JAST (Journal of Animal Science and Technology) TITLE PAGE

Upload this completed form to website with submission

ARTICLE INFORMATION	Fill in information in each box below
Article Type	Research article
Article Title (within 20 words without abbreviations)	Broiler detection system for multi-tier broiler house using a cloud platform
Running Title (within 10 words)	Broiler detection system using cloud platform
Author	Jung-sang Yoo1,2, Taehyeong Kim1,2,3, Chang-hyup Lee1,2, Jung-Sun Gloria Kim1,2, JoonYong Kim1,3, Ghiseok Kim1,2,3, JoongYong Rhee1,2,3
Affiliation	1 Department of Biosystems Engineering, Seoul National University, Seoul 08826, Republic of Korea 2 Convergence Major in Global Smart Farm, Seoul National University, Seoul 08826, Republic of Korea 3 Research Institute of Agriculture and Life Sciences, Seoul National University, Seoul 08826, Republic of Korea
ORCID (for more information, please visit https://orcid.org)	Jung-sang Yoo: https://orcid.org/0009-0000-1435-9207 Taehyeong Kim : https://orcid.org/0000-0002-9608-4362 Chang-hyup Lee : https://orcid.org/0000-0002-8459-7814 Jung-Sun Gloria Kim : https://orcid.org/0000-0003-3017-5146 JoonYong Kim : https://orcid.org/0000-0003-2395-4387 Ghiseok Kim : https://orcid.org/0000-0003-2177-0031 JoongYong Rhee : https://orcid.org/0000-0002-6351-4055
Competing interests	No potential conflict of interest relevant to this article was reported.
Funding sources State funding sources (grants, funding sources, equipment, and supplies). Include name and number of grant if available.	This work was funded by the ministry of Agriculture, Food and Rural Affairs (MAFRA) (No. 319093-04) and the BK21 FOURH, Global Smart Farm Educational Research Center, Seoul National University, Seoul, Korea
Acknowledgements	Not applicable
Availability of data and material	Upon reasonable request, the datasets of this study can be available from the corresponding author
Authors' contributions Please specify the authors' role using this form.	Conceptualization: Taehyeong Kim, JoonYong Kim Data curation: Chang-hyup Lee, Jung-Sun Gloria Kim Formal analysis: Jung-sang Yoo, JoongYong Rhee Methodology: Jung-sang Yoo, JoongYong Rhee Software: Jung-sang Yoo, JoonYong Kim Validation: Ghiseok Kim Investigation: Taehyeong Kim Writing - original draft: Jung-sang Yoo Writing - review & editing: JoongYong Rhee
Ethics approval and consent to participate	This study does not require IRB/IACUC approval because the data in this research was collected by cameras which were already installed in a farm, there was no physical contact or any action that would cause psychological stress with the broilers.

CORRESPONDING AUTHOR CONTACT INFORMATION

For the corresponding author (responsible for correspondence, proofreading, and reprints)	Fill in information in each box below
First name, middle initial, last name	JoongYong Rhee

Email address – this is where your proofs will be sent	jyr@snu.ac.kr
Secondary Email address	
Address	200-2007, 1, Gwanak-ro, Gwanak-gu, Seoul, Republic of Korea
Cell phone number	
Office phone number	+82-2-880-4605
Fax number	

Abstract

Broilers are bred at a higher density than large livestock, and their relatively small size makes monitoring challenging, requiring a reliable, cost-effective automatic monitoring system for precision farming. Particularly in the case of managing large-scale broiler breeding systems, such as multi-tier broiler houses, it is important to reduce investment in computing facilities and to improve productivity through automatic monitoring technology. However, financial ability of farmers to equip and manage the high-specification computing equipment required by these technologies is limited, necessitating development of affordable monitoring systems. We there propose a lightweight broiler-detection model that augments YOLOv5 with two efficiency modules: Bi-directional Feature Pyramid Network (BiFPN), which fuses multi-scale features with learnable weights to sharpen small-object and GhostNet, which replaces many convolutions with inexpensive “ghost” operations to cut parameters. Performance was benchmarked against the original YOLOv5 on a high-spec local GPU machine and CPU based cloud platform. The proposed model attained 90.5% mAP@0.5, marginally below the baseline’s 92.8%, yet inference time dropped by 30% on CPU (50.9 ms) and 50% on GPU (1.8 ms). By implementing lightweight models, farmers can utilize monitoring systems from anywhere via mobile devices, leveraging cloud-based technology. This eliminates the necessity for expensive hardware, offering an affordable and practical solution for improving farming management.

Keywords: Broiler, Multi-tier broiler house, Object detection, Cloud platform

Introduction

The global broiler industry is a cornerstone of livestock production, with annual consumption surpassing 60 billion birds and continuing to grow through large-scale breeding operations [1, 2]. This intensification, while meeting market demand, magnifies critical challenges. The high density of birds in large-scale farms increases the economic risks posed by infectious diseases and heightens concerns regarding animal welfare, demanding more sophisticated management and oversight [3-5].

In response, housing technologies are evolving from conventional floor-based systems to modern multi-tier cage systems. While traditional floor rearing faces difficulties with hygiene management and waste disposal [6-8], multi-tier houses offer enhanced spatial efficiency, automated cleaning, and improved sanitation [9-11]. However, this vertical structure makes direct, manual inspection of every tier impractical for farm workers, creating an urgent need for automated monitoring technologies to ensure flock health and productivity [12].

Precision Livestock Farming (PLF) offers a technological solution, employing sensors and artificial intelligence to automate monitoring tasks [13-15]. Machine vision, in particular, enables real-time analysis of broiler populations, health, and behavior [24-31]. A significant barrier, however, prevents its widespread adoption: the high computational demand of current object detection models. These models often require expensive, high-performance Graphics Processing Units (GPUs), placing them beyond the financial reach of many farmers and hindering the technology's practical application [19].

Cloud platforms present a viable pathway for deploying these AI services, offering scalable infrastructure and reducing maintenance overhead for end-users [20-22]. Yet, this approach introduces its own economic challenge. Cloud services that utilize high-performance GPUs for machine learning are substantially more expensive than standard CPU-based services [23]. Therefore, for a monitoring solution to be both scalable and affordable for the agricultural sector, it must be engineered to perform efficiently on low-cost CPU infrastructure.

This study directly addresses this cost-performance challenge by developing an object detection model for broilers that is optimized for CPU-based cloud environments. To achieve this, we constructed a lightweight architecture using the CPU-efficient GhostNet [32] as a backbone and integrated Bi-directional Feature Pyramid Network (BiFPN) module [33] to preserve high accuracy. We validated our model's performance against existing methods on both a local GPU machine and a CPU-based cloud server. The result is an economical and accessible solution that empowers farmers with real-time, remote monitoring capabilities, thereby advancing the practical application of PLF in both multi-tier and conventional broiler farms.

Materials and Methods

Overall Architecture

Fig. 1 shows the overall architecture of proposed system. Multiple cameras were installed on each floor to cover all areas, which is important to monitor health and distribution of broilers for PLF. The images for the training phase and the cloud-service phase were collected from the cameras. In the training phase, the images for training were sent to a local machine equipped with a GPU to build object detection models. A large number of images were collected as a priority and then sent to a local machine equipped with a GPU. The deep learning-based broiler detection model was optimized using the local machine. After training, the final model was migrated to a CPU-based cloud platform. In the cloud-service phase, once images were captured from the multiple cameras, it was directly sent to the cloud platform and object detection was conducted. Then, the broiler detection results were delivered and visualized on a web browser, mobile, or tablet.

This study does not require IRB/IACUC approval because the data in this research was collected by cameras which were already installed in a farm, there was no physical contact or any action that would cause psychological stress with the broilers.

Data Acquisition and Data Augmentation

The experimental data, as depicted in Fig. 1(a), were collected from a multi-tier broiler house (13.8 m × 70 m × 13.4 m) located in Gangneung city, Gangwon Province, South Korea, in June 2022. Approximately 3,000 broilers (Ross × Ross 708) were raised per tier. The image data were captured using cameras positioned for both the top and side views on each floor. A Raspberry Pi camera (RPI 8MP Camera V2, Raspberry Pi Foundation, UK) was used to obtain images from the top as shown in Fig. 1(b), while images from side were captured using an IP Camera (HIKVISION DS2CD 4 mm, Hangzhou

HIKVISION Digital Technology Co., Ltd, China) at a resolution of 640 × 640 pixels, as shown in Fig. 1(c).

A total of 420 images were captured, with 218 taken from the top view and 202 from the side view. To collect images of broilers in various poses and appearances, the images were taken at 3-hour intervals over a two-day period. The collected images were divided into training, validation, and test sets in a 6:2:2 ratio. Specifically, the training set consisted of 252 images and the validation set comprised 84 images, both of which were used to optimize the deep learning model. The test set included 84 images, evenly split with 42 images from each view, and these were not augmented. On the other hand, the training and validation dataset were augmented through various methods to improve the generalization performance of the object detection. Augmentation can significantly increase the diversity of data that can be used for model training without collecting additional data. In this study, horizontal and vertical flipping, brightness and contrast adjusting, and blurring were randomly applied as shown in Fig. 2. Finally, a total of 2,016 augmented images, 1,056 top views and 960 side views were used for model training and validation. Note that the data augmentation process was applied before the training process to reduce the time spent on learning.

Specification of Local Machine and Cloud Platform

In this study, a local machine with a GPU and the NCP, a cloud platform service, were used. For the local environment, a computer with an Nvidia RTX GPU 3090 (Nvidia Corporation, California, USA) was employed. However, no GPU was used on NCP to prioritize cost efficiency while still meeting affordable computational demands.

The CPU used in NCP was an Intel Xeon E5 v4 2660 (Intel Corporation, California, USA), a model released in 2016 that is commonly found in servers and workstations [34]. Detailed hardware specifications are provided in Table 1.

The broiler detection results were automatically transferred to user devices through FarmOS (version 3.0, FarmOS, Anyang, Republic of Korea). FarmOS is a cloud-based software and an environmental control system that provides information to farmers to help them grow crops and raise livestock.

Broiler Detection Model for the Cloud Environment

To achieve real-time monitoring of broilers in a multi-tier broiler house, improving both the accuracy and inference time for the object detection model is crucial for effectively managing the large volume of images taken by multiple cameras. Most farmers can’t afford a high-performance computer with a GPU, which is necessary for processing images from the cameras. The performance of existing object detection models was verified using GPU-based computers. However, when using a CPU-based cloud, the existing models are not suitable in terms of inference time, as they were developed with a focus on GPU optimization. This difference leads to inefficiencies when operating under CPU environments, highlighting the need for models that are tailored for CPU-based processing. In this study, the existing YOLOv5 model was optimized to improve accuracy and computational cost in a CPU-based cloud environment.

The architecture of YOLOv5, depicted in Fig. 3(a), is composed of five varied models: xlarge (X), large (L), medium (M), small (S), and nano (N). These models differ in size and computational demands, with the X model being the largest and slowest but yielding the highest accuracy, and the N model, the smallest and fastest, more suitable for limited-resource environments or real-time processing needs. A notable aspect across these models is the trade-off between detection accuracy and computational cost, making each one uniquely suited to different application scenarios and hardware setups. The architecture of YOLOv5 is divided into three modules: the backbone, the neck, and the head. The backbone, which utilizes CSP-Darknet53 [35], a derivative of the original Darknet [36], plays a vital role in extracting image features and creating a feature map. This map, a rich multi-dimensional array of extracted features, forms the foundation for subsequent detection processes. The backbone excels in drawing out basic feature representations, which are essential for the detection process. The neck, incorporating Path Aggregation Network for Instance Segmentation (PANet) [23] structure, adeptly merges low- and high-level features from this map. This fusion significantly bolsters the model's performance, leading to richer feature representations, enabling precise bounding box predictions and class identifications. The head of YOLOv5, critical for object detection, processes this enhanced feature map to determine object locations and generate the final output, including classes, objectness scores, and bounding boxes. YOLOv5's approach to object detection is both strategic and effective, employing predefined anchor boxes capable of detecting objects of diverse sizes. The model is enhanced with three specialized detectors—small, medium, and large—to accommodate a wide spectrum of object scales. These three detectors ensure thorough object detection throughout the image. YOLOv5's multi-scale detection framework is pivotal in maintaining both accuracy and efficiency in object identification and classification, catering to various object sizes within the input imagery.

To deploy a lightweight detection model in a CPU-based cloud at large-scale broiler house which has a large number of cameras, this study proposes a customized model derived from YOLOv5 as depicted in Fig. 3(b). CSP-Darknet53 generates a large number of parameters using a CNN, resulting in a significant computational load and is inefficient for deployment on devices without a GPU [37]. To overcome this problem, GhostNet was used to make the object detection model usable on a CPU-based cloud. Unlike CNNs, GhostNet can demonstrate similar performance with far fewer parameters by applying a linear transform [32]. Additionally, the original C3 layer which is based on CNNs was replaced with C3Ghost, and the Conv layer was replaced with GhostConv to improve the model inference time. The characteristics of GhostNet to maintain performance with fewer operations directly reduces computation cost, making it suitable for resource-limited CPU environments. In the neck, BiFPN was used instead of the PANet to improve detection performance. BiFPN, similar to PANet, has a pyramidal structure; however, unlike PANet's unidirectional approach, it introduces additional connections for a bidirectional approach to iteratively improve and fuse features, thereby expecting improved performance [33]. By facilitating efficient feature fusion and scaling, BiFPN enhances accuracy while minimizing a model complexity. Finally, in the head, the small and medium detectors were left. As the size of the broilers in the image was relatively large, the large detector, which is needed to recognize small objects, was removed. The head part uses only a total of six anchor boxes. This design results in a more lightweight model compared to existing models.

Model Evaluation Metrics

The performance of the proposed lightweight detection model was investigated by comparing it with five existing object detection models (EfficientDet, Faster R-CNN, SSD YOLOv5X and YOLOv5N).

The performance of each detection model was evaluated using mAP (mean Average Precision). To calculate mAP, a precision-recall curve and Average Precision (AP) are required. True Positve (TP), False Positive (FP), False Negative (FN), and True Negative (TN) were used to calculate the Precision and Recall. The precision is the ratio of correct detections to all detections that the model claims are correct as shown in Equation (1). Specifically, it is the ratio of detections that were correctly detected as broilers to all detections detected as broilers. Recall is the ratio of correct detections made by the model to the actual correct values as shown in Equation (2). In particular, this is the ratio of detections that the model detects as broilers out of the ground-truth broilers. The precision-recall curve, plotted by varying threshold levels for precision and recall, offers a detailed view of a model's performance at different confidence levels. It is crucial in imbalanced class scenarios, revealing the trade-off between precision and recall. AP represents the area under this precision-recall curve, and mAP is the average AP across all classes. In addition, the accuracy of object detection results varies depending on the threshold of IOU (Intersection over Union). In this study, the average accuracy was calculated for IOU values of 50 and 75.

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

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

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{3}$$

In terms of computational cost, average inference time for each model was measured by the test data set comprising 84 images. The primary metric for this evaluation was the mean time required to perform object detection on each individual image.

Results

Broiler Detection Performance

Fig. 4 shows the accuracy of the six object detection models used in this study as a function of the IOU threshold. Fig. 4(a) presents the performance of the six models at IOU@50. The EfficientDet, Faster R-CNN, and SSD models exhibited faster precision decay as recall increased compared to the YOLOv5X models. The best performing model for IOU@50 was YOLOv5N with an accuracy of 92.6 mAPs. The proposed model showed the second-best performance with 92.4 mAP, which was not a significant performance difference compared to the YOLOv5N model. For IOU@75, the precision value decreased rapidly as the recall value increased for all models, as shown in Fig. 4(b). The best performer was YOLOv5X with an accuracy of 42.4 mAP. The second-best model was YOLOv5N with 41.8 mAP. The proposed model showed the third best performance with 32.2 mAP; however, unlike the results at IOU@50, it showed a significant difference at IOU@75 compared to YOLOv5X. However, the proposed model still showed competitive performance with higher accuracy than the EfficientDet, Faster R-CNN, and SSD models. These trends indicate that lightweight architectures, while competitive at relaxed thresholds, are more sensitive to stricter bounding-box requirements, whereas heavyweight models retain higher precision at the cost of speed.

Computational Cost

Fig. 5 shows the performance differences between the six object detection models used in this study in terms of inference time. Fig. 5(a) demonstrates the performance comparison of the inference time on a GPU machine. The best performers on the GPU machine were the proposed model, YOLOv5N, and SSD. The proposed model showed faster performance than YOLOv5N (3.6 ms), with an average processing time of 1.8 ms per image. Fig. 5(b) presents a performance comparison of the inference time in a CPU-based cloud environment. Similar to the GPU results, the best performers in the CPU cloud were the proposed model, YOLOv5N, and SSD. The proposed model outperformed YOLOv5N (72.1 ms), with an average image processing time of 52 ms. In addition, the inference time rankings for YOLOv5X and

EfficientDet can be inverted between the GPU and CPU environments, highlighting the impact of hardware architecture on the computational costs of object detection models.

Discussion

Lightweight Detection for Broiler Monitoring

Lightweight detection has practical value only if it preserves sufficient accuracy for timely decision-making in barns. High-precision architectures usually come with longer inference times, and latencies approaching even a few hundred milliseconds are impractical for real-time barn monitoring, where alarms must follow abnormal events almost immediately. This situation highlights the familiar balance between speed and accuracy. Achieving real-time performance usually requires leaner models and a small concession in precision.

Many studies have focused on monitoring livestock diseases and behavior using deep learning-based object detection, proposing methods to automate the monitoring of animal behavior for accurate detection and tracking of their condition [38-40]. These approaches allow for real-time monitoring of animal health and welfare, minimizing the need for human intervention and enhancing farm productivity. However, most of these studies have used GPU-based machines, which have limitations in real-time object detection in general farms, primarily due to the high cost and maintenance of such specialized hardware [19]. In this study, a lightweight detection model was developed that improves the accuracy and inference time of object detection on a CPU-based cloud platform, aiming to use object detection in a multi-tier house with multiple cameras, and verified its performance by comparing it with other object detection models.

Comparative Results

Table 2 shows the comparative results of the object detection models and the proposed model. The performance of the proposed model was 90.5 % for IOU@50 and 32.2 % for IOU@75. The COCO dataset benchmark of the object detection models developed in 2021 achieved mAPs of approximately 50–60 for IOU@50 and 30–40 for IOU@75. Therefore, the proposed model performs well for IOU@50 but has an average performance for IOU@75 compared to the models developed in 2021. The overall accuracy of the proposed model lags behind that of the YOLOv5 models, which appears to be a problem caused by reducing the number of detectors in the head part of the existing architecture to two and applying GhostNet.

However, in terms of computational cost, the proposed model significantly reduces the size of the model. In particular, the proposed model was the lightest among the six models at 1.8 MB. The advantage of lightening the size of the model parameters is that it reduces the loading time when the computer loads the deep learning model, as well as does not require high-performance hardware, a critical factor for enabling deployment on diverse and cost-effective IoT devices [41]. This suggests that future IoT systems for barns could be adapted for embedded boards or mobile devices, making them more accessible [42].

Ablation Study

Table 3 shows the ablation study results obtained by changing the architecture of YOLOv5N. First, applying the BiFPN instead of the traditional PANet improves the overall accuracy performance compared to the model using PANet. However, BiFPN also increases the number of parameters, and the size of the model, which is likely due to the increased computation of the model by applying BiFPN, which is a two-way pyramid instead of a one-way pyramid.

Second, by applying GhostNet, the YOLOv5N model reduced the number of parameters by approximately 52 % from 1.76 m to 0.93 m compared to the previous model. The weight of the model was reduced from 3.7 MB to 2.2 MB as the number of parameters decreased. This is likely because the number of unnecessary features for detecting objects was reduced by using Linear Transfer instead of Convolution. However, the overall IOU value decreased in GhostNet compared to the original YOLOv5. The synergistic combination of BiFPN and GhostNet provides additional feature-fusion depth from BiFPN that compensates for the accuracy loss introduced by parameter of GhostNet pruning, yielding a more favorable accuracy–latency balance than either modification alone.

By reducing the number of detectors in the head of the YOLOv5 model, a corresponding reduction in model parameters was achieved, contributing to reduced computational load and faster inference times. Specifically, the number of detectors for large objects was reduced, resulting in a significant decrease in both model size and number of parameters. It was noted that in the absence of small or medium object detectors, connected layers such as GhostConv, Concat and C3Ghost were still required to process information for the large object detector. By contrast, with the removal of the large object detector, as shown in architecture Fig. 3(b), such layers became redundant, further minimizing the complexity and parameter footprint of the model. The removal of the large object detector resulted in a more significant improvement in accuracy than the removal of the small or medium detectors. This improvement was attributed to the predominant presence of large objects, such as chickens, in the evaluation dataset, which were adequately detected without the need for a dedicated large object detector. Conversely, the removal of the small object detector, which is responsible for detecting finer details and smaller objects, was found to be more detrimental to the overall performance of the model than maintaining the original, unmodified model. This approach underscores the importance of detector specificity relative to the scale of objects within the target dataset and suggests that a model optimized for large object detection may forego the complexity traditionally associated with multi-scale detection frameworks.

The impact of data augmentation on model performance was also analyzed. As shown in Table 4, data augmentation led to 0.8% improvement in IOU@50 while 1.2% improvement in IOU@75, demonstrating its effectiveness in improving the performance of the broiler detection model. These results highlight the importance of data augmentation in enhancing the performance of the proposed model, indicating its essential role in strengthening the robustness and generalization of broiler detection tasks.

Cloud-Based Broiler Monitoring

Fig. 6 illustrates the results of object detection using mobile phones. Leveraging the cloud platform allows farmers to access the monitoring results at any time and location. As the data are perpetually uploaded to the cloud, they are safeguarded against losses due to fire or other physical hazards. This continuous data stream could prove to be a valuable resource for future agricultural operations. Compared with prior GPU-based on farm systems, the cloud deployment eliminates local server costs and lets a single dashboard supervise multiple houses, which represents a practical innovation over existing research that often relies on localized, on-premise computing infrastructure [18, 20]. This shift towards a CPU-based cloud model directly addresses the economic challenges of scaling PLF technologies, as GPU instances on cloud platforms remain significantly more expensive [26].

In this study, a lightweight and high-performance broiler detection model was developed that is capable of rapidly processing a large number of images from multiple cameras installed for continuous broiler monitoring. The object detection model, which was improved for use in a cloud environment, demonstrated a more streamlined and faster performance without a significant loss of accuracy compared to YOLOv5N. Although the proposed model demonstrates much faster inference times, its performance is slightly lower than the original object detection model. However, since the broiler monitoring system operates continuously, the speed advantage can compensate for the minor reduction in accuracy. Thanks to this advantage, the design of a cloud platform facilitates fast access for operators through mobile or computer web browsers.

Limitation and Future Work

Images were captured using Raspberry Pi cameras and IP Cameras in a multi-tier broiler house. However, challenges were encountered in regularly collecting images due to environmental factors within the house, such as dust and ammonia, which resulted in insufficient image collection for the experiment. For instance, issues occurred with the Raspberry Pi camera installed in the top view where chickens attacked the exposed camera module, the Raspberry Pi board broke down owing to high concentrations of environmental ammonia, and images could not be obtained on time. Dust accumulation on the camera lens during the experiment also hindered the acquisition of clear images. This issue may pose challenges in larger broiler houses, where even dustier environmental conditions could result in dust accumulating on the camera lens. To get stable and clear images, future endeavors would require a dustproof camera or a suitable alternative to prevent from dust accumulation on the lens. This represents a well-known challenge in applying computer vision systems within real-world farm environments [27]. Furthermore, object detection errors, such as detecting a feeder as a broiler, happened occasionally from the side view. This occurred because of imprecise bounding boxes during the labeling process when chickens were feeding in or around the feeder. In the future, more precise bounding boxes should be drawn and labeled, anticipating to further enhanced performance.

Although this study did not predict diseases by targeting only the images of broilers, the proposed model could help develop an early disease detection system, paving the way for potential future extensions. Our approach is expected to enable the uninterrupted monitoring of broilers via a cloud platform, mitigating the negative impacts of epidemics and diseases on the broiler industry through early detection. Future work could build upon this model to detect subtle visual cues like posture, feather condition, or cosmetic issues, which have been identified as potential early indicators of animal health and welfare [19, 30].

CONCLUSIONS

This study demonstrates that an object detection model, which requires high-end computing, can be effectively implemented in a low-cost cloud-computing environment while maintaining high accuracy by utilizing of lightweight models. Although the proposed model showed slightly lower performance compared to the original YOLOv5, it achieved significant improvements in inference time across both CPU and GPU environments. This allows for the deployment of efficient and fast monitoring systems on low-spec computers, particularly for broiler monitoring. By successfully adapting object detection to resource-constrained environments, this research offers a practical solution for cases where computing power is limited. The proposed model can be integrated into existing farm systems such as real-time livestock monitoring and animal behavior tracking, making it accessible to more farms and improving the efficiency of animal management.

References

1. Li X, Zhao Z, Wu J, Huang Y, Wen J, Sun S, et al. Y-BGD: broiler counting based on multi-object tracking. Comput Electron Agric. 2022;202:107347. doi: 10.1016/j.compag.2022.107347.
2. Chaiban C, Robinson TP, Fèvre EM, Ogola J, Akoko J, Gilbert M, et al. Early intensification of backyard poultry systems in the tropics: a case study. Anim. 2020;14(11):2387-96. doi: 10.1017/S175173112000110X.

3. Lai S, Qin Y, Cowling BJ, Ren X, Wardrop NA, Gilbert M, et al. Global epidemiology of avian influenza A H5N1 virus infection in humans, 1997–2015: a systematic review of individual case data. Lancet Infect Dis. 2016;16(7). doi: 10.1016/S1473-3099(16)00153-5.
4. Espinosa R, Tago D, Treich N. Infectious diseases and meat production. Environ Resour Econ. 2020;76(4):1019-44. doi: 10.1007/s10640-020-00484-3.
5. Brito LF, Oliveira HR, McConn BR, Schinckel AP, Arrazola A, Marchant-Forde JN, et al. Large-scale phenotyping of livestock welfare in commercial production systems: a new frontier in animal breeding. Front Genet. 2020;11:793. doi: 10.3389/fgene.2020.00793.
6. Zhang L, Zhang Y, Liu Y, Cai S, Yuan J, Wang Z, et al. Effects of stocking density on immune function and oxidative stress level of Peking ducks reared on plastic wire-floor. China Poultry. 2015;37(12):31-4.
7. Grimes JL, Smith J, Williams CM. Some alternative litter materials used for growing broilers and turkeys. Worlds Poult Sci J. 2002;58(4):515-26. doi: 10.1079/WPS20020037.
8. Kyakuwaire M, Olupot G, Amoding A, Nkedi-Kizza P, Ateenyi Basamba T. How safe is chicken litter for land application as an organic fertilizer?: a review. Int J Environ Res Public Health. 2019;16(19):3521. doi: 10.3390/ijerph16193521.
9. Wang Z, Gao T, Jiang Z, Min Y, Mo J, Gao Y. Effect of ventilation on distributions, concentrations, and emissions of air pollutants in a manure-belt layer house. J Appl Poult Res. 2014;23(4):763-72. doi: 10.3382/japr.2014-01000.
10. Steenfeldt S, Nielsen BL. Welfare of organic laying hens kept at different indoor stocking densities in a multi-tier aviary system. I: egg laying, and use of veranda and outdoor area. Anim. 2015;9(9):1509-17. doi: 10.1017/S1751731115000713.
11. Fabian-Wheeler E. Unintended impacts on animal welfare and environment of combined farm animal housing with manure storage. 10th Int Livest Environ Symp (ILES X). Am Soc Agric Biol Eng. 2018. doi: 10.13031/iles.18-137.
12. Hao H, Fang P, Duan E, Yang Z, Wang L, Wang H. A dead broiler inspection system for large-scale breeding farms based on deep learning. Agriculture. 2022;12(8):1176. doi: 10.3390/agriculture12081176.
13. Berckmans D. General introduction to precision livestock farming. Anim Front. 2017;7(1):6-11. doi: 10.2527/af.2017.0102.
14. Wathes CM, Kristensen HH, Aerts JM, Berckmans D. Is precision livestock farming an engineer's daydream or nightmare, an animal's friend or foe, and a farmer's panacea or pitfall? Comput Electron Agric. 2008;64(1):2-10. doi: 10.1016/j.compag.2008.05.005.

15. Astill J, Dara RA, Fraser ED, Roberts B, Sharif S. Smart poultry management: smart sensors, big data, and the internet of things. Comput Electron Agric. 2020;170:105291. doi: 10.1016/j.compag.2020.105291.
16. Ma X, Lu X, Huang Y, Yang X, Xu Z, Mo G, et al. An advanced chicken face detection network based on GAN and MAE. Animals. 2022;12(21):3055. doi: 10.3390/ani12213055.
17. Kim T, Lee DH, Kim WS, Zhang BT. Domain adapted broiler density map estimation using negative-patch data augmentation. Biosyst Eng. 2023;231:165-77. doi: 10.1016/j.biosystemseng.2023.06.006.
18. Guo Y, Chai L, Aggrey SE, Oladeinde A, Johnson J, Zock G. A machine vision-based method for monitoring broiler chicken floor distribution. Sensors. 2020;20(11):3179. doi: 10.3390/s20113179.
19. Zhuang X, Zhang T. Detection of sick broilers by digital image processing and deep learning. Biosyst Eng. 2019;179:106-16. doi: 10.1016/j.biosystemseng.2019.01.003.
20. Xiao L, Ding K, Gao Y, Rao X. Behavior-induced health condition monitoring of caged chickens using binocular vision. Comput Electron Agric. 2019;156:254-62. doi: 10.1016/j.compag.2018.11.022.
21. Li G, Zhao Y, Purswell JL, Du Q, Chesser GD Jr, Lowe JW. Analysis of feeding and drinking behaviors of group-reared broilers via image processing. Comput Electron Agric. 2020;175:105596. doi: 10.1016/j.compag.2020.105596.
22. Lee J, Wang J, Crandall D, Šabanović S, Fox G. Real-time, cloud-based object detection for unmanned aerial vehicles. Proc IEEE Int Conf Robot Comput. 2017;36-43. doi: 10.1109/IRC.2017.77.
23. Popović T, Latinović N, Pešić A, Zečević Ž, Krstajić B, Djukanović S. Architecting an IoT-enabled platform for precision agriculture and ecological monitoring: a case study. Comput Electron Agric. 2017;140:255-65. doi: 10.1016/j.compag.2017.06.008.
24. Liu S, Guo L, Webb H, Ya X, Chang X. Internet of Things monitoring system of modern eco-agriculture based on cloud computing. IEEE Access. 2019;7:37050-8. doi: 10.1109/ACCESS.2019.2903720.
25. Jinbo C, Xiangliang C, Han-Chi F, Lam A. Agricultural product monitoring system supported by cloud computing. Clust Comput. 2019;22:8929-38. doi: 10.1007/s10586-018-2022-5.
26. Li H, Ota K, Dong M, Vasilakos AV, Nagano K. Multimedia processing pricing strategy in GPU-accelerated cloud computing. IEEE Trans Cloud Comput. 2017;8(4):1264-73. doi: 10.1109/TCC.2017.2672554.

27. Oliveira DAB, Pereira LGR, Bresolin T, Ferreira REP, Dorea JRR. A review of deep learning algorithms for computer vision systems in livestock. Livest Sci. 2021;253:104700. doi: 10.1016/j.livsci.2021.104700.
28. Szegedy C, Toshev A, Erhan D. Deep neural networks for object detection. Adv Neural Inf Process Syst. 2013;26.Geffen, O., Yitzhaky, Y., Barchilon, N., Druyan, S., Halachmi, I., 2020. A machine vision system to detect and count laying hens in battery cages. Animal, 14(12), 2628–2634. https://doi.org/10.1017/S1751731120001676
29. Geffen O, Yitzhaky Y, Barchilon N, Druyan S, Halachmi I. A machine vision system to detect and count laying hens in battery cages. Anim. 2020;14(12):2628-34. doi: 10.1017/S1751731120001676.
30. Wang J, Shen M, Liu L, Xu Y, Okinda C. Recognition and classification of broiler droppings based on deep convolutional neural network. J Sens. 2019;2019:1-10. doi: 10.1155/2019/3823515.
31. Wang K, Liew JH, Zou Y, Zhou D, Feng J. Panet: Few-shot image semantic segmentation with prototype alignment. Proc IEEE/CVF Int Conf Comput Vis. 2019;9197-206.
32. Han K, Wang Y, Tian Q, Guo J, Xu C, Xu C. Ghostnet: More features from cheap operations. Proc IEEE/CVF Conf Comput Vis Pattern Recognit. 2020;1580-9.
33. Tan M, Pang R, Le QV. Efficientdet: Scalable and efficient object recognition. Proc IEEE/CVF Conf Comput Vis Pattern Recognit. 2020;10781-90.
34. Gan Y, Delimitrou C. The architectural implications of cloud microservices. IEEE Comput Archit Lett. 2018;17(2):155-8. doi: 10.1109/LCA.2018.2839189.
35. Bochkovskiy A, Wang CY, Liao HYM. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.
36. Wang CY, Liao HYM, Wu YH, Chen PY, Hsieh JW, Yeh IH. CSPNet: A new backbone that can enhance learning capability of CNN. Proc IEEE/CVF Conf Comput Vis Pattern Recognit Workshops. 2020;390-1.
37. Han K, Wang Y, Xu C, Guo J, Xu C, Wu E, et al. GhostNets on heterogeneous devices via cheap operations. Int J Comput Vis. 2022;130(4):1050-69. doi: 10.1007/s11263-022-01575-y.
38. Kim M, Choi Y, Lee JN, Sa S, Cho HC. A deep learning-based approach for feeding behavior recognition of weanling pigs. J Anim Sci Technol. 2021;63(6):1453-63. doi: 10.5187/jast.2021.e127.
39. Wang Y, Chen T, Li B, Li Q. Automatic identification and analysis of multi-object cattle rumination based on computer vision. J Anim Sci Technol. 2023;65(3):519-34. doi: 10.5187/jast.2022.e87.

40. Lee J, Kang H. A study of duck detection using deep neural network based on RetinaNet model in smart farming. J Anim Sci Technol. 2024;66(4):846-58. doi: 10.5187/jast.2023.e76.
41. Zaidi SSA, Ansari MS, Aslam A, Kanwal N, Asghar M, Lee B. A survey of modern deep learning-based object detection models. Digit Signal Process. 2022;126:103514. doi: 10.1016/j.dsp.2022.103514.
42. Zhang Y, Cai W, Fan S, Song R, Jin J. Object detection based on YOLOv5 and GhostNet for orchard pests. Inf. 2022;13(11):548. doi: 10.3390/info13110548.

Tables and Figures

Table 1. Specifications of the hardware.

	Local	Naver Cloud Platform
OS	Ubuntu 18.04	
CPU	12th Intel I9-12900	Intel Xeon E5 v4 2660 (x 2)
Memory	32 GB	4 GB
GPU	RTX3090	None

OS, operating system; CPU, central processing unit; GPU, graphics processing unit.

Table 2. Results of the object detection models.

	EfficientDet	Faster R-CNN	SSD	YOLOv5X	YOLOv5N	Proposed model
mAP	80.65	54.76	75.3	87.5	**88.3**	87.7
IOU@50	89.5	88.8	88.6	91.4	**92.8**	90.5
IOU@75	18.4	23.2	26.9	**42.4**	41.8	32.2
Inference time in CPU (ms)	915	3390	225	1449	72.1	**50.9**
Inference time in GPU (ms)	28	39	12	15	3.6	**1.8**
Size (MB)	23.4	369.8	6.5	169.0	3.7	**1.8**

mAP, mean average precision; IOU, intersection over union.

Table 3. Ablation study results for YOLOv5 architectures.

Models	IOU@50	IOU@75	Inference time (ms) CPU	Inference time (ms) GPU	Size (MB)	Params (Millions)
YOLOv5N	**92.8**	41.8	72.1	3.6	3.7	1.76
Without large detector	91.9	**43.4**	**71.3**	**3.5**	**2.8**	**1.31**
Without medium detector	90.0	42.0	71.9	3.6	3.7	1.75
Without small detector	90.4	41.2	72.1	3.6	3.6	1.75
YOLOv5N+BiFPN	**93.0**	**43.8**	76.6	3.6	3.8	1.77
Without large detector	92.0	42.2	**71.9**	**3.4**	**2.9**	**1.32**
Without medium detector	90.8	41.6	75.1	3.5	3.7	1.77
Without small detector	90.5	40.5	72.9	3.5	3.6	1.77
YOLOv5N+GhostNet	**90.6**	**32.9**	63.8	2.0	2.2	0.93
Without large detector	89.8	32.0	**51.9**	**1.8**	**1.9**	**0.71**
Without medium detector	86.5	31.2	53.9	1.9	2.2	0.93
Without small detector	84.4	29.3	53.4	2.0	2.1	0.93
YOLOv5N+GhostNet+BiFPN	**91.2**	**32.8**	61.5	2.0	2.2	0.95
Without large detector	90.5	32.2	**50.9**	**1.8**	**1.8**	**0.73**
Without medium detector	89.5	31.5	54.2	1.9	2.2	0.95
Without small detector	89.0	29.7	54.7	1.9	2.1	0.95

Table 4. Broiler detection performance according to applying data augmentation techniques.

Models	IOU@50	IOU@75
Ours with data augmentation	90.5	32.2
Ours without data augmentation	89.7	31.0

Figure captions

Fig. 1. System architecture. (a) Multi-tier closed broiler house. (b) Images from top. (c) Images from side.

Fig. 2. Samples of the data augmentation.

Fig. 3. Architecture of object detection models. (a) Original architecture of YOLOv5. (b) Architecture of proposed model.

Fig. 4. Precision-recall (PR) curve of models. (a) IOU@50. (b) IOU@75.

Fig. 5. Inference time of models in each environment. (a) GPU-based local. (b) CPU based cloud.

Fig. 6. Visualization of object detection with mobile phone under cloud platform. (a) IP camera in the broiler coop. (b) Select dates. (c) Result display.