Advanced Pest Identification: An Efficient Deep Learning Approach Using VGG Networks

Dr.K.Lakshminadh
Professor, Department of CSE,
Narasaraopeta Engineering College,
Narasaraopeta-522601, India
drklmn7@gmail.com

Divvela Chandu Venkateswara Guptha
Student, Department of CSE,
Narasaraopeta Engineering College,
Narasaraopeta-522601, India
divvelachandu211@gmail.com

Jujjuri Sai
Student, Department of CSE,
Narasaraopeta Engineering College,
Narasaraopeta-522601, India
jujjurisai7989135700@gmail.com

Kandula Rajesh
Student, Department of CSE,
Narasaraopeta Engineering College,
Narasaraopeta-522601, India
rajeshkandula11@gmail.com

Dr. Sireesha Moturi
Asst. Prof., Department of CSE,
Narasaraopeta Engineering College,
Narasaraopeta-522601, India
sireeshamoturi@gmail.com

Yaragani Neelima
Asst. Prof., Department of CSE,
Narasaraopeta Engineering College,
Narasaraopeta-522601, India
neelimayaragani@gmail.com

Dodda Venkata Reddy
Student, Department of CSE,
Narasaraopeta Engineering College,
Narasaraopeta-522601, India
doddavenkatareddy@gmail.com

***Abstract*—Accurate pest identification is crucial for both effective pest management and crop protection. Pests must be found early in order to minimise damage and guarantee crop security. Conventional techniques typically entail visual examination and professional involvement, which might be time-consuming and susceptible to errors by humans. On the other hand, deep learning-powered high-performance systems can now more accurately identify pests thanks to developments in computer vision. In this work, we employed the Keras-based deep learning models VGG16 and VGG19 to construct a passive pest detection system. We greatly improved the efficacy of these models in identifying pest species by using strategies such data augmentation, model optimization, and modification of validated models. The VGG16 model produced an amazing accuracy rate of 99.8% and VGG19 model produced an accuracy of 96.8 % in our testing.**

***Index Terms*—Deep learning, pest identification, convolutional neural networks (CNN), feature extraction, transfer learning, image classification.**

I. Introduction

The world economy depends heavily on agriculture, but pests pose a important challenge to crop production and food security. Pests are responsible for the largest crop losses worldwide, hampering agricultural growth, especially in light of increasing global demand for food [1]. Expert techniques that use flow visualisation are typically laborious, error-prone, and time-consuming. This is because there is a wide form, size and number of pests which is a difficult task to search even by professionals in the field of study known as entomologists.

Advances in computer vision and innovative AI may provide solution in better pest identification in the last few years. Machine learning has been more specific to this classification known as deep learning and has recorded significant innovations in image categorization across various industries ranging from healthcare imaging systems to direct object detection Convolutional Neural networks (CNNs) have recorded exceptional image data analysis and handling.

But it's vital to remember that the application of deep learning to pest detection in agriculture is not as popular at the moment. Some of the challenges relates to efficient insect imaging, including the obtainment of high quality images, variability of image quality due to different conditions, and creation of models, which are able to detect numerous species of insects [2] [3].

This research present a new identification method for insects that will address the above challenges. Utilizing superior methods of data augmentation and converting them with complex deep learning models which are VGG16 & VGG19 enhancing the effectiveness and applicability of the models in the real sense. Our target is to contribute to the improvement of pest diagnostics and offer feasible recommendations that will be employed across the agricultural sector by farmers and the like.

II. Literature Review

Gong et al. [4] has put forward an FCN and DenseNet model with Efficient Channel Attention for rice pest identification at edge. Their model achieved 98. The recognition performance achieved 28% accuracy of 10 pest species, more accurate, and more robust than others.

Yang et al. [5] published a paper on identification of crop pests by using Edge Distance-Entropy with higher accuracy of 100% with utilization of only 60% of the data. This approach decreases the data required by 5% to 15% as well as enhances performance with an Anomaly Feature Detection Strategy.

Li et al. [6], the authors suggested the plant disease and pest recognition by employing the pre-processing and data enhancement approaches. Their method achieved 96.71% Algorithms accuracy achieved on the Plant_Village dataset which is better than traditional CNN models.

To enhance the early jute pest detection system, Talukder et al. [7] proposed the JutePestDetect model that was based on transfer learning. When validated with 17 classes of pest data set, the proposed method attained 99% accuracy which proved better than others.

Mask R-CNN was improved by Rong et al. [8] for identifying and to count pest on yellow plates in the field. It's observable that the presented Feature Pyramid Network (FPN) optimization raised the model's accuracy to above 99%. 4 % detection accuracy, which is two percentage points higher than that of the original. 7%.

The ANN based pest identification system has been proposed in Singh et al., [9] integrated with WSN for smart agriculture. Pest detection is done more accurately in the system leading to an improvement by 3 on the initial measures. It increases the crop management and reduces the environmental impacts

Hu et al. [10] suggested a model for identifying rice pests using multi-scale double-branch GAN.-ResNet. This model combines GANs with ResNet for improved feature learning and obtains a high of 99. 34% accuracy that is higher than traditional networks.

III. Materials and Methodology

A. Dataset Description

The dataset comprises 3,150 images of nine distinct crop pests: Aphids, army worms, boll worms, beetles, grasshoppers, mites, sawflies, mosquitoes and stem borers. These images were downloaded from some publicly available datasets [11] . To keep the image size consistent, every picture was resized to 224 x 224 pixels even if the original image had been of different size. Further, the dataset is divided into labelled categories for every type of pest for easy segregation while training and testing. All images were stored in the JPG format in order to maintain uniformity. Figure 1 shows some images from dataset. Table I describes about images in dataset.

B. Data Augmentation

This is done with the help of data augmentation methods that help to improve the dataset and model performance. These technique included the flipping of the images along the horizontal axis, shearing, rotation of image, brightness modification of the image, zooming and the addition of Gaussian noise to the original image [12]. By applying the above transformations, variance of the data is added hence reduces the problem of imbalance in data and over fitting.

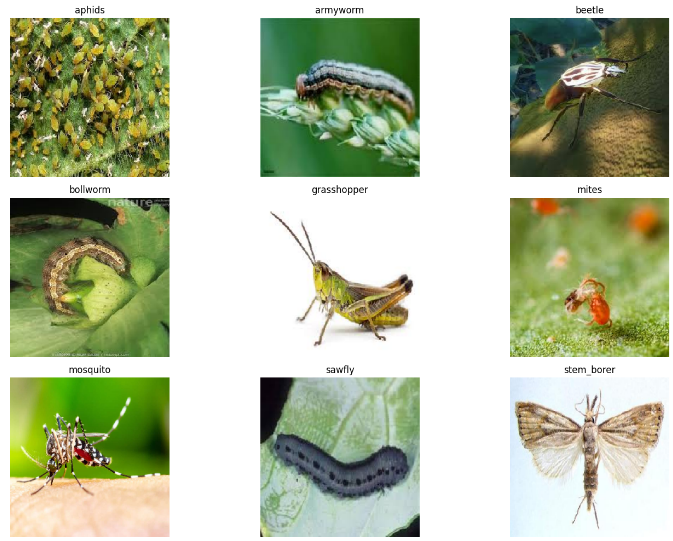

Fig. 1: Sample images of pest

TABLE I: Dataset Description

Scientific Name	Pest Name	Original Image Count
AphidsAphidoidea	Aphids	350
Spodoptera frugiperda	Armyworm	350
Coleoptera	Beetle	350
Helicoverpa armigera	Bollworm	350
Caelifera	Grasshopper	350
Acariformes	Mites	350
Culicidae	Mosquito	350
Symphyta	Sawfly	350
Scirpophaga incertulas	Stem borer	350

Thus, the alterations described above are presented in the given visualizations in Figure 2, that demonstrate the impact of the augmentations on the images indicating how every change transforms the data.

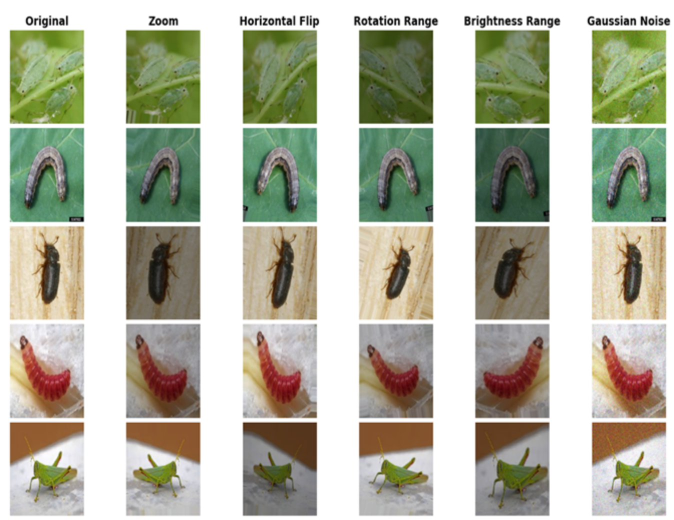

Fig. 2: Sample augmented images

C. System Overview

Our main focus in this effort is to distinguish between various pests that have an impact on agricultural productivity. First, we gathered images from several sources. Consequently, the dataset is divided into sets training and testing. using random sampling. To make data even more variable and numerous, we used several techniques of data augmentation so that models would work properly. During training process, the Adam optimizer was implemented with learning rate being equal to 0. 001.

D. Deep Learning Architectures

We explore the architectures that are employed in our research, specifically focusing on the VGG16 and VGG19 networks. These architectures are famous for the depth of the present model, and feature extraction, which are useful for complicated image classification tasks such as identification of pest.

1) VGG Network
The VGG architecture proposed by Simonyan and Zisserman [13] can be considered one the most popular deep learning models for image recognition tasks. A kind of CNN [14] with its simple and heavy network structure, it only uses very small convolutional kernel of 3×3 and multiple convolutional layers. The two VGG networks, VGG16 and VGG19 [15], is named that because there are 16 and 19 weight layers in the networks.
2) VGG16 Architecture
The VGG16 architecture contains the convolutional layers 13, fully connected layers 3, max-pool layers 5 also with parameter of around 138Ms as in Table II. The layers in Table II are denoted as B1 through B5, rather than Block1 through Block5.

TABLE II: Layered details of VGG16 model

Layer Name	Output Shape	Param #
input_layer (InputLayer)	(224 × 224, 3)	0
b1_convl1 (Conv2D)	(224 × 224, 64)	1,792
b1_convl2 (Conv2D)		36,928
b1_max_pool (MaxPooling2D)	(112 × 112, 64)	0
b2_convl1 (Conv2D)	(112 × 112, 128)	73,856
b2_convl2 (Conv2D)		147,584
b2_max_pool (MaxPooling2D)	(56 × 56, 128)	0
b3_convl1 (Conv2D)	(56 × 56, 256)	295,168
b3_convl2 (Conv2D)		590,080
b3_convl3 (Conv2D)		590,080
b3_max_pool (MaxPooling2D)	(28 × 28, 256)	0
b4_convl1 (Conv2D)	(28 × 28, 512)	1,180,160
b4_convl2 (Conv2D)		2,359,808
b4_convl3 (Conv2D)		2,359,808
b4_max_pool (MaxPooling2D)	(14 × 14, 512)	0
b5_convl1 (Conv2D)	(14 × 14, 512)	2,359,808
b5_convl2 (Conv2D)		2,359,808
b5_convl3 (Conv2D)		2,359,808
b5_max_pool (MaxPooling2D)	(7 × 7, 512)	0
flatten_layer	(25088)	0
dense_layer	(256)	6,422,784
dropout_layer	(256)	0
dense_1_layer	(9)	2,313

The last three convolutional layers with the first two of them containing 4096 channels and a softmax layer for classification are formed. Figure 3 illustrates the detailed architecture of developed model which is mainly constructed for efficient image detection.

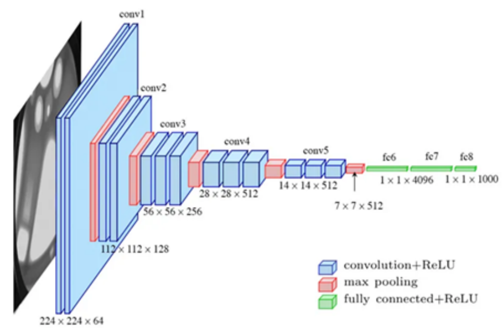

Fig. 3: VGG16 Architecture

3) VGG19 Architecture VGG19 is very much like VGG16 but with a difference in the arrangement of its layers where the number of convolutional layers stands at 16 Table III. It also has 3 fully connected layers and 5 max pool layers, with 4096 channels. The last classification performed by layer softmax. Figure 4 is the architecture about VGG19. The layers in Table III are denoted as B1 through B5, rather than Block1 through Block5.

TABLE III: Layer details of VGG19 model

Layer Name	Output Shape	Param #
input_layer (InputLayer)	(224 × 224, 3)	0
b1_convl1 (Conv2D)	(224 × 224, 64)	1,792
b1_convl2 (Conv2D)		36,928
b1_max_pool (MaxPooling2D)	(112 × 112, 64)	0
b2_convl1 (Conv2D)	(112 × 112, 128)	73,856
b2_convl2 (Conv2D)		147,584
b2_max_pool (MaxPooling2D)	(56 × 56, 128)	0
b3_convl1 (Conv2D)	(56 × 56, 256)	295,168
b3_convl2 (Conv2D)		590,080
b3_convl3 (Conv2D)		590,080
b3_convl4 (Conv2D)		590,080
b3_max_pool (MaxPooling2D)	(28 × 28, 256)	0
b4_convl1 (Conv2D)	(28 × 28, 512)	1,180,160
b4_convl2 (Conv2D)		2,359,808
b4_convl3 (Conv2D)		2,359,808
b4_convl4 (Conv2D)		2,359,808
b4_max_pool (MaxPooling2D)	(14 × 14, 512)	0
b5_convl1 (Conv2D)	(14 × 14, 512)	2,359,808
b5_convl2 (Conv2D)		2,359,808
b5_convl3 (Conv2D)		2,359,808
b5_convl4 (Conv2D)		2,359,808
b5_max_pool (MaxPooling2D)	(7 × 7, 512)	0
flatten_layer	(25088)	0
dense_layer	(256)	6,422,784
dropout_layer	(256)	0
dense_1_layer	(9)	2,313

IV. Results

A. Experimental Setup

The experiment was performed on Google Colaboratory with the help of Colab Pro subscription.

9798-3325-2169-2/25/$3200 ©2025 IEEE

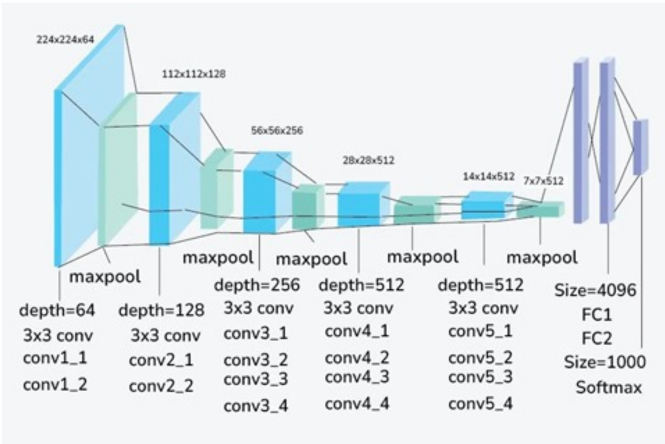

Fig. 4: VGG19 Architecture

- The cloud-based service offered Jupyter Notebook access to the computation resource such as an NVIDIA T4 GPU with capacity of 15GB, a CPU of 12 GB RAM and 100GB disk space.
- The pest dataset was separately uploaded to the Google Drive and then imported into the Colab, which made it easy to use.
- The grid models were built and optimized with Python coding language, where Tensorflow and Keras were used to operate deep learning problems.

B. Formula for Calculating Performance Metrics

We go over the formulas in this section that are used for assess the categorization models' performance [16] [17]. The important measures are Precision, F1-score, recall and Accuracy.

- *Accuracy* equals to the ratio of number of appropriate classifications made to the total number of cases. It is used for evaluating the model's accuracy.

$$\textit{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where TP(True Positives), TN (True Positives), FP(False Positives), FN(False Negatives).
- *Precision* sometimes referred as positive predictive value, measures percentage of actual positive predictions among all the model's positive predictions. It's mainly helpful in situations when false positives are high.

$$\textit{Precision} = \frac{TP}{TP + FP} \tag{2}$$

- *Recall* The percentage of true positive cases that the model accurately recognised is calculated by recall, also known as sensitivity or true positive rate. It is main when the cost of false negatives is considerable

$$\textit{Recall} = \frac{TP}{TP + FN} \tag{3}$$

- *F1-score* is the average of precision, recall. It is helpful when you need for strike a balance between recall and precision because it offers a single statistic that does both.

$$\textit{F1-score} = 2 \times \frac{R \times P}{R + P} \tag{4}$$

Where R stands for Recall and P stands for Precision
- *Epoch* a full iteration over the entire training dataset. The model processes data in batches, updates its weights, and improves performance. Choosing the right number is crucial to avoid underfitting or overfitting. In this work, 50 epochs were chosen for the model to balance learning and prevent overfitting

$$\textit{Number of batches per epoch} = \frac{N}{B} \tag{5}$$

C. Performance Evaluation

TABLE IV: Classification Report of VGG16 model

Class	Precision	F1-Score	Recall	Support
armyworm	1.00	1.00	1.00	50
aphids	1.00	1.00	1.00	50
bollworm	1.00	1.00	1.00	50
beetle	1.00	1.00	1.00	50
mosquito	1.00	1.00	1.00	50
grasshopper	1.00	1.00	1.00	50
mites	1.00	1.00	1.00	50
sawfly	1.00	0.99	0.98	50
stem_borer	0.98	0.99	1.00	50

TABLE V: Comparison of Classification of Different Models

Models	Precision	F1-Score	Recall
VGG16	1.00	1.00	1.00
VGG19	0.97	0.97	0.97
ResNet50 (Self-Attention)	1.00	1.00	1.00
ResNet101 (Self-Attention)	0.98	0.98	0.98
ResNet152 (Self-Attention)	0.97	0.97	0.97
MobileNet	0.96	0.96	0.96
MobileNetV2	0.97	0.97	0.97

TABLE VI: Accuracy, Execution Time of Different Models

Model	Accuracy	Execution Time (GPU)
VGG16 (Proposed approach) ✓	99.78%	4013 seconds
VGG19	96.89%	4451 seconds
ResNet50 (Self-Attention)	99.78%	5650 seconds
ResNet101 (Self-Attention)	97.75%	4830 seconds
ResNet152 (Self-Attention)	96.67%	6806 seconds
MobileNet	96.70%	5125 seconds
MobileNetV2	97.33%	5445 seconds

D. Results

The VGG16 and VGG19 models were both used in this experiment to categorise pest species. While the VGG19 model reached an accuracy of 96.89% with a little longer execution time of 4400 seconds, the VGG16 model achieved an outstanding accuracy of 99.78% with an execution time of 4013 seconds. These models performed quite well on our dataset, largely because to their deep convolutional architectures, which are well-known for their effective feature

9798-3325-2169-2/25/$3200 ©2025 IEEE

extraction techniques. Notably, our VGG16 model performed better than models used in earlier research, like ResNet50 with self-attention, as reported in [17]. Table IV displays the Classification report of VGG16 in pest identification and Table V displays the Comparison of classification reports of different models.

It is for this reason that accuracy and efficiency have enhanced because of the application of data augmentation approaches and hyperparameters fine-tuning on both models' generalization capacities. In addition, we compared our studies to other models such as ResNet101 and MobileNetV2 whereby MobileNetV2 yielded a high accuracy as compared to other models even though its accuracy was marginally lower than the VGG16 model used in this study. As seen in the Table VI, VGG16 was accurate but computationally efficient thus can be a useful tool for precision agriculture jobs such as pest identification. These enhancements establish the efficacy of our method over other models including ResNet50 self-attention [17], by presenting a more precise framework for pest identification.

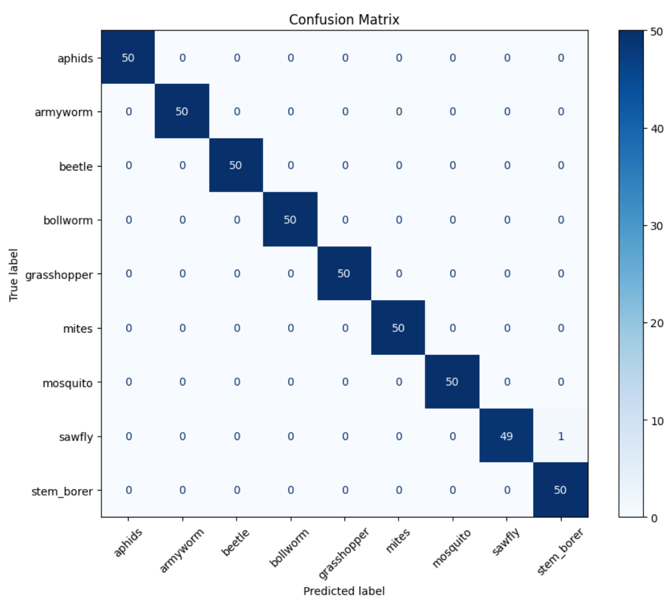

Fig. 5: Confusion Matrix of VGG16 model

This confusion matrix of VGG16 model shown in Figure 5 gives clear cut idea about the classification results and no.of correct and incorrect images classified for each pest species .Performance results, including training, validation accuracy, loss, are illustrated by Figure 6.

This graph showcases the model's performance over training epochs, providing insights into the trends in validation accuracy and loss, which reflect the learning progress and overall effectiveness of VGG16.

The VGG19 model's confusion matrix, displayed in Figure 7, offers a thorough analysis of the classification performance by outlining the proportion of accurate and inaccurate predictions for each pest species and providing details on particular classes where misclassification happened. Figure 8

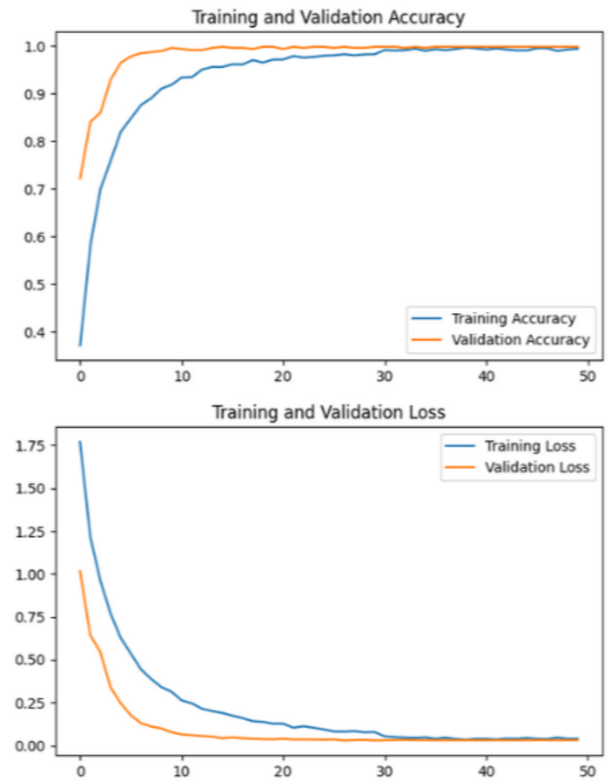

Fig. 6: Performance results of VGG16 model

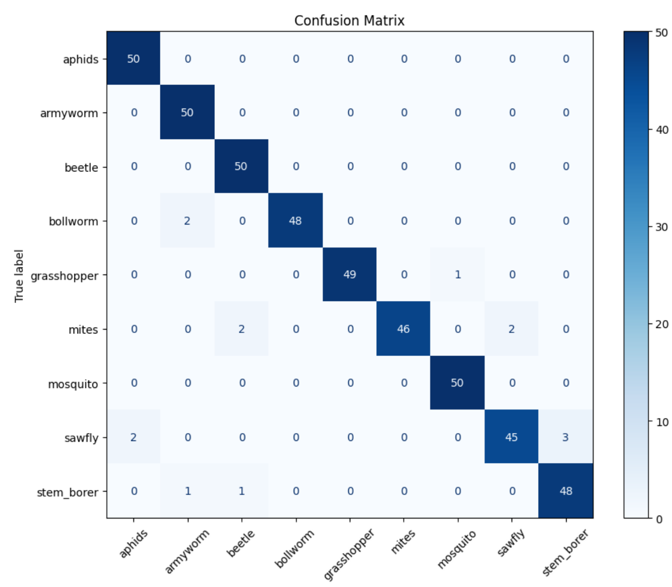

Fig. 7: Confusion Matrix for VGG19 model

shows the training and validation accuracy as well as loss metrics for the VGG19 model. These metrics show how the model learnt, converged, and performed overall throughout the course of the training epochs. During the training phase, these visualisations are essential for comprehending the model's

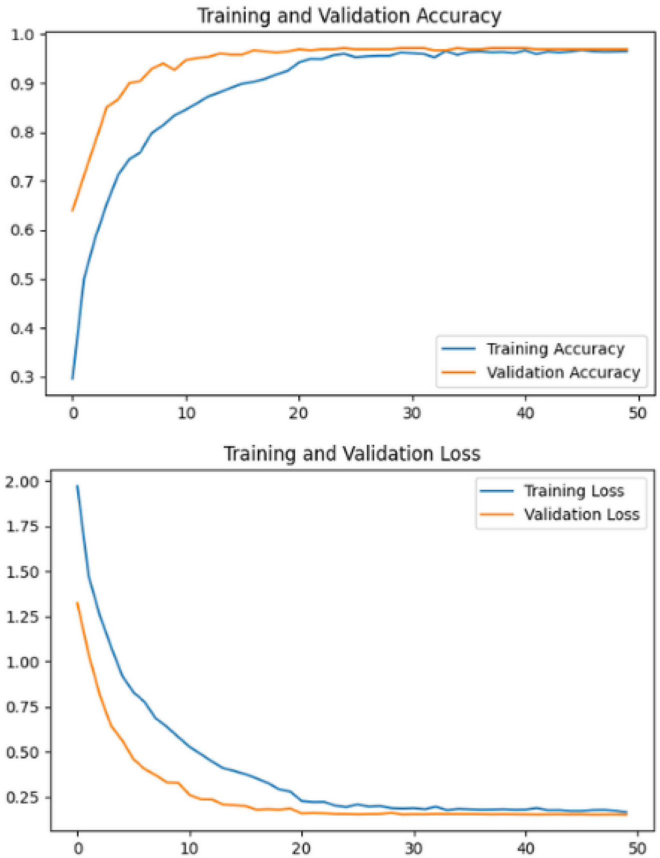

Fig. 8: Performance results of VGG16 model

advantages and disadvantages.

V. Conclusion

This research brings out the potential of deep learning to enhance Pest Identification and recommendation on Agriculture. Using VGG16 and VGG19 approaches, we designed a sensitive detection program that provided accuracy assessments of 99.78% and 96.67%. To provide more data into the dataset which consists of 3150 images of nine different pest species, we have incorporated data augmentation strategies which also assisted in the minimization of over-fitting. Such results recommend that the proposed system can be useful for pest identification in real-life agriculture contexts. It is further possible to build on this work and consider real-time pest identification of a more diverse range of species, thus making it an even more useful tool for sustainable agriculture practices.

References

[1] Spadaro, D., Agustí, N., Ortega, S.F., Hurtado Ruiz, M.A. (2020). Diagnostics and Identification of Diseases, Insects and Mites. In: Gullino, M., Albajes, R., Nicot, P. (eds) Integrated Pest and Disease Management in Greenhouse Crops. Plant Pathology in the 21st Century, vol 9. Springer, Cham. https://doi.org/10.1007/978-3-030-22304-5_8

[2] W. Xu, L. Sun, C. Zhen, B. Liu, Z. Yang, and W. Yang, "Deep learning-based image recognition of agricultural pests," *Applied Sciences*, vol. 12, no. 24, p. 12896, 2022. [Online]. Available: https://doi.org/10.3390/app122412896

[3] Cheng, X., Zhang, Y., Chen, Y., Wu, Y. and Yue, Y., 2017. Pest identification via deep residual learning in complex background. Computers and Electronics in Agriculture, 141, pp.351-356.

[4] Gong, H., Liu, T., Luo, T., Guo, J., Feng, R., Li, J., Ma, X., Mu, Y., Hu, T., Sun, Y. and Li, S., 2023. Based on FCN and DenseNet framework for the research of rice pest identification methods. Agronomy, 13(2), p.410.

[5] Yang, J., Ma, S., Li, Y. and Zhang, Z., 2022. Efficient data-driven crop pest identification based on edge distance-entropy for sustainable agriculture. Sustainability, 14(13), p.7825.

[6] Li, H., Li, S., Yu, J., Han, Y. and Dong, A., 2022, April. Plant disease and insect pest identification based on vision transformer. In International conference on internet of things and machine learning (IoTML 2021) (Vol. 12174, pp. 194-201).

[7] Talukder, M.S.H., Chowdhury, M.R., Sourav, M.S.U., Al Rakin, A., Shuvo, S.A., Sulaiman, R.B., Nipun, M.S., Islam, M., Islam, M.R., Islam, M.A. and Haque, Z., 2023. JutePestDetect: An intelligent approach for jute pest identification using fine-tuned transfer learning. Smart Agricultural Technology, 5, p.100279.

[8] Rong, M., Wang, Z., Ban, B. and Guo, X., 2022. Pest Identification and Counting of Yellow Plate in Field Based on Improved Mask R-CNN. Discrete Dynamics in Nature and Society, 2022(1), p.1913577.

[9] Singh, K.U., Kumar, A., Raja, L., Kumar, V., Singh kushwaha, A.K., Vashney, N. and Chhetri, M., 2022. An Artificial Neural Network-Based Pest Identification and Control in Smart Agriculture Using Wireless Sensor Networks. Journal of Food Quality, 2022(1), p.5801206.

[10] Hu, K., Liu, Y., Nie, J., Zheng, X., Zhang, W., Liu, Y. and Xie, T., 2023. Rice pest identification based on multi-scale double-branch GAN-ResNet. Frontiers in Plant Science, 14, p.1167121.

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

[12] M.Sireesha, S. N. TirumalaRao, Srikanth Vemuru, Optimized Feature Extraction and Hybrid Classification Model for Heart Disease and Breast Cancer Prediction International Journal of Recent Technology and Engineering Vol - 7, No 6, Mar - 2019 ISSN - 2277-3878, Pages – 1754 – 1772

[13] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[14] S. L. Jagannadham, K. L. Nadh and M. Sireesha, "Brain Tumour Detection Using CNN," 2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2021, pp. 734-739, doi: 10.1109/I-SMAC52330.2021.9640875

[15] Mascarenhas, S. and Agarwal, M., 2021, November. A comparison between VGG16, VGG19 and ResNet50 architecture frameworks for Image Classification. In 2021 International conference on disruptive technologies for multi-disciplinary research and applications (CENTCON) (Vol. 1, pp. 96-99). IEEE.

[16] Sunayna, S.S., Rao, S.N.T., Sireesha, M. (2022). Performance Evaluation of Machine Learning Algorithms to Predict Breast Cancer. In: Nayak, J., Behera, H., Naik, B., Vimal, S., Pelusi, D. (eds) Computational Intelligence in Data Mining. Smart Innovation, Systems and Technologies, vol 281. Springer, Singapore. https://doi.org/10.1007/978-981-16-9447-9_25

[17] S. M. Hassan and A. K. Maji, "Pest Identification Based on Fusion of Self-Attention With ResNet," in IEEE Access, vol. 12, pp. 6036-6050, 2024, doi: 10.1109/ACCESS.2024.3351003.