ISBN: 978-628-96613-3-0. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2026.1.1.1833

An Intelligent Vision-Based Feeding and Hydration System for Small Domestic Rodents

Alex Fiallos[1]; Hugo Caizaluisa[1]; José Salazar[1];Angélica Quito[1]; Faruk Abedrabbo[1]
[1]International University of Ecuador UIDE, Ecuador, *alfiallosri@uide.edu.ec*, *hucaizaluisapa@uide.edu.ec*, *josalazarva@uide.edu.ec*, *anquitoca@uide.edu.ec*, *anabedrabboha@uide.edu.ec*

Abstract*– This paper introduces a cost-effective intelligent feeding and hydration platform for small domestic rodents. The system integrates computer vision, mechatronic actuation, and Internet of Things (IoT) connectivity within a unified perception–decision–action framework. A YOLOv8-S model identifies Russian hamsters, Syrian hamsters, and guinea pigs, dispensing species-specific portions of 5 g, 8 g, or 10 g. The ESP32 microcontroller manages a 3D-printed rotary dosing wheel actuated by an SG90 servomotor, verifies dispensed mass using a 1 kg load cell with an HX711 amplifier, and prevents empty-container operation using an infrared sensor. Hydration is controlled by a water-level sensor and dual submersible pumps for filling and drainage. Data and alerts are transmitted via MQTT and visualised through a Blynk mobile interface, enabling real-time monitoring and manual override. Experimental results from 60–40 trial batches demonstrate stable sensing and communication. The vision model achieves 99.667% test accuracy on 82 images from a balanced 819-image dataset, supporting reliable autonomous care.

Keywords-- automation, pets care, ESP32, Image detection.

I. Introduction

Responsible care for small pets, particularly rodents such as hamsters and guinea pigs, requires consistent feeding schedules tailored to their biological and nutritional needs. Irregular feeding schedules or uncontrolled portion sizes may induce stress, behavioural changes, and chronic health issues. Previous studies indicate that food deprivation and chronic stress in rodents are linked to disruptions in circadian rhythms, increased hoarding behaviour, hyperphagia, and weight gain, all of which negatively impact animal welfare [1]–[3].

Automated feeding systems represent an effective alternative to manual feeding by ensuring consistent portion control, minimising human intervention, and enhancing feeding regularity. Developments in embedded systems, low-power sensors, and wireless connectivity have enabled the creation of smart feeders capable of real-time monitoring and remote operation via the Internet of Things (IoT) [A1] [4-4a]. Evidence from both domestic and laboratory settings suggests that these systems reduce stress, support stable feeding patterns, and improve the overall health of small animals [5]–[6].

Despite these technological advances, most commercially available feeding solutions utilise fixed or manually adjustable portions and fail to address species-specific nutritional requirements. This limitation is especially significant for small rodents, whose dietary needs differ considerably between species, frequently leading to overfeeding or inadequate intake [7]–[8]. Therefore, there is a clear need for intelligent feeding systems that can dynamically adjust food portions based on the specific animal being fed.

Recent advancements in computer vision and deep learning have enabled reliable real-time animal recognition across diverse environmental conditions. Convolutional neural networks and single-stage detectors, such as YOLO, have demonstrated high accuracy and rapid inference, with YOLOv8 proving particularly suitable for embedded and distributed IoT applications [9]–[12]. However, most current approaches emphasise perception and do not integrate adaptive mechatronic actuation mechanisms [13]–[14].

To address this gap, Hamsterzinho is introduced as an intelligent feeding and water-supply system that integrates computer vision, IoT communication, and mechatronic control.

Literature review

Numerous studies have investigated the use of microcontrollers and IoT platforms to automate pet feeding. For instance, Camuñas-Martínez et al. developed an ESP32-based feeder with a web interface, which enabled remote monitoring and system control [18]. Jain et al. introduced a smart dispenser designed to track feeding habits and prevent obesity, although this system was restricted to single-pet scenarios [19]. Additional Arduino-based and time-programmed solutions prioritised low cost and ease of implementation but did not incorporate adaptive mechanisms or real-time feedback [20].

Recent research has incorporated artificial intelligence techniques to enhance feeding personalisation. Castillo-Arceo et al. introduced a system that integrates IoT and deep learning to identify pets in real time and automatically adjust food portions, though this approach requires significant computational resources [21]. Similarly, Vijayalakshmi et al. developed AI-based systems that prevent cross-feeding among multiple pets and provide personalised portions, which increases both system complexity and cost [22].

Concurrently, advances in computer vision have enabled the development of robust animal recognition models. Convolutional neural networks have shown strong performance in animal identification and monitoring tasks across diverse lighting and environmental conditions [23]. Within this domain, single-stage object detectors such as YOLO have enabled real-time detection with competitive accuracy [24]. Enhancements in the YOLOv8 architecture have further improved inference speed and model robustness, supporting integration into distributed IoT systems [25].

Finally, in the case of hamsters, most commercially available products are non-automated mechanical dispensers that operate by gravity. Although these devices are practical and inexpensive, they lack sensors and electronic controls and therefore cannot guarantee precise dosing or monitor food levels. Nevertheless, such dispensers represent a suitable base for custom integrations using servomotors and microcontrollers, allowing students or developers to transform them into personalised automatic feeders. However, this type of solution is not formally documented in academic literature [26].

Table I provides a detailed comparison of the characteristics, similarities, and limitations of previously reported work.

TABLE I
SUMMARY OF LITERATURE REVIEW AUTOMATED FEEDING SYSTEMS

Reference	Key Components	Advantages	Limitations
Wood and Bartness [15]	Behavioral analysis, feeding control	Demonstrates impact of feeding regularity on rodents	No automation system proposed
Scherbarth and Parfitt [16]	Feeding schedules, circadian analysis	Links feeding routines with animal welfare	No technological implementation
Moran and Delville [17]	Stress models, metabolic analysis	Shows long-term effects of stress on feeding	Experimental focus only
Camuñas-Martínez et al. [18]	ESP32, sensors, web interface	Remote monitoring and automated feeding	Fixed portion logic
Jain et al. [19]	Embedded system, sensors	Tracks feeding habits, obesity prevention	Single-pet limitation
Wong et al. [20]	Arduino, RTC, solar panel	Low-cost and energy-efficient design	Weather-dependent, no adaptability
Castillo-Arceo et al. [21]	Camera, deep learning, IoT	Real-time pet recognition and portion control	High computational cost
Vijayalakshmi et al. [22]	AI, IoT, multi-pet management	Prevents cross-feeding, tailored portions	Complex and costly system
Kamilaris and Prenafeta-Boldú [23]	CNN-based models	Robust animal recognition	Perception-only focus
Redmon et al. [24]	YOLO object detection	Real-time detection with good accuracy	No actuation or feeding logic
Jocher [25]	YOLOv8 architecture	Improved inference speed and robustness	Requires external processing unit
Wong et al. [26]	Mechanical dispensers, gravity-based systems	Simple, low-cost solution	No sensors or precise dosing

The design and implementation of the proposal, an intelligent automated feeding and hydration system for small domestic rodents, is described. This system integrates computer vision, mechatronic actuation, and IoT communication within a cost-effective platform. A YOLOv8-S–based model enables real-time identification of Russian hamsters, Syrian hamsters, and guinea pigs, enabling automatic adjustment of food portions based on the detected species.

A significant aspect of the mechanical design is the rotary dosing mechanism, which is fabricated using PETG 3D printing. This mechanism features a top inlet connected to the food container and a bottom outlet aligned with the feeding plate. The configuration enables gravity-assisted, repeatable food delivery, reduces clogging, and enhances dosing accuracy.

Beyond food dispensing, the system incorporates an automated water-supply module utilising miniature submersible pumps to maintain stable hydration and controlled drainage. System management is achieved through an ESP32 microcontroller with Wi-Fi connectivity, employing an MQTT-based communication architecture and a Blynk interface for real-time monitoring and manual control. This solution offers a modular, scalable approach to intelligent small-pet care, enhancing feeding and hydration management through accessible automation.

II METHODOLOGY

The Proposal system integrates computer vision, Internet of Things (IoT) communication, and mechatronic actuation within a unified perception–decision–action workflow for automated feeding and hydration of small domestic rodents. The system employs a YOLOv8-based species-recognition model in conjunction with a calibrated dispensing mechanism to deliver precise, species-specific food portions. The following section provides an overview of the system architecture and details the integration between the vision and dispensing modules. In Fig. 1 the functional and no functional requirements are described.

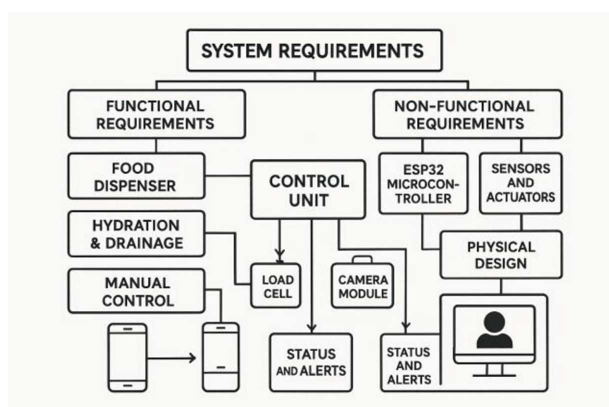


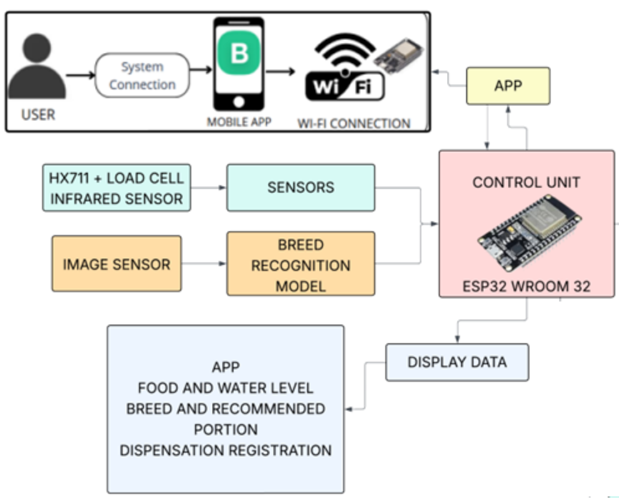
Fig. 1 System Requirements of proposal

System Requirements

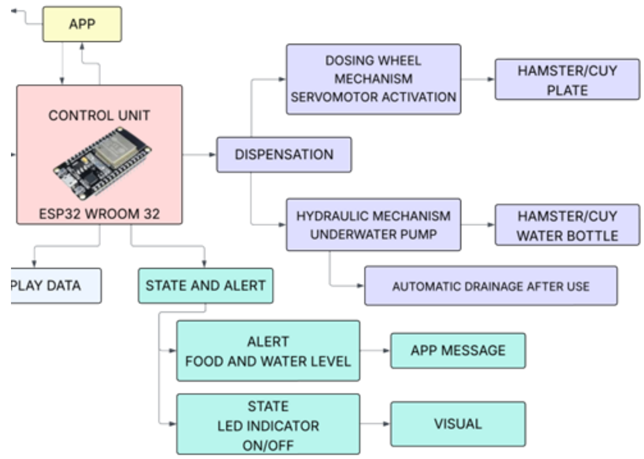
Functional requirements: The system automatically dispenses food and water. A servomotor-driven dosing wheel delivers solid food, while miniature submersible pumps control water supply and drainage based on predefined conditions. A load cell equipped with an HX711 module measures bowl weight to verify portion sizes and ensure adequate food levels. An infrared (IR) sensor monitors food availability in the storage container to prevent operation when the container is empty. Manual activation is available through a web-based interface accessible via the local network. A camera-based module identifies the rodent species and initiates dispensing accordingly.

Non-functional requirements: The platform runs on an ESP32 microcontroller to provide wireless connectivity, sufficient processing performance, and compatibility with low-power components. Sensors and actuators are integrated to minimise interference and reduce operational failures. The physical design is compact, modular, safe for habitats, and uses non-toxic materials such as PLA or PETG. The user interface is intuitive and displays real-time status. Maintenance is minimal (periodic refilling of food and water), and installation is simple without disrupting the animal's environment.

Fig. 2 1 illustrates the block diagram and overall system architecture. A mobile application communicates with the ESP32 microcontroller via Wi-Fi. The control unit integrates sensor data and vision-based recognition to identify the species and determine the appropriate portion size, subsequently actuating the food dispenser and hydration or drainage modules. Status updates and alerts are reported to the user in real time.



a)



b)

Fig. 2 System architecture of proposal:
(a) perception, sensing, and communication layer;
(b) mechatronic food and water dispensing subsystem.

A) *Mechanical Design*

The automated feeding system for hamsters was designed following a functional, compact, and safe approach, adapted to the animal's environment and behaviour. The mechanical structure integrates the food dispensing, weighing, and water management subsystems within a single enclosure, allowing reliable operation and easy installation.

The food dispensing subsystem is based on a rotary dosing wheel driven by a servomotor via a transmission shaft, enabling stable, repeatable delivery of dry food portions. The weighing subsystem, located beneath the food plate, incorporates a load cell that verifies the dispensed mass and supports portion control. In parallel, the water dispensing and drainage subsystem includes a 0.6 L water container, miniature submersible pumps for filling and drainage, a water level sensor, and a hamster drinking bottle, ensuring controlled hydration and hygienic operation.

Fig. 3 shows the real prototype of the Proposal system with numbered labels identifying the food dispensing, weighing, water management, and indicator subsystems. The figure provides a clear visualisation of the spatial arrangement and physical integration of the system components within the enclosure.

Fig.3: Labeled Mechanical Prototype of the Proposal System

In Fig.3 is indicated the parts of the proposal, number 1 is the dispenser of food, number 2 is the food plate, number 3 is the pump of water and number 4 is the container of food.

B) Electronic Design

The system's electronic design focuses on selecting and integrating components to automate the feeding and hydration of hamsters. A central microcontroller coordinates sensor acquisition, control logic, actuation, and wireless communication. Table II summarises the main electronic components used in the Proposal system.

TABLE II
ELECTRONIC COMPONENTS OF THE SYSTEM

Component	Description
ESP32-WROOM-32	ESP32-WROOM-32 (Espress if Systems) microcontroller, dual-core 240 MHz, Wi-Fi and Bluetooth connectivity, powered at 5 V, used for system control, sensor acquisition, and actuator management.
HX711 module	HX711 24-bit analog-to-digital converter for load cells, powered at 5 V, providing high-resolution weight measurements.
1 kg load cell	Strain-gauge load cell with 1 kg maximum capacity and sensitivity ≈ 1.0 mV/V, used to measure the amount of dispensed food.
HW-201 Infrared (IR) sensor	Infrared reflective sensor operating at 5 V, digital output, detection range 2–30 cm, used to detect food presence in the storage compartment.
SG90 servomotor 360°	Continuous-rotation servomotor (generic model, 5 V), PWM-controlled, used to drive the food dispensing mechanism.
RM1 two-channel relay module	RM1 two-channel electromechanical relay module, 5 V control, used to switch the water pumps.
Submersible supply water pump	Mini DC submersible pump (Arduino-compatible), operating at 3–5 V, power 0.4–1.5 W, flow rate 70–120 L/h, maximum head 0.4 m, used for water supply.
ZBCKKING drainage water pump	ZBCKKING DC water pump operating at 5 V, nominal power 5 W, maximum head 2 m, used for water drainage.
Water level sensor	Resistive water level sensor (generic model, 5 V), used to detect low water levels in the system.
LED indicators	LED indicators powered at 5 V with current-limiting resistors, used to display the operational states of the system.

At the circuit level, the ESP32 interfaces with the load cell via the HX711 module for accurate food weight measurement, receives digital signals from the infrared sensor to detect food availability, and monitors the water level sensor to detect low water levels. Actuation is performed through a servomotor for food dispensing and a two-channel relay module that controls the water supply and drainage pumps. Visual feedback is provided by LED indicators that display system status and warning conditions.

Electronic architecture is described at a functional level using a block-based diagram, as shown in Fig. 4. This diagram demonstrates how the ESP32 processes sensor data and translates control decisions into actuation, visual indicators, and wireless communication with the mobile application. The block-based representation is used in place of a detailed electronic schematic to maintain clarity regarding system operation and module interconnections.

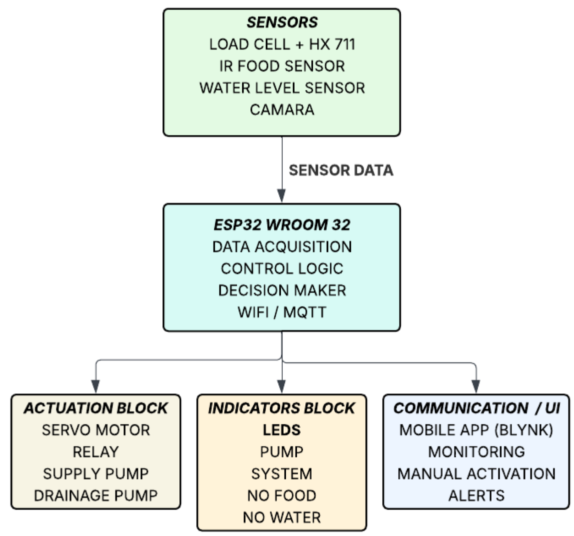


Fig. 4: Functional block diagram of the proposal

C) Detection of the pet

- Hardware Architecture

The hardware architecture of the proposed system is organized into four primary modules: sensing, processing, dispensing, and communication, as shown in Fig. 5. These modules were chosen to provide reliable operation, minimize power consumption, and support real-time performance in autonomous feeding tasks.

The sensing module integrates a digital camera and a weight measurement subsystem. The camera acquires images of the animal during feeding events to facilitate species recognition. Simultaneously, a 1 kg load cell with an HX711 amplifier continuously monitors the food bowl weight, supplying feedback for portion verification.

The processing module is implemented on an ESP32 microcontroller, which manages sensor data acquisition, communicates with the YOLOv8 inference server, and executes the feeding control logic. The ESP32 was selected due to its integrated Wi-Fi connectivity and compatibility with lightweight IoT-based architect he dispensing module comprises a custom 3D-printed rotary feeder actuated by an SG90 servomotor, calibrated to dispense species-specific portions of 5 g, 8 g, and 10 g. The communication module employs the MQTT protocol to transmit detection results, feeding events, and system status information among the ESP32, the inference server, and a web-based user interface developed on the Blynk platform. This configuration enables real-time monitoring, manual activation, and alert notifications. Cations.

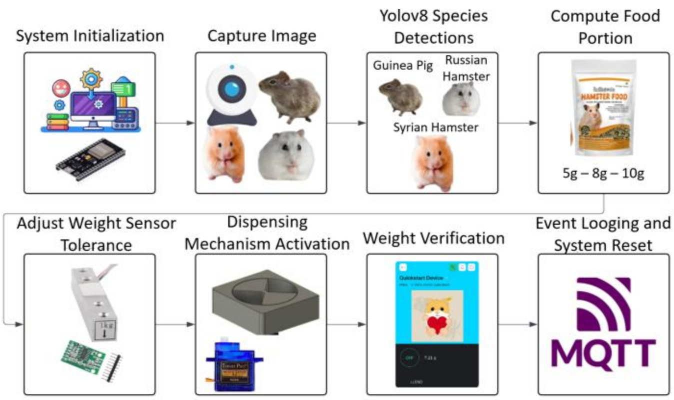


Fig. 5: Operational workflow of the Proposal system

- Image Acquisition and Data Preparation

The YOLOv8-S model, a convolutional neural network (CNN)-based architecture, was employed for real-time detection and classification of small-animal species, specifically Russian hamsters, Syrian hamsters, and guinea pigs. This architecture facilitates robust species recognition across varying lighting conditions, background complexities, and animal poses. It also provides precise bounding-box localisation necessary for reliable detection during feeding events.

A total of 405 original images were collected, with 135 images per class (guinea pig, Russian hamster, and Syrian hamster). To enhance model generalisation and mitigate overfitting, several data augmentation techniques were applied during training. These included horizontal and vertical flipping, random rotations between −15° and +15°, 90° rotations in both directions, and random zooming via cropping from 0% to 15%. Up to three augmented outputs were generated for each training image. This process resulted in a balanced dataset of 819 images (273 per class). The dataset was subsequently divided into training (70%, 573 images), validation (20%, 164 images), and testing (10%, 82 images) subsets. Table 5 provides a detailed breakdown of the dataset composition.

Fig. 6 illustrates the proposed detection pipeline, which begins with image acquisition using a fixed camera positioned above the feeding area. This is followed by data preprocessing, data augmentation, and dataset splitting prior to model training.

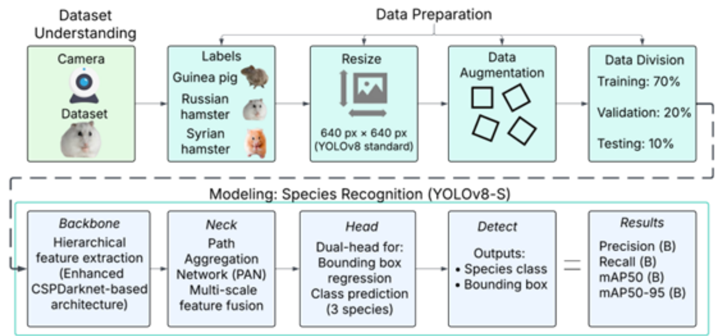


Fig. 6: Image acquisition and species-detection pipeline based on YOLOv8-S

The modelling stage utilises the YOLOv8-S architecture, which comprises a backbone for feature extraction, a multi-scale feature fusion neck, and a dual-head detection layer for bounding-box regression and species classification. The resulting outputs, namely the predicted species and its corresponding bounding box, serve as inputs for the feeding decision logic within the control system.

The dataset was constructed by collecting images of three small-animal species (Russian hamsters, Syrian hamsters, and guinea pigs) from online repositories. This approach ensured variability in illumination, background, animal posture, and camera distance, thereby supporting robust model generalisation.

To better reflect the proposed system's operating conditions, the dataset was supplemented with manually captured images from frontal and top views, consistent with the prototype's camera placement. These images were recorded using the main camera of an iPhone 15, with its technical specifications summarised in Table III.

TABLE III
IPHONE 15 MAIN CAMERA SPECIFICATIONS USED FOR DATASET ACQUISITION

Parameter	Specification
Resolution	8064 × 6048 pixels
Focal Length	26 mm
ISO	125

Data preparation commenced with the assembly of the image dataset, which was manually labelled in Roboflow into three classes: Russian hamsters, Syrian hamsters, and guinea pigs. To enhance viewpoint diversity, both frontal and top-view images were incorporated, and all samples were resized to 640 × 640 pixels to satisfy YOLOv8 input specifications. Robustness was further improved through data augmentation, including random horizontal flips, slight rotations, zoom transformations, and brightness or contrast adjustments to simulate real-world variations in lighting, orientation, and scale. The dataset was subsequently divided into training, validation, and test subsets using a 70/20/10 split to facilitate model learning, hyperparameter tuning on unseen data, and final evaluation. The summarise of the data preparation was detailed in Table IV

TABLE IV
DATASET INFORMATION.

Category	Details
Original Dataset (per class)	135 images
After First Augmentation	Cuy: 213, Ruso: 273, Sirio: 225
Balanced Dataset	819 images (273 per class)
Training (70%)	573 images
Validation (20%)	164 images
Testing (10%)	82 images
Labels	Meaning
Cuy	Guinea Pig
Hamster-Ruso	Russian Hamster
Hamster-Sirio	Syrian Hamster

- Modeling

Species recognition utilised the YOLOv8-S model, a lightweight variant of the YOLOv8 architecture that balances accuracy and computational efficiency for real-time inference in resource-constrained environments [18]. This model is therefore well-suited for embedded and IoT-based applications, including the Proposal system.

The YOLOv8 architecture comprises a backbone for hierarchical feature extraction, a Path Aggregation Network (PAN)-based neck for multi-scale feature fusion, and a dual-head output for bounding-box regression and species classification [19]. In this study, the model was trained on a balanced dataset of 819 images, with equal representation of Russian hamsters, Syrian hamsters, and guinea pigs. All images were resized to 640 × 640 pixels, and data augmentation techniques were implemented to enhance generalisation across variations in lighting, orientation, and background conditions.

- Training

The YOLOv8-S model was trained in a Google Colab environment with sufficient computational resources to ensure efficient, stable training. The primary characteristics of the training environment are presented in Table V, and the training configuration and hyperparameters are provided in Table VI.

TABLE V
TRAINING ENVIRONMENT

Environment	Component	Capacity
Google Colab Pro	RAM	12.7 GB
Google Colab Pro	GPU	NVIDEA T4(15 GB VRAM

TABLE VI
TRAINING PARAMETERS.

Parameter	Value
Model Variant	YOLOv8-S
Epochs	120
Batch Size	32
Image Size	640 pixels
Number of Classes	3
Optimizer	Automatic (Ultralytics default)
Learning Rate	Automatically adjusted

TABLE VII
TRAINING OPTIMIZATION HYPERPARAMETERS.

Hyperparameter	Value
Optimizer	AdamW
Initial Learning Rate (lr0)	0.001
Learning Rate Scheduler	Cosine decay
Early Stopping Patience	15

- Integration of Feeding Logic and Execution of Portion Control

After the YOLOv8-S model identifies the species in the frame, the predicted class is sent to the ESP32 control unit. Using a predefined nutritional table, the ESP32 determines species-specific target portions for Russian hamsters, Syrian hamsters, and guinea pigs, enabling automatic adjustment of the dispensed food quantity.

The ESP32 activates the servomotor-driven dosing wheel to dispense the calculated portion. The load cell, in conjunction with the HX711 module, verifies the dispensed mass in real time. If deviations from the target portion are detected, the system applies corrective micro-adjustments to ensure precise delivery.

Feeding events, such as detected species, dispensed portion, weight verification data, and system status, are transmitted to the mobile application using MQTT. This communication protocol supports remote monitoring, alert generation, and access to feeding history through the user interface.

D) Mobile Application and User Interaction

A mobile application was developed as part of the system methodology to facilitate monitoring, user interaction, and manual control of the proposed platform. As illustrated in Fig. 7, the application provides real-time visualisation of essential system variables, including food weight, water level, dispensing status, and alert notifications.

Communication between the application and the ESP32 microcontroller is established over Wi-Fi using the MQTT protocol, ensuring low-latency data exchange. Manual activation functions serve as a fallback mechanism, enabling direct control of the dispensing and drainage systems when necessary. The application enhances the system's autonomous operation by providing transparency, supervision, and user-level interaction, while all critical control logic is executed locally on the embedded hardware.

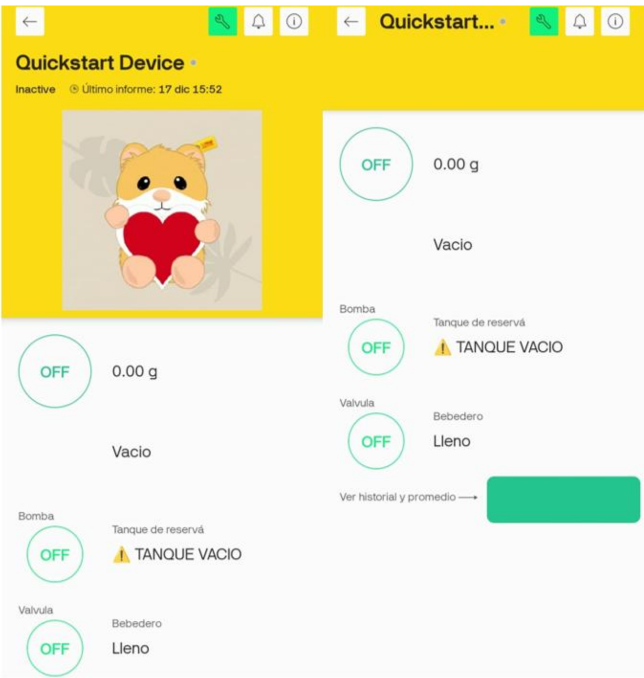


Fig. 7: Mobile application interface for real-time monitoring and manual control of the Proposal

III. TESTING AND RESULTS

To validate the operation of the Proposal system, a set of controlled tests was conducted to evaluate the performance of the mechanical, electronic, hydraulic, communication, and vision-based detection modules, both individually and in integrated operation. You can view the operation in the following video: https://youtu.be/HgKJ78Zd8ZY. The following subsections summarize the test scenarios and results.

A. Load Cell and Servomotor Verification

Sixty controlled tests were performed to verify the load cell's accuracy in detecting the weight placed on the food bowl and to evaluate the system's sensitivity under varying weight levels and detection conditions. The primary parameter assessed was weight (g), with target portions of 5 g, 8 g, and 10 g. Measurements were obtained using a 1 kg load cell and monitored via the ESP32 serial interface, while a reference digital scale provided ground-truth values. Figures 8–10 compare reference and measured weights, showing close agreement with only minor deviations due to sensor resolution and ambient noise, thereby confirming stable and reliable performance.

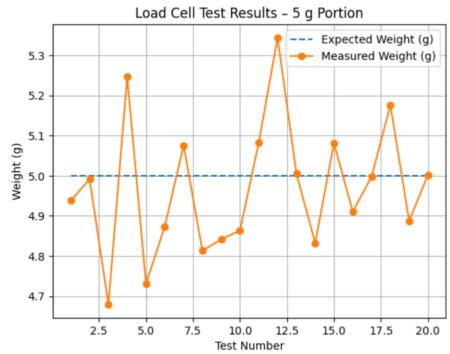


Fig. 8: Load cell test results for 5 g portion

Fig. 9: Load cell test results for 8 g portion

Fig. 10: Load cell test results for 8 g portion

B. Infrared Sensor Reading (Storage Status)

Twenty-six controlled tests were conducted to determine whether the HW-201 infrared sensor accurately detects the absence of food in the storage compartment. The primary parameter assessed was food level (empty, half-full, and full). Sensor readings were monitored via the ESP32 serial interface, and alert latency was measured using a stopwatch. As shown in Fig. 11, the sensor consistently identified empty conditions and triggered alerts within an acceptable time frame. Minor variations in response time were attributed to sensitivity to ambient lighting and differences in reflectance within the container. These results indicate that the infrared sensor provides reliable low-food detection and facilitates prompt user notification.

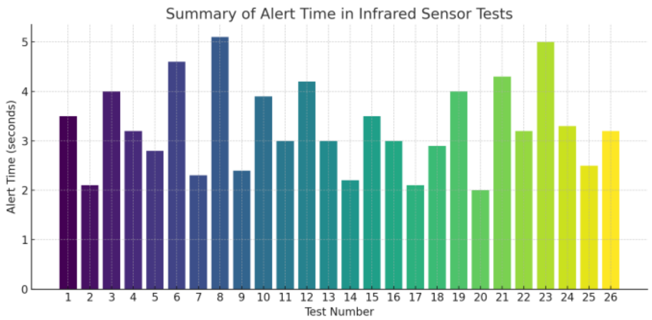


Fig. 11: Summary of Alert Time in Infrared Sensor Tests

C. Data Visualization on the Web Interface

Sixty tests were performed to verify the accurate display of sensor and system data on the Blynk platform. The parameters assessed included food weight, food storage status, detected species, recommended food portion (5 g, 8 g, or 10 g), water-level status, and the response of the manual drainage button. The experimental setup incorporated an ESP32 microcontroller connected via Wi-Fi to the Blynk mobile application, with a load cell circuit, an infrared sensor, a water sensor, and MQTT-based communication. As illustrated in Fig. 12, the interface gradually achieved comprehensive and reliable visualisation of all variables. While initial trials exhibited minor synchronisation delays, subsequent tests demonstrated stable, real-time monitoring following timing and communication optimisations.

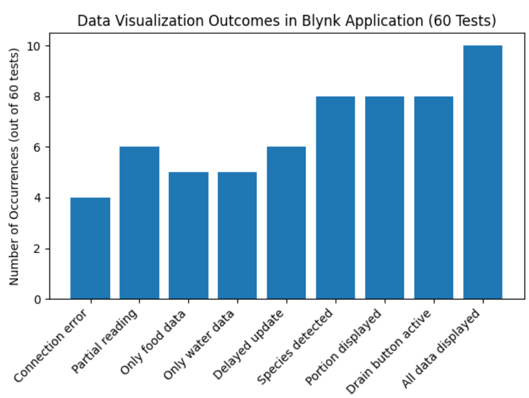


Fig. 12: Data visualization outcomes in the Blynk application during 60 tests

D. Water Pump Filling and Drainage Tests

Thirty tests were conducted to assess the performance of the water dispensing and drainage subsystem, confirming reliable filling of the drinking bottle and effective removal of excess water after use. The parameters evaluated included filling response, drainage activation, and completion, with 15 trials performed for each mode. The experimental setup consisted of submersible pumps, a 0.6 L reservoir, an 80 mL drinking bottle, a water-level sensor, and an ESP32 controller. Response times were measured using a stopwatch. The filling pump consistently delivered water within the expected timeframe, and the drainage pump effectively removed excess water, preventing overflow. Minor leakage and timing variations were observed, primarily attributed to bottle design and pump dynamics. Fig. 13 summarises these results.

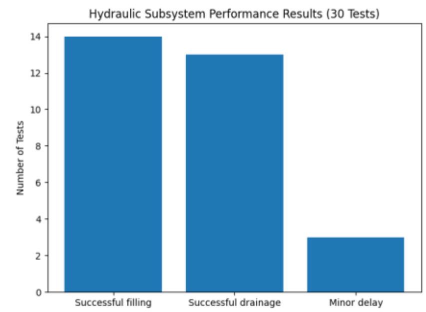


Fig. 13: Hydraulic subsystem performance results during filling and drainage tests (30 tests)

E. MQTT Communication Performance

A total of 40 tests were conducted to evaluate the reliability and responsiveness of the MQTT-based communication architecture connecting the ESP32 microcontroller, the inference server, and the mobile application. The evaluation focused on message transmission success, data consistency, and end-to-end communication latency. The test setup included an ESP32 device, an MQTT broker, an inference server, the Blynk application, and the ESP32 serial monitor for logging and verification. As shown in Fig. 14, initial trials revealed minor synchronisation delays and occasional temporary disconnections, primarily due to network instability. After iterative adjustments to timing and communication parameters, the system demonstrated stable and consistent message exchange in most runs, establishing MQTT as a reliable backbone for real-time coordination in the Proposal system.

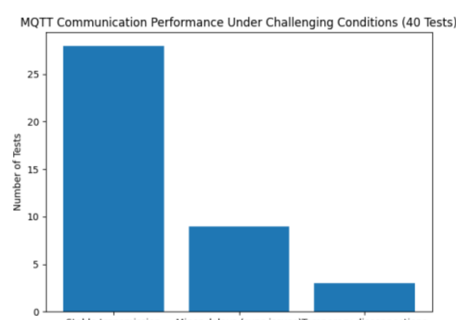


Fig. 14: MQTT communication performance under challenging conditions (40 tests)

F. YOLOv8 Species Detection Accuracy

The YOLOv8-S–based model was evaluated for real-time species-recognition performance and its potential to trigger automated feeding decisions. The assessment considered accuracy, precision, recall, and mean Average Precision (mAP), utilising a balanced dataset of 819 images (273 per class) and an independent test set of 82 images representing Russian hamster, Syrian hamster, and guinea pig. Fig. 15 demonstrates that the training and validation curves exhibit stable convergence without significant overfitting, and that precision, recall, and mAP remain consistently high throughout training. The model attained an overall test accuracy of 99.667%. In addition, Fig. 16 presents the confusion matrix, which confirms a strong concentration of predictions along the diagonal, indicating correct classification across the three species. Misclassifications were infrequent and primarily occurred under conditions of partial occlusion or atypical animal poses, supporting the model's readiness for robust deployment.

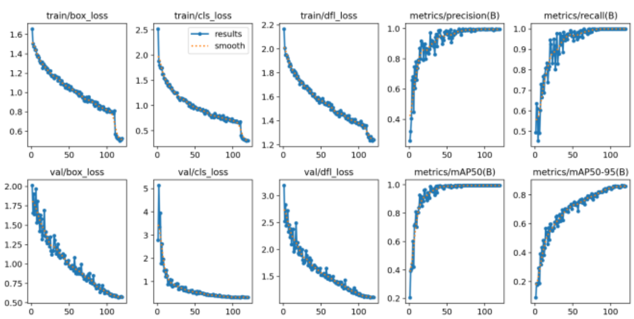


Fig. 15: Training and validation performance curves of the YOLOv8-S model.

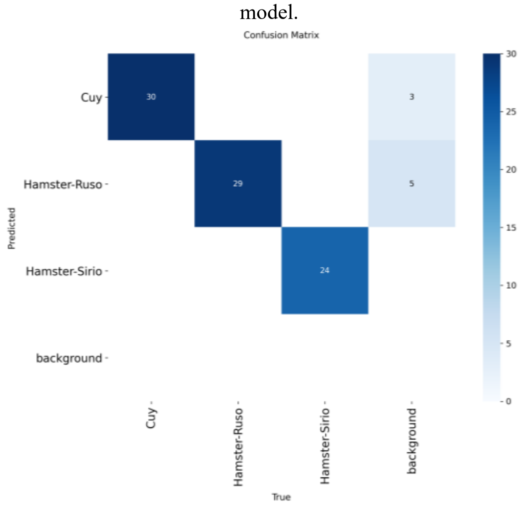


Fig. 16: Confusion Matrix of the YOLOv8-S Species Classification Model

G. Integrated Detection-to-Dispensing Test

Twenty-four tests were performed to evaluate the integration between the species-detection module and the food-dispensing mechanism by determining whether the appropriate portion was delivered for each identified animal. The experimental setup comprised a camera, an ESP32 microcontroller, an SG90 servomotor, a load cell for weight verification, and MQTT-based communication between the inference component and the control unit. This comprehensive evaluation encompassed real-time detection, dispensing actuation, and post-dispense weight verification. As illustrated in Fig. 17, correct portions were dispensed in 18 of 24 trials. The remaining errors were primarily attributed to initial calibration offsets and communication delays. Subsequent calibration and synchronisation adjustments enhanced performance, thereby confirming reliable integrated operation.

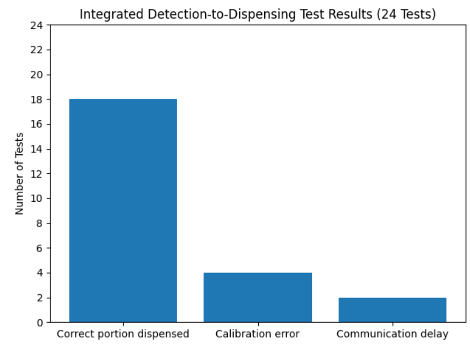


Fig. 17: Integrated Detection-to-Dispensing Test Results

H. General System Functionality

A total of 40 integrated tests were conducted to verify the correct operation of all modules within the Proposal system under real operating conditions. The evaluation assessed the end-to-end integration of the sensing layer (load cell and infrared sensor), the control logic on the ESP32, the actuation components (servomotor and water pumps), the vision-based recognition trigger, the dispensing and hydraulic subsystems, and the communication and user interface stack (MQTT and the Blynk mobile application). As shown in Fig. 18, correct operation occurred in 90% of trials. The remaining cases involved partial or delayed responses during stress scenarios or transient communication latency. No critical failures were observed, which confirms reliable system performance following calibration and optimisation.

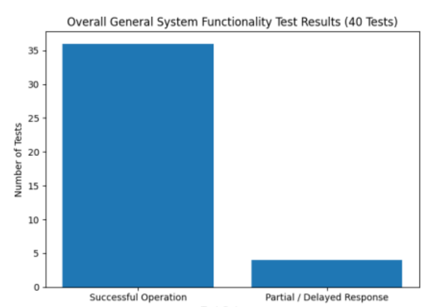


Fig. 18: Overall general system functionality test results (40 tests).

III. CONCLUSIONS

This proposal demonstrates that integrating vision-based species recognition with closed-loop dispensing and hydration control enhances routine consistency for small rodents while maintaining affordability and modularity. Component-level testing verified accurate portion delivery (60 trials), reliable low-food detection (26 trials), and consistent water filling and drainage (30 trials), with minor timing variations attributed to commercial bottle characteristics. The Blynk interface and MQTT layer provided stable, real-time monitoring after timing optimisation, enabling coordinated operation among the ESP32, the inference server, and the mobile application. The YOLOv8-S model achieved a test accuracy of 99.667% across three species, indicating strong suitability for event-driven feeding decisions. End-to-end trials resulted in correct portion delivery in 18 out of 24 runs, and overall integrated functionality reached 90% across 40 trials, with remaining issues primarily due to calibration offsets or transient network latency. Future work will focus on expanding the dataset, reinforcing the enclosure, reducing latency, migrating inference to the device to increase offline autonomy, and validating performance in household environments. The ESP32 serves as an acquisition node that captures and transmits images via the HTTP protocol to an external server (Google Colab with a T4 GPU runtime). There, the YOLOv8-S model performs real-time species identification. The detection label is returned to the ESP32 via MQTT, triggering the dispensing logic for species-specific portions (5 g, 8 g, or 10 g) with a measured end-to-end latency of approximately 1.2 seconds from capture to response. This integration provides a robust, low-cost solution that ensures animal welfare through personalized autonomous care. In conclusion, this project integrates computer vision and IoT to address small pet care through low-cost automation. With 90% operational success, the system supports animal welfare by providing consistent, species-specific nutrition. Remote monitoring via Blynk improves user flexibility and supervision, while the prototype establishes a solid framework for future edge-computing applications in automated rodent husbandry.

References

[1] A. D. Wood and T. J. Bartness, "Food deprivation-induced increases in hoarding by Siberian hamsters are not photoperiod-dependent," Physiology & Behavior, vol. 60, no. 4, pp. 1137–1145, 1996, doi: 10.1016/0031-9384(96)00173-4.

[2] K. L. Scherbarth and J. R. Parfitt, "Effects of restricted feeding on circadian rhythms in Syrian hamsters," Behavioural Brain Research, vol. 310, pp. 229–236, 2016, doi: 10.1016/j.bbr.2016.05.041.

[3] K. M. Moran and Y. Delville, "A hamster model for stress-induced weight gain," Hormones and Behavior, vol. 154, art. no. 105488, 2024, doi: 10.1016/j.yhbeh.2024.105488.

[4] C. Camuñas-Martínez et al., "ESP32-based smart pet feeder with web interface and camera support," Sensors, vol. 23, no. 9, art. no. 4151, 2023, doi: 10.3390/s23094151.

[5] E. E. Bray et al., "Once-daily feeding is associated with better health in companion dogs: Results from the Dog Aging Project," Journal of Gerontology: Biological Sciences, vol. 77, no. 5, pp. 698–705, 2022, doi: 10.1093/gerona/glac191.

[6] Y. Sánchez-Andrade et al., "Smart feeder systems reduce stress and improve health in laboratory rodents," Laboratory Animals, vol. 55, no. 4, pp. 299–308, 2021, doi: 10.1177/0023677220980472.

[7] J. Morales and D. Herrera, "Smart feeding systems for small pets: A review of automated nutritional delivery," Sensors, vol. 21, no. 14, art. no. 4891, 2021.

[8] H. Liu and S. Wang, "Design and development of an automatic feeding robot for small animals," IEEE Access, vol. 8, pp. 115230–115240, 2020, doi: 10.1109/ACCESS.2020.3004375.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, real-time object detection," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788.

[10] L. Zhang and W. Chen, "Real-time animal detection in complex environments using YOLO-based models," Ecological Informatics, vol. 63, art. no. 101299, 2021.

[11] I. Bozkurt and M. Yurtoglu, "Deep learning-based automatic animal classification systems: A review," Artificial Intelligence Review, vol. 54, pp. 927–963, 2021.

[12] G. Jocher, "YOLOv8: Open-source deep learning models for object detection and segmentation," arXiv:2301.10000, 2023.

[13] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, vol. 147, pp. 70–90, 2018.

[14] J. Lee and S. Han, "Automation technologies for pet care: A review and future directions," IEEE Consumer Electronics Magazine, vol. 9, no. 5, pp. 35–44, 2020.

[15] A. D. Wood and T. J. Bartness, "Food deprivation-induced increases in hoarding by Siberian hamsters," Physiology & Behavior, 1996.

[16] K. L. Scherbarth and J. R. Parfitt, "Effects of restricted feeding on circadian rhythms in Syrian hamsters," Behavioural Brain Research, 2016.

[17] K. M. Moran and Y. Delville, "A hamster model for stress-induced weight gain," Hormones and Behavior, 2024.

[18] C. Camuñas-Martínez et al., "ESP32-based smart pet feeder with web interface," Sensors, 2023.

[19] E. Jain et al., "Design of smart pet food dispenser using embedded systems," IEEE ESCI, 2023.

[20] Y. Wong et al., "Design of solar-powered automatic pet feeder based on Arduino and RTC," E3S Web of Conferences, 2024.

[21] O. E. Castillo-Arceo et al., "Design and development of a smart pet feeder with IoT and deep learning," Engineering Proceedings, 2024.

[22] B. Vijayalakshmi et al., "AI- and IoT-enhanced automatic animal food dispenser," IEEE ICAIT, 2024.

[23] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," Computers and Electronics in Agriculture, 2018.

[24] J. Redmon et al., "You Only Look Once: Unified, real-time object detection," CVPR, 2016.

[25] G. Jocher, "YOLOv8: Open-source deep learning models," arXiv, 2023.

[26] L. Zhang and W. Chen, "Real-time animal detection using YOLO-based models," Ecological Informatics, 2021.

[27] Espressif Systems, ESP32 Series Datasheet, Espressif Systems Inc., 2023.

[28] Avia Semiconductor, HX711 24-Bit Analog-to-Digital Converter for Weigh Scales Datasheet, 2022.

[29] Tower Pro, SG90 Micro Servo Motor Technical Specifications, 2021.

[30] Vishay Semiconductors, Infrared Proximity and Obstacle Detection Sensors, Application Note, 2020.

[31] Grundfos, Pump Handbook, McGraw-Hill, 2018.

[32] Texas Instruments, Liquid Level Sensors: Principles and Applications, Application Report, 2019.

[33] Redmon, J. et al., "You Only Look Once: Unified, Real-Time Object Detection," Proc. IEEE CVPR, 2016.

[34] Szeliski, R., Computer Vision: Algorithms and Applications, Springer, 2022.