

# Accurate Pest Identification Using a Hybrid ResNet and Self-Attention Based Deep Learning Model

Roop Lal<sup>1</sup>, Sanjay Singla<sup>2</sup>  
Research Scholar<sup>1</sup>, Professor<sup>2</sup>  
Dept. of CSE

University Institute of Engineering  
Chandigarh University, Mohali-140413, Punjab, India

## Abstract

In farming, it's hard but necessary to quickly and correctly identify pests in order to cure them and protect crops. Insect identification and physical observation are two common ways to find pests, but they take a lot of effort and are easy to get wrong. The proposed techniques have changed the field of computer vision by making it possible to create automatic and effective pest recognition systems. This research evaluated the efficacy of various deep learning models and presented an improved methodology for enhanced extraction of feature. Within suggested methodology, two multitasking methods were recently integrated added to ResNet designs, which greatly improves performance. Based on the outcome, the resultant accuracy of ResNet50-SA, ResNet101-SA, and ResNet152-SA is 99.79%, 88.40%, and 96.67% respectively. According to the other popular DL models, ResNet50-SA is considerably superior. The outcomes show the ResNet with SA works more rapidly because it is more successful at recognizing elements and focusing on important features.

## Index Terms

Pest recognition system, learning network, network module, computer vision

## I. Introduction

The farming industry provides considerably to nations economy. The amount and quality of crops harvested are greatly affected by pest damage. In accordance with the food and agriculture organization (FAO), pest is the major issue for farming leading to global crop losses to 20-40%. Finding and getting rid of pests in farming systems is important for keeping crops healthy and getting the most out of them. Most of the time, people who are experts at identifying pests do so by looking at them, which is not very effective, takes a lot of time, and can be wrong. People in agriculture know that keeping an eye on things all the time is a big problem. Improvements in computer vision and machine learning have made more people interested in using these technologies to automate pest identification [1]. The standard way to accomplish machine learning is to take features from images and use classifiers to sort them. The scale invariant feature transform (SIFT), The Grey level Co-Occurrence Matrix (GLCM) and The Speed Up Robust Feature are some instances for artisanal feature techniques. KNN, SVM, Random Forest and Decision Tree are some of the most effective analyzers.

Wen et al. [2] used local feature extraction to recognize six distinct orchard insects. Using six distinct model with cross-fold validation, the nearest mean classifier (NMC) and SVM achieved the greatest accuracy of 89.5% and 88.4% respectively. Wang et al. [3] used Neural Networks and SVM to extract different orders of geometrical data for autonomous insect recognition. They achieved a 93% of accuracy rate with Neural Networks. By using an SVM model and SIFT-based feature descriptor, Xiao et al. [4] recognized four effective and different vegetable pests with a typical accuracy of 91.56%. Yaakob and Jain [5] recognized insects using shapes and the moment invariant traits. The primary challenge with a hand-crafted approach is finding out the appropriate set of features. The challenges are resolved by the research and creation in ML methods and especially deep learning methods that automatically

retrieves the crucial and efficient features. Deep Learning attention mechanism will help in enhanced pixel inspection and enhance feature learning for pest recognition in categorization. To improve feature extraction, we utilize the self-attention in DL methods in this research. The primary Summary of the paper can be summarized as follows:

1. This research put together nine different groups of pests, each with real-life photos taken in the field. To make the model more robust and add more images to the dataset, different data augmentation methods are used.
2. It suggests an improved ResNet model for identifying pests. The original ResNet design uses a parallel attention strategy to make feature extraction better.
3. Research use a number of complex deep-learning models to test how well the suggested model works. The results show that the proposed model was more accurate.
4. It uses a lot of complicated deep-learning techniques to test how well the suggested model works. The results show that the suggested model was more accurate.

## **II. Literature Review**

Recently, deep learning methods have been used more and more to identify plants, diagnose plant diseases, classify weeds, and find pests. Research on pest identification by deep learning is a nascent domain, with various methodologies recently introduced.

DL architectures like LeNet and AlexNet were introduced by Wang et al., [6] for the categorization of various crop pests. They have investigated how well different convolutional kernels and filter sizes work, and accuracy rate of 90% was achieved. For their study, they built their own dataset of pests which has 30,000 photographs and 82 different pest species. Liu et al. [7] put forward an DL architecture of 8-layer for pest identification which utilizes AlexNet as its base of operation. They have deployed a contrast reason-based approach to identify the pest. The author adjusted parameters including batch size, number of convolutions, To achieve 95.1% accuracy utilizing the convolutions strids, dropout rate and loss function.

Xia et al. [8] put forward an enhanced network architecture derived from the VGG-19 model to classify 24 different species of pests. To accurately identify the pest site on the feature map and eliminate any unwanted surroundings, the researchers used the area proportionate network. Their suggested model surpasses cutting-edge models, including Single Shot Multibox Detector (SSD) and Fast Region-based Convolutional Neural Network (RCNN), in mean Average Precision (mAP) and training time.

Yanfen Li et al. [9] suggested an improved Googlenet design for recognizing different pests in their native habitats. For the purpose to develop the neural network in this research, the researchers gathered the images from the internet as well as utilized the unique recorded images. The models' performance is measured using K-fold cross validation and different data augmentation approach. The suggested approach outperforms DL methods by 6.22% when it comes performance accuracy.

To recognize 24 different pest. Jiao et al. [10] Introduced the anchor free region CNN model, a feature fusion prototype. The recommended approach surpasses by YOLO sensor by 39.4% as well as Faster R-CNN by 15%

For the pest classification, Peng et al. [11] Presented MADN an efficient Dense-Net based architectures. The objective of the MADN is to boost feature extraction. The Selected Kernel Unit, the DenseNet121 base model has been modified through the ACON activation function and the corresponding batch normalization module. The accuracy of the HQIP102 dataset was 5.17% higher than DenseNet121, according to the author Wang et al. [12] utilized VggA, Vgg16, Inception V3, and ResNet50 to detect pest in crops and created our lightweight CNN model named CPAFNet. 92.63% of the 20 distinct kinds of insects could be accurately identified by this model.

Thenmozhi and Reddy [13] used many types built-in DL algorithms such as AlexNet, ResNet, GoogleNet and VGGNet to identify several pests.

The CNN model includes the six layers of convolution, one fully linked, five best pooling as well as one last layer. Three unique datasets-NBAIR, Xie1, and Xie2-where all used for the evaluations, yielding accuracy rates of 96.75%, 96.47% and 95.77% respectively.

Chang et al. [14] Identified 10 different crop insect species against complex background utilizing deep neural network. Their model outperforms SVM and backpropagation, with a 99.67% success rate. The author Khanramaki et al. [15] detected three frequent citrus pest utilizing a variety of DL models. A ten-fold cross-validation procedure, the model accuracy rate of 99.4% was greater than that of numerous other DL models.

Guo et al. [16] developed the self-attention methodology and multi-scale local setting elements for the detection of Chinese farm pest and illness. By using CNNs with various kernel sizes to gather multiscale local context characteristics, the fundamental BiLSTM-CRF model is enhanced. A modified dilated residual network has been proposed by Zhang et al. [17] to detect stored grain pests. A dilated convolution and a residual connection are used together to improve the convolutional vision.

**Table 1:** Comparison Table

| Paper | Technique used                                                           | Image dataset                               | Classes | Performance (%)     |
|-------|--------------------------------------------------------------------------|---------------------------------------------|---------|---------------------|
| [15]  | PCSA integrated from ResNet50 architecture                               | Internet Source Image                       | 10      | 98.17% accuracy     |
|       | ResNet50, InceptionV3                                                    | UAV Captured                                | 13      | 93.82% accuracy     |
|       | Xception, VGG16, VGG19                                                   | Dataset of captured images which is 1774    | 3       | 99.04% accuracy     |
|       | AlexNet, VGG16, ResNet50 Inception ResNet V2                             | The collected images are 550                | 10      | 98.67% accuracy     |
| [13]  | AlexNet, VGG GoogleNet ResNet                                            | Plant picture dataset provided by NBAIR     | 40      | 97.47% accuracy     |
| [12]  | CNN model utilizing the CPAFNet architecture                             | Dataset provided by the CPAF                | 20      | 92.26% accuracy     |
| [25]  | ResNet50, VGG16 as well as GAEnsemble                                    | Plant disease image collection (D0) dataset | 40      | 98.18% accuracy     |
| [26]  | Deep neural network with weight optimizing technique                     | Image collected by IP102 dataset            | 40      | 96 % accuracy       |
| [8]   | RPN integrated from the VGG19 model                                      | Captured image directly from field          | 24      | 89.22(mAP) reported |
| [11]  | Convolutional Neural network connected deeply with parameters            | High-quality images from HQIP102            | 102     | 75.28% accuracy     |
| [9]   | GoogleNet architecture was fine-tuned                                    | Image captured from the source of internet  | 10      | 96.67% accuracy     |
| [10]  | A neural network architecture – AF-RCNN                                  | Captured image                              | 24      | 56.40(aAP) reported |
| [16]  | Self-attention of Convolutional neural network for multi sale            | Plant disease dataset (AgCNER)              | 11      | 94.15% accuracy     |
| [27]  | Convolutional neural network architecture                                | Collected image from internet               | 13      | 91.33% accuracy     |
| [17]  | Modified dilated residual network                                        | Collected from lab and internet             | 6       | 96.72% accuracy     |
| [28]  | Es-MbNet                                                                 | Disease image                               | 9       | 99.37% accuracy     |
| [19]  | Integration of dense and residual connection developed by self-attention | Xiel dataset                                | 24      | -                   |
| [20]  | LSTM as well as YOLOV3                                                   | Collected field image                       | 1       | 90% accuracy        |
| [22]  | MAM-IncNet                                                               | Camellia oleifera leaf images               | 5       | 95.87% accuracy     |

Diseases as well as pests were recognized by Zhang along with Liu [18] using a hybrid methodology that combines self-attention with DenseNet architecture. This study utilizes position SA as well as channel SA to improve the

extraction of lesion characteristics. At a 96.90% accuracy rate, this algorithm could accurately identify six unique orange pest and diseases.

A low-resolution pest identification method that combines quadra-attention with residual and strong connections has been presented by Dai et al. [19] scientist used a (GAN) to make their dataset bigger, which greatly enhanced the performance of model. Using both residual and dense connections keeps more knowledge from previous layers and this made it easier to train deeper models by lowering the number of parameters.

Using artificial intelligence and the IoT, Chan et al. [20] developed a sophisticated pest identification system that classifies 24 distinct insects' groups. The author used LSTM to forecast when pest would show up and YOLOV# to identify them. Of Pest by collecting local meteorological data. The ResNeXt-50 model was 86.95% correct in identifying the pests according to Li et al [21]. The author investigated the effectiveness of the model utilizing various combinations of transfer learning, enhancement of data, and learning rate technique.

Based on their research integrating Cutmix, fine-tuning, and transfer learning improved performance.

Pest and disease affecting the Camellia oleifera plant have been identified by Chen et al. The present research introduces MAM-IncNet, a DL model pests and disease impacting the Camellia oleifera plant have been identified by the Chen et al [22]. This research presents MAM-IncNet, a DL algorithms that utilized an revised inception layer as an extractor of features in the SSD and VGG16 architectures. Having a 95.87% success rate, five distinct kinds of pests had been recognized. Moreover table 1 shows a analysis of the most recent content.

### III. Materials and Methodology

#### A. SYSTEM ARCHITECTUTRE

The author found multiple insects that have an effect on productivity of crops in this work, It split the dataset into sets of training and validation at random utilizing the internet. People are using a number of different data augmentation methods to make the dataset bigger. In order to acquire powerful characteristics that can assist us to recognize various pest, we have used SA methodology in RestNet patterns. The following sections give a short explanation of the suggested model.

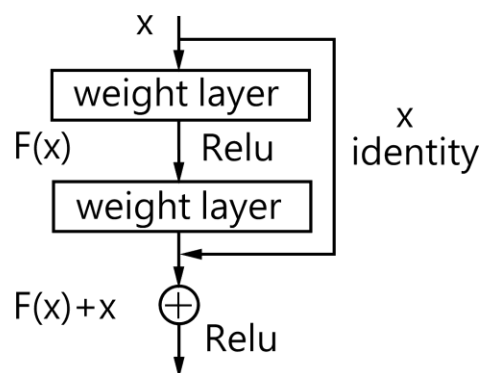


Figure 1 Residual Block

#### B. DEEP LEARNING ARCHITECTURE

The DL architectures working in this study will be addresses in this section. The efficient operation of many RestNet architectures like RestNet50, RestNet101, RestNet152, RestNet50V2, RestNet101V2 as well as RestNet152V2 are evaluated in this research. Additionally, we integrated the ResNet models utilizing attention and suggested an attention based residual model.

##### 1) RESIDUAL NETWORK

Let  $x$  be the residual network inputs and let the  $f(x)$  be the result of using  $x$  for processing two weighted layers.  $H(x)$ , the output of the residual network, is the sum of  $x$  and  $F(x)$ . The final result looks like this:

$$F(x) = W_{n+1} \text{ReLU}(W_n x) \quad (1)$$

$$H(x) = F(x) + x \quad (2)$$

In this case,  $w_n$  and  $w_{n+1}$  are the weights of the two layers. Structure for ResNet one maximum pooling layer, one average global pooling layer, and 48 layers of convolution constitute ResNet50 layers. According to ResNet34, the first layer includes the pooling layer and a single convolutional layer with a  $7 \times 7$  filter. Instead of utilizing the bottleneck block having two layers.

ResNet50 has a three-layer bottleneck block that makes it more accurate than the ResNet34 architecture. There are 16 residual blocks in the design of ResNet50. For this architecture,  $3.8 \times 10^9$  operations demand floating-point calculations (FLOPs). A better version of the ResNet50 design is called ResNet50V2 only the propagating formulation of the entire block connections has been modified to enhanced feature extraction. In ResNetV2, during the additional layer in the residual block, the function that activates ReLU is discarded to make room for a second non-linearity that acts as mapping for identification. Figure 3 shows the flow diagrams of residual connections in the ResNetV1 and ResNetV2 designs next to each other. The block diagram of the ResNet50 architecture is shown in Figure 2. ResNet101 and ResNet152 have the same layer setup as ResNet50, however, their residual blocks are 33 and 50 respectively. The models are deeper and have more variables because of these additional blocks.

### C. SELF-ATTENTION

Self-attention has been very popular in recent years and is now widely used in many areas, including speech recognition, machine translation [30], [31], language processing [32] and plant disease detection [33], [34]. The basic idea is to determine as a set of attention weight for a every element in the feature map. The value of each place in feature map for the eventual production is shown by these weights. The feature map can then grade using the attention weight before being transmitted to the network succeeding layer.

By allowing the network to concentrate on the most crucial portion of the input information, self-attention has assisted CNNs perform better on a variety of image classification and segmentation task. However, it can be very demanding on computers, especially when working with huge input photos, and to accomplish ideal outcomes, excessive parameter might have to be carefully modified.

### D. ResNet USING SELF-ATTENTION

Figure 4 shows the ResNet design with a self-attention layer. BaseNet is the name of the cutting-edge ResNet model's architecture. It blends global average pooling with ResNet models requiring layer for classification. The CNN model features vector was discovered using BaseNet. We utilized to dense layer to establish two parallel attention branches. The first attention concentration probabilities are used to apply to vectors of feature by the second attention branch. The feature vector is multiplied element by element with the attention probability. Finally, we combined the feature vector with the attention vector. The attention module tested in this study is shown in Figure 5. The elements of the planned ResNet50 which includes self-attention, are given in Table 2.

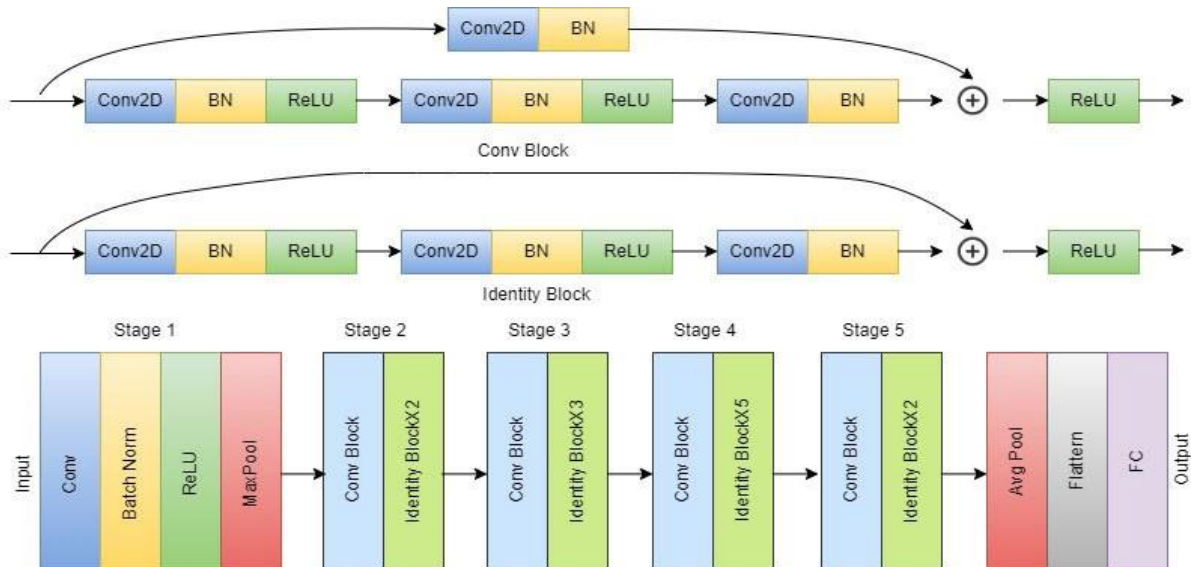


Figure 2: ResNet-50 Module

Table 2: self-attention ResNet details of layers

| Layer                       | Size of kernel                          | Output Range |
|-----------------------------|-----------------------------------------|--------------|
| Input Layer                 | NA                                      | 224x224      |
| Conv Layer 1                | 7x7, 64                                 | 112 X 112    |
| Pooling Layer               | 3 X 3                                   | 56 X 56      |
| Stacked of Conv Layer 1     | 1 x 1, 64<br>3 X 3, 64<br>1 X 1, 256    | X 3 56 X 56  |
| Stacked of Conv Layer 2     | 1 X 1, 128<br>3 X 3, 128<br>1 X 1, 512  | X 4 28 X 28  |
| Stacked of Conv Layer 3     | 1 X 1, 256<br>3 X 3, 256<br>1 X 1, 1024 | X 6 14 X 14  |
| Stacked of Conv Layer 4     | 1 X 1, 512<br>3 X 3, 512<br>1 X 1, 2048 | X 3 7 X 7    |
| Avg Pooling                 | -                                       | 1022         |
| Attention Layer 1           | -                                       | 1022         |
| second layer of attention   | -                                       | 1022         |
| feature fusion              | -                                       | 3070         |
| Fully connected dense layer | -                                       | 7            |

## IV. Results

### A. EXPERIMENTAL SETUP

The equipment and tools utilized during the experiment are mostly covered in this section. Python, particularly version 3.6, is the programming language utilized for the Deep Learning model.

Deep Learning methods such as TensorFlow, OpenCV, and Keras are utilized. We deployed the Google Colab Pro platform with GPU to create the DL technique.

### B. DATASET DESCRIPTION

This research uses open-access datasets on crop pests [35] and has extra pictures from the internet. There are 3,150 pictures of nine different crop pests in the pest collection: aphids, armyworms, bollworms, beetles, grasshoppers, mites, sawflies, mosquitoes, and stem borers. Table 3 gives a full picture of the dataset. Figures 6 and 7 show the sample image from the dataset. The pictures are all the same size, 224 by 224 pixels. For the dataset's future growth, different data augmentation techniques have been used. Data augmentation greatly increases diversity, which helps with the problems that come with data imbalance. To improve the size of the dataset, this study utilizes flipping, contrast improvement, rotation, brightness enhancement and saturation modification. Some of the better pictures are displayed in the figure 8. The dataset used, including the scientific names in explained in detailed in table 3.

### C. PERFORMANCE EVALUATION

In the present study we analyze the model's performance through accuracy as well as categorize cross-entropy loss. We also included the specificity, F1 score, accuracy as well as recall to assess the algorithms performance.

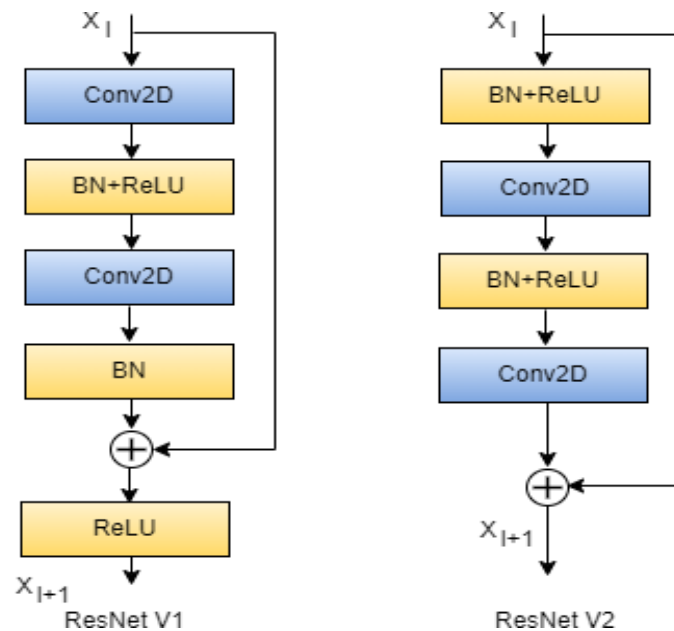


Figure 3. ResNetV1 and ResNetV2 Module

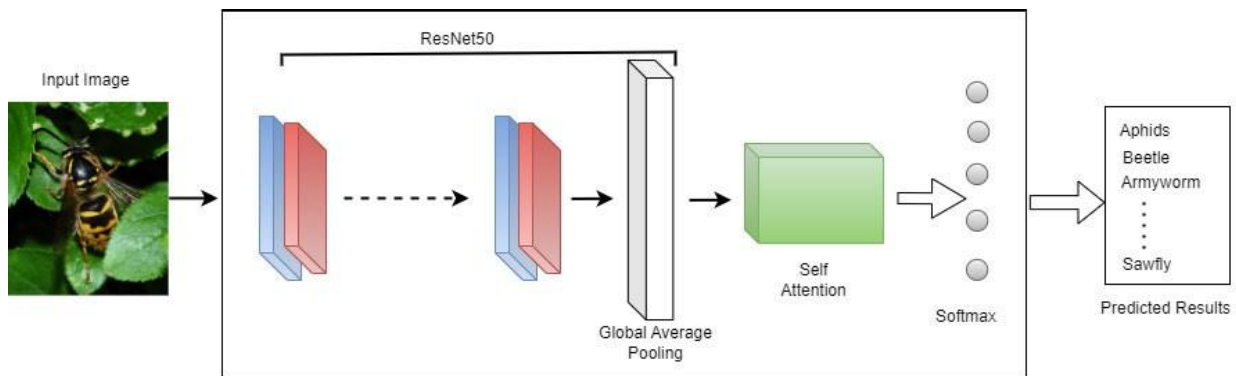


Figure 4. Overall architecture of ResNet with self-attention

The accuracy of the performance metrics is expressed as follows; The following equation is a statistical indicator of how the classifier classify photos.

$$Accuracy = \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \quad (3)$$

*Precision:* The number of positive predictions that are correctly classified as true positives is termed as precision

$$Precision = \frac{T_p}{T_p + F_p} \quad (4)$$

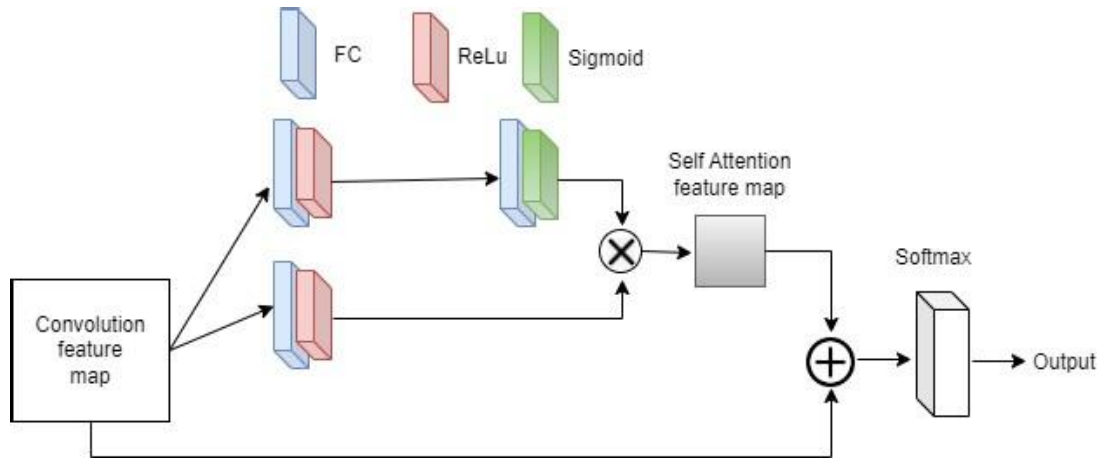


**Figure 5.** Self-attention network

**Table 3:** Pest details

| Pest Name             | Class | Original image | Augmented Image |
|-----------------------|-------|----------------|-----------------|
| Aphid                 | b1    | 300            | 450             |
| Spodoptera frugiperda | b2    | 300            | 450             |
| order Coleoptera      | b3    | 300            | 450             |
| scarce bordered straw | b4    | 300            | 450             |
| locust                | b5    | 300            | 450             |
| lice                  | b6    | 300            | 450             |
| mozzie                | b7    | 300            | 450             |
| Ants                  | b8    | 300            | 450             |
| Larvae                | b9    | 300            | 450             |



(a) Aphids



(b) Beetle



(c) Mites

**Figure 6.** pest's sample collected images

TABLE 4. Hyperparameter utilized

| Algorithm      | Batch size | Training rate | Weight decay | Iterations |
|----------------|------------|---------------|--------------|------------|
| adam optimizer | 34         | 0.001         | 0.0001       | 45-50      |

*Recall*: Also described as sensitivity, it is the ratio of perfectly planned beneficial events to all positive events.

$$Recall = \frac{T_p}{T_p + F_n} \quad (5)$$

*Specificity*: It indicates is the ratio of precisely predicted negative instances to all unsuccessful instances.

$$Specificity = \frac{T_n}{T_n + F_p} \quad (6)$$

*f1-score*: It is the balance of recall and precision

$$Score = 2 \times \frac{precision \times Recall}{precision + Recall} \quad (7)$$



(a) Grasshopper



(b) Armyworm



(c) Sawfly



(a)

(b)

(c)

(d)

(e)

Figure 7 & 8. Augmented images processed by (a) Flipping (b) Contrast adjustment (c) Rotation (d) Changing Brightness (e) Saturation

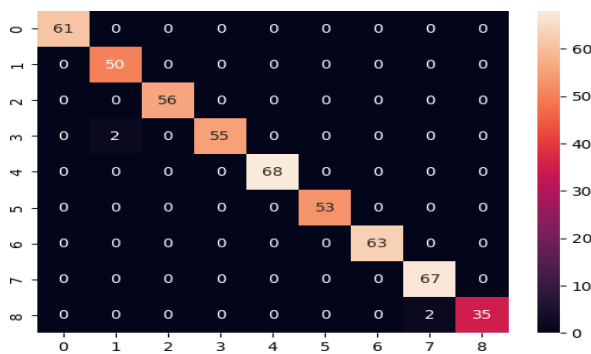
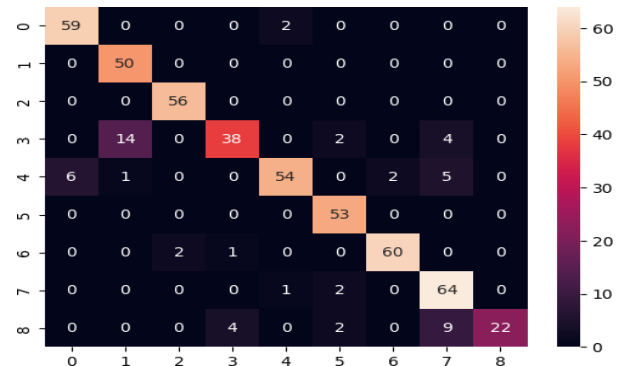
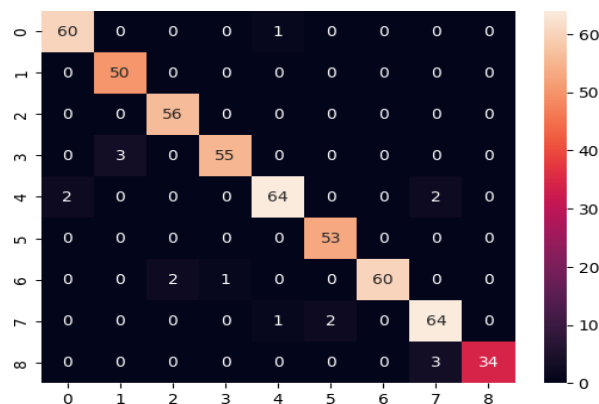
**Table 5.** Comparison Table

| Technique used | Training output | Error of learning rate | Validation performance | Validation error | iterations |
|----------------|-----------------|------------------------|------------------------|------------------|------------|
| M1             | 0.3357          | 1.9307                 | 0.2732                 | 2.1480           | 51         |
| M2             | 0.9861          | 0.0229                 | 0.9478                 | 0.1235           | 51         |
| M3             | 0.3311          | 1.9335                 | 0.2431                 | 2.0228           | 51         |
| M4             | 0.9970          | 0.0187                 | 0.9523                 | 0.1148           | 51         |
| M5             | 0.3428          | 1.9230                 | 0.2352                 | 2.0218           | 51         |
| M6             | 0.9951          | 0.0313                 | 0.9498                 | 0.1245           | 51         |
| M7             | 1.0000          | 0.0122                 | 0.9970                 | 0.0117           | 51         |
| M8             | 0.9717          | 0.0951                 | 0.8838                 | 0.5882           | 51         |
| M9             | 0.9771          | 0.0386                 | 0.9658                 | 0.152            | 51         |

M1- ResNet50, M2- ResNet50-v2, M3- ResNet101, M4- ResNet101-V2, M5- ResNet152, M6- ResNet152-V2, M7- ResNet50 with SA, M8- ResNet101 with SA, M9- ResNet152 with SA

#### D. RESULTS

To test how well the model works, the dataset is split into a training set of 80% and a validation set of 20%. To get the best results, the deep learning models have been fine-tuned by changing the weight decay, learning rate, and batch size. The hyper parameter required to train the model appears in table 4. Table 5 shows the performance metrics for several ResNet models, such as training accuracy, training loss, validation accuracy, and validation loss. Figures 12 to 20 show how accurate and wrong the ResNet models that were used.

**Figure 9.** Confusion matrix of ResNet50-SA**Figure 10.** Confusion matrix of ResNet101-SA**Figure 11.** Confusion matrix of ResNet152-SA.

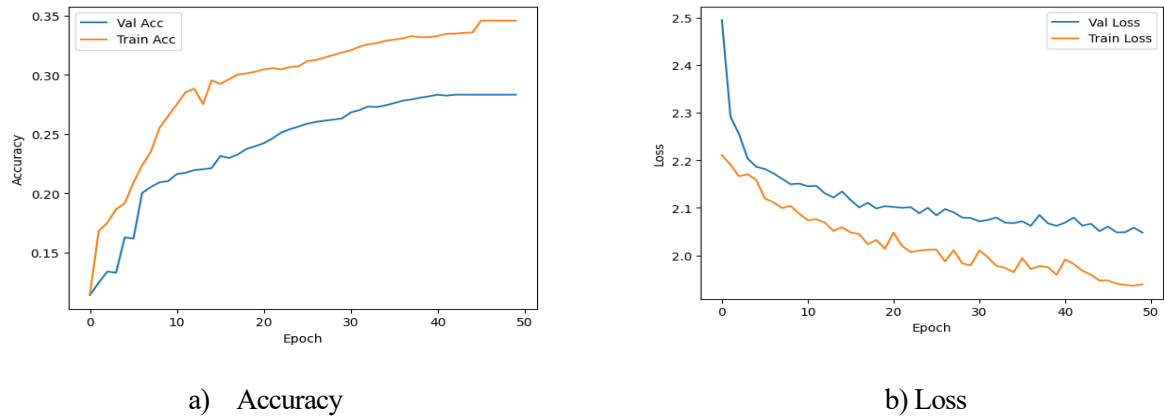


Figure 12. Performance results on ResNet50 model

More metrics for DL models have been evaluated, namely prescion, recall, F1-score, and accuracy. The Precision, recall, accuracy and F1 Score test results are shown in table 6. Table 6 represent the although though the training times for all the models are almost the same, their performances are very different.

We also check how well the proposed strategy works for sorting each type of pest. The experimental results of ResNet50-SA for each pest category are shown in Table 7. Table illustrate how much more correctly the resNet50 – SA technique, we propose to identify images of the pest.

To fully illustrate the effectiveness of the suggested models, we have developed the confusion matrix. The confusion matrix for ResNet50-SA, ResNet101-SA as well as ResNet152-SA are shown in Figures 9-11 respectively. The corrected predicted samples are expressed by the confusion matrix diagonal values, therefore, the better the results, the higher the diagonal value. Figure 9 illustrate the effectiveness of the suggested strategy, with ResNet50-SA outperforming the other two models.

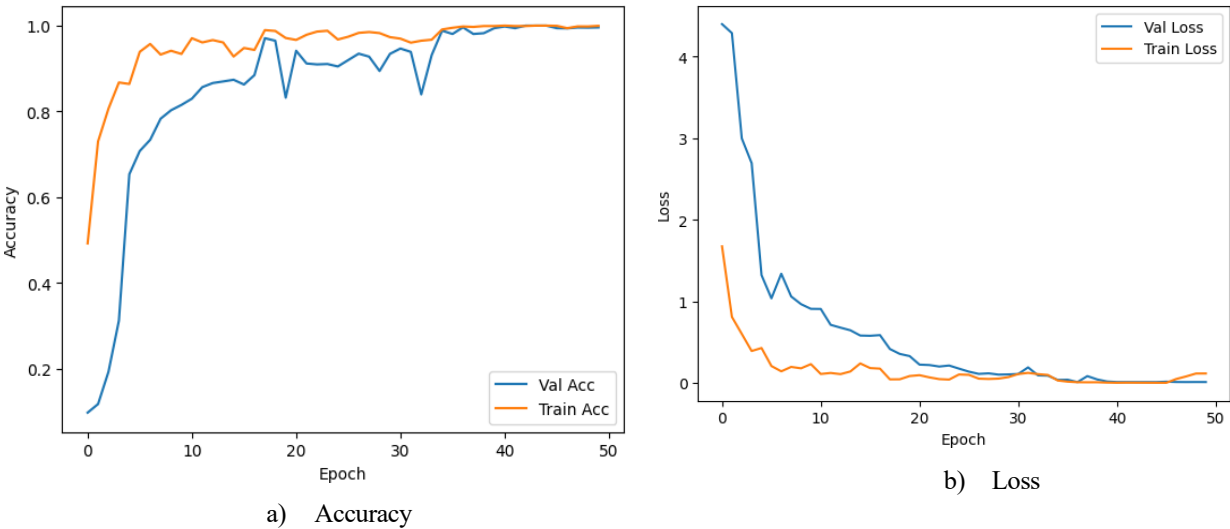


Figure 13. Performance results on ResNet50-SA model

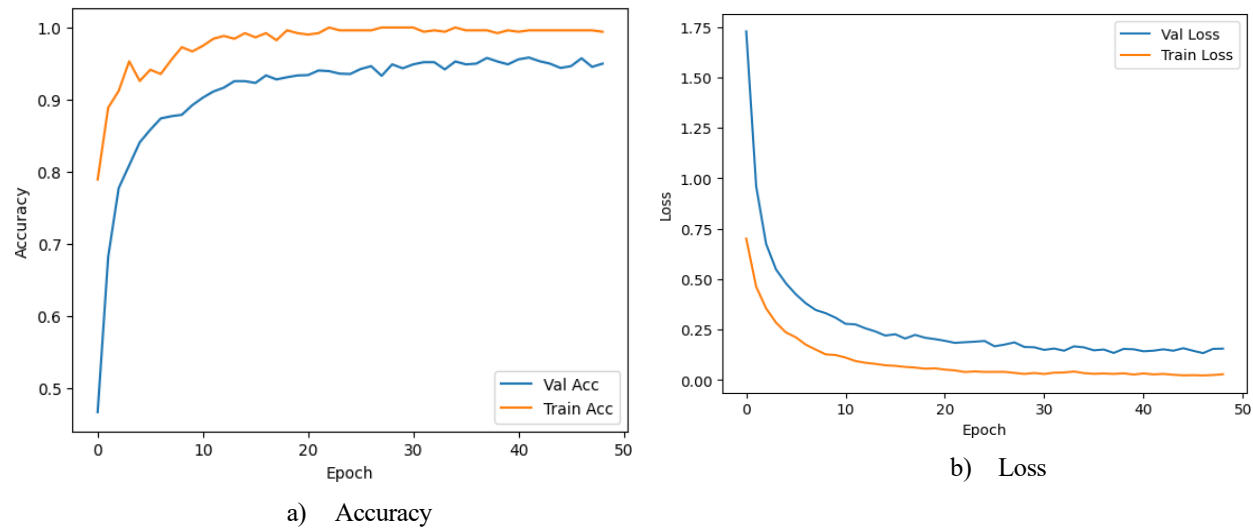


Figure 14. Performance results on ResNet50V2 model

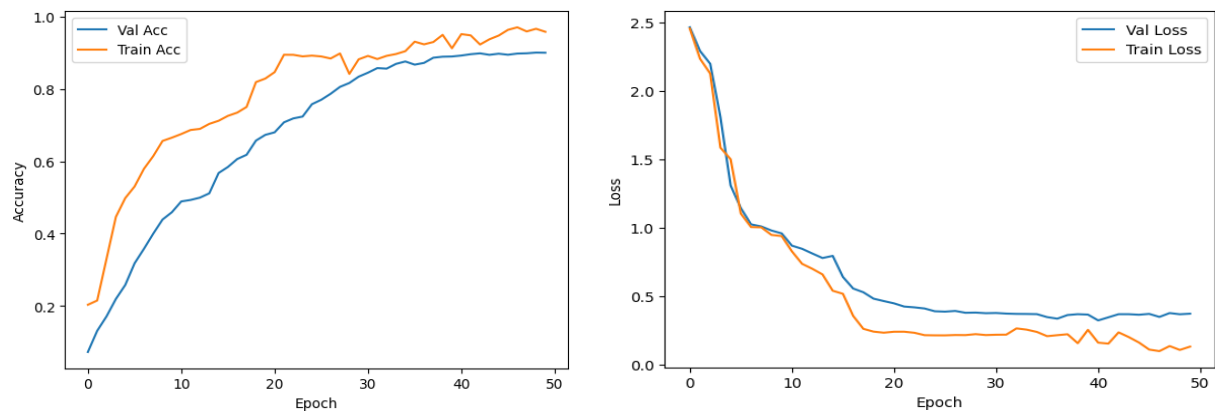


Figure 15. Performance results on ResNet101-SA model.

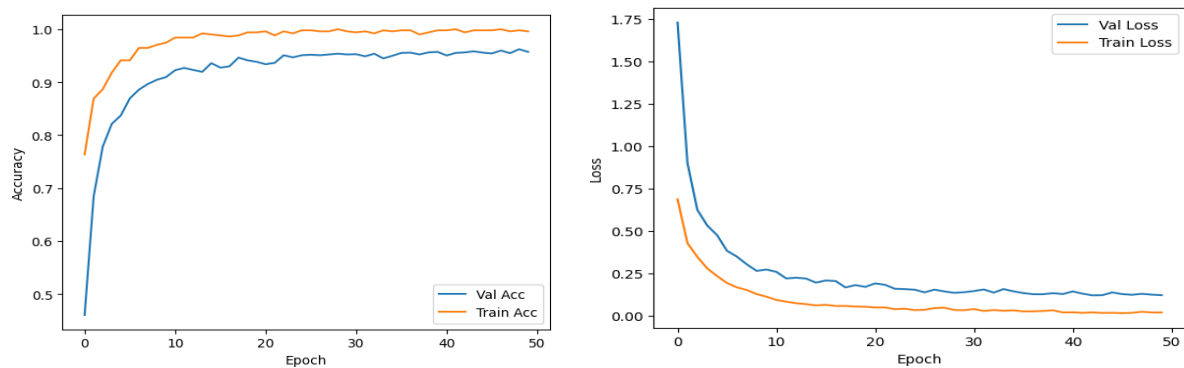


Figure 16. Performance results on ResNet101V2 model.

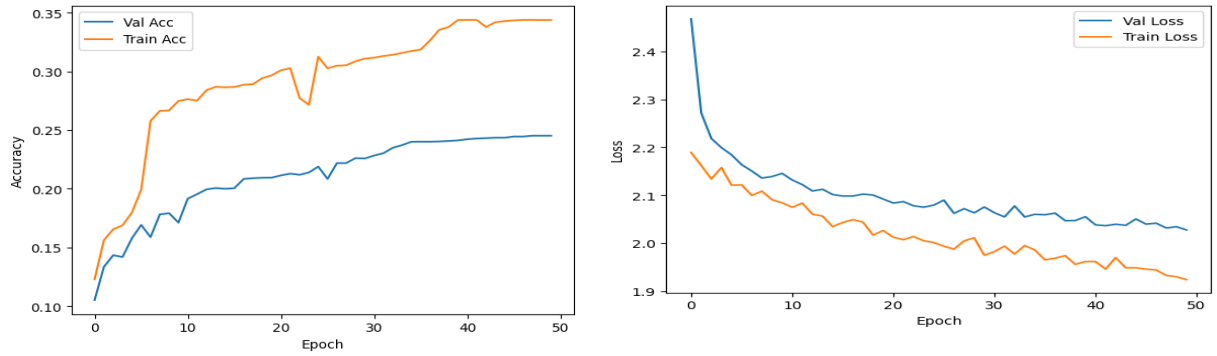


Figure 17. Performance results on ResNet152 model

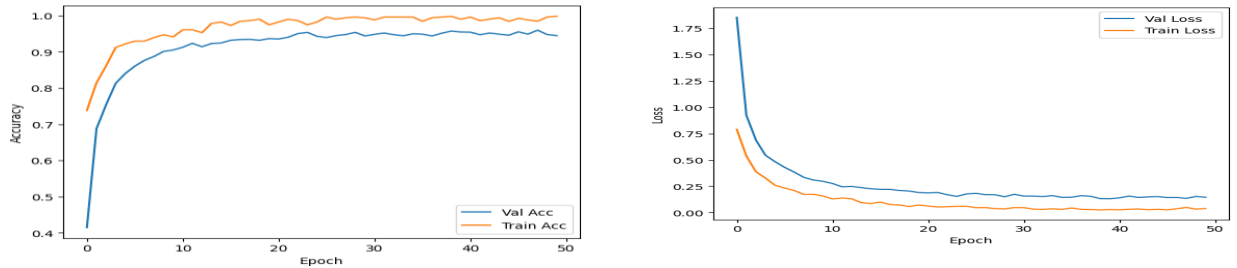


Figure 18. Performance results on ResNet152V2 mode

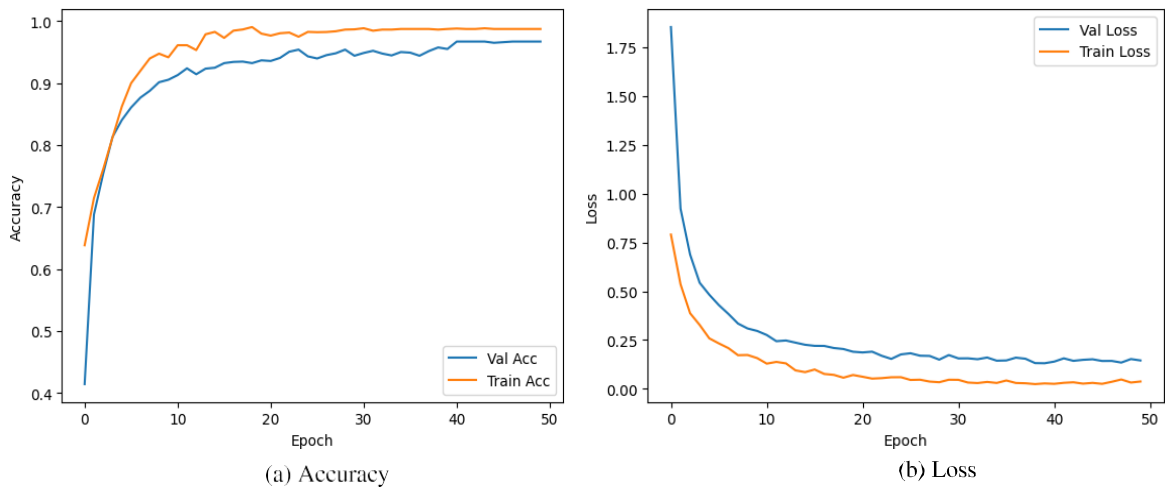


Figure 19. Performance results on ResNet152-SA model.

Table 6. Evaluated parameters of different ResNet architectures

| DL architectures | Precision value | Recall Rate | Specificity Score | F1-accuracy |
|------------------|-----------------|-------------|-------------------|-------------|
| M1               | 0.2667          | 0.3538      | 0.3147            | 0.3034      |
| M2               | 0.9535          | 0.9537      | 0.9552            | 0.9534      |
| M3               | 0.2441          | 0.3591      | 0.3167            | 0.3019      |
| M4               | 0.9480          | 0.9474      | 0.9464            | 0.9476      |
| M5               | 0.2401          | 0.2543      | 0.2512            | 0.2560      |
| M6               | 0.9727          | 0.9724      | 0.9764            | 0.9814      |
| M7               | 0.9833          | 0.9833      | 0.9890            | 0.9923      |
| M8               | 0.8756          | 0.8761      | 0.9764            | 0.8848      |
| M9               | 0.9516          | 0.9510      | 0.9858            | 0.9602      |

M1- ResNet50, M2- ResNet50-v2, M3- ResNet101, M4- ResNet101-V2, M5- ResNet152, M6- ResNet152-V2, M7- ResNet50 with SA, M8- ResNet101 with SA, M9- ResNet152 with SA

**Table 7.** labelling analysis of ResNet50 with SA in pest recognition

|                       | Performance% | Precision value % | Recall Rate % | F1- accuracy % |
|-----------------------|--------------|-------------------|---------------|----------------|
| Aphid                 | 99.9         | 99.9              | 99.9          | 99.9           |
| Spodoptera frugiperda | 98.61        | 99.9              | 95.5          | 97.5           |
| order Coleoptera      | 99.9         | 99.9              | 99.9          | 99.9           |
| scarce bordered straw | 98.61        | 95.5              | 99.9          | 97.5           |
| locust                | 99.9         | 99.9              | 99.9          | 99.9           |
| lice                  | 99.9         | 99.9              | 99.9          | 99.9           |
| mozzie                | 99.9         | 99.9              | 99.9          | 99.9           |
| Ants                  | 98.61        | 99.9              | 96.5          | 98.9           |
| Larvae                | 98.61        | 94.5              | 99.9          | 96.5           |

## E. EFFECTIVENESS OF SELF-ATTENTION

The Table 8 displays the performance analysis of the suggested self- attention-based networks such as ResNet50-SA, Resnet101-SA, ResNet152-SA when compared to the real RestNet algorithms and RestNet with SA models.

**Table 8.** Evaluation compared from different architectures based on deep learning

| Model                                  | Input     | Depth | Parameter | Accuracy (%) |
|----------------------------------------|-----------|-------|-----------|--------------|
| [36]                                   | 224x 224  | 16    | 138.4     | 96.3         |
| [36]                                   | 224 X 224 | 19    | 143.7     | 92.7         |
| InceptionV3[37]                        | 224 X 224 | 189   | 23.9      | 98.24        |
| DenseNet121[38]                        | 224 X 224 | 242   | 8.1       | 97.07        |
| DenseNet201[38]                        | 224 X 224 | 402   | 20.2      | 99.41        |
| NASNetMobile [39]                      | 224 X 224 | 389   | 5.3       | 96.09        |
| MobileNet [40]                         | 224 X 224 | 55    | 4.3       | 95.66        |
| MobileNetV2 [41]                       | 224 X 224 | 105   | 3.5       | 96.30        |
| FR-ResNet [42]                         | 224 X 224 | 50    | 30.78     | 55.24        |
| CNN [26]                               | 416X 416  | NA    | NA        | 96.71        |
| Neural network with parallel-attention | 224 X224  | NA    | NA        | 98.17        |
| GPA-Net [43]                           | 448 X 448 | NA    | 97.3      | 56.9         |
| ResNet50-SA                            | 224 X 224 | 110   | 28.8      | 99.80        |
| ResNet101-SA                           | 224 X 224 | 212   | 47.9      | 88.48        |
| ResNet152-SA                           | 224 X 224 | 314   | 93.64     | 96.68        |

ResNet50, ResNet101, ResNet152 constitute the initial residual networks. Table 5 illustrate how the ResNet model performance is significantly improved by in cooperating self-attention. RestNet50 associated with SA evaluated validated accuracy of 99.80%, while RestNet50 without it has an accuracy of only 23.87%. RestNet101 validation accuracy is modest 25.33% in contrast the accuracy of ResNet101 using self-attention is 88.48%. The performance of ResNet152 is 72% higher than that of ResNet 152 without self-attention. Performance results indicates that the suggested attention-based model improves the model performance by collecting and storing more important information. The parameter is moderately (and insignificantly) increased when self-attention is incorporated into the model.

## V. CONCLUSION

In this research, we offer parallel attention-based ResNet models for crop pest identification. The collection has 3,150 photos organized into nine main groups. Various data augmentation strategies are employed to increase the dataset and prevent the model from overfitting. We tested the effectiveness of ResNet50, ResNet50V2, ResNet101, ResNet101-V2, ResNet152, ResNet152-V2, ResNet50-Self-attention, ResNet101-SA, and ResNet152-SA. Experimental results suggest that ResNet with self-attention enhances performance accuracy, and ResNet50-SA outperforms other deep learning models.

In the future, the suggested approach might be applied in agricultural settings. In addition, real time pest protection involving a broader range of pest species in a important field of study.

## 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. Nanekharan, and M. D. Suzaiddola, "MAM- IncNet: An end-to-end deep learning detector for camellia pest recogni- tion," *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. Nanekharan, 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. Nanekharan, 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. Torrioni, "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, Fe