PEST DETECTION ROBOT

PROJECT REPORT

Submitted by

Surya Narayan S - (CB.SC.U4AIE23251)

Divagar S - (CB.SC.U4AIE23223)

Mounindra P – (CB.SC.U4AIE23273)

Raswanthkrishna M – (CB.SC.U4AIE23266)

Adithyan P V – (CB.SC.U4AIE23206)

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE ENGINEERING

(ARTIFICIAL INTELLIGENCE)

COMPUTER SCIENCE ENGINEERING - ARTIFICIAL INTELIIGENCE

AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112 (INDIA)

APRIL – 2025

COMPUTER SCIENCE ENGINEERING - ARTIFICIAL INTELLIGENCE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

श्रद्धावान् लभते ज्ञानम्

AMRITA VISHWA VIDYAPEETHAM

DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF THE UGC ACT, 1956

BONAFIDE CERTIFICATE

This is to certify that the thesis entitled “Pest detection Robot” submitted by Surya Narayan S (CB.SC.U4AIE23251), Divagar S (CB.SC.U4AIE23223), Mounindra P (CB.SC.U4AIE23273), Adithyan PV (CB.SC.U4AIE23206), Raswanthkrishna M (CB.SC.U4AIE23266), for the award of the Degree of Bachelor of Technology in the “COMPUTER SCIENCE ENGINEERING - ARTIFICIAL INTELLIGENCE” is a Bonafide record of the work carried out by them under my guidance and supervision at Amrita School of Artificial Intelligence, Coimbatore.

Dr. D. Palmani

Dr. Milton Mondal

Project Guide Designation: Senior grade Assistant Professor

Submitted for the university examination held on … … … … …

INTERNAL EXAMINER **EXTERNAL EXAMINER**

AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE

AMRITA VISHWA VIDYAPEETHAM

COIMBATORE - 641 112

DECLARATION

We, Surya Narayan S (CB.SC.U4AIE23251), Divagar S (CB.SC.U4AIE23223), Mounindra P (CB.SC.U4AIE23273), Adithyan PV (CB.SC.U4AIE23206), RaswanthKrishna M (CB.SC.U4AIE23266) hereby declare that this thesis entitled “Pest detection Robot”, is the record of the original work done by us under the guidance of Dr. D. Palmani sir, Dr. Milton Mondal sir, Amrita School of Artificial Intelligence, Coimbatore. To the best of our knowledge, this work has not formed the basis for the award of any degree/diploma/ associateship/fellowship/or a similar award to any candidate in any university.

Signature of the Student

Place: Coimbatore

Date: 22/04/2025

COUNTERSIGNED

Dr. K.P. Soman

Professor and Dean

Amrita School of Artificial Intelligence

Amrita Vishwa Vidyapeetham

Table of Contents

Acknowledgement

We want to express our heartfelt gratitude to all those who supported and directed us during the development of our project, **"Pest detection Robot"**. The project has been a notable learning experience that was not possible without the help and encouragement of many individuals.

Most importantly, we are significantly grateful to our project guides, **Dr. D. Palmani Sir**, and **Dr. Milton Mondal sir** for their constant encouragement, expert advice, and constructive criticism. In areas of robotics and Mathematics for computing, their expert insights are important in shaping the direction and execution of our work.

We are also grateful to the faculty members and employees of Amrita School of Artificial Intelligence, Amrita Vishwa Vidyapeetham, Coimbatore, to provide us with the infrastructure, educational environment and necessary support to successfully fulfill our project.

We thank our honesty for our colleagues, inspiration and teamwork which more manage and rewards challenges during this journey. Their contribution was necessary for the smooth execution of the project.

Finally, we want to express our deep praise for their unwavering support, patience and encouragement for our families, which inspired us to work hard and complete this task with dedication.

This project is a great step in our educational and professional journey, and we are grateful to all those who helped make it possible.

List of Abbreviations

CNN	Convolutional Neural Network
GPU	Graphical Processing Unit
ADAM	Adaptive Moment Estimation (Optimizer)
L/R Wheels	Left/Right Wheels (used in kinematics)
GFLOPS	Giga Floating-Point Operations Per Second (computing performance unit)
MBConv	Mobile Inverted Bottleneck Convolution

Abstract

The concept offers a clever robot system that can identify pests in agricultural areas in real time. Using a PI camera, a fine-tuned deep learning model based on skillnet is used for live video analysis after being trained to identify common farm pests from photos. The system, which is integrated with a mobile robot, records the video frame while the robot moves and uses the difference drive for movement . By effectively combining computer vision, edge computing, and robots, this method seeks to decrease human labor, facilitate early insect identification, and promote precise agriculture.

Chapter 1 :Introduction

Pest management is a critical activity in contemporary agriculture, has a direct impact on the yield of crops, quality of food and economic yield. Traditional methods of detecting insects are reactive, compared to time-consuming, labor-intensive and often preventive. With the advent of Artificial Intelligence (AI) and robotics, there is a valuable opportunity to automate this activity through intelligent systems to assist farmers in detecting pests quickly and accurately. The project presents an integrated solution that combines intensive learning-based image classification with mobile robot navigation to identify real-time pests in fields. Utilizing EfficientNetV2B0, a pre-trained model, the system was fine-tuned to classify different species of insects. The fine-tuned model was utilized to classify pests from live video streams obtained from a Raspberry Pi camera mounted on a mobile robot.

The system relies on a Raspberry Pi 5 that facilitates on-device computation. The aim is to provide a scalable, affordable and region-deployable system that facilitates accurate agriculture, minimizes reliance on human labor, and facilitates early intervention to minimize crop losses. Current work indicates the advantages of using AI and robotics in agricultural insect monitoring.

Amnerkar et al. (2023) suggested an IoT-based pest detection system using Raspberry Pi and the YOLO algorithm, which was able to detect over 90% of pests in real-time. Their research offers proof of employing these technologies in an effort to transform pest management by offering immediate alerts to the farmers. Finally, the EfficientNetV2 architecture by Tan and Le (2021) validates our model choice, achieving accuracy-speed balance for real-time use. Sandhya Devi et al. (2023) demonstrated the choice by showing its effectiveness, with high accuracy; 99.5%, 97.5%, and 80.1% on Cassava, PlantVillage, and IP102 datasets, respectively, while enabling faster convergence than models like InceptionV3 and Vision Transformers (ViT). Their study attests to EfficientNetV2's adaptive regularization and compound scaling, which are crucial in training efficiency as well as pest recognition interpretability.

Paper Title and Authors	Key Contributions	Limitations
Real-Time Detection and Classification of Scirtothrips dorsalis on Fruit Crops with Smartphone-Based	EfficientDet-D0 model trained using the transmittance dataset provided 93.3% accuracy for real-time detection of Scirtothrips dorsalis using a smartphone application. The transmittance lighting setup improved	This system has been tested under controlled experimental conditions only and depends on adhesive trap cards for collection of pests, hence restricting direct

Deep Learning System: Preliminary Results, (Niyigena, G et. al.)	detection accuracy and image quality with an affordable and cost-effective solution. .	use for scanning live plants in agricultural environments
AI-based pest detection and 3alert system for farmers using IoT(S. T. Jaya et. al.)	The system combines acoustic pest detection with AI , sending mobile alerts via Wi-Fi. It features automated pesticide spraying to minimize manual intervention. A website offers real-time pest monitoring and remote system control.	Acoustic detection may face accuracy issues due to environmental noise. Hardware complexity, scalability challenges in large farms.
Automatic pest identification system in the greenhouse based on deep learning and machine vision(Z. Xiaolei et. al.)	The system achieved a 96% detection rate using a superior YOLOv5 model. The surveillance captured variations in pest populations in cherry tomato and strawberry greenhouses, thereby offering practical guidance towards pest management. The integration of fixed traps and deep learning technology emphasizes the potential for technological interventions in agriculture-oriented artificial intelligence systems.	The yellow sticky paper traps-based static pest detection system in the paper relies on monitoring being limited to pests attracted to the traps.
An IoT-Based Solution for Insect Monitoring in Agriculture Using Raspberry Pi and YOLO(P.Amnerkar et. al.)	The fine-tuned YOLOv5 model incorporating DIoU-NMS and data augmentation also achieved a pest detection precision of 96%, a considerable increase from the baseline YOLOv5 precision of 65%. The increase in precision was even more substantial in the identification	It relies on stationary installations and thus can only handle a small number of locations in a greenhouse or farm environment.

	of pests such as leaf miners and fruit flies, with the precision rate achieving 99%. The system further effectively tracked the dynamics of the pests for 40 days across strawberry and cherry tomato greenhouses and exhibited substantial fluctuations in pest occurrence based on crop types. Moreover, edge computing and IoT integration ensured on-device processing with real-time notification to farmers using Wi-Fi.	

Table 1.1: Literature Review Summary

1.2 Problem statement

Farmers face important challenges in detecting pest infections, especially in large agricultural areas. Manual inspection is time consuming , labor-intensive, and is sometimes delayed which leads to more crop damage and unnecessary pesticide usage. There is a growing requirement of a mobile, real -time insect detection system that ensures high accuracy and continuous monitoring. Our project addresses this by developing an integrated syste using a Raspberry PI-controlled mobile robot equipped with a PI camera and an onboard CNN that processes real-time video feeds to detect pests in real time. This solution aims to enhance pest detection efficiency, reduce manual labor, and support sustainable farming practices.

1.3 Objectives

- To develop a CNN-based image classification models capable of identifying several common pest species from the feeds captured by the PI camera.
- Detect pests in real-time from the video feed collected through the robot's onboard camera as its moves across agricultural areas.
- To detect, classify, annotate, and log in pests in a real -time setting using raspberry pi for preprocessing and control.
- To contribute to permanent agriculture by enabling initial pest intervention, reducing manual labor and excessive pesticide use through continuous monitoring.

1.4 Organization of the Report

Chapter 1: Introduction

The problem of pest detection in agriculture shows a robotic solution, inspiration behind developing a brief observation of the project and report structure.

Chapter 2: Background

The CNNs, Raspberry Pi integration, Pi camera use, robotic Kinematics and real -time image processing principles include fundamentals.

Chapter 3: Proposed work

Details system architecture, dataset preprocessing, CNN models for insect classification, PI camera-based image acquisition, using differential drives, robot movement strategy, and integration of these components on raspberry PI platforms.

Chapter 4: Results and Discussion

The model training presents the results, performing the real-time pest detection, and tests the results from both the live feed captured during robot navigation and pre-recorded video streams .

Chapter 5: Conclusion and future work

The project summarize achievements, exposes the capacity of the system in the region's landscapes, and proposes enrichment such as GPS integration, advanced sensor fusion and deployment on large farm parameters.

Chapter 2: Background

2.1 Overview

The recent advancements in artificial intelligence (AI) and robotics have opened new avenues in agriculture, especially for automating operations such as pest monitoring, crop monitoring, and navigation of the field. Pest infestation continues to be a major issue all over the world, and this often results in high crop loss. This necessitates precise, real-time, and automated means. This chapter presents the fundamental technologies, algorithms, and frameworks that comprise the backbone of our proposed system for pest detection robots—convolutional neural networks (CNNs), robot kinematics, and real-time pest detection by embedded systems.

2.2 Convolutional Neural Networks (CNNs)

CNNs are deep learning algorithms that are well suited for image classification and object recognition tasks. Since they can automatically learn hierarchical visual features, CNNs are very effective for the identification of pest species. The key elements of CNN are as follows:

1. **Convolutional Layers**: Detect low to high-level features (edges, textures, shapes) by sliding filters over image patches.

   The output of a convolution operation at a specific position is given by:

$$y(i,j)(t) = \sum_{m=1}^{M} \sum_{n=1}^{N} K(m,n) \cdot x(i+m,\ j+n)$$

- The discrete convolution operation is the dot product of a kernel and a local patch of the image
- **Pooling Layers**: They conduct down-sampling of feature maps to decrease computational complexity and preserve dominant features.

- **Fully Connected Layers**: Flatten the extracted features into a 1D vector and feed it through one or more dense layers, usually culminating in a softmax layer for multi-class classification.

$$soft\max(z_i) = \frac{e^{z_i}}{\sum e^{z_j}}$$

2.3 EfficientNetV2

Our implementation employs EfficientNetV2B0 as the backbone CNN. EfficientNetV2 combines training-aware neural architecture search with a compound scaling strategy to optimize accuracy, model size, and training speed.

- Early layers use Fused-MBConv blocks, which combine expansion and depthwise convolutions as a single operation to reduce memory usage and speed up training efficiency.
- Later layers use regular MBConv blocks coupled with squeeze-and-excitation modules to enhance the recalibration of features.
- With around 7.1 million parameters and 1.46 GFLOPS at resolution 224×224, EfficientNetV2B0 is fairly well-balanced between model complexity and inference speed.
- It starts at low-resolution images with lighter regularization and gradually adds larger images and stronger regularization that allows for fast convergence and stable performance in the end.

2.4 Robotics in Agriculture

Agricultural robots are becoming increasingly popular for the activities of planting, weeding, and pest monitoring. This project utilizes a mobile robot platform for real time pest identification. Some main concepts are:

2.4.1 Differential Drive Kinematics

Differential drive robots are driven on two wheels, which are independently controlled. The robot's motion depends on the individual geometry of those wheels and their velocity settings.

(x_{t+1} ,y_{t+1}) and orientation θ_{t+1} at time t+1 are given by:

$$x_{t+1} = x_t + r \cdot cos\theta$$

$$y_{t+1} = y_t + r \cdot sin\theta$$

$$\theta_{t+1} = \theta_t + \frac{[vr - vl]}{d}$$

Where:

- r: Wheel radius
- vr,vl : Velocities of the right and left wheels
- d: Distance between wheels

2.5 Edge Computing in Agriculture

Edge computing refers to the execution of data processing on the device level itself, instead of via centralized cloud infrastructure. The reduction in latency, ability to provide real-time capabilities, and bandwidth savings are all key benefits for mobile robotic systems in the field, especially those subject to intermittent connectivity.

The architecture is built around the Raspberry Pi 5, a powerful edge computing device that is popular for enhanced processing, high RAM pile, and GPU based hardware acceleration for machine learning workloads. This enables deployment of CNN models for real-time inference optimization.

Chapter 3: Proposed Work

3.1 Overview

A pest detection system trained with deep learning will control the mobile robot platform for automatic pest detection in agricultural fields. The system is capable of running on a Raspberry Pi 5 and is mainly concerned with precise classification, real-time processing, and seamless controlled movement. This chapter describes the architecture and implementation specifics of the three main components making up the system: Perception, Robot Movement, and Execution.

3.2 Methodology

3.2.1 Model Selection

EfficientNetV2B0 was selected as the backbone for the pest classification task since it offers an exceptional trade-off between computational efficiency and classification accuracy. The pre-trained model was originally trained on the ImageNet dataset and fine-tuned to classify pests as one of the following: Beetle, Caterpillar, Earwig, Grasshopper, Slug, Snail, and Weevil.

3.2.2 Model Training

The following processes were carried out to adapt EfficientNetV2B0 to pest classification:

- Images were resized to 224x224 pixels to meet the specific input size of the model.

- The pixel values were normalized to be in the range of [0,1] to facilitate efficient computation and minimize fluctuations during training.

- Afterward, the images were transformed into tensors suitable for the input of TensorFlow-based models.

Modifications made to EfficientNetV2B0:

- The pre-trained EfficientNetV2B0 was used, removing the top layer pertaining to its initial classification.

- The last 10 layers of the EfficientNetV2B0 were unfrozen for task-specific training, whereas the earlier stages were kept frozen to maintain generic pre-trained features.
- A global average pooling layer was stacked on top to reduce the spatial dimension, followed by a fully connected dense layer with 7 output neurons corresponding to the 7 pest classes with softmax activation outputting class probabilities in favor of multi-class classification.
- Categorical cross-entropy was implemented as a loss function for our multi-class classification problem:
  - The Adam optimizer was used for effective gradient descent optimization.
- Callbacks were specified and had the following functions:
  - The training was halted in case the validation accuracy did not improve for a given number of epochs to prevent overfitting;
  - The learning rate was adjusted on-the-fly depending on validation loss to maximize the outcome on training.

3.2.3 Movement of Robot

The robot's motion is facilitated by a differential drive mechanism that allows the two wheels that move on either side of the robot to be independently controlled. This design permits smooth turning, precise movement, and less restriction on various terrains. Velocity inputs are converted into rotations of the wheels via a kinematic model, hence keeping the manoeuvrability and accuracy.

3.2.4 Real-time Pest Detection

The trained model is deployed using a Raspberry Pi camera to perform real-time pest classification.

Camera Feed Capture

Picamera2 module captures streaming live video from the Raspberry Pi's camera, The frames are processed in real-time and converted from RGB (Picamera2 format) to BGR (OpenCV format) because OpenCV images are represented in BGR format.

Frame Preprocessing

Every frame captured is pre-processed so that it becomes compatible with the input requirements of the model:

- Resized to 224 × 224 pixels.
- Normalized into a range between [0,1].
- Converted to tensor batch for inference.

Real Time Inference

The preprocessed frame is fed into the trained EfficientNetV2B0 model for classification.

- Output will be probabilities associated with the pest classes.
- The pest category with the highest confidence score is selected as prediction.

Logging and Annotation

- Predictions and respective confidence scores are timestamped for archival purposes.
- Frames are annotated with pest name and confidence score when the confidence score exceeds a defined threshold (conf > 0.8).
- Annotated frames are displayed in real time on the Raspberry Pi.

Frame Saving

Frames are saved locally for debugging or future analysis; separated logs are maintained for predictions, including timestamped class labels and confidence score.

3.3 System Workflow

3.3.1 Robot Control

- The robot is driven manually and navigates through the agricultural land with little interference by the differential drive system.
- It systematically traverses the field under manual input while looking for pests.

3.3.2 Frame Capture (Simultaneous to Robot Movement)

- Frames are captured periodically by the Pi Camera while the robot is moving across the field. The frames will be the input for pest detection purposes.

- While navigating through the field under manual control, the camera continuously sends frames to the system for processing.

3.3.3 Pest Detection (For Real-Time Analysis)

- Preprocessing of each frame, for instance, resizing or normalization, is followed by the actual pest detection performed using the TensorFlow Lite efficient model, EfficientNetV2B0, on a Raspberry Pi.
- This includes model inference on the frame to determine pests' presence in real time with good accuracy.

3.3.4 Annotation & Logging (Post-Detection)

- Once the detection takes place and a pest is found in the frame, the system annotates that frame by marking the position of the pest, thereby making human interpretation for the frame easier.
- The system also logs the event in respect to key info such as pest type detected by the model and the time at which the pest was detected.
- The annotated frame and logged data can be kept for further analyses and monitoring; thus, a history of pest activity would be maintained.

3.4 Implementation Details

3.4.1 Hardware Components

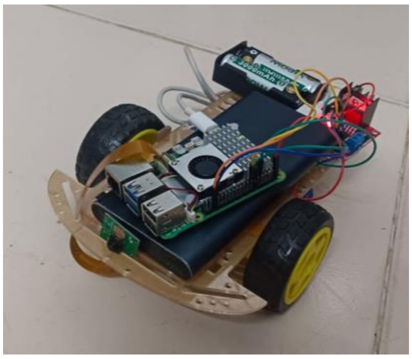
Raspberry Pi 5: The microcontroller board, which conducts pest detection with a TensorFlow model and also regulates the robot movement.

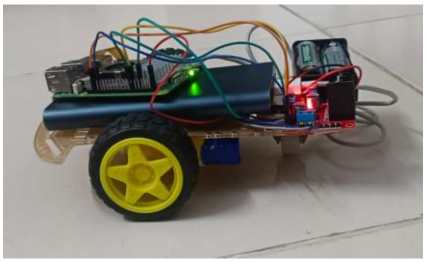
Pi Camera 3 Module: For live video feed capture to detect pests in real time.

DC Motor Chassis: A two-wheel-drive chassis platform housing the DC wheel motors, structural frame, and mounting points for the electronics.

Wheel Motors: Two DC motors mounted on a chassis providing differential-drive locomotion.

Motor Driver: Interface to control the wheels motor speed and direction through the Raspberry Pi GPIO.





a) *Top View* *b) Side View*

Fig 1 a and b: The robot, with a Raspberry Pi 5 with a Pi Camera 3 Module connected to the board, mounted on a 2WD robot chassis with yellow wheels(mounted on the motors) and a motor driver (the pins are also connected to the Pi 5), with a 5V power bank.

3.4.2 Software Stack

EfficientNetV2B0: for pest classification; performs inference and gives predictions on the basis of captured frames.

OpenCV: Video frame capturing, frame annotation, and display.

RPi.GPIO: Library to control of Raspberry Pi GPIO pins to control the motor driver for robot movement.

NumPy: For operations like normalizing and preprocessing the frame data before sending them to the classification model.

Python was used to implement all pest classification, video processing, and robot control script

Chapter 4: Results and Discussion

4.1 Overview

This chapter presents the outcomes of the pest detection system during installation and its performance under real-time detection and on the Raspberry Pi 5 platform.

4.2 Model Results

4.2.1 Performance Metrics

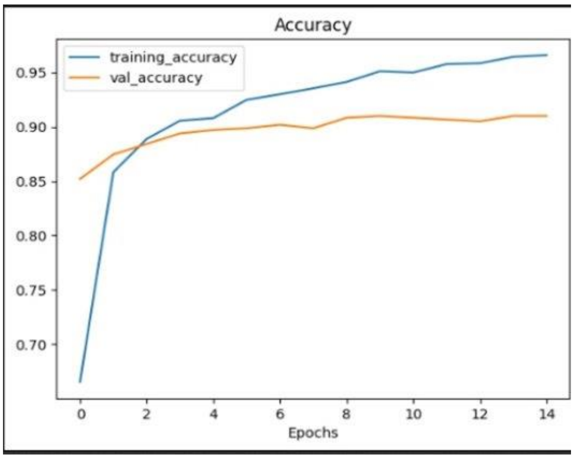


Fig 2: Graph showing the model's training and validation accuracy progression over each epoch.

The training accuracy kept on rising during the course of the epochs, eventually reaching beyond 95% in the last epoch, while validation accuracy also increased steadily for the early epochs, settling to a plateau of around 90%. The training and validation accuracies tracked closely for most of the training process, thus indicating that the model has been able to generalize quite well.

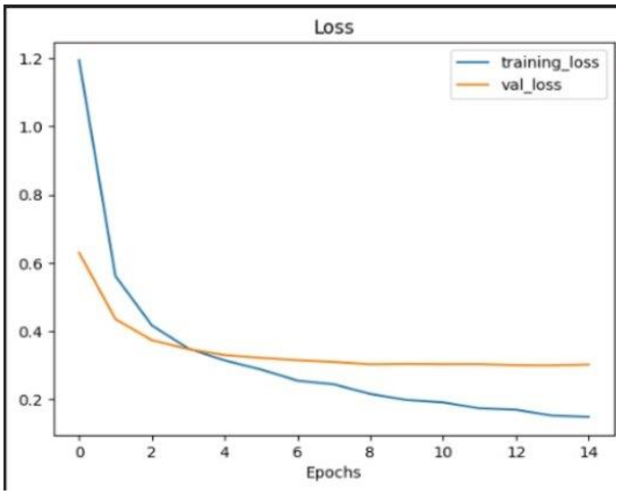


Fig 3: Graph showing the model's training and validation losses decrasing over each epoch.

The training loss diminished steadily across epochs, representing a model which was learning and optimizing well on training data. The initial epochs of validation loss had similar consistent behavior, which indicates that early on, the model generalized well to unseen data.

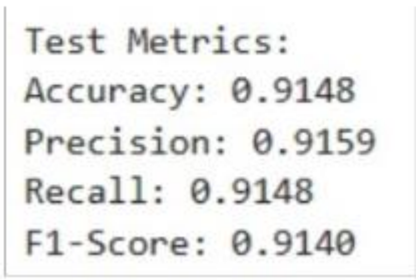

Test Metrics:
Accuracy: 0.9148
Precision: 0.9159
Recall: 0.9148
F1-Score: 0.9140

Fig 4: Accuracy, Precision, Recall and F1 Score on test image set

Evaluating model on test dataset...

Classification Report:

	precision	recall	f1-score	support
beetle	0.91	0.85	0.88	85
catterpillar	0.92	0.82	0.87	105
earwig	0.88	0.86	0.87	76
grasshopper	0.84	0.96	0.90	95
slug	0.93	0.95	0.94	75
snail	0.97	1.00	0.98	95
weevil	0.95	0.98	0.96	91
accuracy			0.91	622
macro avg	0.92	0.91	0.91	622
weighted avg	0.92	0.91	0.91	622

Fig 5: Classification report of model on the test image set

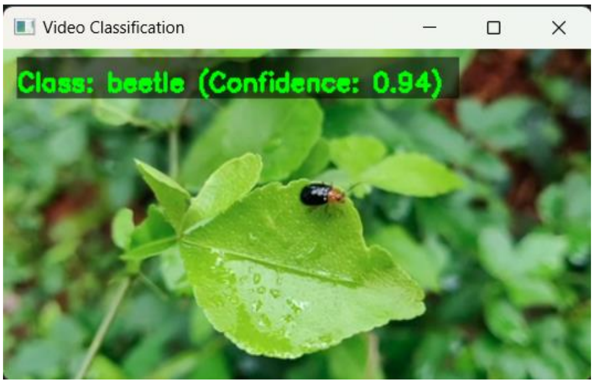
Further evaluation was done using 622 test samples across seven types of pests: beetle, caterpillar, earwig, grasshopper, slug, snail, and weevil.

The overall test accuracy was 91.48%, with a precision of 91.59%, recall of 91.48%, and an F1-score of 91.40%. The performance of individual classes varied slightly, with the class snail obtaining the highest F1-score (0.98), while earwig and caterpillar exhibited slightly lower scores (0.87). These metrics show the model performing quite well, reliably classifying all pest classes, with the macro and weighted averages of the metrics further confirming consistent model behavior.

These results strongly indicate that the model could be practically applied for pest classification tasks with high accuracy and well-balanced performance across classes.

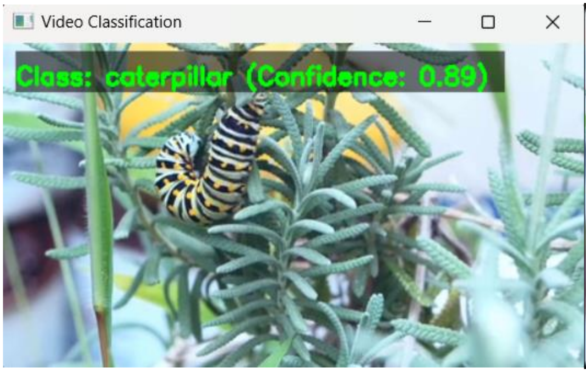
4.3 Video Based Pest Detection

a)

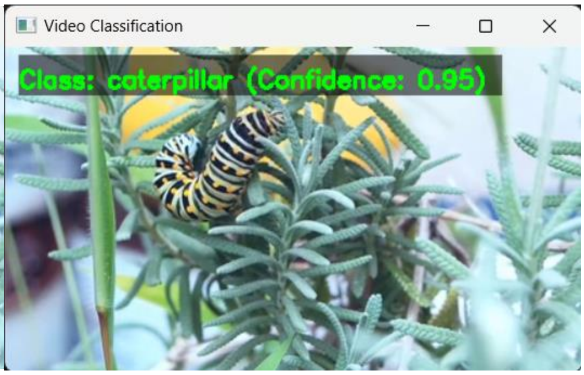


b)

Fig 6 a and b: The model's predictions on a video of beetle. The predicted class and its confidence score are annonated at the top-left corner in green text inside a semi transparent box.



a)



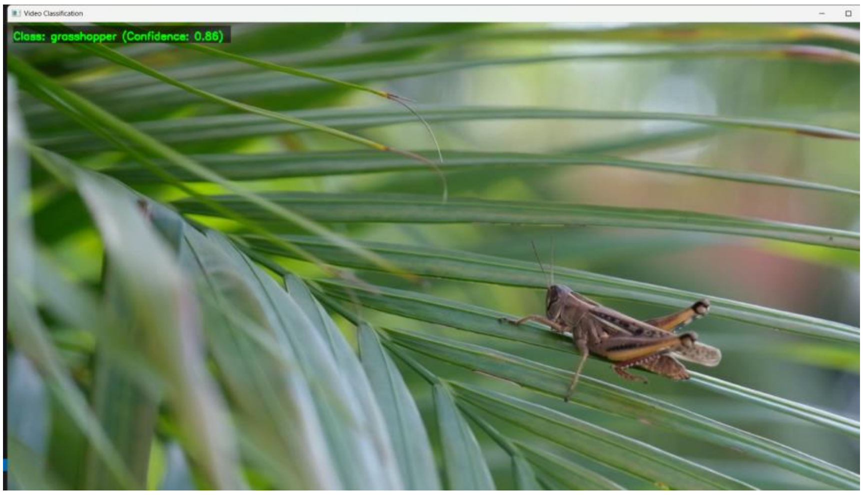
b)

Fig 7 a and b: The model's predictions on a video of a catterpillar.

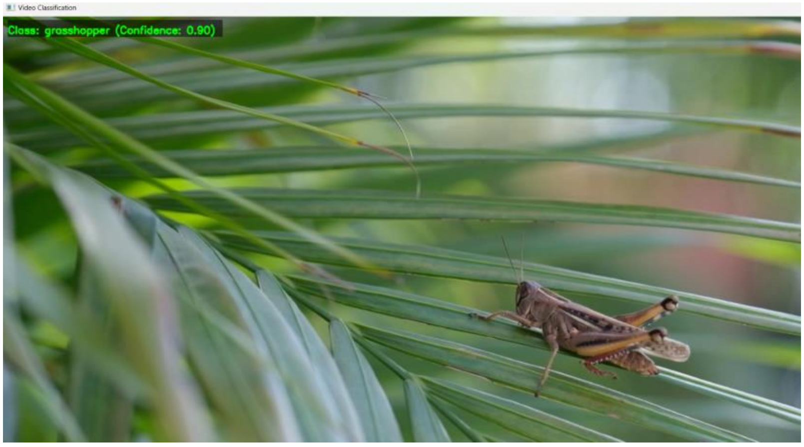
a)

b)

Fig 8 a and b: The model's predictions on a video of an earwig



a)



b)

Fig 9 a and b: The model's predictions on a video of a grasshopper

In addition to the evaluation of the model against the test dataset, videos with grasshopper, beetle, earwig, and caterpillar were classified. The model performed robustly detecting pests over the video frames, consistently identifying the pests with high confidence.

4.4 Real Time Pest Detection

Outputs

- Overlaid on the video stream were predicted pest class and model confidence scores.
- Each prediction was logged for further analysis.

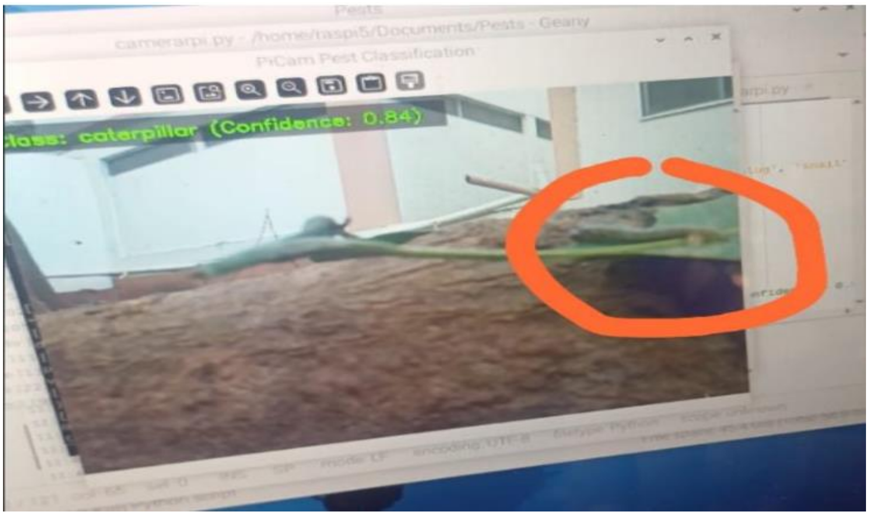


Fig 10:Real-time predictions on a live video feed from PiCam the showing a caterpillar, with the predicted class and confidence score displayed in green text inside a semi-transparent box at the top-left corner of the feed.

Real-time classification was also performed using a live feed from a Pi Camera connected to a Raspberry Pi 5, integrated with the robot. During this test, a live caterpillar obtained from outside was analyzed. The model successfully classified the caterpillar with a confidence score of 0.84.

Chapter 5: Conclusion and Future Work

5.1 Conclusion

The project successfully demonstrated the deployment of a real-time pest detection system based on a mobile robotic platform driven by Raspberry Pi 5 and a TensorFlow Lite-optimized EfficientNet model. The edge-computing-based solution facilitated on-device inference, which lowered the latency and allowed for autonomy in remote agricultural settings.

Key Findings:

- The TF model achieved an inference rate of approximately 6.79 frames per second (FPS) and an average latency of aprox 147.2 ms for video-based and real-time classification.
- For image-based prediction the model demonstrated an average latency of 58.17 ms, indicating better performance when processing individual images.
- The system was able to detect and annotate pests precisely on both live video streams and offline video inputs.
- It has a modular architecture, facilitating easy scalability and upgrade for potential future applications, e.g., adding more sensors or control modules.
- This shows how edge computing devices, deep learning, and robotics could be applied in intelligent applications for pest management in practice.

5.2 Future Work

5.2.1 Enhancements for Pest Detection

- Integrating multispectral or hyperspectral imaging that can detect very subtle pest patterns will give its advantage over an RGB image in terms of improving identification.
- Create algorithms by designing a real-time tracking of pests' movements and behavior, so countermeasures are well specified and much effective.
- Expand the training dataset through the addition of other pest species and images taken within a wider range of environmental conditions for enhanced generalization and robustness of the model.

5.2.2 Advanced Robotic Navigation

- Integrate a terrain-adaptive suspension to improve the performance of the machinery in an uneven agricultural field.
- Utilize GPS or RTK-GPS mode to get the good geolocation of the area mapping and navigation especially in the larger scale areas.
- Adopting different SLAM methods for adaptive path planning and dynamic localization in complex field conditions for simultaneous localization and mapping.

5.2.3 Embedded Countermeasures

- Integrate an automated spraying feature on the robot to immediately perform pest infestation treatment once detected.
- Modify the spraying mechanism so that it refines pesticide amount according to pest type and concentration to allow precise intervention with minimal environmental impacts.

5.2.4 Edge Optimization

- The compression of the CNN for smaller inference times and reducing energy consumption on the edge device.
- Exploration of Alternate High-Performance Edge Devices:
- To analyze other high-performance edge device options for enhanced real time performance of the model.

References

1. Niyigena, G., Lee, S., Kwon, S., Song, D., & Cho, B.-K., "Real-Time Detection and Classification of Scirtothrips dorsalis on Fruit Crops with Smartphone-Based Deep Learning System: Preliminary Results," Insects, vol. 14, no. 6, p. 523, Jun. 2023, doi: 10.3390/insects14060523.
2. S. T., J. Manikandarajan, S. Suhidhana, and T. Subha, "AI-based pest detection and alert system for farmers using IoT," E3S Web of Conferences, vol. 387, 2023, doi: 10.1051/e3sconf/202338705003.

3. Z. Xiaolei, B. Junyi, Z. Xixiang, and W. Xiaochan, "Automatic pest identification system in the greenhouse based on deep learning and machine vision," Frontiers in Plant Science, vol. 14, 2023. [Online]. Available: https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2023.1255719. doi: 10.3389/fpls.2023.1255719.
4. P. Amnerkar, K. Bawane, M. Raut, D. Bargat, B. Damahe, and K. A. Ande, "An IoT-Based Solution for Insect Monitoring in Agriculture Using Raspberry Pi and YOLO," International Research Journal of Modernization in Engineering Technology and Science, vol. 5, no. 4, pp. 1–6, Apr. 2023, doi: 10.56726/IRJMETS36564.
5. Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. arXiv preprint arXiv:1804.02767.
6. He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.
7. R. S. Sandhya Devi, V. R. Vijay Kumar, and P. Sivakumar, "EfficientNetV2 Model for Plant Disease Classification and Pest Recognition," Computer Systems Science and Engineering, vol. 45, no. 2, pp. 2249–2263, 2023, doi: 10.32604/csse.2023.032231.
8. M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," 2021. [Online]. Available: https://arxiv.org/abs/2104.00298.