Pest Detection Robot

C-1

Team Members

- Divagar S – CB.SC.U4AIE23223
- Raswanthkrishna M – CB.SC.U4AIE23266
- Surya Narayan S – CB.SC.U4AIE23251
- Mounindra P – CB.SC.U4AIE23273
- Adithyan P V – CB.SC.U4AIE23206

INTRODUCTION

- Pest management is vital for ensuring crop yield, food quality, and economic stability in modern agriculture.
- Traditional pest detection methods are **manual**, **time-consuming**, and often **reactive**.
- Advancements in **AI and robotics** enable smarter, faster pest detection systems.
- This project combines **deep learning-based image classification** with a **robot** to detect pests in real time.
- The aim is to provide a **scalable**, **cost-effective** solution for **precision agriculture** with timely pest identification and reduced manual effort.

Problem Statement

- Farmers struggle with **timely pest detection**, especially in large and open fields.
- **Manual scouting** is labor-intensive, error-prone, and often delayed—leading to:
  - Reduced crop yield
  - Increased operational costs
- There's a need for a **real-time**, **accurate**, and **field-deployable** pest detection system using a mobile robotic platform.

Objectives

Develop a CNN-Based Pest Classifier
Build a deep learning model to identify different pest species from images with high accuracy.

Implement Real-Time Detection
Use a live video feed from a robot-mounted camera to detect pests in real time.

Annotate and Log Detections
Classify detected pests and store logs for further analysis and model improvement.

Promote Sustainable Agriculture
Reduce manual labor and pesticide usage by enabling early, automated pest detection.

Methodology - Image Classification

Model Selection – EfficientNet
EfficientNet was selected for its excellent trade-off between high accuracy and low computational cost, making it ideal for edge deployment.

Image Preprocessing
Input images are resized to 224×224 pixels, normalized, and converted into tensors compatible with TensorFlow models.

Model Optimization – TensorFlow Lite
The trained model is converted to TensorFlow Lite with quantization to reduce model size and improve inference speed on **Raspberry Pi 5**.

Manual Navigation via Bluetooth
Instead of autonomous navigation, a robot is used to manually explore fields and capture live video feeds for pest detection.

Methodology - Robotic Navigation

The robot uses a **differential drive mechanism** to allow smooth movement in all directions (forward, backward, turns).

Movement is **manually controlled**, using commands sent from a mobile device or remote controller.

A camera mounted on the robot captures live video feed for pest detection while navigating through the field.

Optional: **Proximity sensors** (e.g., ultrasonic) can be added for basic obstacle detection, but navigation decisions remain manual.

Methodology Real-Time Pest Detection

The robot captures **live video frames** using an onboard camera during manual navigation.

Each frame is analyzed by the **EfficientNet-based classifier**, and pests are detected with confidence scores shown on-screen.

All detections are **timestamped and logged**, enabling later analysis and actionable decision-making in pest management.

System Architecture

Control Module

- A **controller** is used to send movement commands for the movement of the robot.
- The **Raspberry Pi** receives these commands and translates them into motor control signals.
- These signals are passed to the **L298N motor driver**, which drives the **DC motors**, allowing the robot to move in all directions using differential drive.

Camera & Video Feed

- A **Pi Camera** mounted on the robot continuously captures **real-time video frames** of crops as the robot moves.
- These frames are sent directly into the Raspberry Pi for processing without any delay.

System Architecture

Image Classification Module

- Each captured frame is **resized to 224×224 pixels**, normalized, and converted into a format suitable for the model.
- The preprocessed frame is passed to the **EfficientNet-based classifier** (deployed using **TensorFlow**) for pest detection.
- If a pest is detected, the **species name and confidence score** are overlaid on the video frame.

Logging & Display

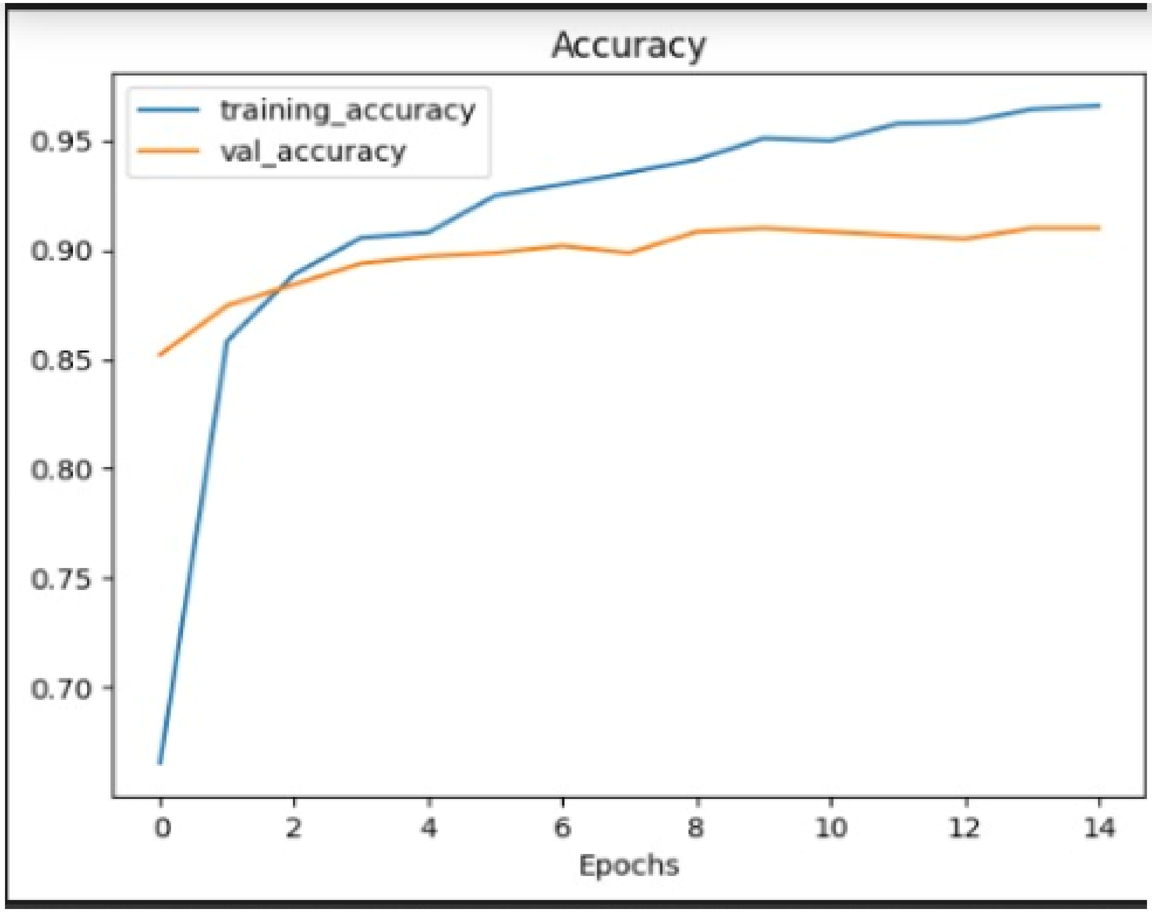
- Each detection is **timestamped and stored in logs**, including class, confidence, and frame number.
- These logs can be used for future analysis and pest trend tracking.
- Optionally, a live preview can be shown on a connected display or streamed to a remote device for real-time monitoring.

Hardware Components Software Stack

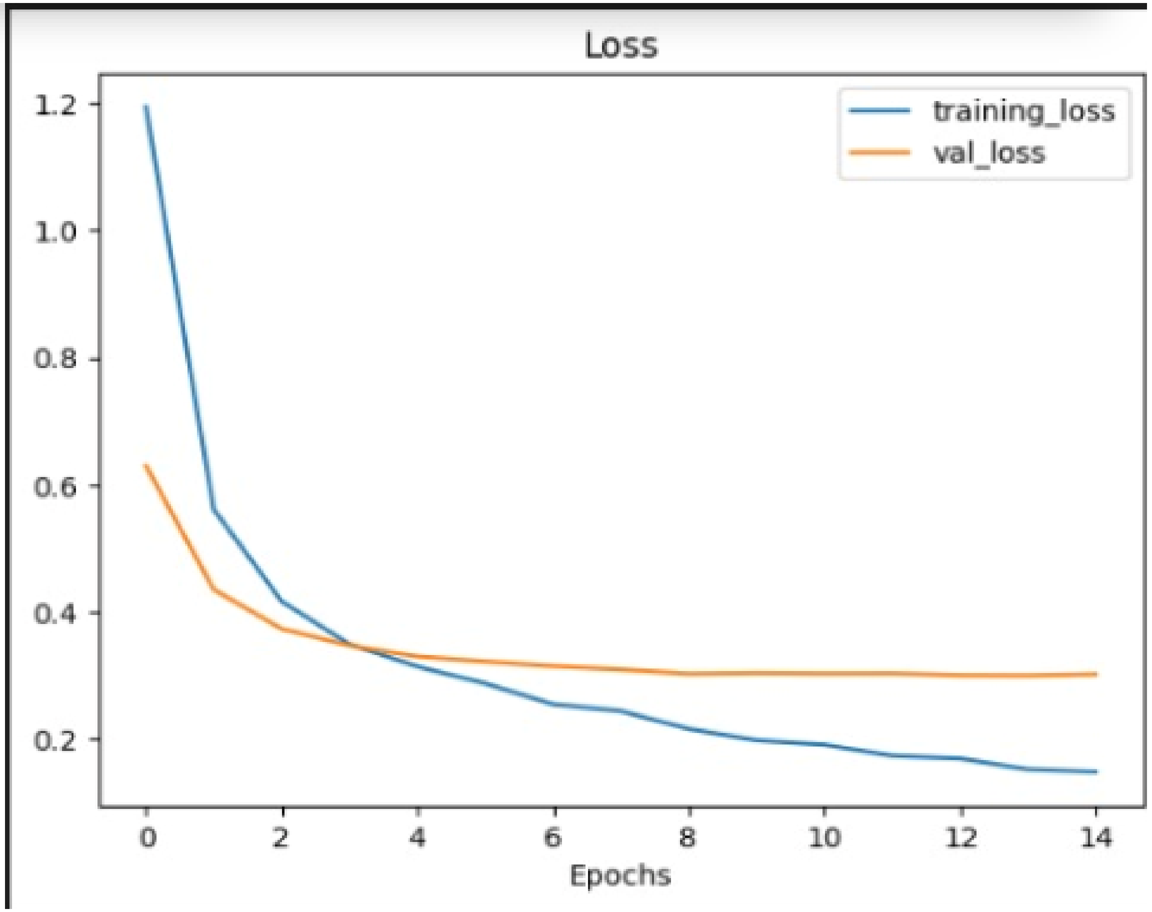
- Raspberry Pi 5
- Pi Camera
- L298N Motor Driver
- DC Motors

- TensorFlow
- OpenCV
- NumPy
- Python

Results

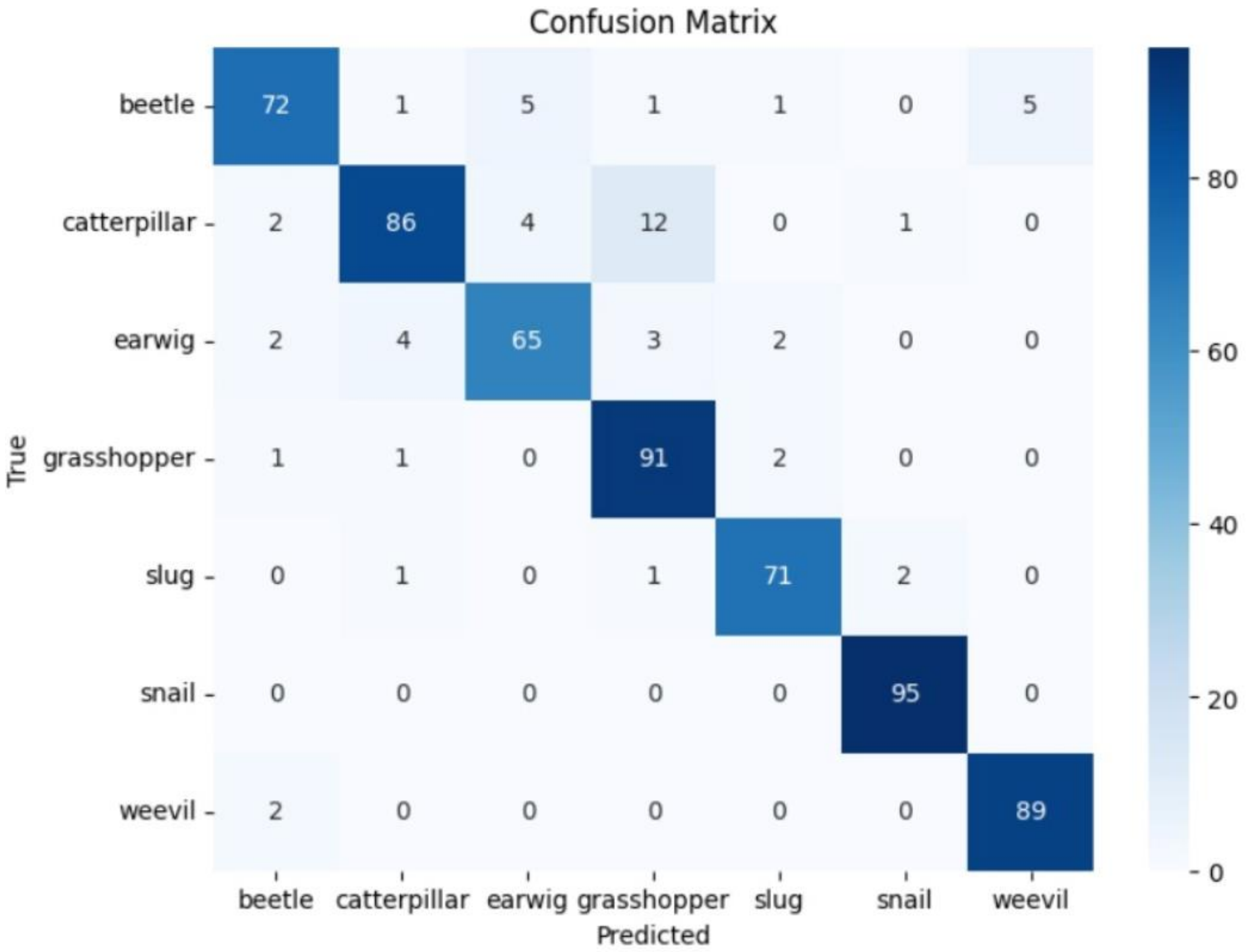


Accuracy
training_accuracy
val_accuracy
0.95
0.90
0.85
0.80
0.75
0.70
0 2 4 6 8 10 12 14
Epochs



Loss
training_loss
val_loss
1.2
1.0
0.8
0.6
0.4
0.2
0 2 4 6 8 10 12 14
Epochs

Confusion Matrix



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

```
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

•The overall test accuracy of the model is **91.48%**, indicating that the model correctly classified 91.48% of the pest samples in the test dataset.

•The **precision** score is **91.59%**, which demonstrates the model's ability to minimize false positives.

•The **recall** score is **91.48%**, reflecting the model's capacity to identify all relevant samples for each pest class.

•The **F1-score** is **91.40%**, balancing precision and recall.

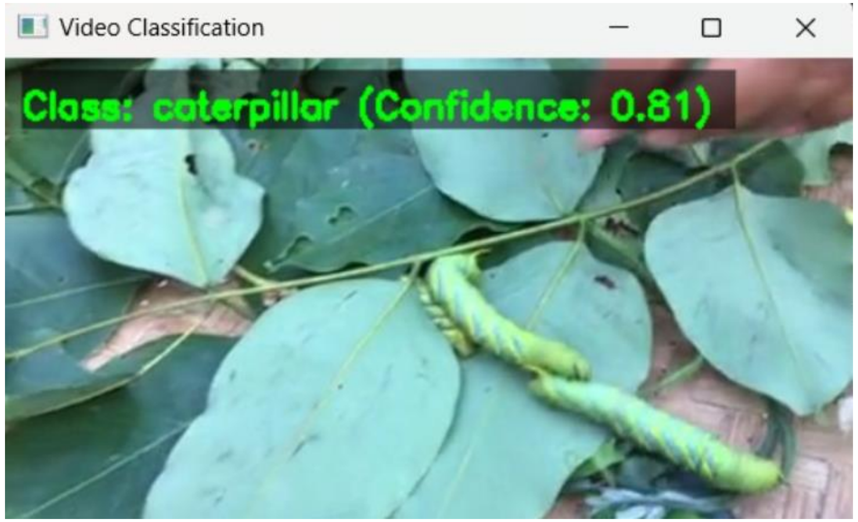
True: caterpillar
Pred: caterpillar, Prob: 0.98

True: snail
Pred: snail, Prob: 1.00

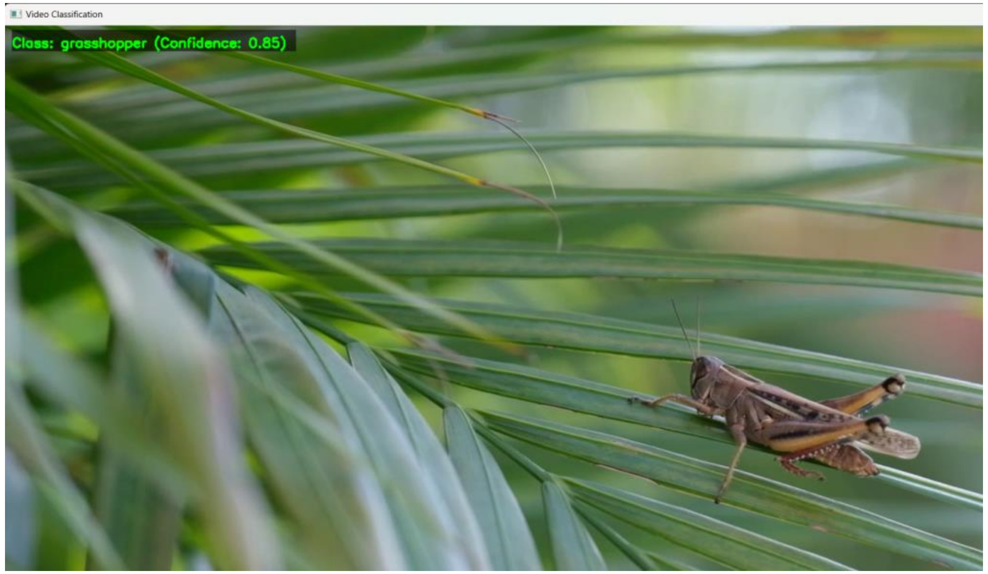
True: beetle
Pred: beetle, Prob: 0.79

True: slug
Pred: slug, Prob: 0.98

True: weevil
Pred: weevil, Prob: 1.00


Video Classification
Class: caterpillar (Confidence: 0.81)

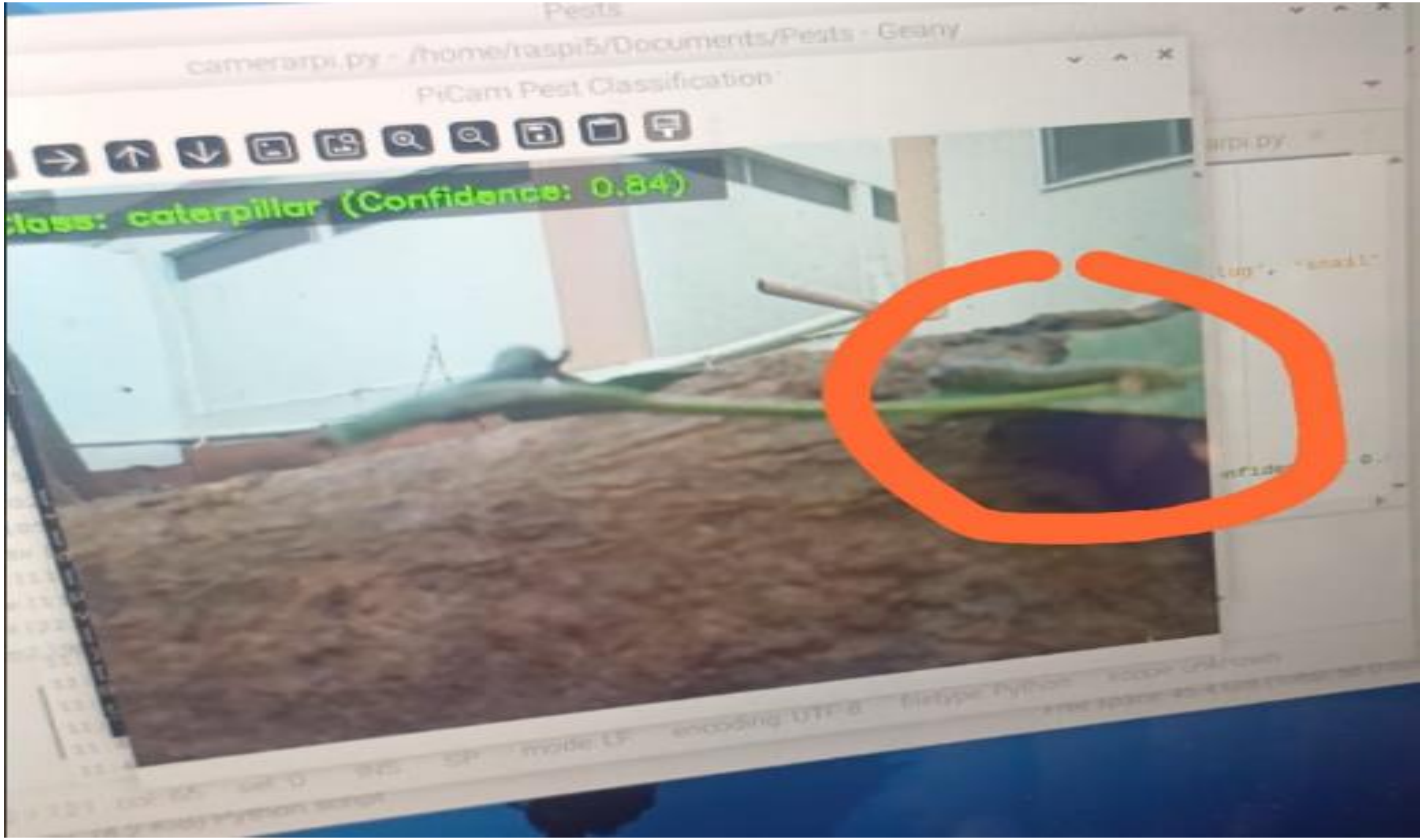

Video Classification
Class: beetle (Confidence: 0.91)


Video Classification
Class: grasshopper (Confidence: 0.85)


Class: earwig (Confidence: 0.82)

Result Video

RESULT



Pests
camerarpi.py - /home/raspi5/Documents/Pests - Geany
PiCam Pest Classification
class: caterpillar (Confidence: 0.84)

Conclusion

The project implemented a real-time pest detection system integrated with a robotic platform. By fine tuning a pre-trained EfficientNet model , the system demonstrated effective and pest classification in live video streams, operational on the Raspberry Pi 5. The synergy of deep learning and edge computing resulted in a cost-effective solution tailored for agricultural applications.

Future Work

The system's navigation can be enhanced by integrating SLAM techniques for adaptable movement across diverse terrains and implementing advanced path-planning algorithms to improve obstacle avoidance and overall efficiency.

Future work can focus on incorporating automated pest control mechanisms, such as pesticide spraying and severity-based responses, to provide targeted interventions based on pest population density and identification.

Edge optimization can be achieved by refining TensorFlow Lite models for faster inference and reduced power consumption, alongside exploring more powerful edge devices like NVIDIA Jetson Nano or Coral Edge TPU.

Extensive field testing, including open-field deployment and large-scale scalability assessments, is essential to validate the system’s performance under real-world conditions and ensure its effectiveness in precision agriculture.

References

Kaur, S., Sharma, M., & Gupta, R. (2023). *Deep Learning for Pest Detection: A Review*. International Journal of Advanced Computer Science and Applications, 14(3), 45-56.

Sharma, R., Kumar, A., & Singh, V. (2022). *Real-time Insect Pest Detection and Classification using YOLOv5*. Proceedings of the International Conference on Computer Vision and Machine Learning, 223-230.

Zhao, X., Li, F., & Zhang, Y. (2020). *Pest Classification in Maize Crops using CNN*. Computer and Electronics in Agriculture, 167, 104-112.

Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*. arXiv preprint arXiv:1804.02767.