Crop Pest Detection and Measures

Harry A Haque[1*], Shiburaj A[2*],Muhammed Basheer K[3*], Prof. Basil Joy[4*]
*Department of Computer Science and Engineering
Mar Athanasius College of Engineering, Kothamangalam, Kerala
[1]harry2255.7@gmail.com, [2]shiburajamabali@gmail.com, [3]basheerkalliyath2006@gmail.com,
[5]basiljoy@mace.ac.in

***Abstract*—Crop pests are a major concern to the agricultural industry because they do significant harm to the environment, society, and economy. The effectiveness of pest management tactics depends on the precise identification of pests, which is made more difficult by the lack of extensive data sets and the inherent challenges of algorithmic interpretation. Inappropriate pesticide use resulting from incorrect pest classification can have a negative impact on crop productivity and the environment. In order to overcome these challenges, we provide ”PestNet,” a state-of-the-art, all-inclusive framework designed for the accurate detection and categorization of agricultural pests. PestNet demonstrates unparalleled accuracy, identifying a wide array of pest species with near-perfect precision. When compared to existing pre-trained deep learning models, PestNet exhibits superior performance, further validated by its remarkable 96.7% accuracy in classifying twenty one distinct pest types within the Kaggle Pest Dataset. The implementation of PestNet promises to revolutionize pest management practices by offering rapid, precise pest identification tools to agricultural professionals and farmers, thereby minimizing economic losses and protecting crop health. By ensuring accurate pest classification, PestNet aims to optimize pesticide usage, contributing to more sustainable agricultural practices and heralding a new era in pest management technology.**

***Index Terms*—PestNet, convolutional layers, Pests**

I. Introduction

A crucial use of technology in agriculture is crop pest identification, which makes use of cutting-edge methods, mainly in the fields of computer vision and machine learning. As the world’s population continues to rise, guaranteeing food security has emerged as a critical issue. However, a number of pests and diseases pose serious risks to agricultural output and health. It might be difficult to identify and take quick action in response to possible crop hazards when using traditional methods of pest monitoring because they are frequently labor- and time-intensive. With the creation of crop pest detection systems, contemporary technology provides creative answers to these problems. The automatic analysis of visual data, such as pictures or videos of crops, to spot indications of pest infestations is made possible by the incorporation of computer vision. Deep learning models in particular, which are machine learning algorithms, are essential for identifying complex patterns and anomalies linked to various pests.Crop pest detection supports precision and sustainable agriculture in addition to helping with pest management. Farmers may optimize pesticide use, minimize environmental effect, and cut costs by precisely detecting and classifying pest threats. Furthermore, a more focused and proactive approach is made possible by these sophisticated systems, which eventually improve crop output, quality, and overall agricultural sustainability. The contemporary trend of boosting agricultural output and food quality at the same time as cutting costs and raising profitability depends heavily on pest management. One of the most frequent sources of agricultural harm in the globe is insect pests. Reducing these losses has the potential to save a large portion of the crop and boost agricultural revenue. Low yields are the result of pest infestations on crops, which also damage the crops and spread a number of illnesses. One of the main causes of yield losses is insect damage to areas used for harvest, such as rice, wheat, and beans. To prevent insects from spreading across large regions, decrease crop losses caused by insects, and reduce insect populations, biocontrol techniques such as insecticides should be employed.Chemicals and pesticides play a major role in pest management. However, they will negatively impact the environment and human health in a number of ways. Furthermore, it’s important to identify the insect because different pest species require different approaches to pest management. The first line of defense against insect pests damaging crops is the ability to identify and categorize insects, to tell the difference between harmful and beneficial ones. However, classifying insects is a challenging task because of their complex structure and the similarities among different insect species.

Conversely, traditional machine learning algorithms have several shortcomings. When the number of crop pest species is restricted, traditional machine learning algorithms have been demonstrated to perform effectively; but, when different features are manually collected, they lose their effectiveness. They require an additional, crucial stage of data preprocessing called feature engineering. Furthermore, it has a limited ability to generalize between datasets. Additionally, the data that is accessible determines how effective they are. The research project is motivated by the continued interest in creating highly accurate automated systems for pest classification, even in the face of a large body of literature on pest recognition and classification.

In general, this approach offers a useful way to identify and categorize agricultural pests. The suggested method improves the performance and dependability of Crop Pest Detection and Classification systems by accurately detecting and classifying the crop pest through the use of sophisticated image processing techniques and machine learning algorithms.

II. RELATED WORKS

Recently, some research studies have been concentrated on pest classification and recognition. The pest classification and recognition research can be divided into ML, DL, and hybrid-based approaches. Hybrid methods include techniques that employ both DL and ML techniques. Many pest classification research works use DL-based techniques, whereas ML-based techniques are rarely used. Below, we highlighted the most recent and relevant research work in automatic pest identification and classification.

Recently, advanced ML-based techniques have effectively performed well in pest categorization and detec- tion [16]–[18]. Multiple classifiers are trained using extracted features from pests, and multiple types of pest images were categorized in these works. In [19], The UAV dataset was used to forecast armyworm contaminated and healthy corn regions, and the armyworm occurrence levels were then categorized. The best combination of image features for recognizing armyworm insects in corn-planted regions was discovered utilizing Gini-importance. The authors compared four types of ML methods: Random Forest (RF), Multilayer Perceptron (MLP), Naive Bayesian (NB), and SVM. The RF model performs the best compared to other ML approaches (MLP, NB, and SVM) for classifying the armyworm pest and normal corn. The authors [20] proposed an automatic pest detection (thrips) system based on SVM for greenhouse pest attack monitoring. Researchers used a novel image pro- cessing technique to detect parasites on strawberry plants. The SVM approach with a different kernel function was applied for parasite classification and thrips identification. The SVM structure was designed using the main diameter to minor diameter ratio as a region index, Hue, Saturation, and Intensify as colour indices. The results demonstrate that utilizing the SVM method with area index and intensify as colour index produces the best classification, with a mean percent error of under 2.25 percent. In [21], the authors used two feature extraction techniques for the identification and categorization of tomato pests, namely, Histogram of Ori- ented Gradient (HOG) and Local Binary Pattern techniques (LBP). HOG outperforms its competitor, according to the comparison results. However, these ML-based pest recogni- tion systems rely on 'handcrafted' features extracted from the actual domain by comparing individual appearances. As a result, the empirical parameters must be manually changed to account for changes in image acquisition conditions. As a result, detecting pests in outside environments using hand- crafted techniques based on colour, structure, and texture remains a serious difficulty.

The importance of DL methods in computer vision has inspired researchers to utilize them for pest recognition and classification [22]–[24]. But unfortunately, DL algorithms in the domain of pest recognition have been constrained by a scarcity of pest image datasets and the inexplicability of DL frameworks. A novel and robust dataset for crop pest recognition was created in [25], and three different DL models were trained to employ TL and fine-tuning. The recognition rate of the three Architectures was greater than 80.00 per cent. Using the gradient-weighted class activation technique, the authors presented appropriate visual descriptions for the most crucial portions of the recognition layers. According to this study, the recognition process concentrates more on visual details than the entire image, and general differences are overlooked. An end-to-end pest detection system combining DL and hyperspectral imaging (HSI) techniques is presented in [26]. This technique can be used to quickly recognize pests for successful pest control. To address noise and duplicate details in the HSI spectral space, one-dimensional convolution and attention techniques across spectral channels are employed to develop a spectral feature extraction unit to effectively use spectrum information. The HSI feature extractor secures rich spectral-spatial information using a three-dimensional convolution branch structure with various resolutions in parallel. The output feature map maintains its higher resolution throughout its use. Each branch contains an adjustable spectral-spatial feature extraction unit that dynamically weights different inputs, limiting the HSI's disproportionate effect and improving the network's feature extraction skills. Pest HSI was acquired utilizing hyperspectral imaging equipment, resulting in a dataset containing nine different pests.

Furthermore, it is commonly known that the hybrid models (DL and ML models) produce better classification results, which can be used to classify insects. In [27], DL models (TL technique) were used to classify eight different types of tomato pests. Using DL models, the extracted pest features were merged with three ML classifiers, i.e., discriminant analysis (DA), SVM, and the k-nearest neighbour approach (KNN). Bayesian optimization was used to effectively tune hyper-parameters. Following image augmentation, the VGG16 framework performed better than the other algo- rithms. The ResNet50 with discriminant analysis classifier had the best accuracy among the CNN and ML frameworks. The authors [28] developed a new DL model TPest-RCNN for pest detection. The faster regional-convolutional neural network is used as a based model in the proposed model. Moreover, VGG16 was used for feature extraction. Then, a region of tiny pests was generated by a region proposal framework. Finally, extracted features and regions of small pests are fed to RoIAlign for classification and detection.

Following a review of prior research, it is discovered that DL models, particularly those used to categorize crop pests, cannot achieve superior identification and classification performance. Instead of using real-world scenes, most models were trained and tested using images captured in highly controlled lab environments. However, in the field, the complicated surroundings, various viewpoints and postures, varying degrees of colour and texture alteration, changes in lighting conditions, and different locations of pests' wings and limbs constitute a considerable barrier to pest recognition. Although few studies addressed automatic detection in natural settings (testing is performed on natural images), most of them focused on a single species or used only one dataset for validation purposes. Detecting pests effectively and quickly and extracting traits independent of viewpoint, scale, and lighting conditions are critical for crop pest recognition. This paper developed an effective DL framework for identifying and classifying crop pests into ten different classes. Also, the data set size is boosted using image rota- tion and data augmentation techniques to obtain generalized results. To test the generalizability of the proposed approach, we tested it on a different dataset with 9 different types of crop pests.

III. Methodology

The outlined methodology details the development of a crop pest detection system for sustainable agriculture. The procedure entails obtaining an image dataset from various Kaggle datasets. We merge multiple datasets sourced from Kaggle to create a comprehensive dataset., specifically focused on various types of pests. These images are then used to train machine learning models to automatically detect and classify pests. During training, image rotation and data augmentation techniques like DeepSMOTE[2] are applied to enhance the dataset size and address class imbalance issues. Convolutional neural networks (CNNs), particularly MobileNetV3[3], are trained on this dataset to detect and classify pests automatically. In parallel, satellite imagery, drones, and aerial photography are used to capture high-resolution images of crop fields, emphasizing pest detection. These images undergo pre-processing to extract features that indicate potential pest infestations.

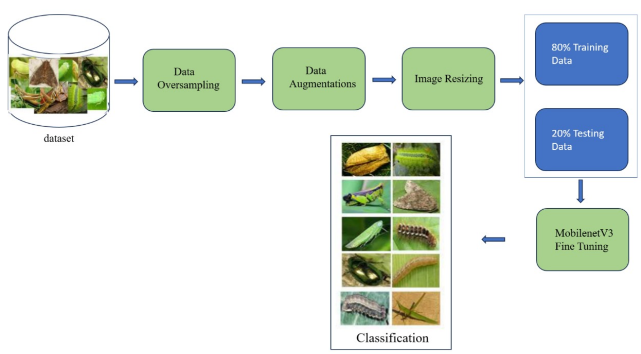

Fig. 1: Proposed method

The MobileNetV3[3] based model used in this methodology has sixteen layers, including thirteen depthwise separable convolutional layers and three fully connected (FC) layers. This design balances efficiency and accuracy, allowing for quick and reliable pest detection. Depthwise separable convolutions significantly reduce the number of parameters, enabling MobileNetV3[3] to operate effectively in resource-constrained environments like mobile devices or edge computing systems. Inverted residual blocks with linear bottlenecks and squeeze-and-excitation modules contribute to the model's robustness and versatility.

Upon successful testing, the system is deployed as a web-based application, providing users with a practical tool for real-time pest detection and classification. This methodology's use of MobileNetV3[3] allows for rapid processing, facilitating real-time decision-making for agricultural management. The lightweight architecture and advanced feature extraction techniques ensure a systematic and robust approach to crop pest detection, promoting sustainable agriculture practices.

A. DeepSMOTE

The initial phase is increasing the training dataset and addressing the class imbalance issues using data augmentation technique DeepSMOTE[2]. DeepSMOTE[2] consists of an encoder/decoder framework, a Synthetic Minority Oversampling Technique (SMOTE) based oversampling method, and a loss function with a reconstruction loss and a penalty term. The DeepSMOTE[2] is based on the deep convolutional GAN (DCGAN) architecture, used a discriminator/generator in a GAN, which is fundamentally similar to an encoder/decoder because the discriminator effectively encodes input (absent the final, fully connected layer) and the generator (decoder) generates output.

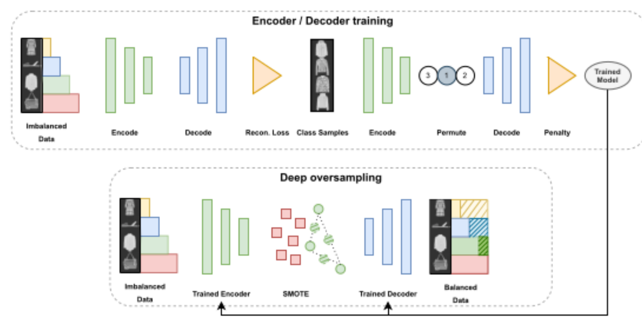

Fig. 2: DeepSmote implementation

B. Encoder/Decoder Framework

The encoder and decoder are trained in an end-to-end fashion. During DeepSMOTE[2] training, an imbalanced dataset is fed to the encoder/decoder in batches. A reconstruction loss is computed on the batched data. All classes are used during training so that the encoder/decoder can learn to reconstruct both majority and minority class images from the imbalanced data. Because there are few minority class examples, majority class examples are used to train the model to learn the basic reconstruction patterns inherent in the data. This approach is based on the assumption that classes share some similar characteristics.

C. Enhanced Loss Function

In addition to a reconstruction loss, the DeepSMOTE[2] loss function contains a penalty term. The penalty term is based on a reconstruction of embedded images. DeepSMOTE's penalty loss is produced in the following fashion. During training, a class (c) is randomly selected from the set of all classes (C). A group of examples is then randomly sampled from c that is equal in number to the batch size. Thus, the number of sampled examples is the same as the number of examples used for reconstruction loss purposes; however, unlike the images used during the reconstruction loss phase of training, the sampled images are all from the same class. The sampled images are then reduced to a lower-dimensional feature space by the encoder. During the decoding phase, the encoded images are not reconstructed by the decoder in the same order as the encoded images. By changing the order of the reconstructed images, which are all from the same class, we effectively introduce variance into the encoding/decoding process. For example, the encoded order of the images may be D0 , D1 , D2 , and the decoded order of the images may be D2 , D0 , D1 . This variance facilitates the generation of images during inference (where an image is encoded, SMOTEd, and the decoded).

Essentially, the permutation step is necessary because DeepSMOTE[2] uses an autoencoder (an encoder plus a decoder). The output of an autoencoder is deterministic with respect to its input, in the sense that an autoencoder can only decode or generate what it encodes. We include variance into the encoding/decoding process by permuting the order of the encoded data. By introducing variance into the encoding process, the decoder gains "practice" at decoding examples that are different from the input data (which a standard decoder in an autoencoder is not trained to do). This "practice" is necessary because during inference, an example is encoded, then it is changed via SMOTE interpolation to a different example, which the decoder must decode. Once DeepSMOTE[2] is trained, images can be generated with the encoder/decoder structure. The encoder reduces the raw input to a lower-dimensional feature space, which is oversampled by SMOTE. The decoder then decodes the SMOTEd features into images, which can augment the training set of a deep learning classifier. The main difference between the DeepSMOTE[2] training and generation phases is that during the data generation phase, SMOTE is substituted for the order permutation step. SMOTE is used during data generation to introduce variance, whereas during training, variance is introduced by permuting the order of the training examples that are encoded and then decoded and also through the penalty loss.

D. Artificial Image Generation

Once DeepSMOTE is trained, images can be generated with the encoder/decoder structure. The encoder reduces the raw input to a lower-dimensional feature space, which is oversampled by SMOTE. The decoder then decodes the SMOTEd features into images, which can augment the training set of a deep learning classifier. The main difference between the DeepSMOTE[2] training and generation phases is that during the data generation phase, SMOTE is substituted for the order permutation step. SMOTE is used during data generation to introduce variance, whereas during training, variance is introduced by permuting the order of the training examples that are encoded and then decoded and also through the penalty loss. SMOTE itself does not require training because it is nonparametric.

E. Data Augmentation

The unavailability of a large amount of data for training the DL frameworks is one of the challenges when intending to use DL approaches to pest recognition and classification problems. More crop pest data is difficult and expensive to get, both in terms of time and resources. Data augmentation, or increasing the amount of available data without acquiring new data by applying multiple processes to current data, has been proven advantageous in image classification [12]. The ImageNet classifier challenge winners adopted this method [13], [14] and used it academically to improve training data and reduce overfitting [15]. Due to the limited number of images in the dataset, we applied image rotations, and data augmentation approaches in this study. For this purpose, we rotated all of the dataset's images (both training and testing) by 90 degrees twice. The dataset's image count was raised threefold through this image rotation procedure. Additionally, the images in the training set were rotated, arbitrarily translated up to thirty pixels vertically and horizontally. They randomly translated the images between [0.9 and 1.1] to create additional images. It's also worth noting that sets of augmented images are dynamically created during each training phase. The number of images in the training set was significantly expanded using this data augmentation method, enabling more effective use of our DL model by training with a much higher number of training images. Furthermore, the augmented images are only used to train the proposed framework, not to test it; hence, only real images from the dataset are utilized to test the learned framework.

F. Image Resizing

The input images in the datasets are of different sizes. To ensure uniformity and speed up the processing, we applied certain pre-processing to resize the input images to 224×224 pixels according to the input image requirements of our model.

G. Dataset Partitioning

For each experiment, the dataset is separated into training and testing sets. More precisely, 80% of the dataset was used for model training, and 20% was used for testing.

H. MobileNetV3 Architecture Details

MobileNetV3[3], a lightweight yet robust convolutional neural network (CNN), forms the backbone of our methodology for real-time pest recognition and classification. This architecture is optimized for efficient object detection, making it suitable for deployment in resource-constrained environments, such as mobile devices or edge computing systems often used in agricultural settings.

MobileNetV3's design incorporates sixteen learnable layers, consisting of thirteen depthwise separable convolutional layers and three fully connected (FC) layers. This structure allows the model to maintain high performance while remaining lightweight, facilitating rapid detection and classification of crop pests. The architecture integrates depthwise separable convolutions, inverted residual blocks with linear bottlenecks, and squeeze-and-excitation modules to optimize feature extraction and computational efficiency.

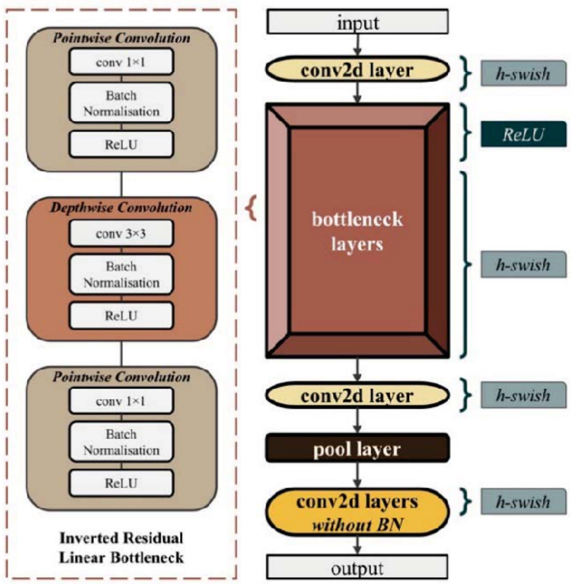

Fig. 3: MobileNetV3 Architecture

The input layer of MobileNetV3[3] receives images with a resolution of 224 x 224 pixels. These images are then processed through a series of convolutional operations designed to extract critical features for identifying pests. The depthwise separable convolutions reduce computational load, while the inverted residual blocks with linear bottlenecks enhance information flow through the network. The squeeze-and-excitation modules dynamically adjust channel-wise attention, allowing the model to focus on relevant features for accurate pest recognition and classification.

I. Fine Tuning

To fine-tune MobileNetV3[3] for pest classification, we utilized a custom dataset with labeled images of various crop pests. This process involved customizing the model's parameters to align with our specific dataset. Data augmentation techniques, such as random cropping, flipping, etc., were employed to artificially expand the dataset size and improve model robustness against variations in pest appearance and background conditions. Batch normalization layers were used to stabilize training by addressing internal covariate shift, a phenomenon where the distribution of input data changes during training. By introducing dense layers with appropriate activation functions, like the ReLU used here, the model gains the capacity to learn intricate patterns and relationships within the data. These dense layers typically contain a large number of neurons (units) – the 256 units used here represent a good balance between model complexity and efficiency.

The fully connected layers near the end of the architecture transform the high-level features extracted by MobileNetV3[3] into a format suitable for classification. Two dense layers with 256 units each are appended to the pretrained model's output. These layers utilize the rectified linear unit (ReLU) activation function to introduce non-linearity. This allows the model to capture complex, non-linear relationships within the pest image data. To prevent overfitting, dropout regularization is applied after each dense layer. This regularization technique randomly drops a certain percentage of neurons during training, forcing the model to rely on a broader set of features and enhancing its generalization capability to unseen data. The final output layer uses a softmax function to generate a probability distribution across pest categories. This probability distribution provides not only the most likely pest classification but also the confidence scores for each category, enabling precise identification and classification.

IV. Results

We provide a detailed description of the findings of numerous studies conducted to determine the efficacy of our pest classification model.

A. Dataset

To assess the performance of the proposed framework, we combine multiple dataset presented in kaggle. It comprises twenty one distinct pest categories primarily seen in tea plants and other plants throughout Europe and Central Asia. More specifically, the dataset's twenty one different pests images include Aphids, Africanized Honey Bees (Killer Bees), Armyworm, Beetle, Bollworm, Brown Marmorated Stink Bugs, Cabbage Loopers, Citrus Canker, Colorado Potato Beetles, Corn Borers, Corn Earworms, Fall Armyworms, Fruit Flies, Grasshopper, Mites, Sawfly, Spider Mites, Stem Borer, Thrips, Tomato Hornworms and Western Corn Rootworms. A sample representation of each pest type is shown in Fig. 1. The total number of pests images in our dataset is 8400. Each class of pests consists of 400 images. The number of images against each pest's category is shown in Table 1. Fig. 2 shows sample images after rotations. We rotated the images of the dataset twice by 90 degrees. The number of images against each pest's category after rotation is also mentioned in Table 3. The pest images in the dataset were gathered from Mendeley and other online sources and include pest images acquired with a Single Lens Reflex camera (SLR). The rest of the images were from Insert Images, IPM Images, Dave's Garden, and others. The dataset contains RGB images of different resolutions. The size, posture, angle, lighting conditions, and backgrounds of the sample images vary greatly.

B. Evaluation Matrics

The following assessment measures are used to assess the proposed method's performance: Accuracy, Precision, Sensitivity (Recall) and F1 score. The accuracy of the presented framework is given by equation 1, defined as the number of correctly detected or classified images to the total number

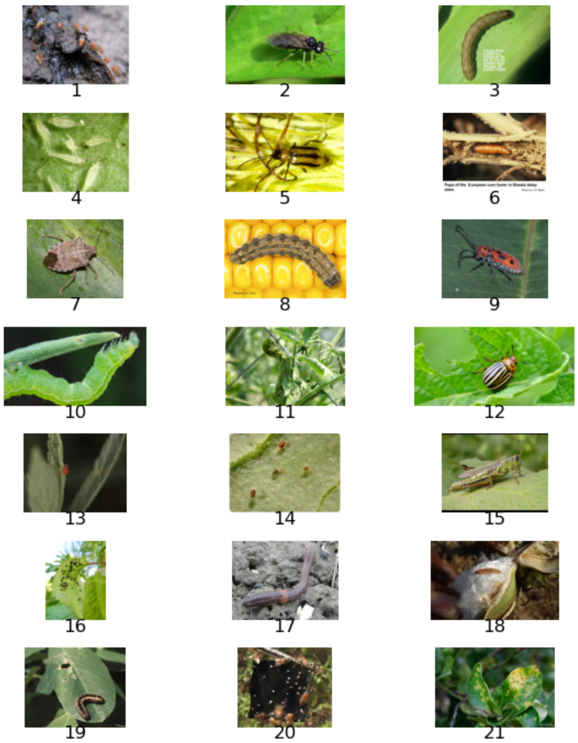

Fig. 4: Sample images from our dataset (1) Fruit Flies, (2) Sawflly, (3) Armyworm, (4) Thrips, (5) Western Corn Rootworms, (6) Corn Borers, (7) Brown Marmorated Stink Bugs, (8) Corn Earworms, (9) Beetle, (10) Cabbage Loopers, (11) Tomato Hornworms, (12) Colorado Potato Beetles, (13) Mites, (14) Spider Mites, (15) Grasshopper, (16) Aphids, (17) Stem Borer, (18) Bollworm, (19) Fall Armyworms, (20) Africanized Honey Bees (Killer Bees), (21) Citrus Canker

Fig. 5: Sample images from our dataset after rotations

of sample images. The precision of the proposed model is identified as the numberof correctly detected or classified images to the total number of positive images detected (correctly or erroneously) by the model. The recall is calculated as the number of correctly classified images to the total number of images in the dataset. Whereas F1 score combines precisions and recall and calculates the weighted average of both. The equations to estimate these metrics are:

$$f(x) = \max(x1, x2, x3, \ldots, xk) \tag{1}$$

$$\text{Accuracy} = \frac{TN + TP}{TS} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$\text{F1 score} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

TP, TS, FP, TN, and FN stand for true positive, total samples, false positive, true negative, and false negative, respectively.

C. Experimental Setup

The proposed method is tested and validated on a system with an Intel (R) Core (TM) i5-5200U processor and 8GB of RAM. Models require input images to be shrunk; thus, input images are resized accordingly. 80% of the images were used for train- ing, while 20% were utilized for testing. For all the exper- iments, the training and testing sets are being used to train and test the proposed approach and other contemporary models using the same experimental settings for pest recognition and classification. A set of tests are car- ried out to assess the proposed framework for multiclass classification pest classification's classification performance.

D. Performance Evaluation On Pest Recognition And Classification

This experiment aims to verify the usefulness and effectiveness of the proposed pest recognition and classification method. We used all 8400 crop pest images of our dataset in this experiment (6720 pest images for model training and the rest 1680 images for model testing). The total number of iterations in the training stage for our method is 100 iterations per epoch, and the number of epochs is 100. We also created a confusion matrix assessment to precisely explain the proposed technique's classification performance in terms of actual and predicted classes. The proposed approach confusion matrix is shown in Figure 6. It is concluded from the confusion matrix that the proposed system achieves the optimal results with a true- positive rate of 100% for all the pest classes in our dataset, indicating that the proposed framework correctly classified all pest image samples. The loss function shows how well our framework can predict the dataset. which means we can obtain satisfactory results even at lower classification epochs. The proposed approach attained the ideal accuracy, precision, recall, and F1-score of 100%, demonstrating its effectiveness for multiclass classification of pest images.

To further identify the effectiveness and validity of the proposed approach, precise recognition and classification of many crop pests are required. For this purpose, we assess the usefulness of the proposed approach in identifying the class of each crop pest. Figure 7 shows the precision, recall, and

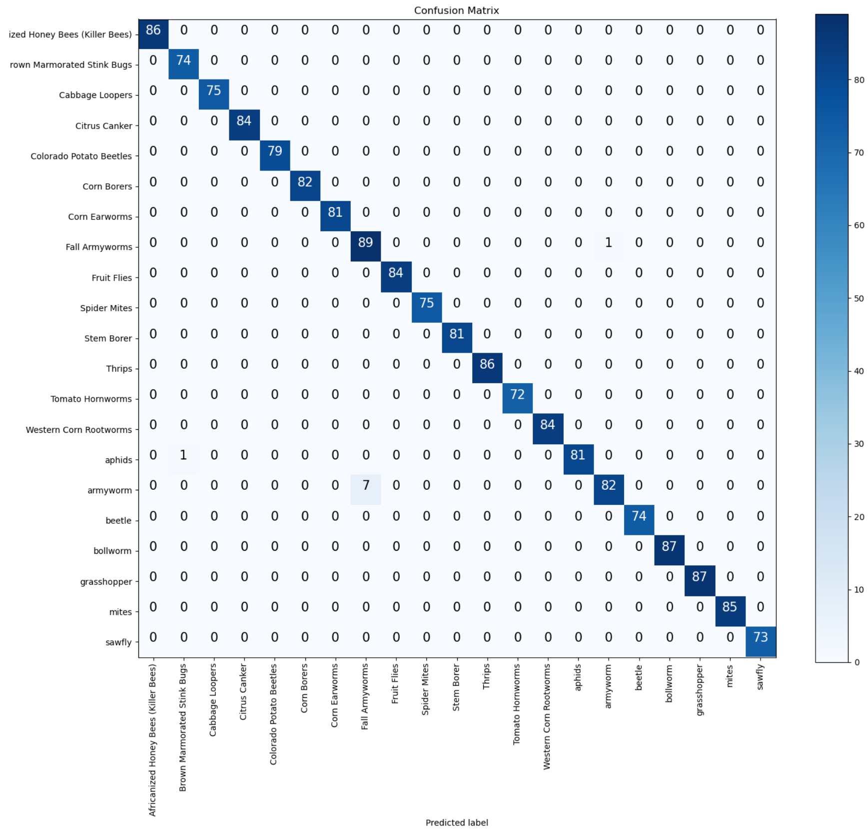

Fig. 6: Confusion Matrix

	precision	recall	f1-score
Africanized Honey Bees (Killer Bees)	1.00	1.00	1.00
Brown Marmorated Stink Bugs	1.00	1.00	1.00
Cabbage Loopers	1.00	1.00	1.00
Citrus Canker	1.00	1.00	1.00
Colorado Potato Beetles	1.00	1.00	1.00
Corn Borers	1.00	1.00	1.00
Corn Earworms	1.00	1.00	1.00
Fall Armyworms	0.93	0.94	0.94
Fruit Flies	1.00	1.00	1.00
Spider Mites	1.00	0.97	0.98
Stem Borer	1.00	1.00	1.00
Thrips	1.00	1.00	1.00
Tomato Hornworms	1.00	1.00	1.00
Western Corn Rootworms	1.00	1.00	1.00
aphids	0.99	1.00	0.99
armyworm	0.94	0.93	0.94
beetle	1.00	1.00	1.00
bollworm	1.00	1.00	1.00
grasshopper	1.00	1.00	1.00
mites	0.99	1.00	0.99
sawfly	1.00	1.00	1.00

Fig. 7: Class-wise performance

F1-score performance of the proposed approach in class-wise crop pest classification. The proposed technique provides state-of-the-art performance in terms of all evaluation parameters, as revealed in Figure 7. The results show that all pests images are correctly classified, resulting in optimal accuracy. The robustness of the introduced DL model, which better reflects each class, is the critical cause for the improved pest recognition accuracy.

E. Model Comparison

The key aim of this experiment is to validate the efficiency of the proposed technique for pest recognition and classification before and after applying the oversampling method DeepSmote. For this purpose, We assessed the classification performance of our proposed framework in comparison with the pre-trained DL model PestDetNet[1]. PestDetNet demonstrated superior performance compared to two other DL models, namely SqueezeNet[4] and GoogleNet[5]. Therefore, our evaluation primarily focused on PestDetNet. The framework are trained on many images from the Deng et al. (2018) dataset. Figure 9 & 10 shows the training and validation accuracy of our proposed model and the PestDetNet Model. Figure 11 & 12 shows the training and validation loss of our proposed model and the PestDetNet Model. From the findings shown in the figures, it is evident that the PestDetNet achieved the lowest performance results compared to the proposed model in terms of all performance metrics. It is essential to mention that the PestDetNet achieved the same classification results in all performance measures. Based on the results, we noticed that the proposed framework performed better than the PestDetNet framework by achieving accuracy, precision, recall, and an F1 score.

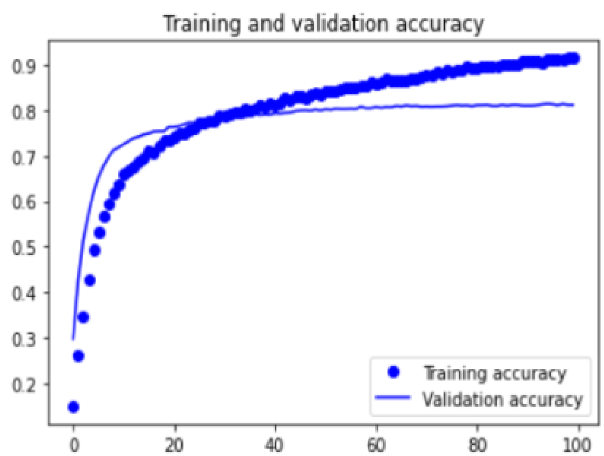

Fig. 8: PestDetNet Model Accuracy

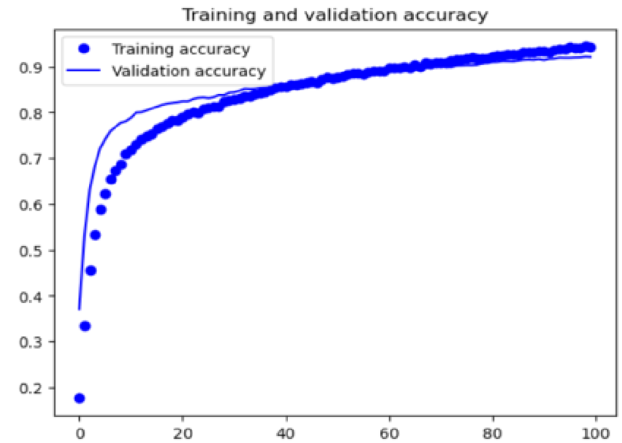

Fig. 9: Proposed method Accuracy

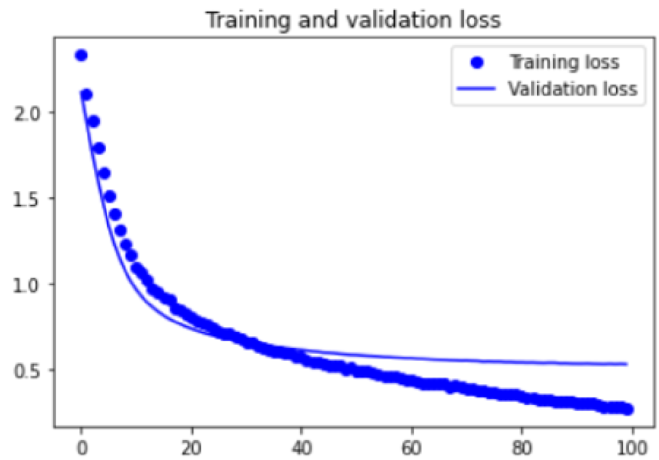

Fig. 10: PestDetNet Model Loss

Furthermore, we experimented with comparing the proposed model and existing state-of-the-art pest recognition and classification methods to verify the proposed model's superiority. We compared the proposed approach to the most recent DL frameworks and presented the results in Table 10. For pest recognition, Nanni et al. [5], proposed ensem- bles of CNNs based on multiple architectures (ResNet50, EfficientNetB0, ShuffleNet, GoogleNet, DenseNet201, and MobileNetv2) optimized with various Adam versions. Two novel Adam algorithms based on DGrad for deep network optimization are proposed, each with a scaling factor in the learning rate. Six CNN models with different optimization functions are trained on the Deng (SMALL) dataset, big IP102, and Xie2 (D0) pest datasets. On the Deng dataset, the best scoring ensemble competed with domain experts' classifications. It achieved the best results on all three pest datasets: 89.28 per cent on Deng, 74.11 per cent on IP102, and 94.81 per cent on Xie2. In [6], the authors presented a deep CNN framework for classifying insects on three widely accessible insect datasets. The first insect dataset employed was the National Bureau of Agricultural Insect Resources (NBAIR), which comprises 40 classes of crop pest photos. In contrast, the second and third datasets (Xie1, Xie2) com- prise 24 and

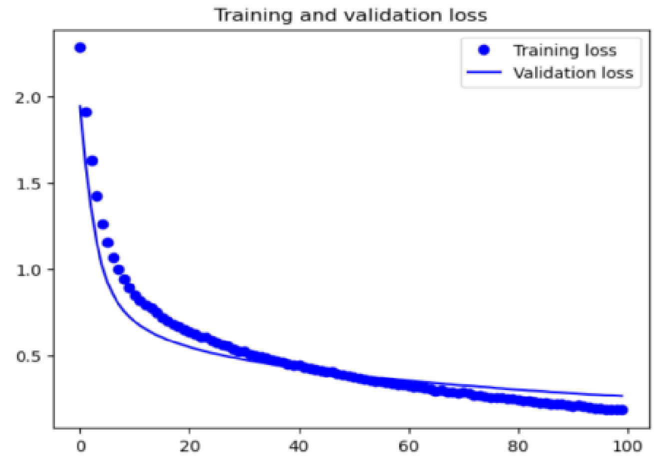

Fig. 11: Proposed method Loss

Model	Accuracy	Precision	Recall	F1-score
PestDetNet	93.6	93.2	93.2	93.2
Proposed Model	96.7	96.3	96.3	96.3

Fig. 12: performance comparison

40 classes of pests, respectively. Data augmen- tation methods such as scaling, reflection, translation, and rotation are used to prevent the network from overfitting. To increase accuracy, the effectiveness of hyperparameters was investigated in the proposed model. The proposed CNN framework achieved the best accuracy of 92.34%, 90.47%, and 93.97% for the NBAIR insect dataset, Xie insect dataset, and Xie2 insect dataset, respectively. In [7], DL models were used to categorize eight different types of tomato pests. The extracted pest features are merged with three ML clas- sifiers using the DL models, including discriminant analy- sis (DA), SVM, and k-nearest neighbour (KNN) approach. Bayesian optimization was used to tune hyper-parameters automatically. With an accuracy of 90.95 per cent after image augmentation, the VGG16 model performed better than the other models. The ResNet50 with DA framework achieved 91.73 per cent classification accuracy in the CNN + ML models. This comparison also shows how successful the proposed model is compared to other approaches. It's worth nothing that these methods are more computation- ally expensive than the proposed because they use deeper frameworks, which can inevitably lead to overfitting. These findings indicate the effectiveness of the proposed tech- nique and its additional advantages, such as computing efficiency. We may say that the proposed approach is more efficient and effective in identifying and classifying pests images.

V. Conclusion and Future Work

Globally, contamination is the major cause of agricultural loss and financial loss. The detection and removal of exotic insects would be greatly accelerated if invading insects could be identified automatically. This paper presented a MobileNet framework for effective pest recognition and classification. The accuracy of 96% for pest recognition and classification has confirmed the superiority of the proposed framework over contemporary methods. Moreover, experimental results on the standard Kaggle dataset (Pest Dataset)have confirmed the effectiveness and robustness of the proposed framework for pest recognition and classification. However, only major pests are investigated in this study. There are many different types of insects, and there are distinctions between larvae and adults. For example, noctuid pests have only been identified during their most dangerous stage of development (i.e., the larval stage). In future, we are interested in expanding the classification size by including more pests types to be effectively identified by the proposed MobilNet framework. This research could help specialists and farmers identify pests more quickly and effectively, thus reducing economic and crop output losses.

TABLE I: Comparison of Different Methods

Work	Dataset	Method	Accuracy
Loris Nanni et al.,[5]	Deng et al.,	Ensemble CNN	89.28%
K. Thenmozhi et al.,[6]	Xeil(24 class)	Deep CNN	92.34%
Huang, M.L et al.,[7]	IPM Images	ResNet50 with DA	91.73%
Wang Dawei et al.,[8]	Deng et al.,	LCP plus SVM	85.5%
PestDetNet et al.,[9]	Deng et al.,	PestDetNet Model	93.6%
Proposed method	Multiple pest datasets	MobileNetV3	96.7%

References

[1] Naeem Ullah, Javed Ali Khan, Lubna Abdulaziz Alharbi, Asaf Raza, Wahab Khan, Ijaz Ahmad, "An Efficient Approach for Crops Pests Recognition and Classification Based on Novel DeepPestNet Deep Learning Model" , Volume 10, may 2022

[2] Damien Dablain , Bartosz Krawczyk, "DeepSMOTE: Fusing Deep Learning and SMOTE for Imbalanced Data", vol. 34, no. 9, september 2023

[3] Siying Qian,Chenran Ning,Yuepeng Hu, "MobileNetV3 for Image Classification", pp. 490-497, doi: 10.1109/ICBAIE52039.2021.9389905., September 2023

[4] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50X fewer parameters", 2016, arXiv:1602.07360.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions" in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.

[6] L. Nanni, A. Manfè, G. Maguolo, A. Lumini, and S. Brahnam, "High performing ensemble of convolutional neural networks for insect pest image detection" Ecol. Informat., vol. 67, Mar. 2022, Art. no. 101515.

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

[8] M.-L. Huang, T.-C. Chuang, and Y.-C. Liao, "Application of transfer learning and image augmentation technology for tomato pest identification" Sustain. Comput., Informat. Syst., vol. 33, Jan. 2022, Art. no. 100646.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going deeper with convolutions" in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 1–9.

[10] W. Dawei, D. Limiao, N. Jiangong, G. Jiyue, Z. Hongfei, and H. Zhongzhi, "Recognition pest by image-based transfer learning" J. Sci. Food Agricult., vol. 99, no. 10, pp. 4524–4531, Aug. 2019.

[11] L. Deng, Y. Wang, Z. Han, and R. Yu, "Research on insect pest image detection and recognition based on bio-inspired methods" Biosyst. Eng., vol. 169, pp. 139–148, May 2018.

[12] W. Li, D. Wang, M. Li, Y. Gao, J. Wu, and X. Yang, ”Field detection of tiny pests from sticky trap images using deep learning in agricultural green- house” Comput. Electron. Agricult., vol. 183, Apr. 2021, Art. no. 106048

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ”ImageNet classification with deep convolutional neural networks” in Proc. Adv. Neural Inf. Process. Syst., vol. 25, 2012, pp. 1–9.

[14] 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.

[15] S. C. Wong, A. Gatt, V. Stamatescu, and M. D. McDonnell, ”Understanding data augmentation for classification: When to warp?” in Proc. Int. Conf. Digit. Image Comput., Techn. Appl. (DICTA), Nov. 2016, pp. 1–6.

[16] S. Yousefi, F. Haghighian, M. Naghdyzadegan-Jahromi, and H. R. Pourghasemi, ”Pest-infected oak trees identify using remote sensing-based classification algorithms” in Computers in Earth and Environmental Sciences. Amsterdam, The Netherlands: Elsevier, Jan. 2022, pp. 363–376.

[17] D. Wei, J. Chen, T. Luo, T. Long, and H. Wang, ”Classification of crop pests based on multi-scale feature fusion,” Comput. Electron. Agricult., vol. 194, Mar. 2022, Art. no. 106736.

[18] L. Jiao, C. Xie, P. Chen, J. Du, R. Li, and J. Zhang, ”Adaptive feature fusion pyramid network for multi-classes agricultural pest detection” Comput. Electron. Agricult., vol. 195, Apr. 2022, Art. no. 106827.

[19] W. Tao, X. Wang, J. Xue, W. Su, M. Zhang, D. Yin, D. Zhu, Z. Xie, and Y. Zhang, ”Monitoring the damage of armyworm as a pest in summer corn by unmanned aerial vehicle imaging,” Pest Manage. Sci., vol. 78, no. 6, pp. 2265–2276, Mar. 2022.

[20] M. A. Ebrahimi, M. H. Khoshtaghaza, S. Minaei, and B. Jamshidi, ”Vision-based pest detection based on SVM classification method,” Comput. Electron. Agricult., vol. 137, pp. 52–58, May 2017.

[21] G. Pattnaik and K. Parvathi, ”Automatic detection and classification of tomato pests using support vector machine based on HOG and LBP feature extraction technique,” in Progress in Advanced Computing and Intelligent Engineering. Singapore: Springer, 2021, pp. 49–55.

[22] K. N. Dharmasastha, K. S. Banu, G. Kalaichevlan, B. Lincy, and B. K. Tripathy, ”Classification of pest in tomato plants using CNN,” in Proc. Int. Conf. Metaheuristics Softw. Eng. Appl. (METASOFT), Bhubaneswar, India, 2022, pp. 1–10.

[23] T. Wang, L. Zhao, B. Li, X. Liu, W. Xu, and J. Li, ”Recognition and counting of typical apple pests based on deep learning,” Ecol. Informat., vol. 68, May 2022, Art. no. 101556.

[24] D. J. A. Rustia, J. Chao, L. Chiu, Y. Wu, J. Chung, J. Hsu, and T. Lin, ”Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method,” J. Appl. Entomol., vol. 145, no. 3, pp. 206–222, Apr. 2021.

[25] K. Wang, K. Chen, H. Du, S. Liu, J. Xu, J. Zhao, H. Chen, Y. Liu, and Y. Liu, ”New image dataset and new negative sample judgment method for crop pest recognition based on deep learning models,” Ecol. Informat., vol. 69, Jul. 2022, Art. no. 101620.

[26] Z. Xiao, K. Yin, L. Geng, J. Wu, F. Zhang, and Y. Liu, ”Pest identification via hyperspectral image and deep learning,” Signal, Image Video Process., vol. 6, pp. 1–8, Jan. 2022.

[27] M.-L. Huang, T.-C. Chuang, and Y.-C. Liao, “Application of transfer learning and image augmentation technology for tomato pest identification,” Sustain. Comput., Informat. Syst., vol. 33, Jan. 2022, Art. no. 100646.

[28] W. Li, D. Wang, M. Li, Y. Gao, J. Wu, and X. Yang, ”Field detection of tiny pests from sticky trap images using deep learning in agricultural green- house,” Comput. Electron. Agricult., vol. 183, Apr. 2021, Art. no. 106048.