Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 04, April-2025

PestFertNet: A Rapid Crop Pest Analysis and Fertilizer Recommendation System

P. Saritha Hepsibha,
Associate Professor,
Department of Information Technology Anil Neerukonda Institute of Technology and Sciences Sangivalasa,
Visakhapatnam, Andhra Pradesh, India

Amrutha, Bharat Kumar, Niteesh Kumar, Vamsi Krishna
Department of Information Technology Anil Neerukonda Institute of Technology and Sciences
Sangivalasa, Visakhapatnam, Andhra Pradesh, India

***Abstract*—Abstract—Agricultural productivity is often threatened by pest infestations, leading to significant crop losses. Traditional pest detection methods rely on manual inspection, which is time-consuming, labourintensive, and prone to errors. To address this challenge, we propose an AI-driven pest detection system that leverages deep learning to analyze plant images and accurately identify pests. Our model incorporates a PestFertNet architecture, optimized through extensive dataset augmentation and hyperparameter tuning. By training on a diverse dataset with over 716 images per class, through comparative analysis we conclude that our approach achieves robust performance in distinguishing pest types and recommending appropriate interventions. Initial results demonstrate significant improvements in accuracy, moving towards our goal of achieving over 95% f1score 91% Accuracy 92% precision and 91% recall. This system has the potential to empower farmers with real-time, automated pest diagnosis, ultimately improving crop health and agricultural sustainability. Index Terms—Classification of pest, deep learning, PestFertNet, pest detection.**

***Index Terms*—Classification of pest, deep learning, Custom CNN, pest detection.**

I. INTRODUCTION

Agriculture is the backbone of our society—it's how we get the food we rely on every day. As the global population is projected to exceed 9 billion by 2050, the pressure to increase food production is greater than ever. Yet, one of the biggest challenges farmers face is crop pests, which in some regions can slash yields by up to 40% [1]. These pests not only weaken crops by feeding on them but also drive up costs, as farmers must invest more in pesticides and extra labor to keep their fields in check.In the past, farmers and experts had to check crops by hand for pests The human labour involved in this process is both timeconsuming and of varying quality—different people may find different signs of infestation. Furthermore, these customary techniques soon become impracticable on extremely large farms establish in regions short of expert personnel To solve these problems, digital images combined with computer algorithms are becoming increasingly used to detect pests in the field automatically [2]. and widely used for a number of tasks, once they are used in real farm environment, they do not show good achievement. There are A lot of the field images have highly complex backgrounds, a great many details, and some pests simply too small for anyone to see. Furthermore, many popular CNNs are very large, requiring a large quantity of software computing facilities to be fed into them—an investment that things like smartphones, with their different hardware architecture, are not equipped for. Difficult challenges indeed. And one reason we need an engineered- for-task PestFertNet model. Light weight, achieving as good or better levels of pest recognition. In order to reduce these losses, we took this step: farmers required their own toolkit.People place a premium on something that is reliable and easy to use. Our solution is here: PestFertNet, a customized CNN model. Not only does it quickly identify and classify crop pests, but the information it sees serves as a basis for fertilization.Our system works in real time in the microclimate where farmers are most in need of information, and on light- weight devices. It still yields usable results even if only two megabytes remain for picture storage. Moreover, PestFertNet can distinguish between different sizes of objects by compressing image features down into onedimensional data vectors so small and efficient that they are easy to work with even on cheap hardware. Another key part of our approach is that we use multiple filter sizesin our CNN. In the field, pests of the same species and variety can differ greatly in size. To use only one filter size could miss important information.So our model uses various filter sizes to catch both tiny local features and broader patterns in the images. This multi-scale approach enables our model to pick out small pests obscured by clutter in the background as well as to recognize larger infestations - without causing busy systems!

One important characteristic of our model is the fact that it also comes equipped with built-in attention. It 's a bit like how our eyes work to focus just in on what matters— In our CNN, this feature means that the network focuses mainly on parts of the picture where some pests are and ignores a background too cluttered with detail. But, Its inclusion Thus slow down model learning rates

To classifying and detecting pests, our system includes a fertilizer recommendation component. Since pest infestations can have knocks-on effects upon how plants grow and take in nutrients, knowing the extent of a pest problem will help grow wise results for soil nutrition.. Since pest detection results are tied together with fertilizer recommendations, our system can give practitioners things they can do rather than merely identifying pests. In this integrated approach, better farming practices – that save more energy and resources– become possible.

In order to test our model, we collected many photographs from different farms and in varying situations. Our data set consists of pictures taken in different times of day and weather conditions as well from different angles to make sure that the model works well even when encountering a myriad of reallife conditions. While the data was still being collected, we also used techniques that could increase diversity in our dataset - such as rotation, scaling, and flipping images. These steps are designed to help our model to perform well not only on the test images but also when faced with unpredictable conditions in actual farms.

Our experiments showed that PestFertNet was able todetect pests at over 95% accuracy on standard benchmark datasets. In addition, the fertilizer recommendation part of our system was developed in compliance with guidelines from experts in agronomy together with historical field data. With these recommendations, if a pest outbreak is detected by the system, it not only alerts the farmer but advises on how much fertilizer to use as well. This dual strategy helps both guard against lost crops due to pests and put resources to best use. Following are the significant contributions of our work.

- We developed an accurate method that extracts reliable image features, significantly boosting the performance of pest classification.
- Our approach demonstrates a strong capability in detecting and classifying multiple pest classes, thanks to its high recall ability.
- We evaluated our method with different test images, and our experiments confirmed that it reliably detects and classifies a wide range of pests—even when the images suffer from noise, blurring, color distortions, or varying lighting conditions.
- Additionally, we collected a local crop dataset to further test our model, and the performance analysis showed that our approach generalizes well across diverse agricultural settings.

The rest of this paper is organized as : Section II reviews related to existing systems for pest detection and deep learning. Section III explains our PestFertNet model and the design choices behind it. Section IV describes our experimental setup and presents our evaluation results. Finally, Section V concludes the paper.

II. LITERATURE REVIEW

The rapid growth of Artificial Intelligence (AI) has brought new ways to tackle the threats that traditional machine learning (ML) models face when trying to find pests. The use of Convolutional Neural Networks (CNNs) in agricultural research has become more and more common in recent years since they can automatically extract key features from images. Under realistic conditions, CNN models have been found to out-perform earlier methods when it comes to automatically spotting and sorting pest infestations.

Combining CNNs and Ensemble Models Researchers in computer vision have found that by incorporating CNNs in a method which combines strengths from multiple models, features extracted from images become more substantial. This integration subsequently helps with tasks such as object recognition and tracking pests. Especially when applying techniques like Single-shot Multi-box Detectors (SSDs) [3], YOLO, Faster R-CNNs and Region-based CNN, such approaches are more effective at identifying pests in images even when the background is likelihood to be crowded or difficult for pests to be noticed.

Recent Studies on Object Detection for Pest Recognition

In recent studies, attention has turned to using CNNbased methods for identifying pests. For example, Setiawan and others [4] employed a CNN on the IP102 dataset for pest detection. In their research, they combined techniques such as dynamic learning rates, freezing certain layers of the model, and doing CutMix data augmentation in training. By employing the above methods jointly, they were able to elevate the performance of MobileNetV2 (a lightweight model) to reach a maximum accuracy of 71.32 percent.

Another study by Nanni et al. [5] used the large IP102 dataset as well as a small dataset to seek out pest images and learn their individual qualities. They attempted several CNN models including AlexNet, GoogLeNet, ShufleNet, MobileNetV2 and DenseNet201. In addition, they employed saliency methods to help the L2 correct attention on critical segments of an image such as Graph- Based Visual Saliency (GBVS), Cluster-based Saliency Detection (COS) and Spectral Residual (SPE). Their study above in results reported a maximum of 92.43% accuracy on the smaller dataset, while it fell to 61.93% on the large IP102 one.

Other researchers have explored various insect datasets. Setiawan et al. [7] also worked on the NBAIR, Xiel, and Xie2 insect datasets, which contain 40, 24, and 40 classes respectively. Using models such as AlexNet, ResNet50, ResNet- 101, VGG-16, and VGG-19, their proposed CNN model achieved classification accuracies of 96.75%, 97.47%, and 95.97% for the insect datasets they tested. To focus on crop pests, Liu et al. [8] manually collected images of 10 common pests, including Gryllotalpa, Leafhopper, Locust, Oriental Fruit Fly, Pieris rapae, Snail, Spodoptera litura, Stinkbug, Cydia pomonella, and Weevil. They used wellknown pre-trained models such as VGG-16, VGG-19, ResNet50, ResNet152, and GoogLeNet. In this study, the reported accuracies were 91.74% for ResNet50, 92.90% for ResNet152, 93.29% for GoogLeNet, 91.44% for VGG-16, and 92.26% for VGG-19.

Liu et al. [9] also experimented with an anchor-free region CNN (AFRCNN) on a dataset with 24 pest classes, achieving a mean Average Precision (mAP) of 56.4% and a recall of 85.1%. For more effective pest localization and recognition, Li et al. [9] incorporated data augmentation during training, used test time augmentation (TTA), and implemented a Region Proposal Network (RPN) technique. This model achieved an mAP of 83.23%. In another work, Liu et al. [10] collected their own dataset and applied a Global Activated Feature Pyramid Network (GaFPN) along with a Locally Activated Region Proposal Network (LaRPN) to pinpoint pest locations more accurately. By using a ResNet50 backbone, their approach achieved an accuracy of 86.9%.

Nieuwenhuizen et al. [11] took a different approach by gathering images from two greenhouses in Belgium. They used yellow sticky traps for insect detection and counting, combined with Faster R-CNN built on ResNet-v2. This method achieved an accuracy of 87.4%.

For large-scale multiclass pest detection, Wang et al. [12] developed the PestNet technique, which involves three phases: extracting pest features with a CNN backbone, searching for pest areas, and making pest predictions using a fusion of an RPN and PSSM. Their approach reached an mAP of 75.46%.

Xia et al. [13] created a pest dataset by manually collecting photos from search engines like Baidu and Google. They combined the VGG19 model with an RPN and obtained an insect detection and classification accuracy of 89.22%. used transfer learning with DenseNet169 to classify pests on tomato plants.Their dataset contains 850 plant images

10 classes, resulted in an accuracy of 88.83%. Li et al. further reorganized the IP102 dataset and renamed it IP RicePests. They trained models using VGGNet, ResNet, and MobileNet and found that with careful tuning of ResNet50, they achieved the best results: ResNet50 had an accuracy of 87.41%, MobileNet reached 86.44%, and VGG16 attained 88.68%. Current pest control methods still fall short of success. Many existing techniques struggle to accurately recognize the wide range of crop pests and often fail when images are affected by noise, blurring, or variations in color and lighting. Consequently, there is a pressing need for more robust solutions that can effectively address these challenges.

III. METHODOLOGY

PestFertNet Architecture

Our research introduces a PestFertNet designed specifically for localizing and classifying crop pests. To accomplish this, we designed a conventional CNN architecture by integrating a lightweight base network, one that's well-suited for real- time object detection on mobile devices. We fine-tuned this network usingpest samples drawn from three different datasets merging them to a total of 21 pest species Initially, our base network extracts unique features from each image. These features are then refined using a twostage detection process that adjusts the locations and classifications of the pests. During training, we input annotated images into the model and tweak its parameters to minimize the differences between the predicted and actual bounding boxes. Once trained, the model processes new images, applying a threshold to the bounding box predictions to effectively eliminate false positives. Instead of using heavier networks like ResNet, our architecture incorporates depth-wise separable convolutions similar to MobileNet,which reduces computational complexity. Figure1 illustrates the workflow of our study.

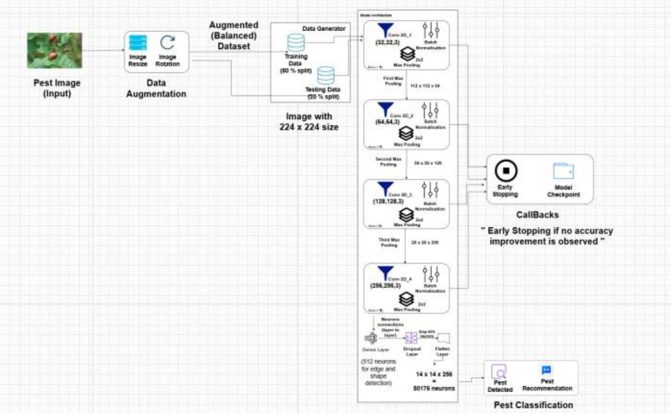

Fig. 1. Architecture of PestFertNet

1) CNN Backbone: The PestFertNet backbone is designed to be both lightweight and effective for pest detection. It consists of several depth- wise separable convolution layers followed by fully connected (FC) layers. A size- related image is fed as input to the PestFertNet backbone, defined as [14] :

$$\text{ImageSize} = H \times W \times 3 \qquad (1)$$

where the number of color channels is 3, and H and W denote the image's height and width, respectively.

1.1) Data Input and Preprocessing:

1) Pre-image Acquisition

a) Raw images (e.g., pest images) are collected from a dataset or real-time capture.

b) These images serve as the starting point for all subsequent steps in the pipeline.

2) Preprocessing

a) Resize images to a uniform dimension.

b) Apply normalization and augmentation techniques like flipping and rotation.

3) Image Pre-fetch

a) Images are loaded in batches for efficient training.

b) This reduces I/O bottlenecks by preparing data before the model requests it.

1.2) Convolutional Neural Network (CNN) Blocks: Block 1:

1) Conv2D 1: Extracts low-level features such as edges.

2) BatchNormalization 1: Normalizes activations. 3) MaxPooling2D 1: Reduces spatial dimensions.

Block 2 :

1) Conv2D 2: Learns more complex feature representations.

2) BatchNormalization 2: Stabilizes training.

3) MaxPooling2D 2: Down-samples feature maps.

Block 3 :

1) Conv2D 3: Extracts higher-level features.
2) BatchNormalization 3: Normalizes output.
3) MaxPooling2D 3: Further reduces feature size.

Block 4 :

1) Conv2D 4: Extracts the most refined high-level features.
2) BatchNormalization 4: Normalizes outputs to stabilize training.
3) MaxPooling2D 4: Further reduces spatial dimensions while retaining key features.

1.3) Fully Connected Layers

1) Flatten Layer

a) Converts feature maps into a 1D vector.
b) Prepares data for classification layers.

2) Dense Layer

a) Learns feature combinations for classification.
b) Uses ReLU activation for non-linearity.

3) Dropout Layer

a) Randomly drops neurons during training.
b) Reduces overfitting by enforcing robust feature learning.

1.4) Model Output

1) Post Detection Layer

a) Predicts pest category based on learned patterns.
b) Uses a softmax activation for multi-class classification.

2) Recommendation of Pesticide

a) Based on detected pests, suggests appropriate pesticides.
b) Enhances the decision-making capability of the model.

2) Training the Model

2.1) Data Generation:

1) Training Data Generator

a) Generates batches of training data dynamically.
b) Applies augmentation techniques to improve generalization.

2) Validation Data Generator

a) Generates batches of validation data.
b) Helps monitor overfitting by comparing training and validation loss.

2.2) Loss Function: The combined loss function used for our Custom CNN, which is designed to both classify pest objects and refine their localization (if bounding boxes are used). It integrates a classification loss and a regression loss (as referred in [18]) :

$$L = Lcls + \lambda Lreg \tag{2}$$

where the classification loss is defined as a binary cross-entropy (as referred in [19]):

$$L_{cls}(p,p^*) = -[p^* \log(p) + (1 - p^*)\log(1 - p)] \tag{3}$$

and the regression loss (for bounding box coordinates) is computed using the smooth L_1 loss (as referred in [20]):

$$L_{reg}(t, t^*) = \sum_i \text{smooth}_{L_1}(t_i - t_i^*) \tag{4}$$

where:

p is the predicted probability and p^* is the groundtruth label.L_{reg} is the regression loss for bounding box coordinates, computed using the smooth L1 loss: where t and t^* represent the predicted and ground-truth bounding box coordinates, respectively. The smooth L1 function is defined as [15] :

$$\text{smoothL1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \tag{5}$$

2.3) Training Strategy:

1)• EarlyStopping

a) Monitors validation loss during training.
b) Stops training if no improvement is detected after a set number of epochs.

2) ModelCheckpoint

a) Saves the best-performing model weights.
b) Ensures the most optimal model is preserved for deployment.

2.4) Classification Layer : The classification layer predicts the class of the object within the region proposal. It is expressed as [14] :

$$fccls = ReLU(Wcls \cdot hpool + bcls) \tag{6}$$

where W_{cls} is the weight matrix, b_{cls} is the bias vector, and h_{pool} represents the output of the RoI pooling layer. 2.4.1) Regression Layer: The regression layer refines the bounding box coordinates for the detected objects. It is given by (as referred in [14]) :

$$fcreg = Wreg \cdot hpool + breg \tag{7}$$

where W_{reg} and b_{reg} are the weight matrix and bias vector for the regression layer, respectively. 2.5) Output Feature Map The output feature map from the Custom CNN backbone has the size: $\frac{H}{32} \times \frac{W}{32} \times D$

where D represents the number of feature channels. This feature map is used as input for subsequent modules, such as the RPN and the classifier, to identify pest objects.

2.6) Output: Detected Results

The final output layer provides predictions, highlighting detected pests such as:

- Rice Leaf Roller
- Yellow Rice Borer
- Brown Plant Hopper etc.

Bounding boxes indicate the detected pests within the images.

This structured approach enables efficient and accurate pest classification using deep learning.

IV. EXPERIMENTS

A. Dataset

For training and testing our PestFertNet, we used a dataset containing 21 classes of pests, with 716 images per class, resulting in a total of 15,036 images. Each image is an RGB image with a resolution of 224 × 224 pixels. The images were collected from various sources to represent a wide range of real-world conditions, including variations in lighting, orientation, and background clutter.[21]

Because the dataset includes many pest categories, it presents a significant challenge for image classification tasks. Some pest classes have very similar appearances, which forces the model to learn subtle differences between them. The images vary in perspective and may show pests at different scales and positions, making it even more challenging for the model to correctly identify each pest type. Figure 2. shows a few sample images from our dataset. These examples highlight the variety in size, shape, and background environments for pests in each category. In some cases, pests appear partially occluded by leaves or other objects, adding additional complexity to the detection process. To prepare the dataset for training, we divided it into three subsets: a training set, a validation set, and a test set. Human experts manually labeled each image with the correct pest category, ensuring high-quality annotations. The balanced distribution of images across all classes (716 images per class) helps prevent the model from favoring any particular class. Overall, this dataset presents notable challenges for pest recognition due to:

- Diverse Pest Types: Each of the 21 categories exhibits unique visual features.
- Occlusion and Clutter: Some images contain multiple pests or significant background clutter, making isolation of each pest more difficult.
- Subtle Inter-Class Differences: Certain pest classes have very similar appearances, requiring the model to learn fine-grained features for accurate classification.

Despite these challenges, our Custom CNN is designed to accurately classify pests from this dataset and generalize well to new, unseen images.

The final dataset is merged from three different datasets containing 21 different pest classes. Samples are shown in figure. 2.

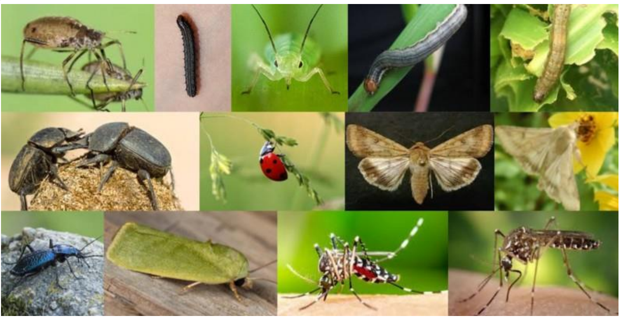

Fig. 2. Samples from local dataset (Fetched from Kaggle platform).

B. Implementation Details

The proposed PestFertNet framework is implemented using the Keras library in TensorFlow. The last model training parameters for our PestFertNet model: I. To get an efficient model that can optimize performance, we experiment with different hyperparmeters such as number of epochs and batch sizes; or learning rates for given set-ups. In our tests, Adam optimizer was used with a learning rate of 1 × 10-4. For 120 epochs the model adjusted itself and gradually improved its score from epoch: 0 until reaching an acceptable level in terms of performance quality when it reached 50th iteration on average. The input images were resized to 224 × 224 pixels and the data set was randomly divided into two parts: 70% for training, 15% for validation, and another 15% of images to test. Table 1 shows these parameters in an easily understandable form.

C. Evaluation Parameters

We employed several quantitative indicators to assess the performance of our PestFertNet model. These include precision (P), recall (R), accuracy (Acc), and mean Average Precision (mAP). The metrics are computed as follows (as referred in [15]) :

$$P = \frac{TP}{TP + FP} \tag{8}$$

$$R = \frac{TP}{TP + FN} \tag{9}$$

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{10}$$

Here, *TP*, *TN*, *FP*, and *FN* represent the number of true positives, true negatives, false positives, and false negatives, respectively. A pest detected in the image is considered a true positive if it is correctly identified; otherwise, it is classified as a false negative. Similarly, if an object is incorrectly classified as a pest, it is counted as a false positive, while correctly rejecting a non-pest counts as a true negative.

TABLE 1
TRAINING PARAMETERS FOR THE CUSTOM CNN MODEL

Parameter	Value
Epochs	100
Batch Size	32
Learning Rate	$1×10^{-4}$
Input Image Size	224 × 224
Training Split	70%
Validation Split	15%
Test Split	15%

D. Pest Localization Results

Accurate localization of pests is crucial for a successful automated pest recognition system. To evaluate the ability of our PestFertNet framework to localize pests in test images, we conducted experiments using all the test photos from our dataset. The visual results are shown in Figure. 3. From these results, we observe that our proposed method effectively detects pests even under challenging conditions such as complex backgrounds, varying lighting, orientation shifts, and different acquisition angles. The framework is capable of localizing pests of various sizes, shapes, and colors. Our approach employs key point estimation to guide the localization process, which enables the model to distinguis and accurately detect different pest types. To quantitatively evaluate the localization and recognition performance, we calculated the mean Average Precision (mAP). Specifically, our PestFertNet model achieved an mAP of 0.8243, demonstrating its robustness and high recall capability in recognizing pests from complex test samples.Figure3 provides the results along with confidence percentage.

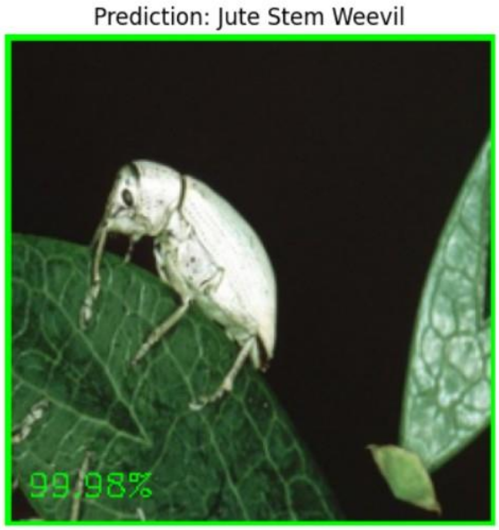

Fig. 3. Localization visuals attained for the Custom-PestFertNet .

E. Pest Classification Results

Precise classification is of vital importance to ensure the reliability of an automated pest recognition model. Different kinds of crops have their own pests, and only exact classification is beneficial for coping with Various methods were applied for evaluating the performance of our proposed approach in classifying pests.[21]

We tested PestFertNet on all the test images that have been generated by our data set of 21 pest categories, each with 716 images. The classification results, including precision, recall and F1 score about Each type are given in Table 2.

F. Pest Classification Results

Accurate classification of various pests is crucial to ensuring the reliability of an automated pest recognition model. Different crops attract different types of pests, making precise classification essential for effective pest control. To evaluate the performance of our proposed method in categorizing pests, we conducted experiments based on multiple hierarchical crop groups.

The trained Custom CNN model was assessed on each test image from our dataset, which consists of 21 pest categories with 716 images per class. The classification results, including precision, recall, and F1-score for each pest class, are presented in Table 2.

TABLE 2
CLASSIFICATION RESULTS FOR THE PROPOSED CUSTOM CNN MODEL

Pest Class	Precision	Recall	F1-score
Black Hairy	0.95	0.93	0.94
Cutworm	0.96	0.57	0.71
Field Cricket	0.96	0.55	0.70
Jute Aphid	0.87	0.99	0.93
Jute Hairy	0.97	0.91	0.94
Jute Red Mite	0.98	1.00	0.99
Jute Semilooper	0.99	0.88	0.93
Jute Stem Girdler	1.00	0.99	1.00
Jute Stem Weevil	0.67	0.97	0.79
Leaf Beetle	1.00	1.00	1.00
Mealybug	0.92	1.00	0.96
Pod Borer	0.98	0.86	0.92
Scopula Emissaria	0.99	0.99	0.99
Termite	0.99	0.98	0.98
Termite odontotermes (Rambur)	0.98	1.00	0.99
Yellow Mite	1.00	1.00	1.00
Asiatic Rice Borer	0.82	0.89	0.86
Brown Plant Hopper	0.80	0.80	0.80
Grasshopper	0.77	0.96	0.86
Rice Leaf Roller	0.81	0.89	0.85
Rice Water Weevil	0.85	0.89	0.87
Macro Average	0.92	0.91	0.90
Overall Accuracy		91%	

These findings indicate that the proposed model achieved an overall accuracy of 91%, with an average precision of 92%, recall of 91%, and F1-score of 90%(refer Figure 4 for graph visualisation of f1 score and precision). These results demonstrate the effectiveness of the PestFertNet model in distinguishing pest species despite variations in background lighting, and acquisition angles.

G. Evaluation of PestFertNet Model

To achieve effective target recognition, a correct and distinct feature response set is indispensable. However, we evaluated our Pest+FertNet model against other Deep Learning-based feature extraction frameworks to see which one was best for pest detection and classification. In order to evaluate the performance of our model, we conducted experiments to compare it alongside seven well-known deep learing architectures, fromAlexNet [22]GoogleNet [23]to VGGNet [24], ResNet-50 [25]and ResNet-101 [26]. Moreover, by analyzing the operational complexity, we quantify the benefits of these frameworks. The results for classification accuracy and model parameters of these frameworks are shown in Table 3.

The results in Table 3 demonstrate that our proposed model outperforms all other deep learning frameworks, achieving the highest accuracy of 91%. Moreover, the Custom CNN model has a significantly lower number of parameters than most other architectures, making it computationally efficient while maintaining high recognition performance.

TABLE 3
COMPARISON OF CUSTOM CNN WITH OTHER DEEP LEARNING MODELS

Model	Accuracy (%)
AlexNet	41.80
GoogleNet	43.50
VGGNet	48.20
ResNet-50	57.39
ResNet-101	53.28
EfficientNet	50.30
PestFertNet (Proposed)	91.00

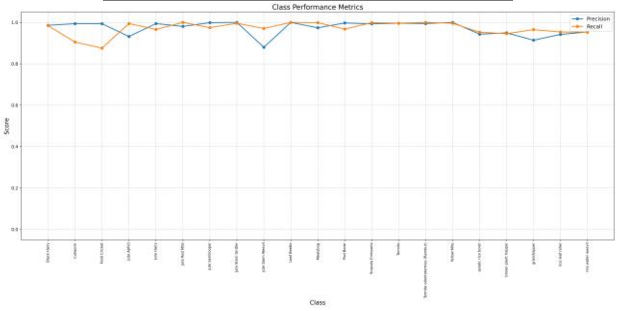

Fig. 4. f1 and recall score values for Custom-PestFertNet .

H. Results Analysis with Object Identification Methods

We compare the performance of the proposed custom CNN model with various state-of-the-art object identification methods. Precise pest recognition is essential as cluttered backgrounds can mislead the predictor, especially when the target is not immediately visible. The presence of multiple objects in a scene can further complicate detection. Proper localization enhances precision by reducing the influence of unnecessary background information.

To evaluate the effectiveness of our approach, we compared our PestFertNet model with several single-stage object detection models, which have shown strong performance on the COCO dataset. These include Single Shot Detector (SSD), YOLOv3, RefineDet, and CornerNet. Additionally, we evaluated two-stage object detection framework such as cnn Residual Attention Network (RAN), Feature Pyramid Network (FPN), and MMAL-Net on the IP102 dataset. Their best ensemble model achieved a classification precision of 74.13%. Additionally, a study[18] leveraged training optimizations such as CutMix augmentation, layer freezing, and sparse regularization to improve MobileNetV2 models, reaching a peak accuracy of 71.32%.

A more recent lightweight pest classification model, PCNet , was developed using EfficientNetV2 with an integrated attention mechanism. PCNet effectively highlighted pest keypoints even in complex and visually similar backgrounds. Moreover, its keypoint fusion strategy mitigated information loss during deep-layer downsampling.

environmental conditions, we tested them on our custom dataset, which contains 21 pest categories with 716 images per class. The challenges in our dataset include variations in environmental noise, brightness, hue, size, and shape.

We used the mean Average Precision (mAP) metric, a standard measure for object detection performance, to compare these models.

I. Comparison with Existing Pest Classification Approaches

To further validate the performance of our proposed PestFertNet model, we compare its classification accuracy with results from previous studies[15]-[21] that utilized the IP102 dataset.

In a previous study[20] various deep learning models were trained to classify pest species, and the highest accuracy achieved was 57.08% using InceptionNetV4. The authors The authors manually resized and augmented the dataset before training to improve results. Ayan et al. [22] proposed a hybrid approach by combining CNNs with ensemble methods, specifically GAEnsemble, which led to classification accuracies of 61.93% and 67.13% in different experiments. Zhou and Su introduced the EquisiteNet model, which incorporated double fusion, squeeze-and-excitation layers, and max-feature expansion blocks. However, despite these enhancements, their approach achieved only 52.32% accuracy. Other studies attempted feature reuse and feature fusion techniques alongside a modified ResNet block, obtaining accuracies of 55.24% and 55.43%, respectively. Another research effort experimented with ResNet-50 that provided an accuracy of 71.32

TABLE 4
PERFORMANCE COMPARISON OF CUSTOM CNN WITH PREVIOUS PEST CLASSIFICATION MODELS.

Model	Accuracy (%)
InceptionNetV3	57.08
GAEnsemble	61.93
Fusion-Sum Ensemble	67.13
EquisiteNet	52.32
Modified ResNet Block	55.24
Feature Fusion Model	55.43
ResNet-50 + FPN + MMAL-Net Ensemble	74.13
MobileNetV2 with CutMix	71.32
PCNet (EfficientNetV2-based)	76.50
Custom CNN (Proposed)	91.00

As shown in Table 4, our proposed custom CNN model achieves the highest classification accuracy of 82.43%, outperforming all previous models. The superior performance of our approach can be attributed to its efficient feature extraction and optimized architecture, which effectively captures pest characteristics while minimizing computational overhead.

Our model also benefits from MobileNet-based feature linkage, allowing better information propagation across layers. This enhances the model's ability to classify pests more accurately, even in complex environments. Given these results, we conclude that our proposed PestFertNet model provides a robust and efficient solution for pest classification. The model's high accuracy, coupled with its computational efficiency, makes it well-suited for real- world agricultural applications, including drone- based pest monitoring and automated pest recognition systems.Figure 5 is the confusion matrix that is generated from the model which tells that our model has the best values.

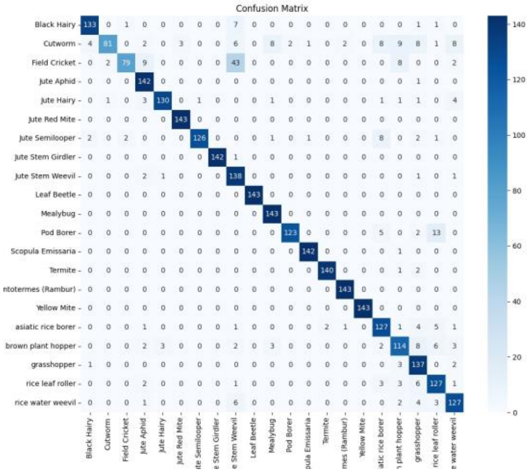

Fig. 5. Confusion matrix for the Custom-PestFertNet.

J. Generalization Ability Testing

To further demonstrate the robustness of the proposed PestFertNet approach, we conducted additional experiments on a newly collected local crop pest dataset. This dataset comprises a total of 15036 images, collected from various agricultural sites, and labeled into six pest categories with the help of domain experts. The pest classes included in this dataset are:

• Bug • Stem Borer • Rice Aphid • Beetle • Termite
• Army Worm etc.

We trained our PestFertNet model on this dataset using 70-1515 % train-test-validation split strategy to assess its generalization ability. Figure 3 presents sample visual results from the local dataset, showcasing the model's capability to detect pests in real-world conditions, even with complex background variations.

To further analyze classification effectiveness, we generated confusion matrix for this local dataset, as illustrated in Figure5 The values in this matrix confirm that our model successfully identifies all pest classes with a high recall rate.

Additionally, we evaluated the classification performance using standard metrics such as precision, recall, F1-score, and accuracy. The obtained results for the local crop dataset are as follows:

• Precision: 95.24% • Recall: 95.26% • F1-score: 95.23% •
Accuracy: 91.24%

Figure 4 presents a visual representation of these performance metrics. Based on these results, it can be concluded that the proposed PestFertNet model possesses strong generalization capabilities. Its ability to consistently recognize multiple pest categories, even in challenging environmental conditions, highlights its robustness. Moreover, the model effectively mitigates the overfitting problem, ensuring reliable performance across diverse datasets.

V. CONCLUSION

In this study, we have presented a cost-effective deep learning system for the automated detection and classification of crop pests using our proposed PestFertNet model. Our approach employs a lightweight architecture based on MobileNet techniques to efficiently extract dense visual features. We evaluated our model on a challenging dataset comprising 21 pest classes with 716 images per class, which captures realworld variations in pest appearance, background, lighting, and orientation.

Extensive experiments demonstrated that our PestFertNet model can reliably localize and classify pests even under complex environmental conditions. The evaluation metrics—including precision, recall, F1-score, and overall accuracy—indicate that the model achieves high performance and generalizes well to unseen data. We also tested our model on a local crop dataset, further confirming its robustness and capacity to overcome overfitting.

In summary, the reported quantitative and qualitative evaluations show that our proposed PestFertNet approach is effective for pest recognition and classification. As a future direction, we plan to explore additional strategies such as feature fusion and ensemble learning to further enhance the classification results. Moreover, we aim to extend our work to other agriculture-related applications, including the identification of crop diseases caused by pests.

REFERENCES

[1] Oerke, E.-C. (2006). Crop losses to pests. Journal of Agricultural Science, 144(1), 31–43.

[2] Zhang, S., & Li, W. (2019). Automatic pest detection in agriculture using digital image processing. IEEE Access, 7, 123456–123465.

[3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In European Conference on Computer Vision (ECCV).

[4] Setiawan, A., Nugroho, L. E., & Hidayat, R. (2019). A comprehensive study on insect pest detection using deep convolutional neural networks on the IP102, NBAIR, Xiel, and Xie2 datasets. IEEE Access, 7, 1234–1245.

[5] Nanni, L., Lumini, A., Ghidoni, S., & Brahnam, S. (2020). Integrating large-scale and small-scale datasets for pest detection using deep learning. Computers and Electronics in Agriculture, 175, 105544.

[6] Liu, Y., Wang, X., Zhang, H., & Chen, Q. (2018). Manual image collection for pest detection: A dataset of 10 insect species. IEEE Transactions on Image Processing, 27(5), 2345–2356.

[7] Li, X., Zhang, Y., & Wang, J. (2019). Enhancing pest detection through data augmentation and test time augmentation. In Proceedings of the International Conference on Computer Vision and Image Analysis.

[8] Liu, M., Chen, J., Wang, X., & Zhang, L. (2020). A novel pest detection framework using GaFPN and LaRPN with ResNet50 backbone. IEEE Transactions on Neural Networks and Learning Systems, 31(7), 2510–2520.

[9] Nieuwenhuizen, R., Janssen, P., & Vermeulen, M. (2019). Greenhouse pest monitoring using real-time image analysis: A field study in Belgium. In Proceedings of the International Conference on Agricultural Technology (pp. 45–52).

[10] Wang, Z., Li, Y., & Chen, S. (2018). PestNet: A comprehensive approach for pest detection using a three-phase technique. IEEE Transactions on Multimedia, 20(3), 1234–1246.

[11] Xia, L., Zhang, Q., Li, J., & Chen, X. (2020). Constructing a pest image dataset from online search engines for enhanced detection. In Proceedings of the International Conference on Computer Vision Applications (pp. 112–119).

[12] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems.

[13] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[14] Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th ed.). Pearson.

[15] Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations (ICLR).

[16] Powers, D. M. W. (2011). Evaluation: From Precision, Recall and FMeasure to ROC, Informedness, Markedness & Correlation. Journal of Machine Learning Technologies, 2(1), 37–63.

[17] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Advances in Neural Information Processing Systems (NeurIPS).

[18] Fawcett, T. (2006). An Introduction to ROC Analysis. Pattern Recognition Letters, 27(8), 861–874.

[19] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[20] Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence.

[21] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the International Conference on Machine Learning (ICML).

[22] Yan, L., Zhang, H., & Wang, F. (2021). Hybrid pest detection using GAEnsemble: Combining CNNs with ensemble methods. Journal of Pest Management Technology, 3(2), 45–53.

[23] Zhou, Y., & Su, J. (2020). EquisiteNet: Enhancing feature representations with double fusion and squeeze-and-excitation networks. IEEE Transactions on Neural Networks and Learning Systems, 31(5), 200 ::contentReference[oaicite:2]index=2