Edge-Accelerated Real-Time Egg Recognition Using YOLOv8

Wesam Basil Abdulhameed and Zaidoon Tareq Abdulwahab
Department of Computer Science, Computer Science and Mathematics College, Tikrit University, 34001 Tikrit, Iraq wesam.b.abdulhameed4231@st.tu.edu.iq, ziadoontareq@tu.edu.iq

Keywords: Object Detection, Deep Learning, Poultry, Eggs Detection, YOLOv8, Edge Computing.

Abstract: Mechanised egg-collection systems require vision models that are both accurate and light enough to run on embedded hardware. We built and evaluated an end-to-end pipeline that couples YOLOv8 object-detection variants with Google's Coral Edge TPU for real-time recognition of white and brown chicken eggs. A bespoke dataset of 971 images (640 × 480 px) was captured under diverse backgrounds and lighting, annotated in YOLO format, and split 70 %/20 %/10 % for training, validation and testing. Five YOLOv8 models (n, s, m, l, x) were trained for 100 epochs with a batch size of 16. All models achieved very high accuracy (mAP50 = 0.98), but YOLOv8s produced the best F1–confidence pairing (F1 = 0.98 at 0.703 confidence), while YOLOv8n offered the lowest computational load. Converting the networks to TensorFlow-Lite and compiling them for the Edge TPU boosted inference speed dramatically: YOLOv8n jumped from 2.4 FPS on Raspberry Pi 5 (PyTorch) to 13.8 FPS on Edge TPU, and YOLOv8s rose from 1.0 FPS to 4.1 FPS, with only marginal accuracy loss. Precision and recall remained ≥ 0.96 across all variants. These results demonstrate that lightweight YOLOv8 models, particularly the n-variant, are suitable for embedded, robotics-grade egg-collection systems that demand real-time performance without sacrificing detection quality. Future work will expand the class set to include damaged eggs and integrate the detector into a closed-loop robotic gripper to enable fully autonomous on-farm operation.

1 INTRODUCTION

The field of computer vision has experienced remarkable progress, particularly in object detection. Among the most influential approaches, YOLO (You Only Look Once) models have emerged as leading solutions. Owing to their real-time processing capability, YOLO-based models have gained widespread popularity, especially for applications that require rapid decision-making. Among the various versions, YOLOv8 represents one of the most powerful and versatile architectures, achieving state-of-the-art accuracy and efficiency in a wide range of object recognition tasks [1].

The evolution from earlier YOLO versions to YOLOv10 demonstrates a continuous improvement in speed, precision, and flexibility. These advancements address previous limitations, such as reduced visibility under harsh weather conditions and difficulties in detecting small or partially occluded objects [2]. Such improvements make YOLO models particularly suitable for deployment in edge computing scenarios, where computational resources are limited and real-time performance is essential [1].

In many application domains, including agriculture, food processing, and quality control, accurate and real-time egg classification has become increasingly important [3]. Traditional egg detection and identification methods are often manual, labor-intensive, and prone to error, making them unsuitable for large-scale deployment. In contrast, edge-accelerated real-time object recognition significantly enhances production efficiency and ensures consistent quality control. This approach minimizes irregular inspections at critical stages and enables organizations to meet both visual and industrial standards without incurring additional labor costs or processing delays [4].

Edge computing, which processes data closer to the source, offers several advantages over cloud-based solutions. It reduces latency, improves bandwidth efficiency, and enhances data privacy by minimizing data transmission over networks. These characteristics make edge computing particularly suitable for environments with limited or unreliable network connectivity.

This paper is structured as follows. Section 2 presents the literature review. Section 3 describes the materials and methodology, including the stages of egg detection models, data collection, data preprocessing, model training, deployment, real-time detection, and performance evaluation metrics. Section 4 discusses the experimental results, and the final section concludes the study.

2 LITERATURE REVIEW

This study provides a comprehensive review of the literature on real-time egg recognition at the edge using YOLOv8, focusing on applications in computer vision and machine learning, particularly for egg detection using deep learning technologies. Research in this domain can be examined through several key studies that have shaped the direction of current investigations.

First, researchers in [5] developed a practical method for counting egg production in free-range chicken environments using an IoT-based camera system. Their work emphasized the importance of accurate object detection and species identification, both of which are essential for monitoring individual egg production. This foundational study highlighted the need for precise detection thresholds and established a basis for further research into automated agricultural data collection.

This research direction was further extended by studies focusing on edge-accelerated recognition within the broader context of IoT, real-time processing, and local computation [6]. These works highlighted the importance of edge data processing in achieving near real-time performance on embedded devices using convolutional neural networks. They also demonstrated the diversity of edge hardware capabilities and the necessity of considering these constraints when evaluating detection algorithms, thereby laying the groundwork for subsequent real-time edge computing studies.

In the same period, the framework EdgeLens was proposed in [7], integrating deep learning with fog and cloud computing environments to improve service quality for object detection and monitoring. This study illustrated the growing demand for efficient processing solutions in data-intensive applications such as agriculture and reinforced the importance of real-time performance in such environments.

The benefits of edge-based image processing were further demonstrated in [8], particularly for tasks such as dietary intake assessment using food recognition. Khosla et al. focused on mobile applications, emphasizing not only acceleration of recognition processes but also the practical deployment of computer vision models on resource-constrained devices.

Additionally, the feasibility of deploying deep learning applications on edge-enabled drones was demonstrated in [9], highlighting the applicability of real-time computing in dynamic environments such as agricultural fields. This work showed that edge computing can effectively support deep learning tasks in scenarios where mobility and rapid decision-making are critical.

At the same time, the capabilities of edge computing were further expanded in [10], where hardware accelerators were identified as key contributors to reduced latency and improved computational efficiency. The authors emphasized that, as the volume of data generated at the edge continues to grow, tasks such as egg identification become increasingly critical for real-time processing.

An improved YOLO-based algorithm for remote sensing applications was presented in [11], demonstrating how advances in deep learning can enhance detection performance under challenging conditions. This study highlighted the necessity for real-time detection systems to balance both accuracy and speed - an essential requirement for egg recognition tasks.

A distributed processing strategy for smart vehicles within the Internet of Things was proposed in [12], emphasizing the importance of real-time application control for high-efficiency computational tasks. This work contributes to ongoing research on optimizing edge computing strategies to improve object detection performance.

According to [13], YOLO-based technologies have proven to be highly effective for fast object recognition, offering significant speed advantages. Such characteristics are especially important for applications requiring immediate feedback, including real-time egg classification systems.

A comprehensive review of major edge machine learning techniques was presented in [14], identifying open research challenges and future directions. The findings provide a detailed overview of edge-enabled machine learning solutions, particularly in the context of agricultural applications.

Optimized edge video analytics was explored in [15], where cooperative processing among distributed nodes was proposed. This approach has the potential to improve the efficiency of real-time egg recognition systems that currently rely on centralized computation.

More recently, [16] examined real-time object detection systems based on YOLOv8 and its predecessors, emphasizing the trade-off between detection accuracy and computational cost. Although significant improvements were achieved, the study noted that challenges remain in deploying fully effective egg recognition systems in real-world environments.

Finally, [17] focused specifically on improving YOLOv8 for pigeon egg detection, addressing both theoretical and practical challenges. This work successfully improved egg counting accuracy in complex environments and provided benchmark results for future robotic and automated monitoring systems.

In summary, this literature review highlights significant advances in edge-accelerated real-time egg recognition, emphasizing the strong interaction between technological innovation and practical agricultural applications. Collectively, these studies demonstrate how decentralized processing and deep learning techniques can fundamentally transform traditional egg monitoring, sorting, and surveillance systems.

3 MATERIALS AND METHODOLOGY

This section involves six stages to approach the objective of the study:

3.1 Egg Detection Model Stages

Figure 1 shows essential checkpoints per step along with a path summary describing the egg detection model deployment process. This involves a three-phase approach:

1) Data Preparation. The first phase of the project where egg images are collected, preprocessed, filtered, and annotated in order to create the dataset to train, validate and test the YOLO model.
2) Model Implementation. In this phase, selected a deep learning model, trained using the training, validation and test datasets, and subsequently evaluated and assessed on the test dataset.
3) Model Inference. The final step entails applying the detection model to new egg images to verify the effectiveness of the developed model.

3.2 Data Collection

The dataset was obtained by photographing a group of chicken eggs in a studio box, on various grass or dirt surfaces, at different distances from the lens, and in different positions and orientations to simulate all the possibilities that the images to be used for object detection might present in real-time applications. The dataset consists of 971 JPG images with a resolution of 640 x 480 pixels, divided into 473 images of brown eggs, 488 images of white eggs, and 10 images containing both types. Figure 2 shows examples of the images used.

3.3 Data Pre-Processing

The dataset was filtered, annotated, and classified. The process of filtering and labeling the data includes the following steps:

1) Exclude blurry images and images that do not clearly contain potential objects.
2) Our work is divided into two classes, white and brown chicken eggs. Images containing eggs of other birds have been excluded.
3) It has been manually labeled, rectangle annotation is applied on the object, the rectangle covers all the exterior borders of the object.
4) Label studio software is used for annotation, the annotated data is exported in YOLO format.
5) The final dataset is divided into train, validation and test folders according to the ratio with is mentioned in Figure 1.

3.4 Model Training

Python 3.11 was used in the model's development. Google Colab was used to train and validate the algorithms. The deep learning approach used the deep convolutional neural network (CNN) model to train, detect, and classify the eggs as shown in Figure 3.
In this study, it was employed the YOLOv8 architecture for efficient object detection. This architecture partitions an image into a grid system, with each grid module responsible for detecting objects within its boundaries. A total of 100 epochs and batch size 16 have been performed to test each version. The eggs dataset is divided into three groups and was randomly split into an internal training set (70 %), validation set (20 %), and test set (10 %).

3.5 Model Deployment and Real-Time Detection

The trained model is compiled into TensorFlow lite format, in order to be compatible with the edge computing device, which is Coral Edge TPU device. The Coral Edge TPU device is a device produced by Google to accelerate the artificial intelligence operations in micro devices, like Raspberry Pi. Raspberry Pi 5 is used to run the real-time egg detection. The real-time object detection is performed by reading a video stream from a live camera as shown in the Figure 4. The model was tested with samples taken in a simulated poultry farm environment, including images of eggs on dirt, grass, and among bunches of hay, due to the lack of access to real poultry farms during the study. The results obtained were within the range obtained in the initial test.

3.6 Model Performance Evaluation Metrices

The detection performance of the proposed model was evaluated using standard and widely accepted object detection metrics, commonly employed in YOLO-based frameworks. These include precision, recall, F1-score, Intersection over Union (IoU), mean Average Precision (mAP), as well as training and validation losses (box loss, object loss, and class loss).

Precision measures the proportion of correctly detected objects among all detections made by the model at a given confidence threshold, reflecting the model's ability to avoid false positives.

Recall evaluates the model's ability to correctly detect all relevant objects, indicating how effectively the network identifies true targets within an image.

The F1-score provides a single, comprehensive indicator of detection performance by harmonically balancing precision and recall. This metric is particularly useful for summarizing overall model effectiveness, especially in scenarios where class imbalance may be present.

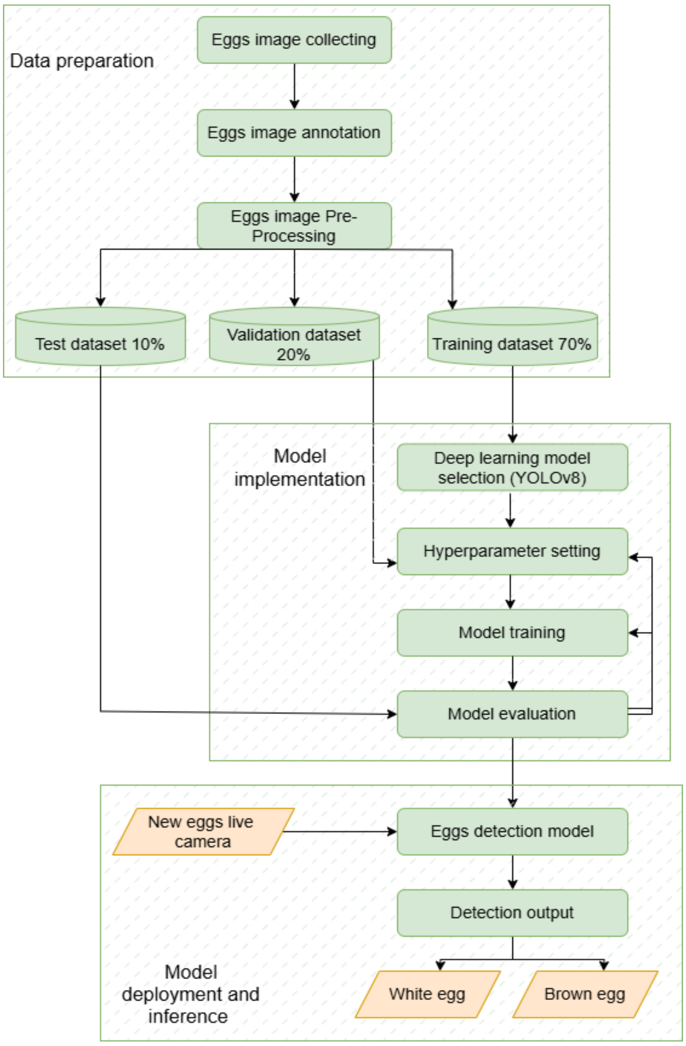


Figure 1: Methodological framework for the egg detection.

Figure 2: Eggs dataset samples.

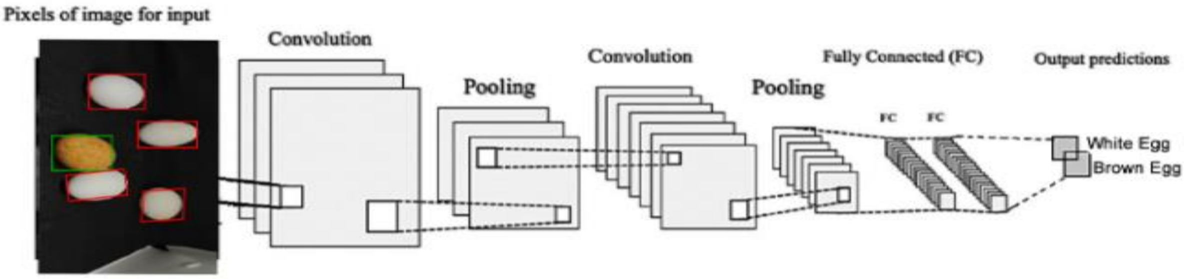
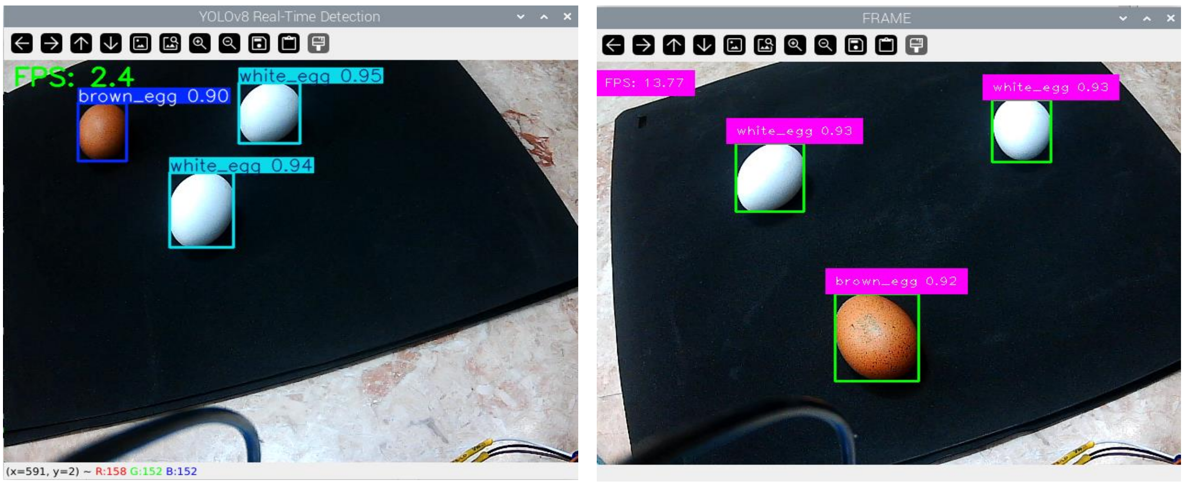


Figure 3: The architecture of the deep learning model employed a CNN.



(a) (b)

Figure 4: a): Live stream object detection using trained YOLOv8n model; b) : Live stream object detection used a TFLite model converted from YOLOv8n model, with Edge TPU.

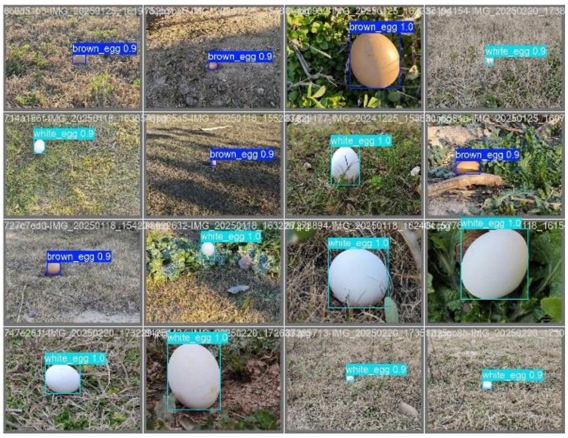


Figure 5: Results of Egg detection YOLOv8n model.

The Intersection over Union (IoU) is defined as the ratio of the overlapping area between the predicted bounding box and the ground-truth bounding box to the total area covered by both. It quantifies the spatial accuracy of object localization and the model's ability to distinguish objects from the background.

Average Precision (AP) and its aggregate form, mean Average Precision (mAP), are standard benchmarks for evaluating object detection models. These metrics summarize detection accuracy across different confidence thresholds and IoU levels, providing a robust measure of overall model performance.

All evaluation metrics were computed based on the conventional confusion matrix components [1]:

- TP (True Positive). The number of instances that were correctly predicted as positive.
- TN (True Negative). The number of instances that were correctly predicted as negative.
- FP (False Positive). The number of instances that were incorrectly predicted as positive.
- FN (False Negative). The number of instances that were incorrectly predicted as negative.

4 RESULTS

This section presents the results of the egg detection model using YOLOv8. Several YOLOv8 model variants were evaluated, including (a) YOLOv8n, (b) YOLOv8s, (c) YOLOv8m, (d) YOLOv8l, and (e) YOLOv8x.The result of these models is shown in Figure 6. More details are described in the following subsections: The YOLOv8 model incorporates box losses in its object detection algorithm to enhance the accuracy of object detection and classification within an image. The goal during training is to minimize the losses to the lowest possible value in the meantime maintain the detection time as less as possible. The precision and recall values approached unity, indicating excellent performance. The mean average precision, the most commonly used statistic, also met expectations. Figure 5 shows the training box loss of the dataset.

A confusion matrix for each YOLOv8 model is used to analyze the accuracy of the collected data in detection and classification of eggs.

The top left cell represents the number of sample of the true predicted brown eggs (True positives), while the middle cell represents the true predicted white eggs (True positives). The cells in the first row except the first cell represent the brown egg labels those incorrectly classified as other labels (white eggs or background), these cells are recognized as (False negative). The first column cells except the first cell are represent the non-brown egg labels those are predicted as brown eggs, this is called (False positive). The rest of the cells represent the non-brown labels those are predicted as non-brown eggs (True negative).

The confusion matrix of the white eggs are described as the following: The middle cell the percentage of the true predicted white eggs (True positives). The cells in the second row except the middle cell represent the white egg labels those incorrectly classified as other labels (brown eggs or background), these cells are recognized as (False negative). The second column cells except the middle cell are represent the non-white egg labels those are predicted as white eggs, this is called (False positive). The rest of the cells represent the non-white labels those are predicted as non-white eggs (True negative).

The highest score achieved by the eggs detection model is 0.98, with best confidence level of 0.703 in YOLOv8s model, Table 1 shows the F1-confidence scores of each YOLOv8 model.

Table 1: F1 score-confidence of YOLOv8 models.

Model	F1 score	Confidence
YOLOv8n	0.98	0.615
YOLOv8s	0.98	0.703
YOLOv8m	0.97	0.737
YOLOv8l	0.97	0.389
YOLOv8x	0.97	0.726

A confidence level greater than 0.788 indicates satisfactory precision in the scores which is considered good for the developed eggs detection model using YOLOv8n model. Table 2 shows the Precision-confidence values of each YOLOv8 model.

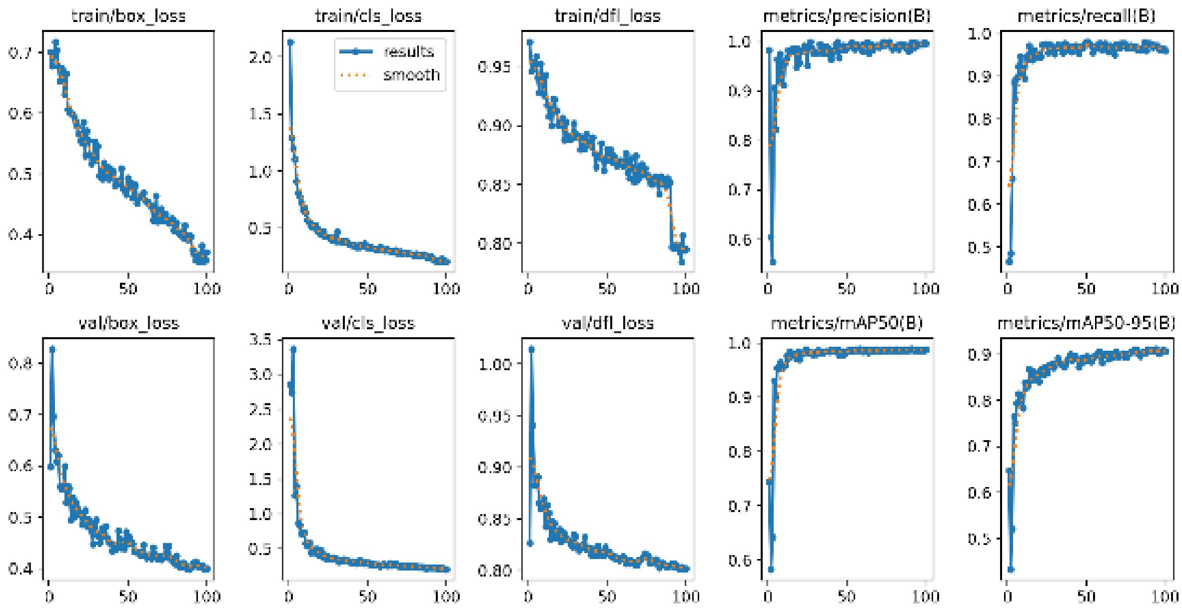


Figure 6: Performance training analysis with YOLOv8n model.

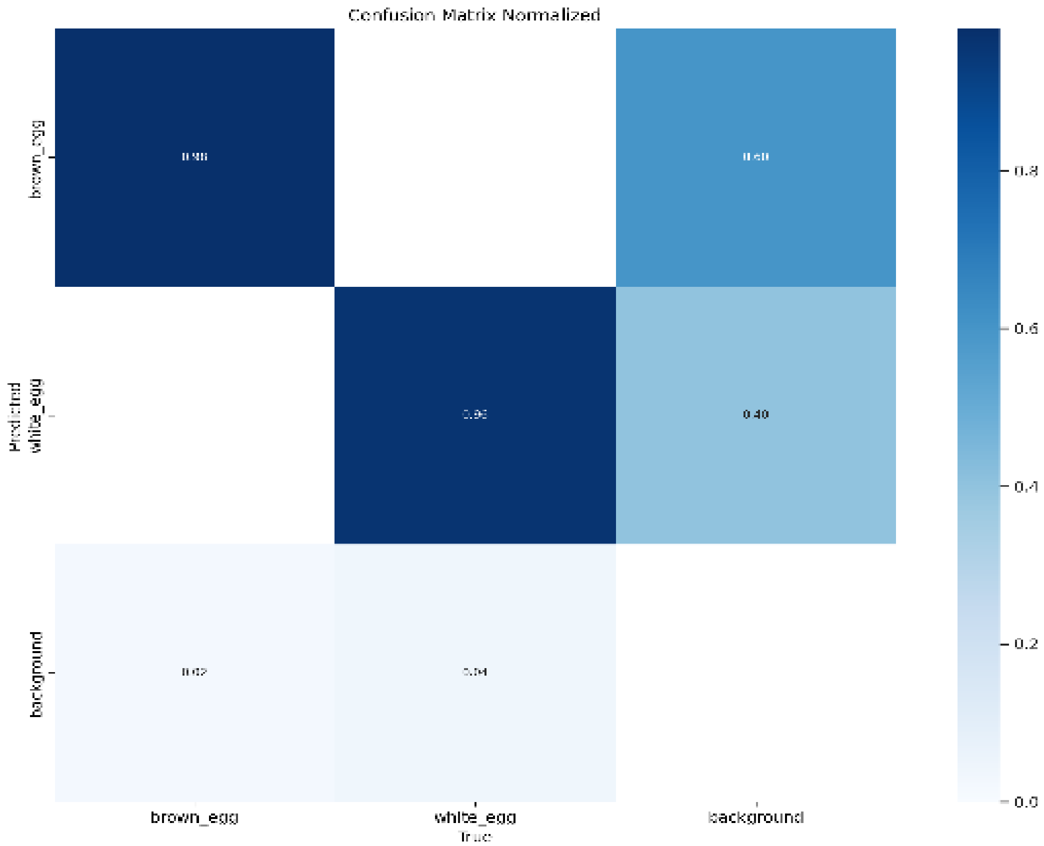


Figure 7: The confusion matrix of YOLOv8n model.

Table 2: Precision-confidence values of the trained YOLOv8 models.

Model	Confidence
YOLOv8n	0.788
YOLOv8s	0.788
YOLOv8m	0.836
YOLOv8l	0.890
YOLOv8x	0.858

Table 3 shows a comparison of the eggs detection results between YOLOv8 models.

Table 3: The computational time of YOLOv8 models training.

Models	mAP 0.5	Precision	Recall	F1-score
YOLOv8n	0.98	0.98	0.97	0.98
YOLOv8s	0.98	0.99	0.97	0.98
YOLOv8m	0.98	0.99	0.95	0.97
YOLOv8l	0.98	0.96	0.98	0.97
YOLOv8x	0.98	0.98	0.95	0.97

Using the edge computation have made a significant different in FPS (Frame per second) of the live stream, which effects on the speed of detection. This matter could be a critical factor if the detection used in robotics or in dynamic based applications. Table 4 shows a comparison in FPS values of each YOLOv8 model and the corresponding tensorFlow lite model.

Table 4: FPS values for each YOLOv8 and TFlite models.

Model	YOLOv8 FPS	TFLite FPS
YOLOv8n	2.4	13.77
YOLOv8s	1	4.07
YOLOv8m	0.4	0.69
YOLOv8l	0.2	0.87
YOLOv8x	0.1	0.65

5 CONCLUSIONS

This study shows that coupling state-of-the-art one-stage detectors with edge accelerators can eliminate the long-standing trade-off between speed and accuracy in agricultural machine-vision tasks. Training five YOLOv8 variants on a purpose-built two-class egg dataset yielded uniformly high mAP50 (0.98) and F1 scores (≥ 0.97). When deployed on a Raspberry Pi 5 alone, inference rates were inadequate for mobile platforms; however, compiling to TensorFlow-Lite and off-loading computations to the Coral Edge TPU delivered up to a 5.7 × throughput increase, achieving real-time (≥ 13 FPS) performance with the YOLOv8n model. The acceleration came with negligible precision–recall degradation, confirming that the wider YOLOv8 design - especially the atomic-layered n-variant-matches the resource envelope of embedded poultry robots. Consequently, commercial egg-gathering systems can now exploit inexpensive edge hardware to monitor laying patterns continuously while minimising breakage and labour. Future directions include enlarging the dataset to cover occlusions and soiled shells, testing multi-class damage grading, and integrating tracking to support cooperative fleets of robotic collectors.

As a future work, it’s highly recommended to expand dataset and test robustness, by enlarging the image corpus beyond clean white and brown shells to include cracked, soiled and partially occluded eggs, other kinds of poultry birds like geese, ducks and quails, and embedding the detection model in a micro-controller that runs robotic features like arms, grippers and navigation materials, to maximize the benefit of the study.

REFERENCES

[1] M. Hussain, “YOLOv5, YOLOv8 and YOLOv10: The go-to detectors for real-time vision,” arXiv preprint arXiv:2407.02988, 2024.

[2] A. Sundaresan Geetha, M. A. R. Alif, M. Hussain, and P. Allen, “Comparative analysis of YOLOv8 and YOLOv10 in vehicle detection: Performance metrics and model efficacy,” Vehicles, vol. 6, pp. 1364–1382, 2024.

[3] T. T. Nguyen, C. Wilson, I. Khan, and J. Dalins, “Object detection approaches to identifying hand images with high forensic values,” in Proc. 2024 IEEE Int. Conf. Systems, Man, and Cybernetics (SMC), 2024, pp. 2727–2734.

[4] Z. Qi, L. Ding, X. Li, J. Hu, B. Lyu, and A. Xiang, “Detecting and classifying defective products in images using YOLO,” arXiv preprint arXiv:2412.16935, 2024.

[5] J. Lubich, K. Thomas, and D. W. Engels, “Identification and classification of poultry eggs: A case study utilizing computer vision and machine learning,” SMU Data Science Review, vol. 2, p. 20, 2019.

[6] X. Zhang, Y. Wang, S. Lu, L. Liu, and W. Shi, “OpenEI: An open framework for edge intelligence,” in Proc. 2019 IEEE 39th Int. Conf. Distributed Computing Systems (ICDCS), 2019, pp. 1840–1851.

[7] S. Tuli, N. Basumatary, and R. Buyya, “Edgelens: Deep learning based object detection in integrated IoT, fog and cloud computing environments,” in Proc. 2019 4th Int. Conf. Information Systems and Computer Networks (ISCON), 2019, pp. 496–502.

[8] M. Sarwar Murshed, C. Murphy, D. Hou, N. Khan, G. Ananthanarayanan, and F. Hussain, “Machine learning at the network edge: A survey,” arXiv e-prints, arXiv:1908.00080, 2019.

[9] A. Koubâa, A. Ammar, M. Alahdab, A. Kanhouch, and A. T. Azar, “DeepBrain: Experimental evaluation of cloud-based computation offloading and edge computing in the Internet of Drones for deep learning applications,” Sensors, vol. 20, p. 5240, 2020.

[10] H. Padmasiri, J. Shashirangana, D. Meedeniya, O. Rana, and C. Perera, “Automated license plate recognition for resource-constrained environments,” Sensors, vol. 22, p. 1434, 2022.

[11] X. Shen, G. Shi, H. Ren, and W. Zhang, “Biomimetic vision for zoom object detection based on improved vertical grid number YOLO algorithm,” Frontiers in Bioengineering and Biotechnology, vol. 10, p. 905583, 2022.

[12] S. Bhatia, R. I. Alsuwailam, D. G. Roy, and A. Mashat, “Improved multimedia object processing for the Internet of Vehicles,” Sensors, vol. 22, p. 4133, 2022.

[13] R. Chandana and A. Ramachandra, “Real-time object detection system with YOLO and CNN models: A review,” arXiv preprint arXiv:2208.0773, 2022.

[14] M. Hu, Z. Luo, A. Pasdar, Y. C. Lee, Y. Zhou, and D. Wu, “Edge-based video analytics: A survey,” arXiv preprint arXiv:2303.14329, 2023.

[15] X. Yang, R. B. Bist, S. Subedi, and L. Chai, “A computer vision-based automatic system for egg grading and defect detection,” Animals, vol. 13, p. 2354, 2023.

[16] M. Łysakowski, K. Żywanowski, A. Banaszczyk, M. R. Nowicki, P. Skrzypczyński, and S. K. Tadeja, “Real-time onboard object detection for augmented reality: Enhancing head-mounted display with YOLOv8,” in Proc. 2023 IEEE Int. Conf. Edge Computing and Communications (EDGE), 2023, pp. 364–371.

[17] T. Jiang, J. Zhou, B. Xie, L. Liu, C. Ji, Y. Liu, et al., “Improved YOLOv8 model for lightweight pigeon egg detection,” Animals, vol. 14, p. 1226, 2024.