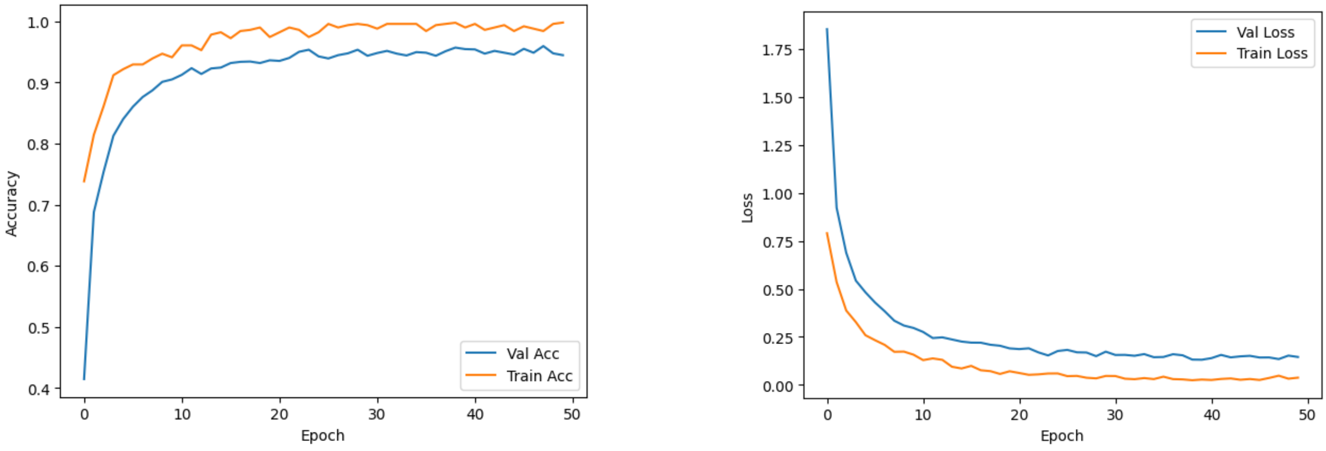
FIGURE 17. Performance results on ResNet101V2 model.



(a) Accuracy

(b) Loss

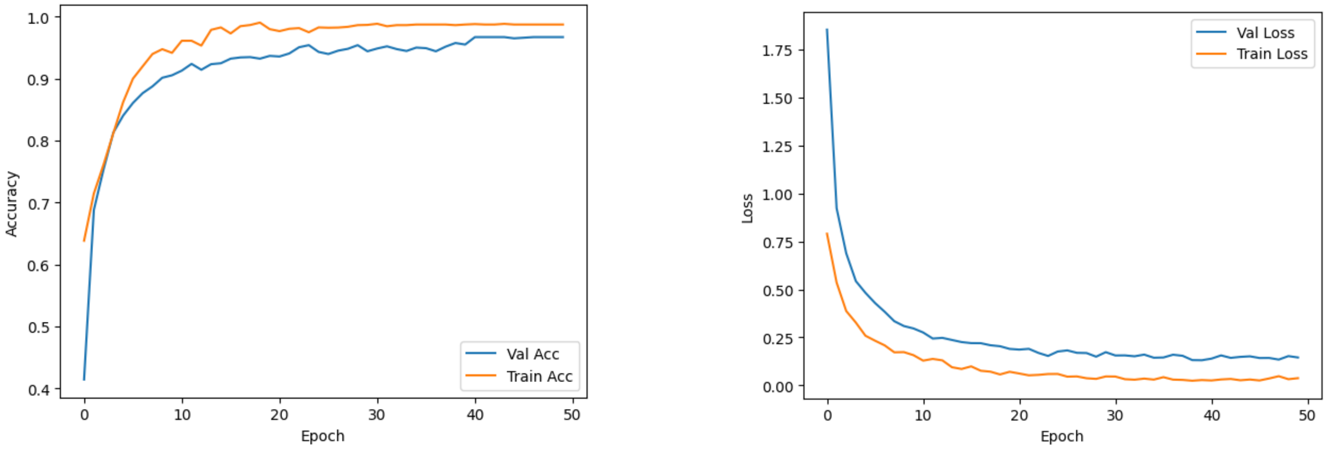
FIGURE 18. Performance results on ResNet152 model.



(a) Accuracy

(b) Loss

FIGURE 19. **Performance results on ResNet152V2 model.**



(a) Accuracy

(b) Loss

FIGURE 20. **Performance results on ResNet152-SA model.**

TABLE 6. **Performance matrices of different ResNet models.**

Model	Precision	Recall	Specificity	F1-score
ResNet50	0.2767	0.3638	0.3247	0.3143
ResNet50V2	0.9635	0.9637	0.9652	0.9634
ResNet101	0.2541	0.3691	0.3267	0.3009
ResNet101V2	0.9580	0.9574	0.9564	0.9576
ResNet152	0.2501	0.2643	0.2612	0.2570
ResNet152V2	0.9827	0.9824	0.9834	0.9824
ResNet50-SA	0.9933	0.9933	0.9990	0.9933
ResNet101-SA	0.8856	0.8861	0.9864	0.8858
ResNet152-SA	0.9616	0.9610	0.9958	0.9612

TABLE 7. **Classification results of ResNet50-SA in pest identification.**

Categories	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Aphids	100	100	100	100
Armyworm	99.61	100	96	98
Beetle	100	100	100	100
Bollworm	99.61	96	100	98
Grasshopper	100	100	100	100
Mites	100	100	100	100
Mosquito	100	100	100	100
Sawfly	99.61	100	97	99
Stem borer	99.61	95	100	97

respectively. In the confusion matrix, the diagonal value of the confusion matrix represents the correctly predicted samples. Thus, the larger the diagonal value, the better the result. Figure9 - Figure11 shows that the proposed approach performed well, and among the three models, ResNet50-SA gives better performances.

E. EFFECTIVENESS OF SELF-ATTENTION

The original ResNet models and the ResNet with self-attention model are compared in terms of performance. Table 5 displays the performance results of the suggested self-attention-based residual networks (ResNet50-SA, ResNet101-SA, and ResNet152-SA) in comparison

TABLE 8. Performance comparison of different DL architectures.

Model	Input Size	Depth	Parameter	Accuracy(%)
VGG16 [36]	224 × 224	16	138.4	96.3
VGG19 [36]	224 × 224	19	143.7	92.7
InceptionV3 [37]	224 × 224	189	23.9	98.24
DenseNet121 [38]	224 × 224	242	8.1	97.07
DenseNet201 [38]	224 × 224	402	20.2	99.41
NASNetMobile [39]	224 × 224	389	5.3	96.09
MobileNet [40]	224 × 224	55	4.3	95.66
MobileNetV2 [41]	224 × 224	105	3.5	96.30
FR-ResNet [42]	224 × 224	50	30.78	55.24
IoT based CNN [26]	416 × 416	NA	NA	96.71
CNN with parallel-attention [23]	224 × 224	NA	NA	98.17
GPA-Net [43]	448 × 448	NA	97.3	56.9
ResNet50-SA	224 × 224	110	28.8	99.80
ResNet101-SA	224 × 224	212	47.9	88.48
ResNet152-SA	224 × 224	314	93.64	96.68

to the original residual networks (ResNet50, ResNet101, and ResNet152). Table 5 shows that adding self-attention to the ResNet model significantly improves the model's performance. ResNet50 with self-attention provides 99.80% validation accuracy compared to ResNet50 without it providing only 23.87%. ResNet101 has a validation accuracy of just 25.33%; however, ResNet101 with self-attention has an accuracy of 88.48%. There is a 72% improvement in ResNet152's performance when compared to ResNet152 without self-attention. The suggested attention-based model extracts and retains more significant information, which improves the model's performance, according to performance results. By including self-attention in the model, the parameter is slightly (and negligibly) increased.

F. PERFORMANCE COMPARISON WITH VARIOUS MODELS

The proposed model is compared with several state-of-art deep learning architectures, including VGG16, VGG19, Inception-V3, DenseNet121, and DenseNet201, and also with some existing techniques proposed by the researchers in order to evaluate the robustness of the proposed model. Additionally, we examined the performances of some small deep-learning models like NASNetMobile, MobileNet, and MobileNetV2. Performance comparison of the several deep learning models with the proposed ResNet-SA model is shown in Table 8. The proposed model provides greater performance accuracy and can be observed in Table 8. In comparison to VGG16, VGG19, InceptionV3, and Densenet121, ResNet50 with self attention provides performance accuracy of 99.80%, which is 3.5%, 7.1%, 1.6%, and 2.7% higher, respectively. Although DenseNet201 has a performance accuracy of 99.41%, its depth is significantly more than that of other deep learning models.

V. CONCLUSION

This work demonstrates the potential of deep learning to revolutionize agricultural pest diagnosis. To identify crop pests in this work, we have proposed parallel attention-based ResNet models. A total of 3150 images from 9 distinct classes make up the dataset. For increasing the dataset size and decreasing model overfitting, various data augmentation methods are applied. We have evaluated the performances of ResNet50, ResNet50V2, ResNet101, ResNet101V2, ResNet152, ResNet152V2, ResNet50-SA, ResNet101-SA, ResNet152-SA. ResNet with self-attention improves performance accuracy, according to experimental findings, and ResNet50-SA provides the highest performance accuracy when compared to other deep learning models. In future work, the proposed model can be carried out in practical deployment in agriculture. Additionally, real-time pest identification with a wider variety of pest categories is an important area of exploration.

REFERENCES

[1] J. Liu and X. Wang, "Plant diseases and pests detection based on deep learning: A review," *Plant Methods*, vol. 17, no. 1, pp. 1–18, Dec. 2021.

[2] C. Wen, D. E. Guyer, and W. Li, "Local feature-based identification and classification for orchard insects," *Biosyst. Eng.*, vol. 104, no. 3, pp. 299–307, Nov. 2009.

[3] J. Wang, C. Lin, L. Ji, and A. Liang, "A new automatic identification system of insect images at the order level," *Knowl.-Based Syst.*, vol. 33, pp. 102–110, Sep. 2012.

[4] D. Xiao, J. Feng, T. Lin, C. Pang, and Y. Ye, "Classification and recognition scheme for vegetable pests based on the BOF-SVM model," *Int. J. Agricult. Biol. Eng.*, vol. 11, no. 3, pp. 190–196, 2018.

[5] S. N. Yaakob and L. Jain, "An insect classification analysis based on shape features using quality threshold ARTMAP and moment invariant," *Appl. Intell.*, vol. 37, pp. 12–30, 2012, doi: 10.1007/s10489-011-0310-3.

[6] R. Wang, J. Zhang, W. Dong, J. Yu, C. Xie, R. Li, T. Chen, and H. Chen, "A crop pests image classification algorithm based on deep convolutional neural network," *TELKOMNIKA, Telecommun. Comput. Electron. Control*, vol. 15, no. 3, pp. 1239–1246, 2017.

[7] Z. Liu, J. Gao, G. Yang, H. Zhang, and Y. He, "Localization and classification of paddy field pests using a saliency map and deep convolutional neural network," *Sci. Rep.*, vol. 6, no. 1, Feb. 2016, Art. no. 20410.

[8] D. Xia, P. Chen, B. Wang, J. Zhang, and C. Xie, "Insect detection and classification based on an improved convolutional neural network," *Sensors*, vol. 18, no. 12, p. 4169, Nov. 2018.

[9] Y. Li, H. Wang, L. M. Dang, A. Sadeghi-Niaraki, and H. Moon, "Crop pest recognition in natural scenes using convolutional neural networks," *Comput. Electron. Agricult.*, vol. 169, Feb. 2020, Art. no. 105174.

[10] L. Jiao, S. Dong, S. Zhang, C. Xie, and H. Wang, "AF-RCNN: An anchor-free convolutional neural network for multi-categories agricultural pest detection," *Comput. Electron. Agricult.*, vol. 174, Jul. 2020, Art. no. 105522.

[11] H. Peng, H. Xu, Z. Gao, Z. Zhou, X. Tian, Q. Deng, H. He, and C. Xian, "Crop pest image classification based on improved densely connected convolutional network," *Frontiers Plant Sci.*, vol. 14, Apr. 2023, Art. no. 1133060.

[12] J. Wang, Y. Li, H. Feng, L. Ren, X. Du, and J. Wu, "Common pests image recognition based on deep convolutional neural network," *Comput. Electron. Agricult.*, vol. 179, Dec. 2020, Art. no. 105834.

[13] K. Thenmozhi and U. Srinivasulu Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," *Comput. Electron. Agricult.*, vol. 164, Sep. 2019, Art. no. 104906.

[14] X. Cheng, Y. Zhang, Y. Chen, Y. Wu, and Y. Yue, "Pest identification via deep residual learning in complex background," *Comput. Electron. Agricult.*, vol. 141, pp. 351–356, Sep. 2017.

[15] M. Khanramaki, E. A. Asli-Ardeh, and E. Kozegar, "Citrus pests classification using an ensemble of deep learning models," *Comput. Electron. Agricult.*, vol. 186, Jul. 2021, Art. no. 106192.

[16] X. Guo, H. Zhou, J. Su, X. Hao, Z. Tang, L. Diao, and L. Li, "Chinese agricultural diseases and pests named entity recognition with multi-scale local context features and self-attention mechanism," *Comput. Electron. Agricult.*, vol. 179, Dec. 2020, Art. no. 105830.

[17] Y. Zhang, W. Zhong, and H. Pan, "Identification of stored grain pests by modified residual network," *Comput. Electron. Agricult.*, vol. 182, Mar. 2021, Art. no. 105983.

[18] Y. Zhang and Y. P. Liu, "Identification of navel orange diseases and pests based on the fusion of DenseNet and self-attention mechanism," *Comput. Intell. Neurosci.*, vol. 2021, pp. 1–12, Sep. 2021.

[19] Q. Dai, X. Cheng, Y. Qiao, and Y. Zhang, "Agricultural pest super-resolution and identification with attention enhanced residual and dense fusion generative and adversarial network," *IEEE Access*, vol. 8, pp. 81943–81959, 2020.

[20] C.-J. Chen, Y.-Y. Huang, Y.-S. Li, C.-Y. Chang, and Y.-M. Huang, "An AIoT based smart agricultural system for pests detection," *IEEE Access*, vol. 8, pp. 180750–180761, 2020.

[21] C. Li, T. Zhen, and Z. Li, "Image classification of pests with residual neural network based on transfer learning," *Appl. Sci.*, vol. 12, no. 9, p. 4356, Apr. 2022.

[22] J. Chen, W. Chen, Y. A. Nanehkaran, and M. D. Suzauddola, "MAM-IncNet: An end-to-end deep learning detector for camellia pest recognition," *Multimedia Tools Appl.*, pp. 1–16, Sep. 2023.

[23] S. Zhao, J. Liu, Z. Bai, C. Hu, and Y. Jin, "Crop pest recognition in real agricultural environment using convolutional neural networks by a parallel attention mechanism," *Frontiers Plant Sci.*, vol. 13, Feb. 2022, Art. no. 839572.

[24] E. C. Tetila, B. B. Machado, G. Astolfi, N. A. D. S. Belete, W. P. Amorim, A. R. Roel, and H. Pistori, "Detection and classification of soybean pests using deep learning with UAV images," *Comput. Electron. Agricult.*, vol. 179, Dec. 2020, Art. no. 105836.

[25] E. Ayan, H. Erbay, and F. Varçın, "Crop pest classification with a genetic algorithm-based weighted ensemble of deep convolutional neural networks," *Comput. Electron. Agricult.*, vol. 179, Dec. 2020, Art. no. 105809.

[26] B. Prasath and M. Akila, "IoT-based pest detection and classification using deep features with enhanced deep learning strategies," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 105985.

[27] Y. A. Nanehkaran, D. Zhang, J. Chen, Y. Tian, and N. Al-Nabhan, "Recognition of plant leaf diseases based on computer vision," *J. Ambient Intell. Humanized Comput.*, pp. 1–18, Sep. 2020.

[28] J. Chen, A. Zeb, Y. A. Nanehkaran, and D. Zhang, "Stacking ensemble model of deep learning for plant disease recognition," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 9, pp. 12359–12372, Sep. 2023.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[30] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," 2016, *arXiv:1606.01933*.

[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[32] A. Galassi, M. Lippi, and P. Torroni, "Attention in natural language processing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 10, pp. 4291–4308, Oct. 2021.

[33] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[34] W. Zeng and M. Li, "Crop leaf disease recognition based on self-attention convolutional neural network," *Comput. Electron. Agricult.*, vol. 172, May 2020, Art. no. 105341.

[35] *Pest Dataset*. Accessed: Sep. 15, 2023. [Online]. Available: https://www.kaggle.com/datasets/simranvolunesia/pest-dataset

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[37] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[38] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.

[39] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8697–8710.

[40] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[41] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.

[42] F. Ren, W. Liu, and G. Wu, "Feature reuse residual networks for insect pest recognition," *IEEE Access*, vol. 7, pp. 122758–122768, 2019.

[43] S. Lin, Y. Xiu, J. Kong, C. Yang, and C. Zhao, "An effective pyramid neural network based on graph-related attentions structure for fine-grained disease and pest identification in intelligent agriculture," *Agriculture*, vol. 13, no. 3, p. 567, Feb. 2023.

SK MAHMUDUL HASSAN received the B.Tech., M.Tech., and Ph.D. degrees in information technology from North Eastern Hill University (NEHU), Shillong, India, in 2015, 2017, and 2023, respectively. Currently, he is an Assistant Professor with the School of Computer Science and Engineering, Vellore Institute of Technology, Andhra Pradesh, India. His research interests include computer vision, artificial intelligence, machine learning, and image processing.

ARNAB KUMAR MAJI (Senior Member, IEEE) received the B.E. degree in information science and engineering from Visvesvaraya Technological University (VTU), in 2003, the M.Tech. degree in information technology from Bengal Engineering and Science University, Shibpur (Currently IIEST, Shibpur), in 2006, and the Ph.D. degree from Assam University, Silchar (Central University of India), in 2016. He has approximately 19 years of professional experience. He is currently an Associate Professor with the Department of Information Technology, North Eastern Hill University, Shillong (Central University of India). He has published around 40 numbers of articles in different reputed SCIE/SCOPUS Indexed International Journals, more than 12 numbers of articles as book chapters, 30 numbers of papers as conference proceedings, and authored three numbers of books with several international publishers, such as Elsevier, Springer, IEEE, MDPI, IGI Global, and McMilan International. Eight numbers of Ph.D. scholars are successfully guided by him. He has also guided successfully 18 numbers of M.Tech. thesis. His research interests include computer vision and natural language processing. He is also a reviewer of several reputed international journals and the guest editor of one Springer journal.