Building a Robust, Autonomous Pest-Control Vehicle for Real-World Agricultural Deployment

Bennett Huang, Jason Pan

***Abstract*— Globally, pests destroy over 40% of crops annually. Current application methods include inefficient hand-spraying and the use of machines that conduct wasteful mass-spraying which pollutes the environment and harms plants.**

In this paper, we present an autonomous pest-control vehicle that applies pesticides only when pests are detected, conducting targeted pest-control in real-world agricultural environments. We first designed a custom detection pipeline using SIFT-based keypoints and a YOLOv5-inspired network to recognize small pests accurately in real-time. Then, once a pest is detected, the vehicle will turn a cam piece that presses down a spray nozzle, precisely spraying a small amount of pesticides. The spraying mechanism is attached to two linear actuators, allowing it to reach varying plant heights. In our tests, the vehicle's precise application yielded similar results compared to traditional mass-spraying while reducing pesticide use by over 90%.

I. Introduction

According to the Food and Agricultural Organization (FAO), pests and their associated plant diseases account for up to 40% of global crop losses, representing hundreds of billions of dollars in economic damage [1]. Furthermore, MIT's Water and Food Systems Lab states that only 2% of pesticide chemicals used actually reach target pests, with most of the waste harming the environment and workers [2]. These limitations highlight the need for systems that can deliver precise pest-management in real-world agricultural environments.

However, detecting and classifying pests remains a challenge due to their small size and similar details. Current methods like the Convolutional Block Attention Module (CBAM) and the Regional-based Convolutional Neural Network (R-CNN) rely on progressively down-sampling feature maps, losing fine-grained spatial detail in the process. We propose a cheap deployment vehicle that houses a custom pest-detection model to autonomously apply pesticides to crop regions only when pests are detected.

II. Methodology

A. Pest-Detection Model

Given an image, the first thing our system does is convert the color to grayscale. Since SIFT-based keypoint extraction and later filtering steps rely on gradient, texture, and intensity information rather than color, this preserves accuracy. After applying grayscale, we then use a two-stage noise filtering setup to enhance textures and edges while removing irrelevant clutter and blurriness. First, a median filter with a 3 x 3 kernel is applied to remove salt-and-pepper noise while preserving edges. Then, a bilateral filter (with a spatial kernel of 9, sigmaColor of 75, and sigmaSpace of 75) is applied to reduce Gaussian noise [7].

We now extract keypoints using a modified variant of the Scale-Invariant Feature Transform (SIFT), where we redefine the keypoint detection objective to better capture small, texture-rich pest structures. In standard SIFT, keypoints are identified as extrema in the Difference-of-Gaussians (DoG) scale space, which primarily reflects intensity contrast and often favors larger, high-contrast background regions [8], [9]. To address this, we calculate the Shannon entropy of an 11-by-11 patch around each keypoint after keypoint proposition, discarding ones that have a score below the 65th percentile. Another way to solve this problem is to introduce a composite response function that integrates both intensity variation and local structural complexity directly into the extrema detection stage of SIFT.

Specifically, for each pixel at scale s, we define a modified response:

$$R(x, y, s) = D(x, y, s) \cdot (1 + \beta \cdot \tilde{H}(x, y)) \cdot \exp(-\alpha s)$$

where $D(x, y, s)$ is the standard DoG response, $\tilde{H}(x, y)$ is the normalized local Shannon entropy computed over a small neighborhood, and α and β are scaling constraints. The entropy term promotes regions with high micro-texture variation, while the exponential term biases detection toward finer spatial scales, which are more likely to correspond to small pests [11]. We used the simpler first version for testing.

For every keypoint that is now left over, SIFT assigns an orientation based on the dominant gradient direction. In the end, each keypoint will be represented by a 128-dimensional descriptor [10]. Because this process is computationally intensive, we cap the number of identified keypoints at 50 to 100 per image, prioritizing those with the highest response scores, and significantly improving its speed.

However, to avoid overly aggressive filtering that could eliminate accurate keypoints, we applied an edge-based restoration step using a Canny edge map. The edge detector identifies edges by locating regions of rapid intensity change, which it does by using gradient magnitude and non-maximum suppression to produce thin, well-localized contours [12]. Any filtered keypoints that coincide with strong edges are reinstated, as edges often delineate object boundaries even if their overall entropy is low. With this additional filtering process, we now have a set of promising points in areas of high interest.

Now, the remaining keypoints are grouped into candidate regions using the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [13]. For each cluster,

we compute the bounding box by taking the minimum and maximum x- and y-coordinates of its keypoints. The score of each cluster is then defined as the sum of its associated entropy values. Clusters are then ordered from the highest score to the lowest score, and a non-maximum suppression (NMS)-like step is applied to ensure that only non-overlapping, high-value boxes remain. These boxes are then finally passed to YOLOv5 for the final object detection.

The final detection leverages the You Only Look Once (YOLOv5s) model to classify and localize objects within the candidate regions produced prior. Each bounding box will be applied with an adaptive padding of 0.05 proportional to the box size and then cropped out of the original images to ensure the target object is fully covered. These cropped regions are then passed in one-by-one through the YOLOv5, which uses a convolutional backbone and feature pyramid network (FPN) to extract multiscale features [14], [15]. This model only needs a single forward pass for each image or cropped section, making it extremely efficient. We further use the small model (YOLOv5s) to balance speed and accuracy.

The model is pre-trained on the Common Objects in Context (COCO) dataset [16] and then fine-tuned on our own image datasets. We trained the model primarily on caterpillars and also other types of pests like aphids and whiteflies. During training, data augmentation was implemented to improve the model's robustness and prevent overfitting.

B. Pest-Control Vehicle

The deployment vehicle is designed as a two-platform system. The lower platform (Platform 1) houses all the essential electronics and executes vehicle movement (See fig:13).

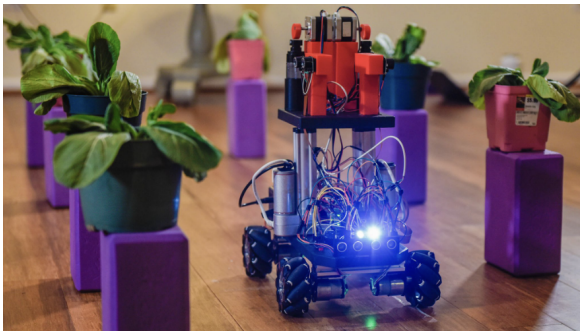

Fig. 1. Dual-platform pest-control vehicle carrying out pest management.

The top platform (Platform 2) was mounted above the lower one and houses the spraying system designed to precisely spray pesticides and contains precision cameras for pest-detection. The two platforms are connected by two 12-volt 5-amp VEVOR linear actuators, which allow the upper platform to move vertically. The linear actuators can reach heights of over a full foot from the base, allowing the spraying system (on the second platform) to reach different levels of plant height for smaller crops.

The vehicle's base utilizes a four-wheel mecanum wheel system for precise movement [20]. Mounted on the first level of the vehicle are four 3.3-volt HC-SR04 ultrasonic sensors located on the front and sides of the vehicle [21]. Ultrasonic sensors use sound waves to detect nearby obstacles and determine how far they are from the vehicle. This allows our vehicle to map out open areas and avoid blocked regions by scanning its surroundings. Combining ultrasonic sensors with our mecanum wheels allows the vehicle to move safely throughout the entire field while avoiding obstacles [22], [23].

On the second platform, the vehicle uses two Raspberry Pi V3 camera modules to scan the field and look for pests. This specific camera version gives us a large resolution 2304 x 1296 pixels, but more importantly, it has powered autofocus. This allows the camera to better focus and therefore capture objects even with the external vehicle movement going on simultaneously. On the two sides of each Raspberry Pi camera are infrared LED lights that each have an internal light sensor [24]. When the environment gets dark, the infrared LED lights will light up, allowing the Raspberry Pi camera to continue detecting pests. This makes the vehicle usable during the day and night.

Then, once pests are detected, we use a motor-cam spraying system to spray the pesticides. This consists of a 12-volt DC high-torque stepper motor with an irregular red cam piece attached to it that was custom designed and 3D printed. Below it is the spray bottle containing the pesticides (See fig:13). Essentially, due to the irregular shape of the cam piece, the cam presses down on the sprayer exactly once each time the motor makes a full rotation, spraying the solution in a controlled manner. This allows us to accurately control the amount of pesticides sprayed based on the number of times we tell the motor to rotate.

III. Results

We now compare the overall pest-detection model with other prominent models that can easily be added onto YOLOv5 on small objects. These images consisted of a variety of small pests, and were also specifically chosen to be taken in real-world agricultural environments, allowing us to evaluate the efficacy of our model in the wild.

Model	Precision	Recall	F1	Confidence	IoU
YOLOv5s - S	13.3	**16.0**	14.0	48.1	25.6
YOLOv5s + MSI - S	15.0	28.0	18.1	51.4	49.9
YOLOv5s + TTA - S	**8.7**	24.0	**12.5**	**47.1**	**25.2**
YOLOv5s + SAHI - S	25.6	72.0	32.9	**85.7**	**79.3**
Created Model - S	**70.0**	**90.0**	**76.0**	73.9	75.8

Fig. 2. Performances of different types of small object detection tools with YOLOv5 (MSI, TTA, SAHI, SIFT) on detecting and classifying three different types of pests.

Finally, we tested the entire vehicle system in a simulated environment. We set up multiple crop rows, and planted pests onto some of the plants. This was recorded in a demonstration video and we are happy to provide it upon request in the future. Ongoing work focuses on validating the system under real-world field conditions.

REFERENCES

[1] Food and Agriculture Organization of the United Nations. Climate Change Fans Spread of Pests and Threatens Plants and Crops. 2021.

[2] MIT Abdul Latif Jameel Water and Food Systems Lab (J-WAFS). Reducing Runoff and Environmental Impact of Agricultural Sprays. 2017.

[3] WelkinU. Among Us Player Tracking with YOLOv5 and SIFT. GitHub repository.

[4] Z. Jin et al. DWCA-YOLOv5: An Improved Single Shot Detector. *Journal of Sensors*, 2021.

[5] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. CBAM: Convolutional Block Attention Module. In *ECCV*, 2018.

[6] E. Akbarnezhad and F. Naserizadeh. Improving Camouflage Object Detection Using U-Net and VGG16 with CBAM. In *Proc. QICAR*, 2024.

[7] J. Shen, Q. Huang, Z. Ding, and J. Liu. A Comparative Analysis of Image Denoising Filters. In *Proceedings of the 2025 5th International Conference on Computer Network Security and Software Engineering (CNSSE)*, pages 128–132, 2025.

[8] V. Mousavi, M. Varshosaz, and F. Remondino. Using Information Content to Select Keypoints for UAV Image Matching. *Remote Sensing*, 2021.

[9] Y. Tian, Q. Ye, and D. Doermann. YOLOv12: Attention-Centric Real-Time Object Detectors. *arXiv*, 2025.

[10] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[11] C. Cheng Single-image Background Removal with Entropy Filtering. In *SCITEPRESS Conference Proceedings*, pages 1030–1037, 2021.

[12] K. Muntarina, R. Mostafiz, F. Khanom, S. B. Shorif, and M. S. Uddin. MultiResEdge: A Deep Learning-Based Edge Detection Approach. *Intelligent Systems with Applications*, 20:200274, 2023.

[13] J. Glas Espinel, V. Alvarado Espinel, S. León Abad, and J. Parra Fonseca. DBSCAN Clustering Algorithm Based on Density. In *IEEE Xplore Conference Proceedings*, 2024.

[14] R. Khanam and M. Hussain. What is YOLOv5: A Deep Look into the Internal Features of the Popular Object Detector. *arXiv*, 2024.

[15] S. Praveen and Y. Jung. CBAM-STN-TPS-YOLO: Enhancing Agricultural Object Detection Through Spatially Adaptive Attention. *arXiv*, 2025.

[16] T.-Y. Lin, M. Maire, S. Belongie, et al. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014.

[17] M. Kimura. Understanding Test-Time Augmentation. *arXiv*, 2024.

[18] X. Liang, J. Xiang, S. Qin, Y. Xiao, L. Chen, D. Zou, H. Ma, D. Huang, Y. Huang, and W. Wei. Small Target Detection Algorithm Based on SAHI-Improved-YOLOv8 for UAV Imagery: A Case Study of Tree Pit Detection. *Smart Agricultural Technology*, 12:101181, 2025.

[19] F. C. Akyon, S. O. Altinuc, and A. Temizel. Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection. In *Proc. ICIP*, 2022.

[20] Q. Jia, M. Wang, S. Liu, J. Ge, and C. Gu. Research and Development of Mecanum-Wheeled Omnidirectional Mobile Robot Implemented by Multiple Control Methods. In *International Conference on Mechatronics and Machine Vision in Practice*, 2017.

[21] S. A. K. Tareen, Z. Saleem, M. Koeda, and K. Watanabe. Logic Level Conversion Method in Serial Data System. *Procedia Engineering*, 29:1539–1543, 2012.

[22] X. Wu, G. Wang, and N. Shen. Research on Obstacle Avoidance Optimization and Path Planning of Autonomous Vehicles Based on Attention Mechanism Combined With Multimodal Information Decision-Making of Robots. *Frontiers in Neurorobotics*, 17:1269447, 2023.

[23] A. Topiwala, P. Inani, and A. Kathpal. Frontier-Based Exploration for Autonomous Robots. *arXiv*, 2018.

[24] Adafruit Industries. Adafruit DC and Stepper Motor HAT for Raspberry Pi. 2025.