Journal of Computer Science and Frontier Technologies
ISSN Print: 3104-4204
ISSN Online: 3104-4212
CD ISSN: 3104-4220
https://doi.org/10.63313/JCSFT.9107

Vision Based Robotic Autonomous Refueling System for Highway Service Areas

Rongrong Zhu. Xiaofang Zhang*
School of Intelligent Engineering, Shaoguan University, Shaoguan 512000, China
Email: findlabzrr@163.com

How to cite this paper: Zhu, R., & Zhang, X. (2026). Vision based robotic autonomous refueling system for highway service areas. Journal of Computer Science and Frontier Technologies, 4(3), 51–58. ISSN Print: 3104-4204, ISSN Online: 3104-4212.
https://doi.org/10.63313/JCSFT.9107
Published: 2026-10-26

Copyright © 2026 by author(s) and Erytis Publishing Limited.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
http://creativecommons.org/licenses/by/4.0/

Abstract

Highway service-area fueling stations face recurring challenges in labor cost, centralized management, and operational safety. This paper presents a prototype autonomous visual unmanned gas-station system that combines YOLOv5s object detection, a Raspberry Pi-controlled two-degree-of-freedom manipulator, and an ESP32-based Internet-of-Things (IoT) interface. A camera detects the fuel-inlet proxy target, and its image coordinates are transmitted through a serial link to an incremental PI controller that aligns the nozzle. Users select a fueling amount through a mobile interface connected to the Bafa Cloud platform. A 900-image dataset was collected under varied viewpoints and backgrounds; the model was trained for 400 epochs with an 80/20 train–validation split. The prototype achieved detection accuracies above 80% on the tested target shapes, while nozzle positioning errors remained below 3 cm. The results demonstrate the feasibility of integrating lightweight deep learning, embedded control, and cloud interaction for unattended fueling, while also revealing the need for larger real-vehicle datasets and additional safety sensing.

Keywords

Autonomous fueling; Computer vision; YOLOv5; Robotic manipulator; Internet of Things

1. Introduction

The rapid growth of highway travel has increased demand for reliable fueling at service areas. Conventional stations require repeated staffing, are difficult to manage consistently across locations, and expose operators to mechanical and flammable environments. Autonomous fueling can reduce these burdens, but it requires robust vehicle localization, nozzle alignment, actuator control, and user interaction in one coordinated system. Modern one-stage detectors provide a practical starting point for such real-time visual localization [1, 2].

This work develops and evaluates a low-cost prototype for highway service areas. The contribution is an end-to-end architecture in which a YOLOv5s detector

estimates the fuel-inlet position, a Raspberry Pi drives a servo manipulator through serial communication and incremental PI control, and an ESP32 links fueling commands to a cloud platform and mobile interface. YOLOv5 provides an openly documented implementation family suitable for rapid prototype development [3]. Unlike a detector-only study, the prototype connects perception to physical positioning and user-level operation.

2. System Architecture

The system is organized into four layers: visual perception, embedded motion control, fueling actuation, and IoT interaction. A USB camera streams frames to a computer running Python, OpenCV, and PyTorch. YOLOv5s returns the target bounding box and center coordinates. These coordinates are sent at 9600 baud to a Raspberry Pi 4B, which drives a two-degree-of-freedom servo platform through a PCA9685 controller. An ESP32 provides Wi-Fi connectivity to the Bafa Cloud TCP/HTTP service, where a mobile App Inventor interface sends the selected fueling amount. This cloud-connected arrangement follows the broader IoT pattern of linking sensing, computation, and remote services [4].

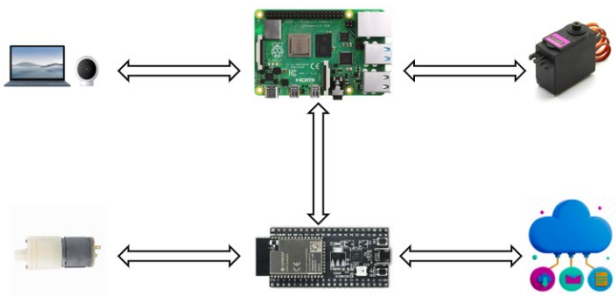


Figure 1. Hardware and communication architecture of the prototype

The design emphasizes inexpensive, readily available components: a 1080p USB camera, Raspberry Pi 4B, MG996R metal-gear servos, PCA9685 servo driver, ESP32 module, and relay interface. This modular split allows perception and cloud services to run independently from the real-time actuator loop, simplifying maintenance and later replacement of individual modules.

3. Visual Detection and Dataset

YOLOv5 is a one-stage detector with an input-backbone-neck-head pipeline. The YOLOv5s variant was selected because it offers the lowest computational demand among the tested model sizes and is therefore suitable for a resource-constrained prototype. Its design follows the real-time, single-stage detection paradigm established by YOLO-type models. Mosaic augmentation, adaptive anchor computation, and adaptive image scaling improve robustness to viewpoint and target-size changes. The backbone extracts features with Focus/C3/SPP-style

modules, while the neck fuses multi-scale features using spatial pyramid pooling and path aggregation.

The detector evaluates overlap between a predicted box B and a ground-truth box A using intersection-over-union (IoU), defined in (1). To provide a useful gradient when the boxes do not overlap, generalized IoU (GIoU) is used in (2), where C is the smallest enclosing box containing A and B [5]. This loss formulation is relevant to the present task because the target opening occupies a small region and early predictions can be substantially displaced.

$$IoU(A,B) = \frac{area(A \cap B)}{area(A \cup B)} \quad (1)$$

$$GIoU(A,B) = IoU(A,B) - \frac{area(C - (A \cup B))}{area(C)} \quad (2)$$

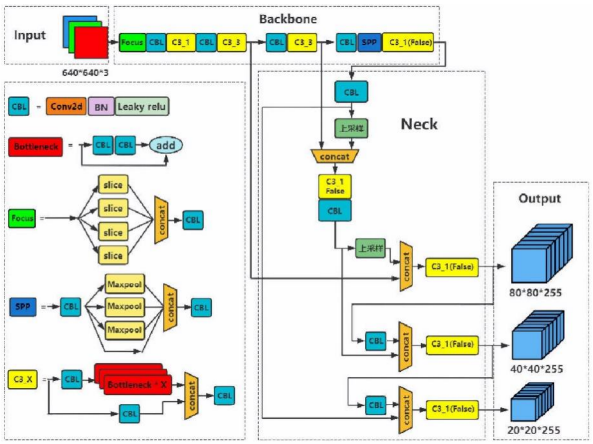


Figure 2. YOLOv5 network structure used for target localization

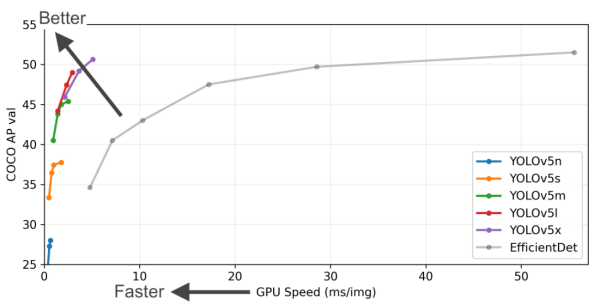


Figure 3. Comparison of YOLOv5 model variants in speed and accuracy

Because collecting real fuel-inlet images is difficult during prototype development, plastic bottle openings were used as geometric proxies. Images were captured at different angles, distances, and backgrounds and supplemented with online samples, producing 900 labeled images. LabelImg annotations were converted to YOLO text format; 80% of the images were used for training and 20% for validation. This variation in training views is consistent with the role of data augmentation in

improving visual-model generalization [6]. Training used PyTorch 1.12.1 and Python 3.8 on an Intel i5-9300H CPU with an NVIDIA GTX 1650 GPU. The YOLOv5s weights were initialized from yolov5s.pt, with a batch size of 4 and 400 epochs.

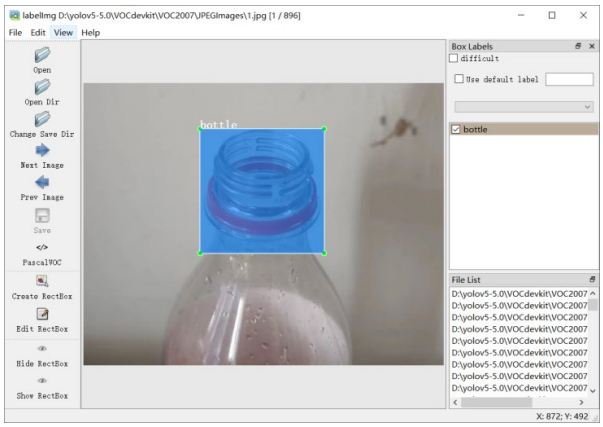


Figure 4. Example of the annotated training image

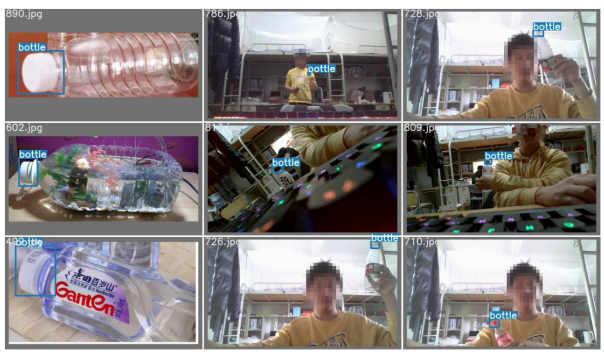


Figure 5. Representative samples in the collected dataset

4. Embedded Control

The detector provides a target center (x, y), while the camera reference point is set to (400, 300). The Raspberry Pi computes the horizontal and vertical errors and updates the servo commands using an incremental PI controller. In the general incremental PID form, the control increment depends on the current and two previous errors; the derivative term was set to zero in this prototype to reduce sensitivity to image noise. This use of PI/PID feedback follows established controller-design practice [7]. Once the target is within the alignment tolerance, the relay and nozzle actuation sequence can be enabled.

The incremental controller can be written as:

$$\Delta u(k) = K_p[e(k) - e(k-1)] + K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2)] \quad (3)$$

where e(k) is the image-coordinate error at step k. In this implementation Kd = 0, so the controller behaves as an incremental PI regulator. The target is considered aligned when both coordinate errors fall within the experimentally selected

tolerance.

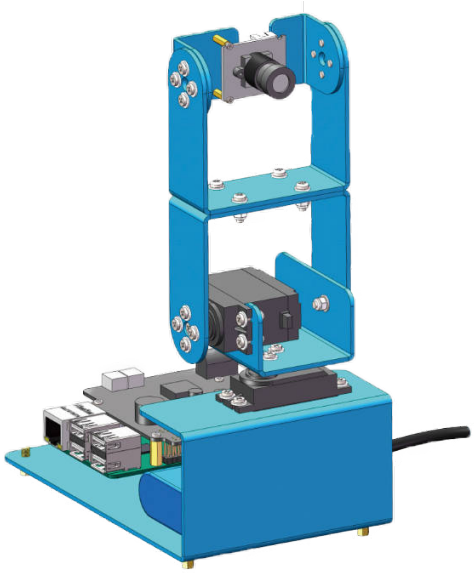


Figure 6. Two-degree-of-freedom servo manipulator

For user interaction, the ESP32 connects the station to Bafa Cloud through Wi-Fi. The mobile interface, implemented with App Inventor, transmits a selected amount through the cloud channel. The ESP32 parses the message, maps the amount to a fueling command, and controls the station relay. This arrangement separates account or interface logic from the embedded actuator logic and supports remote status exchange.

5. Experimental Results

Training curves showed that the intersection-over-union (IoU) based mAP at 0.5 increased as training progressed, while box-regression loss decreased. The 400-epoch model was selected because it produced more stable detections than the 120- and 200-epoch alternatives. Across the tested target shapes, recognition accuracy exceeded 80%, indicating that the detector could provide a usable position estimate for the downstream controller.

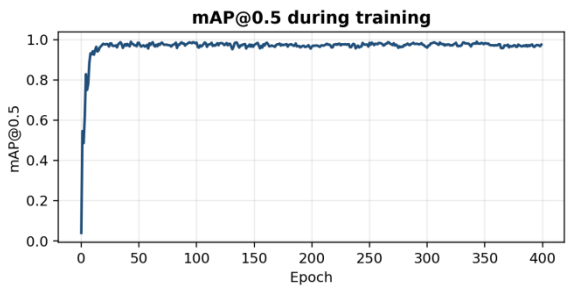


Figure 7. mAP@0.5 over the 400 training epochs

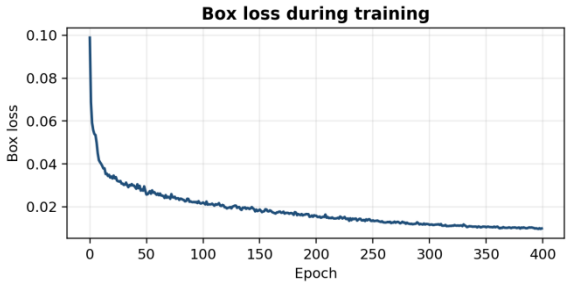


Figure 8. Box-regression loss over the 400 training epochs

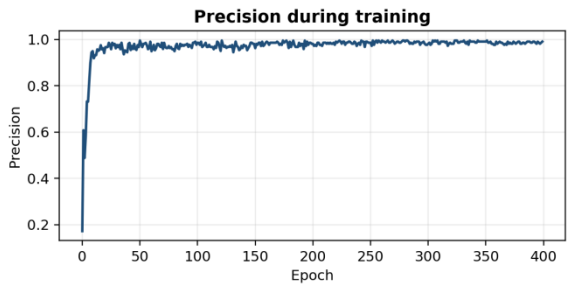


Figure 9. Precision curve over the 400 training epochs

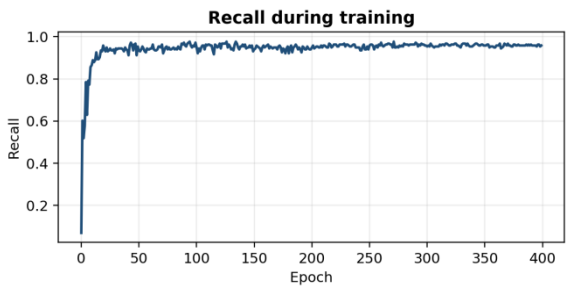


Figure 10. Recall curve over the 400 training epochs

The training diagnostics provide complementary evidence. The mAP@0.5 curve reflects the proportion of predictions whose intersection-over-union with the annotation exceeds 0.5. The box-loss curve tracks localization error during optimization, whereas precision and recall describe the trade-off between false positives and missed targets. Taken together, the curves indicate that the detector converged progressively rather than relying on a single favorable frame. In practical deployment, the confidence threshold and non-maximum suppression settings should be tuned jointly with the manipulator tolerance, because a visually plausible box is not sufficient if its center produces an unsafe nozzle trajectory.



Figure 11. Detection results on representative target images

In the physical positioning test, the difference between the nozzle position and the target center was below 3 cm for the tested cases. A complete user flow was demonstrated: the user opens the mobile interface, enters a fueling amount, the command reaches the ESP32, and the manipulator aligns the nozzle with the detected target. These results validate the integration path from image acquisition to cloud command and mechanical response, but they should be interpreted as prototype-level evidence rather than field performance.

Table 1. Core prototype test results reconstructed from the source thesis

Test Case	Accuracy	Error (Cm)
1	0.82	1.8
2	0.91	1.4
3	0.95	2.3
4	0.94	1.8
5	0.85	2.0
6	0.93	1.9
7	0.88	1.7
8	0.84	2.4
9	0.97	2.1
10	0.98	1.1

6. Discussion and Limitations

The prototype shows that a low-cost combination of YOLOv5s, Raspberry Pi control, and ESP32 IoT communication can support autonomous fueling functions. The main engineering advantage is the clear separation of perception, motion, and cloud layers. However, the evaluation used bottle openings as proxies rather than a diverse real-vehicle dataset, and the reported accuracy was measured on a limited set of target shapes. The current design also lacks dedicated fire, leak, human-presence, and emergency-stop sensing. Such functions require a safety lifecycle and suitable flammable-gas detection rather than software-only safeguards [8,9]. Future work should collect annotated images from real vehicles and lighting conditions, add redundant safety interlocks, improve nozzle–vehicle geometric calibration, and evaluate latency and reliability over long unattended operation.

7. Conclusion

An autonomous visual unmanned gas-station prototype for highway service areas was designed and tested. YOLOv5s localized the fuel-inlet proxy, a Raspberry Pi and servo manipulator converted image coordinates into nozzle motion, and an ESP32/Bafa Cloud link enabled mobile fueling commands. With 900 training images, the prototype achieved above-80% detection accuracy on tested shapes and

positioning errors below 3 cm. The work provides a practical integration baseline for future real-vehicle, safety-certified autonomous fueling systems.

References

[1] Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934.

[3] G. Jocher et al., "ultralytics/yolov5: v3.1," Zenodo, Oct. 2020, doi: 10.5281/zenodo.4154370.

[4] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," IEEE Commun. Surveys Tuts., vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.

[5] H. Rezatofighi et al., "Generalized Intersection over Union: A Metric and a Loss for Bounding Box Regression," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), 2019, pp. 658–666, doi: 10.1109/CVPR.2019.00075.

[6] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," J. Big Data, vol. 6, art. no. 60, 2019, doi: 10.1186/s40537-019-0197-0.

[7] K. J. Åström and T. Hägglund, PID Controllers: Theory, Design, and Tuning, 2nd ed. Research Triangle Park, NC, USA: Instrument Society of America, 1995.

[8] International Electrotechnical Commission, Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems—Part 1: General Requirements, IEC 61508-1:2010, 2010.

[9] International Electrotechnical Commission, Explosive Atmospheres—Part 29-1: Gas Detectors—Performance Requirements of Detectors for Flammable Gases, IEC 60079-29-1:2016, 2016.