Smart Surveillance for Swiftlet Farming: IoT-Driven Real-Time Pest Detection with YOLOv10

Depi Ginting
Department of Electrical and Computer Engineering
Universitas Syiah Kuala
Banda Aceh, Indonesia
depi23@mhs.usk.ac.id

Khairun Saddami*
Department of Electrical and Computer Engineering
Universitas Syiah Kuala
Banda Aceh, Indonesia
khairun.saddami@usk.ac.id

Ramzi Adriman
Department of Electrical and Computer Engineering
Universitas Syiah Kuala
Banda Aceh, Indonesia
ramzi.adriman@usk.ac.id

Kurnianingsih
Department of Electrical and Computer Engineering
Politeknik Negeri Semarang
Semarang, Indonesia
kurnianingsih@polines.ac.id

Sunu Wibirama
Department of Electrical and Information Engineering,
Faculty of Engineering, Universitas Gadjah Mada, Indonesia
sunu@ugm.ac.id

Abstract—Pest disturbances in swiftlet houses reduce edible bird's nest (EBN) production, a valuable commodity in Southeast Asia. Manual pest monitoring is often inefficient and disruptive to the sensitive environments of birds. However, to the best of our knowledge, no study has incorporated computer vision technology and the IoT for smart surveillance in swiftlet farming. To address this gap, we propose a YOLOv10-based detection system to identify small pests in real time under low-light conditions. The proposed system features a Python-based UI, a Region of Interest (ROI) function to improve detection focus, and IoT integration via a Telegram Bot that sends image-based alerts upon detection. The infrared CCTV cameras captured 2,011 images, which were augmented through rotation, resulting in 3,992 images. Six YOLOv10 model variants (n, s, m, b, l, and x) were evaluated. Based on our experimental results, the 'b' variant exhibited the best performance, with an mAP50 of 0.9936 and the lowest latency. Evaluation using a 50-minute video demonstrated accurate and rapid pest identification with cockroaches as the main pests. The experimental study showed the effectiveness of the system in monitoring swiftlet farming while reducing environmental disturbance to birds.

Keywords: EBN, Pest detection, Swiftlet House, YOLOv10, TelegramBot

I. INTRODUCTION

Edible Bird's Nest (EBN), also known as swiftlet nest, is a high-value food product in Southeast Asia [1], [2]. EBN is associated with various health benefits, including anti-aging, bone-strengthening, and neuroprotective properties [3]. Swiftlets make their nests using saliva [4]. They breed in buildings called swiftlet houses, which are made to resemble their natural homes, such as caves [5]. Optimal conditions within these swiftlet houses include a temperature range of 26–35°C, humidity levels of 80–90%, low air velocity, and lighting below 5 lux [6]. Initially, swiftlet houses were constructed primarily in coastal areas. However, they have been increasingly established in residential zones because of the growing swiftlet population [7].

One of the primary factors contributing to decreased swiftlet nest production is pest disturbance, which include rats, insects, reptiles, and owls [8], [9], [10]. Automatic doors have been successfully implemented to prevent owls from entering, thereby eliminating the need for human intervention [11]. Nevertheless, pest monitoring and nest damage inspection are still conducted manually by swiftlet house managers [10]. This presents a challenge because swiftlets are particularly sensitive to human presence and environmental disturbances [12]. Consequently, the frequency of human entry into swiftlet houses must be minimized to avoid disrupting the nesting process [7]. Consequently, pests frequently go unnoticed between inspection periods, resulting in decreased nest productivity [10].

Implementing infrared CCTV cameras has proven to be an effective approach to this problem because they are capable of generating grayscale images even in low-light environments within swiftlet houses [6], [13]. Furthermore, Internet of Things (IoT)-based systems are increasingly being implemented to monitor environmental conditions, such as temperature, humidity, light, and sound. These approaches eliminate the need for direct human presence within a swiftlet house [14], [15]. In conjunction with these advancements, developments in computer vision technology have facilitated real-time object detection using algorithms such as YOLO [16].

*Corresponding Author: khairun.saddami@usk.ac.id

Previous studies have demonstrated that YOLOv3 can detect soybean pests with an accuracy of up to 72.92% [16]. YOLOv7 can be integrated with Telegram-based applications for notifications and has shown effectiveness in detecting ships with an accuracy of 94% [17]. YOLOv7 has also exhibited high reliability in detecting objects in infrared (grayscale) images, achieving a mean average precision (mAP) of 95% while supporting real-time performance [18]. YOLOv10, a more recent iteration of the YOLO algorithm, introduces enhancements in efficiency and accuracy without requiring the Non-Maximum Suppression (NMS) process, utilizing a dual-label assignment strategy that reduces computational redundancy while maintaining high accuracy [19]. This model is lightweight and faster than its predecessors, YOLOv9 and RT-DETR, and excels in detecting small and complex objects [20], [21], [22].

YOLOv10's capability to detect small objects, coupled with its high computational efficiency and adaptability to low-light conditions, makes it an ideal solution for pest detection in swiftlet houses. This system can be integrated with Internet of Things (IoT) technology and a Telegram bot, enabling real-time notifications to swiftlet house owners and facilitating prompt decision-making. However, no study has incorporated computer vision technology and the IoT for smart surveillance in swiflet farming.

To address this research gap, we propose a smart pest detection system utilizing YOLOv10, the IoT, and a Telegram bot. The proposed system aims to deliver early warnings and support effective pest management without manual intervention. The system was tested using primary data obtained from the swift houses. The following sections detail the methodology, experimental setup, and evaluation of the proposed system.

II. Materials and Method

This study was conducted to develop an automated pest detection system by integrating the YOLOv10 with Internet of Things (IoT) technology through a Telegram-based notification bot. The research took place at a swiftlet house, with a specific focus on detecting cockroaches as the primary pest. The development process involved a systematic workflow comprising hardware and software selection, image data acquisition and preprocessing, model training and optimization, system integration, and performance evaluation under real-world conditions.

A. Tools

The hardware used in this study comprised an ROG Zephyrus G15 laptop, which was equipped with an AMD Ryzen 9 5900HS processor, 16 GB of RAM, and an NVIDIA RTX 3050 4 GB graphics card. Additionally, a Hikvision 2CE16KOT IR CCTV camera with a 5 MP resolution, a 4-channel Turbo HD DVR, an EasyCap USB 2.0 device for video signal conversion, and a BNC-to-RCA converter were employed. The primary software components included Python programming language, Roboflow for dataset management and annotation, and Telegram Bot for automated notifications. Model training was conducted using Google Colab Pro, employing a g2-standard-16 virtual machine configuration that featured 16 vCPUs, 64 GB of RAM, and an NVIDIA L4 GPU with 24 GB of VRAM. The trained model was automatically saved to Google Drive.

B. Dataset

The dataset processing (Fig. 1) began with the identification of cockroaches as the primary target pest using infrared (IR) CCTV footage installed within the swiftlet houses. The recorded video was subsequently extracted into image frames using video processing software. These extracted images underwent preprocessing and augmentation stages to enhance data variability. Each original image underwent rotation augmentation at angles of +15°, -15°, +90°, and -90°, resulting in four additional images per original image. This process expanded our dataset from 2,011 original images to a total of 3,992 images. Some images were not augmented due to quality considerations.

TABLE I. Hyperparameters configuration for Yolov10

Hyper-parameter	Value
Epochs	200
Optimizer	SGD
Batch size	16
Image Size	640x640
Momentum	0.937
Learning rate	10^{-2}

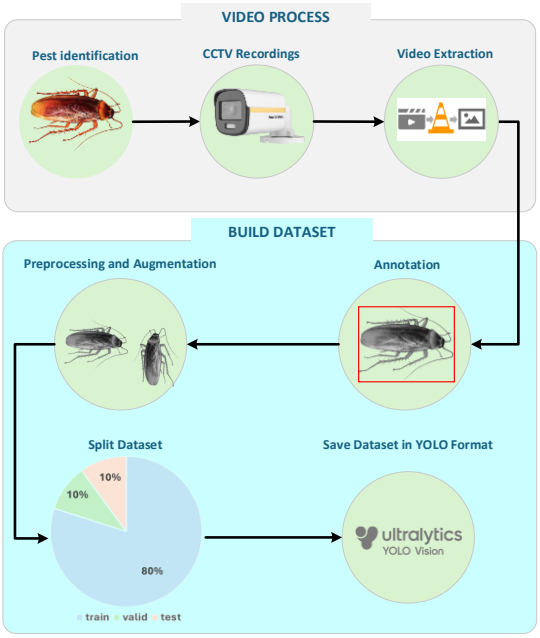

Fig. 1. Dataset processing workflow.

The annotation process involved marking the presence of pest objects with bounding boxes on the Roboflow platform. All images were then resized to a standard resolution of 640×640 pixels and saved in a format compatible with the YOLO architecture [23]. To ensure a comprehensive and unbiased evaluation of the model's performance, the dataset was divided into three segments: 80% for training, 10% for validation, and 10% for testing.

C. Model Training and Validation

The YOLOv10 model was trained using pretrained weights from six model variants: Nano (n), Small (s), Medium (m), Baseline (b), Large (l), and Extra Large (x). Each variant was specifically designed to provide an optimal balance between inference speed and detection accuracy, with the

complexity of the model architecture progressively increasing from variant n to x. The training process used images with a resolution of 640×640 pixels to align with the input architecture of YOLOv10. The training was conducted over 200 epochs with a batch size of 16. The optimization algorithm employed was Stochastic Gradient Descent (SGD), characterized by a momentum value of 0.937 and an initial learning rate of 0.01, as detailed in Table 1. This training was performed using Google Colab Pro to ensure the availability of adequate computational resources. Model validation occurred concurrently during the training process to facilitate real-time performance monitoring.

Model performance was evaluated using precision, recall, F1-score and mean Average Precision (mAP) as shown in Eq. (1)-(5). In object detection, precision and recall rely on the concepts of true positives (TP), false positives (FP), and false negatives (FN). A detection is considered a true positive when the model correctly identifies an object. The predicted bounding box sufficiently overlaps the ground truth (typically measured using an Intersection over Union (IoU) threshold of 0.5 or higher) and the predicted class label matches the actual class. A false positive occurs when the model predicts an object that either does not exist, has an incorrect class label, or has insufficient overlap with any ground truth object (IoU below the threshold). Conversely, a false negative refers to a case where an object present in the image is not detected by the model at all. Distribution focal loss (dfl) is a specialized loss function that helps the model better handle the distribution of bounding box coordinates by treating them as classification rather than regression problems.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$AP = \sum_{n} (R_n - R_{n-1}) \cdot P_n \tag{3}$$

where R_n represents the recall at the nth threshold, R_{n-1} is the recall at the previous threshold, and P_n is the precision at the n[th] threshold.

$$\text{F1 } Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4}$$

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i \tag{5}$$

$$Latency = Timestamp\ End - Timestamp\ Begin \tag{6}$$

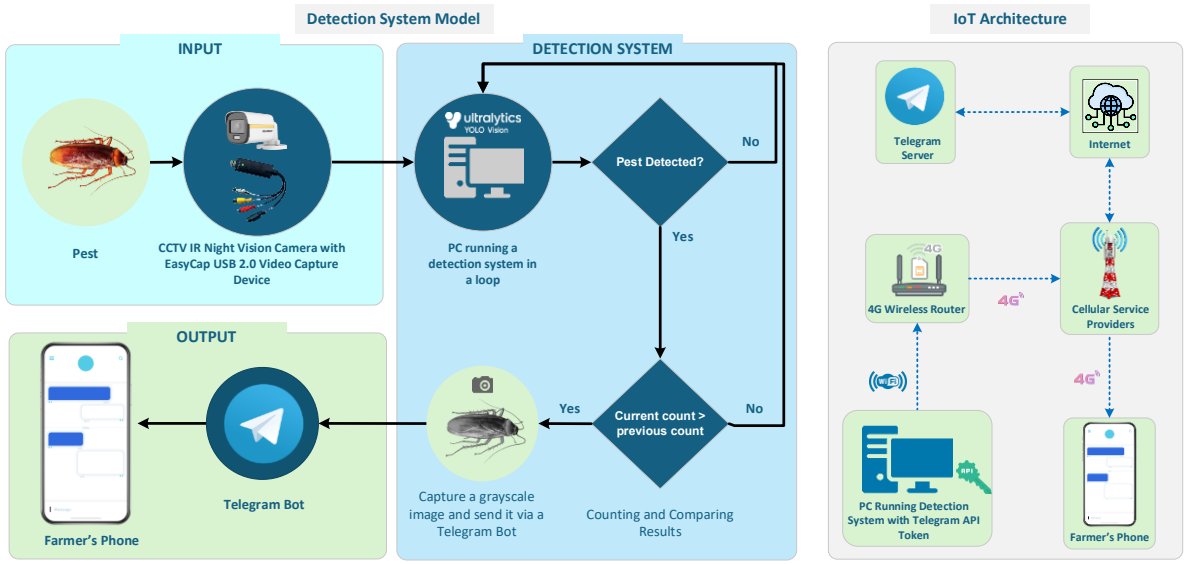

Fig. 2. The proposed system in this study.

D. *Development of the proposed system*

The object detection system was developed using a Python-based desktop user interface (UI) utilizing the Tkinter library. We developed an intuitive graphical interface that displayed results of real-time detection as shown in Fig. 2. An IR CCTV camera transmitted video signals to the PC via an EasyCap device. Upon detecting an object, the system captured an image in grayscale format and sent a notification to a Telegram bot linked to the smartphone of the swiftlet farmer.

The system included a "Run Startup" button, options for selecting different YOLO model variants, and a Region of Interest (ROI) feature to limit the detection area. It was designed to prevent redundant notifications by tracking the number of detected objects, sending alerts only when the count increases. The object count was reset every 24 hours or upon system restart.

E. *Evaluation of the the proposed system*

The evaluation examined the detection performance of each YOLOv10 variant by using an unfamiliar test dataset. The metrics assessed included precision, recall, F1-score, and mean average precision (mAP), all of which were derived from the confusion matrix. A comprehensive system evaluation measured detection latency (YOLO latency), notification latency (Telegram latency), and the total latency from detection to user notification. Testing was conducted using a video exceeding 50 minutes in length, simulating real-world conditions in the swiftlet house. The captured video from CCTV footage was not part of the dataset and was exclusively used for the real-time performance evaluation of the system. The 50-minute test video was recorded under actual swiftlet house conditions with varying lighting levels

(0.5-5 lux) and contained 157 instances of cockroach appearances at different locations. The detection results of the system were manually verified against a frame-by-frame analysis by human experts to confirm the accuracy. Having established the experimental framework, we analyzed the results of our YOLOv10-based pest detection system.

III. Results and Discussion

A. Dataset

Infrared CCTV recordings from within the swiftlet house confirmed that cockroaches image acquisition was effective even under low-light conditions. The camera, placed in key areas such as wooden surfaces, walls, and nesting boards, successfully captured clear images of cockroaches activities using its 24-hour infrared feature. The augmentation process doubled the dataset from 2,011 to 3,992 images, with varied angles improving the model's detection capability. Annotation results are shown in Fig. 3. The dataset was split into 80% training or around of 3,194 images, 10% or 400 images as validation, and 10% or 400 images as testing, and published via Roboflow [24].

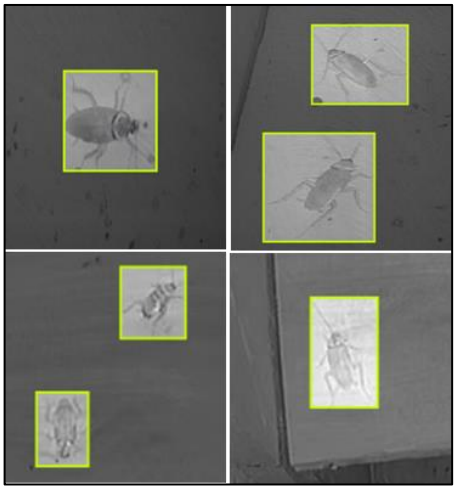

Fig. 3. Cockroach dataset used in this study.

B. Training and validation results

The YOLOv10 training for cockroach detection within the swiftlet house was successfully executed across six model variants: n, s, m, b, l, and x. Each variant demonstrated varying performance based on the hyperparameter configurations, as detailed in Table 1.

TABLE II. Training results of six Yolov10 variants

Model Variants	Training Time	Best Epoch (mAP50)	Model Size (MB)
YOLOv10n	2h 41m 17s	199	5.64
YOLOv10s	5h 40m 50s	184	16.16
YOLOv10m	4h 21m 18s	167	32.71
YOLOv10b	5h 24m 37s	145	40.49
YOLOv10l	6h 57m 14s	172	50.98
YOLOv10x	8h 5m 49s	158	62.60

Table 2 provides a summary of the training duration, optimal epoch based on the mAP50 value, and model size for the six YOLOv10 variants. The YOLOv10-n variant achieved the shortest training time of 2 hours and 41 minutes, while the x variant exhibited the longest training duration, exceeding 8 hours. As the complexity of the models increased from variant n to x, the file size also significantly escalated from 5.64 MB to 62.60 MB. We discovered that the s variant took longer to train compared to the m and b variants because the training was conducted on a free cloud service platform. The computation resources are reduce to lower resource.

The validation results of the six variants are presented in Table 3. All models demonstrated excellent detection performance, with mAP50 values exceeding 0.9900. The s variant provided the most balanced results, achieving a precision of 0.989, a recall of 0.9811, and the highest F1 score of 0.9851. Meanwhile, the x variant recorded the highest precision at 0.9900, but its recall was relatively lower at 0.9673, resulting in an F1 score of 0.9785, comparable to that of the n variant (0.9784).

TABLE III. Validation results of six Yolov10 variants

Model Variants	Precision	Recall	F1 Score	mAP50
YOLOv10n	0.9824	0.9743	0.9784	0.9909
YOLOv10s	0.989	0.9811	0.9851	0.9932
YOLOv10m	0.9782	0.9794	0.9788	0.9929
YOLOv10b	0.9778	0.9869	0.9823	0.9936
YOLOv10l	0.9863	0.976	0.9811	0.9939
YOLOv10x	0.99	0.9673	0.9785	0.9937

The b and l variants also demonstrated competitive performance, with F1 scores of 0.9823 and 0.9811, respectively, and exceptionally high mAP50 values of 0.9936 and 0.9939. These results indicate that models with medium to high complexity are capable of delivering excellent detection accuracy. Based on the results, all YOLOv10 variants could be effectively utilized for cockroaches detection in swiftlet houses. When choosing the most appropriate variant, it is important to take into account the system's specific needs, especially in terms of the computational resources available and the requirement for fast inference.

TABLE IV. Performance comparison between several models

Model Variants	Precision	Recall	F1 Score	mAP50
YOLOv10-s	0.9890	0.9811	**0.9851**	0.9932
YOLOv10-b	0.9778	**0.9869**	0.9823	0.9936
YOLOv10-l	0.9863	0.976	0.9811	**0.9939**
YOLOv10-x	**0.9900**	0.9673	0.9785	0.9937
YOLOv8-m	0.9779	0.9648	0.9713	0.9875
RT-DETR-l	0.9869	0.9824	0.9846	0.9906

Table 4 shows the comparison of the best evaluation measures of YOLOv10 variants outperform YOLOv8-m and RT-DETR-l in mAP50, with YOLOv10-l achieving the highest score (0.9939). YOLOv10-x offers the best precision (0.9900), while YOLOv10-b provides the most balanced performance, combining high recall (0.9869), precision (0.9778), and low latency. Although YOLOv10-l and -x are slightly more accurate, their computational demands may limit real-time deployment. In contrast, YOLOv10-b offers an optimal trade-off between accuracy and efficiency, making it well-suited for practical, low-latency applications such as real-time pest detection in swiftlet houses.

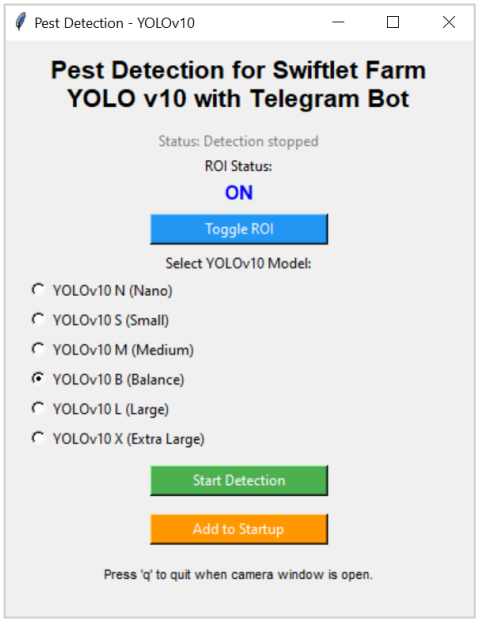

Fig. 4. User interface of the proposed system.

C. *Real-time detection using the proposed system*

Figure 4 illustrates the user interface (UI) of the cockroaches detection system based on YOLOv10. Users are allowed to select from six model variants, ranging from Nano (n) to Extra Large (x), using radio button options. The "Start Detection" button initiates the detection process, while the "Add to Startup" button configures the application to launch automatically upon powering on the computer. For user convenience, a note indicates that pressing the 'q' key will exit detection mode.

The ROI feature offers two display modes: ROI OFF and ROI ON. In ROI OFF mode, the entire camera view is utilized for detection as shown in Fig 5. Furthermore, as shown in Fig. 6, only a specific highlighted area (displayed in translucent purple) is targeted for detection in ROI ON. This mode enhances system efficiency by minimizing false detections outside critical areas and reducing computational load, as only a portion of each frame is processed. This feature is particularly advantageous in swiftlet houses, where cockroaches activities are concentrated in defined areas. By focusing the detection area, the system increases speed while mitigating distractions from irrelevant movements. Overall, the ROI mode significantly enhances system efficiency and supports a more focused and reliable detection process.

Additionally, the system is integrated with a Telegram bot that delivers real-time notifications containing detection images, complete with bounding boxes and timestamps (see Fig 7). Notifications are triggered only when the number of detected cockroaches increases. Furthermore, the system automatically resets the detection count every 24 hours or following a power outage, ensuring that notifications remain pertinent for the swiftlet house owner.

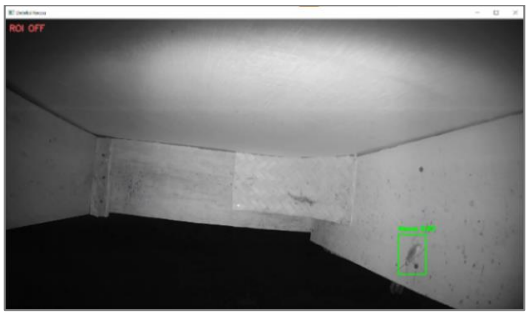

Fig. 5. Region of interest (ROI) mode: OFF.

Fig. 6. Region of interest (ROI) mode: ON.

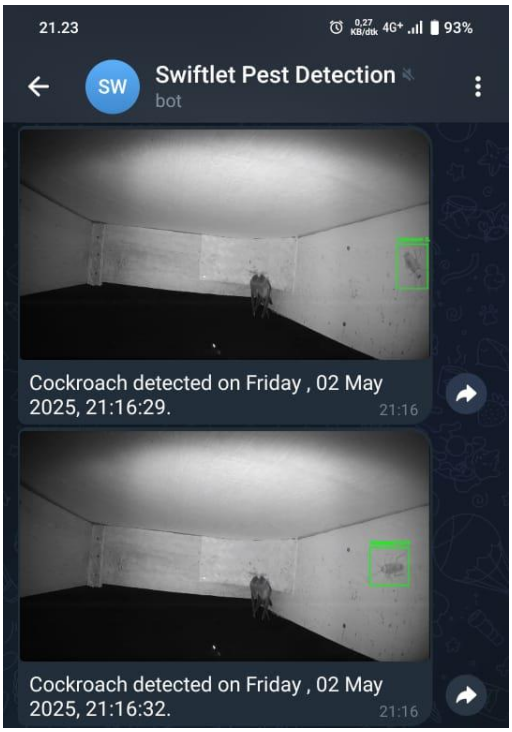

Fig. 7. Telegram notification based on results of detection.

TABLE V. PERFORMANCE EVALUATION OF SYSTEMS' LATENCY

Model variant	Total Detection messages	Average Detection Latency (ms)	Average Telegram Latency (ms)	Average Total Latency (ms)	Average Google Ping (ms)
YOLOv10-n	157	29	2241	2283	38
YOLOv10-s	153	27	2894	2934	84
YOLOv10-m	151	26	2004	2043	46
YOLOv10-b	153	26	1799	1838	38
YOLOv10-l	153	26	1834	1874	39
YOLOv10-x	150	25	1996	2034	39
YOLOv8-m	246	58	1741	1815	41
RT-DETR-l	230	59	1892	1966	40

The latency results in Table 5 represent the average detection and notification times for each model variant. YOLO-based detection was consistently fast, with processing times between 25–29 ms per frame. All variants accurately detected cockroaches, with total counts ranging from 150 to 157. However, Telegram notification latency varied due to network conditions, as indicated by ping times to Google. YOLOv10-s had the highest notification delay at 2,894 ms, while YOLOv10-b performed best with the shortest latency of 1,838 ms, making it the most suitable for real-time deployment. The system uses a state-based mechanism that triggers alerts only when cockroach counts increase, avoiding redundant messages. A 20-second reset mechanism was used during testing, while in real use, the counter resets every 24 hours or after a power outage. Limitations include the system's focus on a single pest class and sensitivity to network conditions. Future work includes multi-pest detection, improved robustness, and edge computing integration.

IV. CONCLUSIONS

This study proposed a novel pest detection system for swiftlet houses based on the YOLOv10 models. Evaluation results demonstrated exceptional detection performance across all models, with mAP50 values exceeding 0.99 and F1 scores surpassing 0.97. Although more complex models required longer training times and had larger file sizes, their validation performance remained relatively comparable to that of lighter models. The system was tested on a real-time video and efficiently delivered real-time notifications via Telegram, with the YOLOv10-b variant achieving the lowest total latency. Furthermore, the system was equipped with a Region of Interest (ROI) feature to enhance detection efficiency and an anti-spam mechanism that sends notifications only when the number of detected cockroaches increases. Based on our experimental results, the proposed system is a reliable and effective for field implementation.

REFERENCES

[1] X. Wang, D. Hu, F. Liao, S. Chen, Y. Meng, J. Dai, X. Zhang, and C. Liu, "Comparative proteomic analysis of edible bird's nest from different origins," Scientific Reports, vol. 13, no. 1, p. 15859, Sep. 2023.

[2] A. Sungsiri, S. Nonsiri, and A. Monsakul, "The classification of edible-nest swiftlets using deep learning," in Proceedings of the 2022 6th International Conference on Information Technology (InCIT), Bangkok, Thailand, Nov. 2022, pp. 404–409.

[3] Y. Ito, K. I. Matsumoto, A. Usup, and Y. Yamamoto, "A sustainable way of agricultural livelihood: edible bird's nests in Indonesia," Ecosystem Health and Sustainability, vol. 7, no. 1, p. 1960200, Jan. 2021.

[4] Z. Hou, P. He, M. U. Imam, J. Qi, S. Tang, C. Song, Q. Zhang, and W. Zhang, "Edible bird's nest prevents menopause‐related memory and cognitive decline in rats via increased hippocampal Sirtuin‐1 expression," Oxidative Medicine and Cellular Longevity, vol. 2017, p. 7205082, Oct. 2017.

[5] S. P. Loh, S. H. Cheng, and W. Mohamed, "Edible bird's nest as a potential cognitive enhancer," Frontiers in Neurology, vol. 13, p. 865671, Apr. 2022.

[6] A. N. Ramli, S. Z. Badrulzaman, R. V. Patil, N. I. Azelee, N. H. Manas, and A. W. Aminan, "Beyond tradition: A novel approach for edible bird nest cleaning and its processing," Veterinary Research Communications, vol. 48, no. 1, pp. 29–37, Jan. 2024.

[7] W. W. Lee and W. K. Lai, "A novel flower pollination algorithm for auto-grading of edible bird's nest," in Proceedings of the 2021 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), Shah Alam, Malaysia, Jun. 2021, pp. 140–145.

[8] E. S. H. Quah and J. L. Chong, "Reptile predators of swiftlets (genus Aerodramus) with a focus on their impact on the swiftlet farming industry," Herpetology Notes, vol. 16, pp. 457–462, Jul. 2023.

[9] B. Usmanto and N. A. K. Dewi, "Monitoring and automation system of swiftlet house using internet of things (IoT) based," Jurnal Teknologi Komputer dan Sistem Informasi, vol. 6, no. 1, pp. 72–79, 2023,

[10] M. R. Yaacob, W. I. W. Khairy, A. R. Munirah, A. J. N. F. Nabilah, and H. Zulhazman, "Pest disturbance in edible bird nest swiftlet house," AIP Conference Proceedings, vol. 2347, no. 1, p. 030081, Apr. 2021

[11] K. Kartika, M. Misriana, and J. Julsam, "Owl pest security doors in swiftlet breeding houses based on microcontrollers," in Proceedings of National Conference Politeknik Negeri Lhokseumawe, vol. 5, no. 1, pp. 173–176, 2021.

[12] N. A. S. A. Shuyuti, E. Salami, M. Dahari, H. Arof, and H. Ramiah, "Application of artificial intelligence in particle and impurity detection and removal: A survey," IEEE Access, vol. 12, pp. 31498–31514, Jan. 2024

[13] A. Anggrawan, S. Hadi, C. Satria, and B. K. Triwijoyo, "Development of an internet of things-based air temperature and humidity control system for swiftlet houses using microcontroller ESP32," in Proceedings of the 2023 6th International Conference of Computer and Informatics Engineering (IC2IE), Banda Aceh, Indonesia, Sep. 2023, pp. 374–380

[14] A. R. Ibrahim, N. H. N. Ibrahim, A. N. Harun, M. R. M. Kassim, S. E. Kamaruddin, and G. Witjaksono, "Bird counting and climate monitoring using LoRaWAN in swiftlet farming for IR4.0 applications," in Proceedings of the 2018 2nd International Conference on Smart Sensors and Application (ICSSA), Kuching, Malaysia, Jul. 2018, pp. 33–37.

[15] A. R. Ibrahim, N. H. N. Ibrahim, A. N. Harun, M. R. M. Kassim, S. E. Kamaruddin, L. K. Meng, C. J. Hui, and G. Witjaksono, "Automated monitoring and LoRaWAN control mechanism for swiftlet bird house," in Proceedings of the 2018 International Conference on Intelligent and Advanced System (ICIAS), Kuala Lumpur, Malaysia, Aug. 2018, pp. 1–5.

[16] E. C. Tetila, T. S. Cavalcanti, M. P. Cacique, C. H. Tardelli, R. T. Souza, and R. R. Paz, "YOLO performance analysis for real-time detection of soybean pests," Smart Agricultural Technology, vol. 7, p. 100405, Jan. 2024.

[17] M. Abrar and D. P. Caniago, "Implementation of deep learning using YOLOv7 and Telegram notifications for preventing illegal fishing in the waters of Batam," Indonesian Journal of Computer Science, vol. 12, no. 5, pp. 2460–2473, 2023.

[18] Z. Zhang, J. Huang, G. Hei, and W. Wang, "YOLO-IR-Free: An improved algorithm for real-time detection of vehicles in infrared images," Sensors, vol. 23, no. 21, p. 8873, Nov. 2023.

[19] A. Wang, Z. Huang, J. Zhang, L. Xie, and Q. Tian, "YOLOv10: Real-time end-to-end object detection," Advances in Neural Information Processing Systems, vol. 37, pp. 107984-108011, 2024.

[20] F. Martinez, J. B. Romaine, P. Johnson, A. Cardona, and P. Millan, "Novel fusion technique for high-performance automated crop edge detection in smart agriculture," IEEE Access, vol. 13, pp. 45632–45641, Apr. 2025

[21] H. Y. Kim, T. J. Yi, and J. Y. Lee, "An attention-based convolutional neural network with spatial transformer module for automated optical inspection of small objects," IEEE Transactions on Instrumentation and Measurement, vol. 74, pp. 1–12, Jan. 2025.

[22] Y. Zhang, Y. Wei, H. Zhang, S. Zhang, and Z. Li, "External defect recognition of lightning arresters based on an improved YOLOv10 model," in Proceedings of the 2024 5th International Symposium on New Energy and Electrical Technology (ISNEET), Changsha, China, Dec. 2024, pp. 276–281.

[23] A. Afdhal, K. Saddami, M. Arief, S. Sugiarto, Z. Fuadi, and N. Nasaruddin, "MXT-YOLOv7t: An efficient real-time object detection for autonomous driving in mixed traffic environments," IEEE Access, vol. 12, pp. 23045–23057, Feb. 2024

[24] D. Hama, "Kecoa Night Vision Dataset," Apr. 2025. [Online]. Available: https://universe.roboflow.com/deteksi-hama-ey5lz/kecoa-night-vision