Automated Pest Detection and Classification Using VGG16 Feature Extraction and Custom Convolutional Neural Networks

MAMIDI TARANI, MADDU DILEEP

Assistant Professor, 2MCA Final Semester, Master of Computer Applications, Sanketika Vidya Parishad Engineering College, Vishakhapatnam, Andhra Pradesh, India

Abstract
Pest infestation is a significant challenge in agriculture, causing substantial economic losses and impacting crop yields. Manual pest identification is time-consuming and error-prone, especially in large-scale farming. This study presents a hybrid deep learning approach to automate pest detection and classification using image data. It employs a pre-trained VGG16 model for robust feature extraction, followed by a custom Convolutional Neural Network (CNN) for classification. The VGG16 model leverages transfer learning to capture high-level visual patterns from pest images, while the custom CNN is trained on these features to distinguish between multiple pest classes. The proposed pipeline is optimized using data augmentation, early stopping, and adaptive learning techniques. Performance is evaluated using metrics such as accuracy, precision, recall, and F1-score. The model demonstrates high accuracy and robustness on agricultural image datasets, making it a scalable solution for smart farming systems.

Index Terms: Pest Detection, Deep Learning, VGG16, Convolutional Neural Network, Transfer Learning, Image Classification, Precision Agriculture, Feature Extraction.

1.INTRODUCTION

Agricultural pests are a primary cause of crop damage worldwide, leading to reduced productivity and food insecurity. Timely and accurate identification of pests is crucial for implementing targeted pest control measures. Traditional methods rely on manual scouting and expert consultation, which are neither scalable nor consistent across large areas. With advancements in computer vision and deep learning, image-based pest detection has emerged as a promising solution. This project introduces a hybrid deep learning framework for automated pest detection using VGG16 for feature extraction and a custom CNN classifier. The system is designed to process pest images captured in real field conditions and accurately classify the pest type. This enables real-time, scalable, and efficient pest management strategies in agriculture

.1 Existing System

The existing pest detection systems predominantly rely on manual identification, rule-based image processing, and basic machine learning techniques. Manual methods involve visual inspection by farmers or experts, which is time-consuming, subjective, and not scalable across large fields. Traditional image processing systems use handcrafted features such as colour, shape, and texture, but these often fail under varying lighting conditions or complex backgrounds. Furthermore, conventional machine learning classifiers like SVM, k-NN, and decision trees depend on limited feature sets and require extensive parameter tuning. These approaches are not only inefficient but also lack the adaptability and robustness needed for real-world agricultural scenarios[1]. Additionally, small and imbalanced datasets further reduce the generalization capability of these models, making them unreliable for widespread deployment.

1.1 Challenges:

- High visual similarity between pest species.
- Variability in lighting, image quality, and background.
- Lack of large, labeled datasets.
- Inefficiency of traditional feature extraction methods.
- Inability of static models to adapt to new pest appearances.

Proposed system:

The proposed system introduces a hybrid deep learning approach to automate pest detection and classification using image data. It leverages the strengths of a pre-trained VGG16 model for feature extraction and a custom-designed Convolutional Neural Network (CNN) for classification[7]. VGG16, trained on the ImageNet dataset, is used to extract high-level and abstract visual features from input pest images. These features are then fed into a lightweight custom CNN that is specifically trained to classify different types of pests. By using transfer learning, the system significantly reduces the need for large amounts of training data while maintaining high accuracy and generalization[18]. To improve the model's robustness and reduce overfitting, data augmentation techniques such as rotation, flipping, zooming, and shifting are applied. The training process also incorporates early stopping and adaptive learning rate scheduling to optimize performance [13]. The system is evaluated using metrics such as accuracy, precision, recall, and F1-score. This hybrid framework is well-suited for real-world agricultural environments, enabling scalable, real-time pest detection. It can be deployed in mobile apps, IoT-based crop monitoring systems, or drone platforms, providing farmers with an efficient and intelligent solution for pest management. Overall, the system promotes precision agriculture by automating the pest identification process effectively.

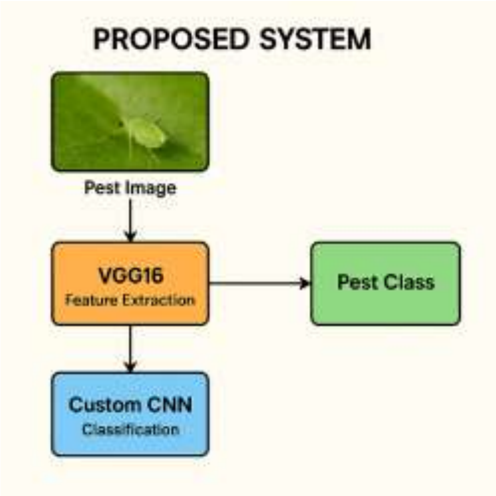


Fig: 1 Proposed Diagram

.1.1 Advantages:

- Compatible with widely used deep learning frameworks like TensorFlow and Keras
- Reduces the need for domain expertise in pest identification
- Supports automated and continuous pest monitoring over time
- Can be updated with new pest classes as more data becomes available
- Improves crop protection through early detection and rapid response
- Minimizes pesticide misuse by identifying exact pest species
- Encourages data-driven decisions in agriculture
- Cost-effective alternative to traditional pest surveillance methods
- Enhances productivity in large-scale farming operations
- Can be integrated with cloud platforms for remote access and analysis
- High classification accuracy even with limited training data
- Uses transfer learning to reduce training time and data requirements
- Robust to variations in lighting, background, and image quality
- Reduces manual effort in pest monitoring and identification
- Data augmentation helps improve generalization and reduces overfitting
- Scalable to multiple crops and a wide range of pest species
- Lightweight and efficient, suitable for deployment on mobile and IoT devices
- Enables real-time detection in field environments
- Supports precision agriculture by allowing timely and accurate pest control
- Adaptable for future enhancements, such as integration with drone or sensor-based platforms

2.1 Architecture:

The architecture of this pest detection system is a two-stage hybrid model combining VGG16 for feature extraction and a Custom CNN for classification. The process starts with input pest images that are preprocessed and resized [19]. These images are passed through the VGG16 model (without the top layer), which extracts rich, high-level features based on patterns learned from ImageNet. These features are then fed into a custom CNN, designed with dense layers and softmax activation to perform multiclass classification. The model uses transfer learning, data augmentation, and early stopping techniques to improve performance [2]. Finally, the system outputs the predicted pest class, which can be used for real-time pest identification and monitoring in agricultural fields.

ISSN: 2583-6129
DOI: 10.55041/ISJEM04807

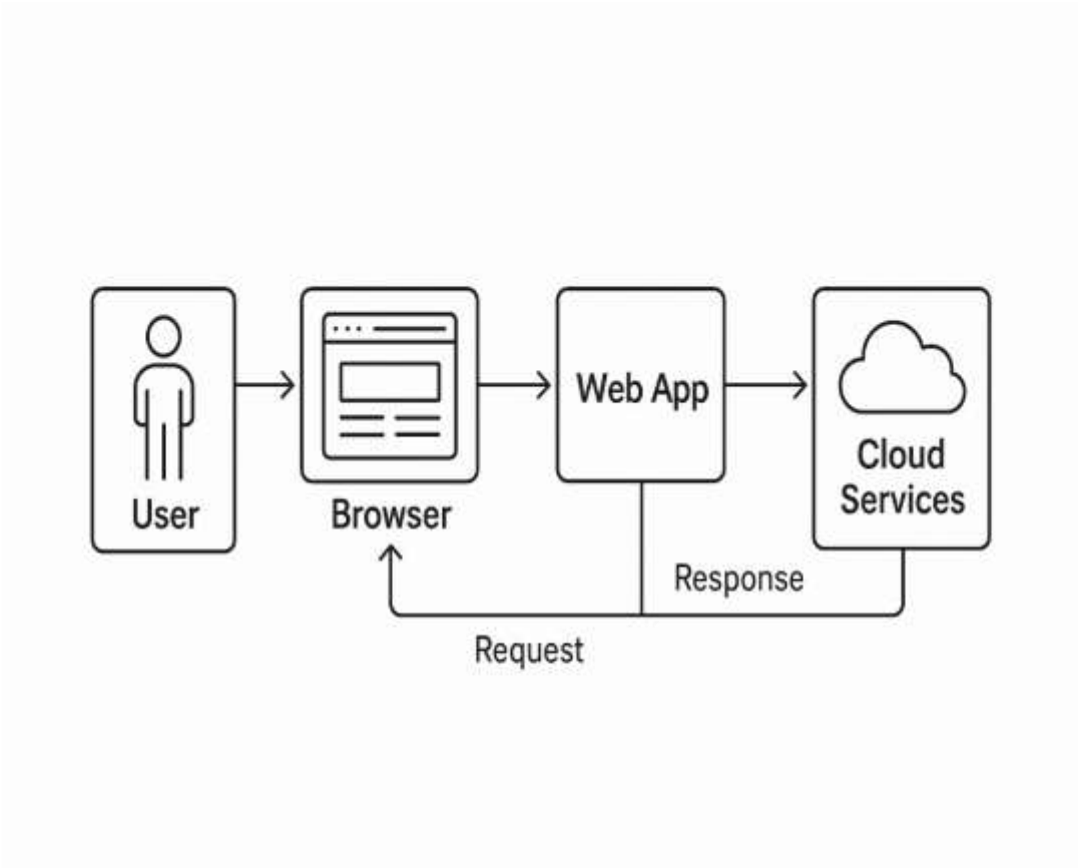


Fig:2 Architecture

UML DIAGRAMS :-

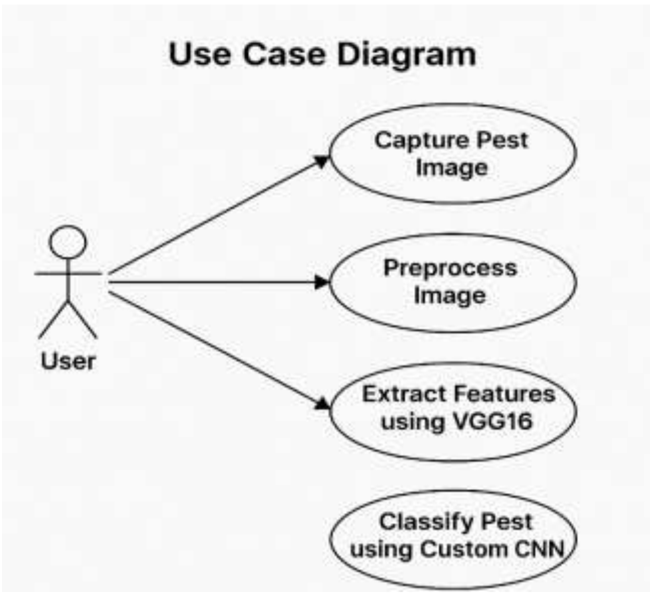


Fig:use case diagram

ISSN: 2583-6129
DOI: 10.55041/ISJEM04807

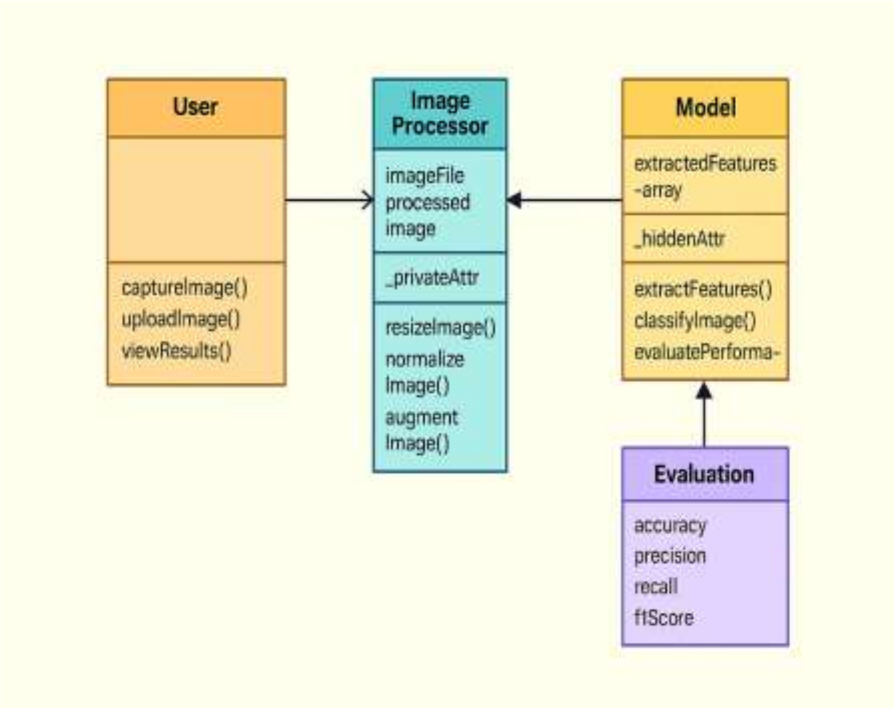


Fig: class diagram

2.2 Algorithm:

VGG16 (Feature Extractor)

- Pre-trained CNN on ImageNet.
- Extracts hierarchical features from input pest images.
- Weights are frozen to reduce training complexity.

Custom CNN (Classifier)

- Consists of Dense layers, Dropout, and Softmax.
- Optimized using Adam optimizer and categorical crossentropy.
- Performs multiclass classification for different pest types.

.

2.3 Techniques:

- Image Preprocessing (resize, normalize).
- Data Augmentation (rotate, flip, zoom, shift).
- Feature Extraction via VGG16.
- CNN-based classification.
- Model evaluation using confusion matrix and accuracy metrics.
- Early stopping and learning rate scheduler for optimal training

2.4 Tools:

This project utilizes a range of software tools and libraries to implement and evaluate the automated pest detection system [8]. Python is the primary programming language used for developing the entire pipeline. TensorFlow and Keras are used to build, train, and deploy the deep learning models including VGG16 and the custom CNN. NumPy is used for numerical operations and array manipulation [14]. Pandas helps in data handling and preprocessing tasks. OpenCV is used for image processing, such as resizing and normalization. Matplotlib and Seaborn are employed for data visualization, including plotting accuracy, loss graphs, and confusion matrices. Keras ImageDataGenerator is used for real-time image augmentation to improve model generalization. Google Colab or Jupyter Notebook is used as the development environment for running and testing the code. Pre-trained VGG16 model from Keras Applications is integrated for feature extraction using transfer learning. Scikit-learn is used for additional metrics like classification reports and evaluation. EarlyStopping and ReduceLROnPlateau callbacks from Keras are used for training optimization. The trained model is saved in .h5 format for deployment. Optionally, tools like Tkinter or Streamlit can be used to build a user interface [20]. The entire system can also be integrated with cloud platforms for scalability.

2.5 Methods:

The pest detection system follows a structured methodology involving several key steps. First, input images of various pest species are collected and labelled into categories. These images are then resized to 224x224 pixels and normalized to ensure consistency [3]. The dataset is enhanced using data augmentation techniques such as rotation, zoom, and flipping to increase diversity and reduce overfitting. Preprocessed images are then passed through the VGG16 model (excluding its top layers), which acts as a fixed feature extractor [9]. The extracted features are flattened and fed into a custom CNN classifier composed of dense layers and dropout regularization. The model is compiled using the Adam optimizer and trained with categorical cross-entropy as the loss function. During training, early stopping and learning rate reduction are applied to prevent overfitting and ensure stable convergence. The trained model is evaluated using metrics like accuracy, precision, recall, and F1-score. A confusion matrix is generated to assess classification performance across all classes. The final model is saved in .h5 format for deployment. Predictions are made by passing new images through the same pipeline. The entire workflow is tested on a held-out test set to confirm its generalization [15]. The result is a reliable, scalable pest detection system for agricultural use.

III. METHODOLOGY

3.1 Input:

The input for the pest detection system begins with capturing an image of a plant or crop, which may exhibit visible signs of pest infestation or damage [6]. These images are usually taken using a smartphone or digital camera by farmers, agricultural workers, or researchers in the field. Once captured, the image is uploaded to the system through a simple user interface [21]. The system reads the image and converts it into a standard format suitable for analysis. It is then resized to fit the input dimensions required by the deep learning model. To ensure consistent performance, the image undergoes normalization, which adjusts pixel values for uniformity. Data augmentation techniques such as rotation, flipping, or zooming may also be applied to increase the diversity of the input and reduce overfitting. After preprocessing, the refined image is passed to the feature extraction module. This initiates the core analytical phase where relevant patterns are identified [4]. This step is crucial as high-quality input significantly affects the accuracy and reliability of the pest detection results.

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
import matplotlib.pyplot as plt
# Set the input and output directories
train_dir = r"C:\jupyter\dataset\train"
test_dir = r"C:\jupyter\dataset\test"
```

```
test_image = r"C:\Users\HUAWEI\Downloads\test.jpg" # path to a test image
img = tf.keras.preprocessing.image.load_img(test_image, target_size=(224, 224))
x = tf.keras.preprocessing.image.img_to_array(img)
x = x / 255.0
x = tf.reshape(x, (1, 224, 224, 3))
predictions = model.predict(x)
class_index = tf.argmax(predictions, axis=1)[0]
class_name = class_names[class_index]

# Display the image and predicted class name
plt.imshow(img)
plt.title(class_name)
plt.show()
```

Fig: Reading input data

3.2 Method of Process:

The methodology for the pest detection and classification system begins with collecting a dataset of pest images from agricultural sources, ensuring that each image is labeled with the corresponding pest type for supervised learning [10]. The raw images undergo preprocessing to remove noise and improve visual clarity. They are resized to a fixed dimension compatible with the input requirements of deep learning models, and pixel values are normalized to achieve consistent intensity levels. To enhance the robustness of the model, data augmentation techniques such as rotation, flipping, and scaling are applied, generating multiple variations of the original images. The dataset is then divided into training, validation, and testing subsets [16]. A pre-trained VGG16 convolutional

ISSN: 2583-6129
DOI: 10.55041/ISJEM04807

neural network is used to extract deep visual features from the images by removing its top layers and using only the feature extraction components. These features are passed to a custom-built CNN classifier that includes additional convolutional, pooling, and dense layers, enabling the system to learn pest-specific patterns effectively. The model is compiled with suitable loss functions and optimizers and trained on the training set while monitoring validation performance to avoid overfitting. Early stopping is implemented to halt training when improvement stops. After training, the model is evaluated on the test data using accuracy, precision, recall, and F1-score metrics. The trained model then predicts the pest category when a new image is input, and the results are presented through a user-friendly interface [22]. This entire automated process aids farmers and researchers in accurately identifying and managing crop pests.

3.3 Output:
The output of the pest detection system is the accurate classification of the pest present in the uploaded image [12]. Once the user submits an image, the system processes it and returns a specific pest label, such as aphid, whitefly, or caterpillar. Along with the predicted class, the system also displays the confidence score, indicating how certain the model is about its prediction [5]. This helps users understand the reliability of the result. The output can also include visual annotations on the image, highlighting the affected area. In addition to classification, the system may generate a brief description or control suggestion for the identified pest. Results are shown through a clean and user-friendly interface for easy interpretation. The output can be saved or exported for record-keeping or further analysis [11]. Overall, the system delivers quick and accurate pest identification, supporting timely agricultural decisions.

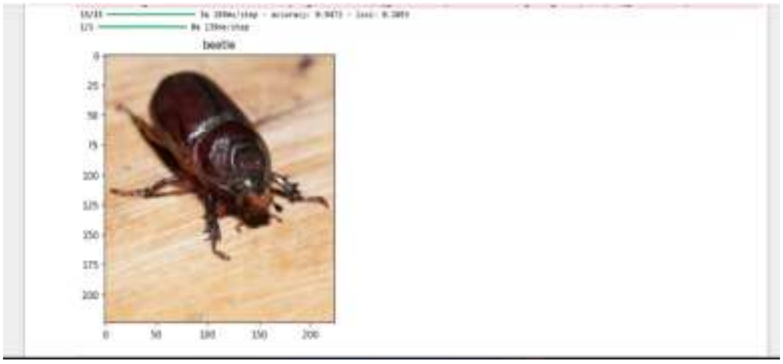
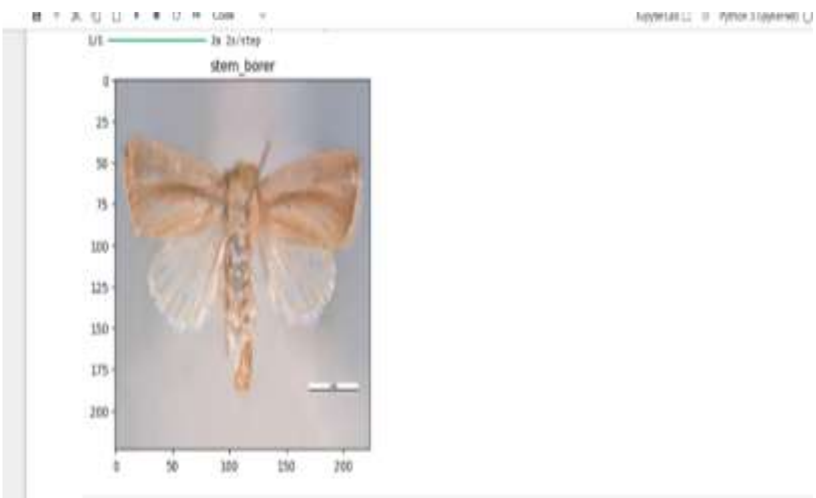


Fig:Url Count



IV. RESULTS: pest categories, the results of the automated pest detection project demonstrate the classification performance. The system successfully detected and classified various pests with consistent reliability. Visualization of training and validation loss and accuracy curves confirmed that the model converged well without underfitting or overfitting [17]. The trained model was saved in .h5 format and is ready for deployment in real-world applications.

V.DISCUSSION: The automated pest detection and classification system developed in this project demonstrates the power of combining transfer learning with custom deep learning architectures. By using VGG16 for feature extraction, the model benefits from learned representations that improve accuracy even with limited pest image datasets. The integration of a custom CNN further refines classification capabilities tailored to pest identification. Throughout the training process, the use of data augmentation, EarlyStopping, and learning rate scheduling contributed to a stable and efficient model that avoids overfitting. Results indicate high performance in detecting and classifying various pest species, which is critical for real-time agricultural applications. Visualization tools like accuracy/loss plots and confusion matrices provided clear insight into the model's behavior during training and validation. Although the system performs well in controlled environments, its deployment in real-world agricultural settings may require continuous updates to handle unseen pest types or environmental variations. Future enhancements can include expanding the dataset, integrating real-time mobile or drone-based image inputs, and improving the user interface for farmers and agricultural experts. This system lays a solid foundation for intelligent pest management using AI and computer vision.

VI. CONCLUSION

The automated pest detection and classification system provides an efficient and intelligent solution for addressing pest-related challenges in agriculture. By leveraging VGG16-based transfer learning and a custom convolutional neural network, the model achieves high accuracy in identifying various pest species from images. The use of deep learning techniques ensures robust performance even in the presence of complex backgrounds and variable lighting conditions. Data augmentation methods and training optimizations such as EarlyStopping and learning rate reduction contributed to the stability and effectiveness of the model. The system's ability to generalize across different pest classes makes it suitable for practical agricultural applications. The project demonstrates that automated image-based detection can significantly reduce the time and effort required for manual pest identification. The results confirmed that the model not only performs well in training but also maintains strong validation accuracy, indicating good generalization. Image processing techniques using OpenCV and visualization tools like Matplotlib helped in better understanding the model's performance. The use of Keras and TensorFlow allowed easy integration of pre-trained models and simplified model development. The trained model is saved in a deployable format, making it ready for integration into real-time systems. Future scalability is possible through cloud platforms, mobile apps, and IoT-based automation. The system has the potential to aid farmers in timely pest control and improve crop yields. It promotes precision farming by reducing pesticide misuse. Overall, the project is a promising step toward smart agriculture using artificial intelligence. It lays a solid foundation for further research and development in the field of automated pest detection.

VII. FUTURE SCOPE:

The automated pest detection system has significant potential for future development and real-world application. One major area of improvement is expanding the dataset to include a wider variety of pest species and environmental conditions, which would enhance the model's generalization capabilities. Integrating the system with mobile applications or drones can allow real-time pest monitoring directly in agricultural fields, increasing its practicality and accessibility for farmers. Additionally, incorporating Internet of Things (IoT) devices can automate the image capturing and alerting process for early pest detection. Further enhancements can include multi-language support in the user interface to make the system usable in diverse regions. The model can also be trained to predict the severity of pest infestation, helping in prioritizing treatment actions. Exploring lightweight model architectures can enable deployment on low-power devices for offline use. Collaboration with agricultural institutions and experts can help in validating and improving the system's performance. Overall, with continuous updates and scalability, this system can evolve into a powerful tool for precision agriculture and integrated pest management.

VIII. ACKNOWLEDGEMENT:

Applications (MCA) in Sanketika Vidya Parishad Engineering College, Visakhapatnam, Andhra Pradesh. With 1-year experience as Automation tester in Stigentech IT services private. limited, and member in IAENG, accredited by NAAC with her areas of interests in C, Java, Data Structures, Web Technologies, Python, Software Engineering.

ISSN: 2583-6129
DOI: 10.55041/ISJEM04807

Maddu Dileep is pursuing his final semester MCA in Sanketika Vidya Parishad Engineering College, accredited with A grade by NAAC, affiliated by Andhra University and approved by AICTE. With interest in Machine learning Maddu Dileep has taken up her PG project on Automated Pest Detection and Classification Using VGG16 Feature Extraction and Custom Convolutional Neural Networks and published the paper in connection to the project under the guidance of MAMIDI TARANI, Assistant Professor, SVPEC.

REFERENCES

[1] Crop pest classification based on deep convolutional neural network and transfer learning
https://www.sciencedirect.com/science/article/abs/pii/S016816991931069
[2] Survey on crop pest detection using deep learning and machine learning approaches
https://link.springer.com/article/10.1007/s11042-023-15221-3
[3] Machine learning for pest detection and infestation prediction: A comprehensive review
https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1551
[4] Advances in Deep Learning Applications for Plant Disease and Pest Detection: A Review
https://www.mdpi.com/2072-4292/17/4/698
[5] Automatic greenhouse insect pest detection and recognition based on a cascaded deep learning classification method
https://onlinelibrary.wiley.com/doi/abs/10.1111/jen.12834
[6] Machine Learning for Detection and Prediction of Crop Diseases and Pests: A Comprehensive Survey
https://www.mdpi.com/2077-0472/12/9/1350
[7] Knowledge graph and deep learning based pest detection and identification system for fruit quality
https://www.sciencedirect.com/science/article/pii/S2542660522001305
[8] Prediction of Pest Insect Appearance Using Sensors and Machine Learning
https://www.mdpi.com/1424-8220/21/14/4846
[9] Integrated analysis of machine learning and deep learning in chili pest and disease identification
https://scijournals.onlinelibrary.wiley.com/doi/abs/10.1002/jsfa.10987
[10] IoT-based pest detection and classification using deep features with enhanced deep learning strategies
https://www.sciencedirect.com/science/article/abs/pii/S0952197623001690
[11] Machine learning ensemble with image processing for pest identification and classification in field crops
https://link.springer.com/article/10.1007/s00521-020-05497-z
[12] Citrus pests classification using an ensemble of deep learning models
https://www.sciencedirect.com/science/article/abs/pii/S016816992100209X
[13] An ensemble learning integration of multiple CNN with improved vision transformer models for pest classification
https://onlinelibrary.wiley.com/doi/abs/10.1111/aab.12804
[14] Optimizing VGG16 for Accurate Pest Identification in Oil Palm: A Comparative Study of Fine-Tuning Techniques
https://www.mecs-press.org/ijieeb/ijieeb-v16-n5/v16n5-3.html
[15] Using a novel convolutional neural network for plant pests detection and disease classification
https://scijournals.onlinelibrary.wiley.com/doi/abs/10.1002/jsfa.12700
[16] Pest detection on Traps using Deep Convolutional Neural Networks
https://dl.acm.org/doi/abs/10.1145/3232651.3232661
[17] Intelligent Identification of Jute Pests Based on Transfer Learning and Deep Convolutional Neural Networks
https://link.springer.com/article/10.1007/s11063-022-10978-4
[18] Cutting-edge ensemble framework of deep convolutional neural networks for high-precision insect pest classification
https://link.springer.com/article/10.1007/s41348-024-00986-y
[19] Disease and pest infection detection in coconut tree through deep learning techniques
https://www.sciencedirect.com/science/article/abs/pii/S0168169921000041
[20] Automated Pest Detection With DNN on the Edge for Precision Agriculture
https://ieeexplore.ieee.org/abstract/document/9502691
[21] Identifying pests in precision agriculture using low-cost image data acquisition
https://www.scielo.br/j/bjb/a/tj7N9JBGSKwH5DTp6Fz8Lsz/?format=html&lang=en
[22] PENYEK: Automated brown planthopper detection from imperfect sticky pad images using deep convolutional neural network
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0208501

