Lightweight Vision Models for Egg Fertility Detection

Earl John Flores
Don Mariano Marcos Memorial State University, Bacnotan, La Union, Philippines
earljohn.flores@dmmmsu.edu.ph (corresponding author)

Received: 21 October 2025 | Revised: 14 November 2025 and 2 December 2025 | Accepted: 3 December 2025

 DOI: https://doi.org/10.48084/etasr.15693

ABSTRACT

The Philippine chicken industry relies heavily on effective egg fertility detection to sustain its growth. Traditional manual candling techniques are prone to human error and inefficiency; thus, there is a need for automation. While deep learning models such as Convolutional Neural Networks (CNNs) and YOLO have shown great potential for real-time object detection, a noteworthy gap in prior research is their application to native Philippine chickens due to a lack of publicly available image datasets. The current study fills this gap by developing and optimizing a YOLOv11 model for egg fertility detection in native Philippine chickens. A grid search was implemented to tune key parameters, such as the learning rate, optimizer (SGD, Adam, RMSProp), and weight decay, to improve the detector. Overall, careful tuning increases the model's performance. The best configuration used SGD with a 0.01 learning rate and a 0.0001 weight decay. The tuned model outperformed the baseline model in terms of mAP50-95, precision, and recall, while having faster inference speed. This focused tuning makes YOLOv11 more reliable in detecting the fertility of chicken eggs.

Keywords-chicken egg fertility; computer vision; deep learning; YOLO

I. INTRODUCTION

The poultry industry is at the core of Philippine agriculture. It provides affordable protein to millions of Filipinos and contributes much to the nation's economy. The sector has been expanding steadily, with the single production of chicken eggs reaching 207,760 tons in the second quarter of 2025, representing a 4.8% increase over the same period the previous year. This sustained growth underscores the necessity for more effective and efficient methods of detecting egg fertility, as the Philippines is a strong regional leader [1]. Traditionally, fertility assessment depends on candling, where eggs are checked by shining a bright light through them. This method's shortcomings are: human fatigue, subjectivity, and inconsistent accuracy, often resulting in errors and decreased production [2, 3]. These challenges have pushed the industry to find more objective and automated solutions.

Advancements in artificial intelligence, particularly in deep learning, have made automatic fertility detection possible. CNNs are very effective in image data analysis and detecting minute features that may elude the human eye easily. When big enough training datasets are used, CNNs can achieve highly accurate fertility classification. Their results can also be enhanced by hyperspectral imaging or physiological information, since systems can analyze multiple dimensions of data simultaneously [4-6]. Object-detection models, especially YOLO, are of great value for real-time industrial use [7]. The YOLO models are optimized for speed and are hence ideal for the identification of egg fertility on fast-moving conveyor systems. They are also efficient enough to run on devices with limited computing power, with high accuracy and precision [8-10].

While both CNN and YOLO models have the potential for fast, non-destructive fertility checks, reducing manual candling, their full potential is realized when they are combined with other tools to support a completely automated egg sorting and classification system [11]. However, challenges persist; CNNs need large and varied training datasets, which may be difficult and time-consuming to collect. On the other hand, YOLO models sometimes face difficulty in predictions when images have messy or cluttered backgrounds. Both models also tend to require heavy hardware, making them far less accessible to small or backyard poultry producers [12, 13].

Another current knowledge gap is the absence of available datasets with images of eggs from native Philippine chicken breeds. This dataset should cover variations in egg size, shell color, and real-world scenarios common in backyard farms. Only a few studies have investigated ways to fine-tune the YOLO hyperparameters for improved performance. To help fill these gaps, the proposed system employs a YOLO-based egg fertility detection model enhanced with hyperparameter optimization through grid search. The present study determines an optimal combination of learning rate, weight decay, and optimizer settings, outperforming the default configuration of YOLOv11.

II. RELATED STUDIES

Effective hatcheries require a fast and accurate method of determining the developmental potential of an egg. This method saves time, reduces disease risks, and keeps the operation profitable. Manual checking of eggs, for instance, by candling is time-consuming, subjective, and not practical on a large scale. Newer systems involve the use of non-invasive automated tools that do not harm the eggs in any way.

Work on recognizing fertile eggs has shifted from hand-crafted features to deep learning; the models learn from the patterns themselves. Current systems deploy CNNs with attention and detection or segmentation tools [4]. Transfer learning is helpful when there are not enough labeled images for training. It has been demonstrated that this type of model can achieve very high accuracy [5, 14].

The CNNs have been trained directly on images acquired from cameras [15], as well as physiological signals extracted from heart rate waveforms and photoplethysmography visuals [16]. Several end-to-end optimization methods have achieved an accuracy of over 90% in different controlled experimental scenarios. Employing attention modules in conjunction with multiple feature extraction techniques at variable scales has proved instrumental in highlighting subtle vascular patterns and signs of early development. Specifically, the utilization of dense spatial attention and squeeze-and-excitation style modules is shown to reduce misclassification of embryos with weak or late-onset growth [17].

Transfer learning is one of the most popular approaches being used in this area. Fine-tuning traditional backbone architectures, such as VGG16, ResNet50, InceptionNet, and MobileNet, improves performance when the dataset size is small. Authors in [18] presented the results of four different backbone models on a dataset of 200 images and showed that InceptionNet outperformed the rest, with a test accuracy of 98%, a sensitivity of 100%, and a specificity of 96%. Mask R-CNN has also been used to perform egg detection, segmentation, and classification in incubator settings simultaneously [4]. These studies have shown very robust localization and segmentation performance at the commonly used intersection-over-union thresholds in day-three test imagery.

Comparative studies using single-shot detectors and YOLO variants imply that thermal imaging of quails is efficient, with YOLOv4 and YOLOv5 reaching very high mean Average Precision (mAP) metrics [10]. Furthermore, the SSD-MobileNet has a balance between speed and precision, ideal for low-energy or embedded hardware setups [12].

III. METHODOLOGY

The development and evaluation of the YOLO model for classifying native chicken eggs as fertile or infertile were conducted using Python programming in the JupyterLab environment. The system was developed and optimized using several libraries, including PyTorch for deep learning model implementation and the Ultralytics package for YOLO model configuration and training.

A. Egg Image Data Collection

The image acquisition for this study was conducted using 113 fertile eggs, obtained from native chicken breeds. The eggs were incubated in an automatic incubator calibrated to maintain a constant temperature of 37.8 °C and equipped with an automatic rotation mechanism to promote optimal embryonic development. Due to the limited capacity of a single incubator, the eggs were distributed between two separate units. Candling was performed on the seventh and tenth days of incubation to evaluate the fertility status of the eggs. An improvised imaging apparatus was developed to capture images with consistency and high reproducibility. To introduce variability, some eggs were imaged with the apparatus open, allowing different backgrounds, lighting, and environmental conditions to be recorded. The setup consisted of a Logitech C920 PRO HD webcam, a base-mounted egg candler providing uniform backlighting, and a stabilizer to hold the eggs securely in place. The webcam's high-definition lens, wide field of view, and automatic light correction features facilitated the acquisition of clear, high-resolution images.

Consistent with standard candling protocols, eggs were classified as fertile if branched blood vessels or the silhouette of an embryo were visible under illumination. Infertile eggs, characterized by unobstructed light passage and the absence of embryonic structures, were removed from the incubator. A sample of the egg images is illustrated in Figure 1. To ensure adequate representation of infertile samples, three trays of commercially sourced eggs were imaged using the same procedure and equipment, thus maintaining uniformity in lighting and orientation across all data.

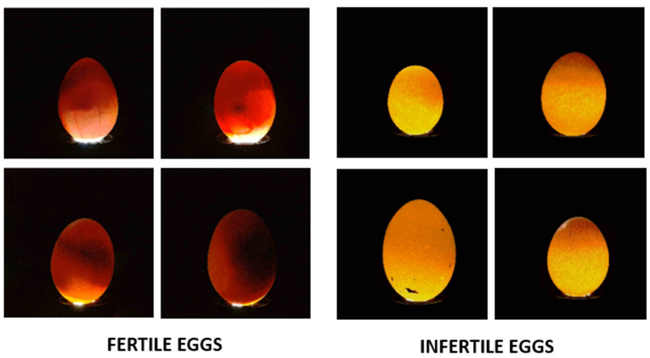

Fig. 1. Sample egg images.

This dataset consisted of 1,300 images in total, 75% of which were set aside for the training set and the remaining 25% for validation. There were 977 images of eggs in the training set, and an additional 47 background images in order to help the YOLO model learn non-object patterns and, hence, decrease false positives and improve generalization performance. Data splitting was done on a per-session basis. Each session represents a different group of eggs, and each session does not span both the training and validation sets. This means that the model was evaluated on images it did not see in training, which were taken under different conditions and on different days to avoid overfitting and support a more realistic evaluation of model performance. Table I presents the composition of the dataset used in the present study.

The dataset used in this study has been uploaded to Kaggle as a publicly accessible resource [19]. Before uploading, the dataset underwent cleaning and refinement. Several images were removed to protect the researcher's data privacy and to ensure a consistent visual standard. The dataset version available online reflects these revisions.

TABLE I. DATASET COMPOSITION

Image data	Train	Validation
Fertile egg	514	182
Infertile egg	416	141
Background	47	0
Total	977	323

B. Data Preparation and Annotation

Following image collection, all images and their corresponding labels were manually verified to ensure labeling accuracy and dataset quality. This labeling process was performed with the support of a poultry farmer who is experienced in manual egg candling. Each image was viewed and annotated using the DigitalSreeni Image Annotator [14], which enabled precise creation of bounding boxes around each egg and assignment of fertility labels. The fertility labels were developed from the candling, done on the seventh and tenth days of incubation, whereby all eggs showing visible embryonic development or vascular structures were labeled as fertile, and those without such features were labeled and removed from the incubator. Eggs having blood rings or showing evidence of embryonic death were also removed from the incubator and excluded from the dataset. Following the labeling by the researcher, the poultry expert independently reviewed all the annotations, confirming consistency in the labeling and resolving discrepancies by consensus. The final annotations were formatted in YAML format, according to the input specifications required by the YOLO models for full compatibility with the object detection framework that would later be used for model training.

C. Model Development and Training

The YOLO framework [20, 21] revolutionized object detection by formulating it as a single-stage regression problem, unlike earlier two-stage detection pipelines [15, 16]. The most recent version, YOLOv11, even presents significant enhancements that improve real-time performance with newer architectural components to increase accuracy, speed, and computational efficiency [22]. Capabilities have also expanded from classical object detection to include tasks such as posture estimation and instance segmentation. Structurally, YOLOv11 is composed of three main modules: a backbone responsible for feature extraction, a neck that enables multi-scale feature aggregation, and a prediction head that generates the final detection outputs [23].

The models were developed and trained on a high-performance laptop with an AMD Ryzen 7 processor, 16 GB of RAM, 512 GB of storage, and an NVIDIA GeForce RTX 3050 Ti GPU with 4 GB of VRAM to support the demands of accelerated deep learning computations. The software environment was controlled using Anaconda to ensure compatible and non-conflicting package configurations. Core libraries included PyTorch for model implementation and Ultralytics for configuration, training, and evaluation of the YOLO architecture. This work makes use of the YOLOv11-nano variant for egg fertility detection. First, this study presented a performance baseline established by training a baseline model with YOLOv11's default hyperparameters, summarized in Table II. This baseline is used as a reference to evaluate subsequent improvements. The hyperparameter optimization was performed using a grid search approach. The optimized parameters, which include the optimizer type, initial learning rate (lr0), and weight decay, are presented in Table III. Grid search conducted systematic explorations of the parameter space for those combinations of parameters that produced the highest validation accuracy. In general, grid search is one of the most well-known methods for boosting the results of deep learning models. For stable convergence and low variance across different runs, 27 training runs were performed, each for 100 epochs.

TABLE II. BASELINE PARAMETERS

Hyperparameter	Default value
Epochs	100
Batch size	16
Image size	640 × 640
Learning rate	0.01
Weight decay	0.0005
Optimizer	'auto'

TABLE III. HYPERPARAMETER SEARCH RANGE

Hyperparameter	Grid search values
Initial learning rate (lr0)	[0.01, 0.001, 0.0001]
Optimizer	Adam, RMSProp, SGD
Weight decay	[0.0001, 0.0005, 0.001]

D. Model Evaluation

Model performance was assessed using the designated validation set. Standard object detection metrics were employed, including mAP, precision, recall, F1-score, and total training time. For the tuned models, the optimal hyperparameter configuration was selected based on the highest mAP value. The optimal configuration was then compared with the baseline model to determine whether improvements in accuracy or computational efficiency were achieved.

IV. RESULTS AND DISCUSSION

A. Model Training Summary

The summarized results of tuning the YOLOv11 model, displayed in Table IV, show how changing the learning rate, optimizer, and weight decay affects detection performance. Among all possible combinations, the model maintained an excellent mAP50 around 0.995; that is, it detects most of the objects at a basic IoU threshold of 0.5. The more detailed metric, mAP50-95, reflects the precision in localization across a range of IoUs, which changes slightly across this tuning, underlining where the difference comes in. The maximum recorded mAP50-95 is 0.9803, recorded with the SGD optimizer, learning rate = 0.01, and weight decay = 0.0001. At the same time, this setting produces practically perfect precision, 0.9995; recall, 0.9998; and F1-score, 0.9997, making this configuration the best overall on this dataset.

TABLE IV. SUMMARY OF PERFORMANCE METRICS OF THE TUNED MODELS

Learning rate (lr0)	Optimizer	Weight decay	mAP50	mAP50-95	precision	recall	F1-Score	Training time (s)
0.01	Adam	0.0001	0.995	0.976495	0.998033	1	0.999015	1811.44
0.01	Adam	0.0005	0.995	0.974804	0.993245	0.999062	0.996145	1793.76
0.01	Adam	0.001	0.995	0.979402	0.998403	1	0.999201	1782.32
0.001	Adam	0.0001	0.995	0.972789	0.993085	0.992937	0.993011	1795.62
0.001	Adam	0.0005	0.995	0.975411	0.999386	1	0.999693	1799.22
0.001	Adam	0.001	0.995	0.975824	0.994543	0.99784	0.996189	1841.21
0.0001	Adam	0.0001	0.995	0.971544	0.995349	0.997499	0.996423	1819.3
0.0001	Adam	0.0005	0.995	0.974298	0.996936	0.999715	0.998324	1813.99
0.0001	Adam	0.001	0.995	0.972354	0.996892	0.9988	0.997845	1831.45
0.01	RMSProp	0.0001	0.500478	0.273394	0.040672	1	0.078164	1841.38
0.01	RMSProp	0.0005	0.466952	0.182272	0.008533	0.997253	0.01692	1776.03
0.01	RMSProp	0.001	0.480786	0.149943	0.603489	0.496454	0.544764	1794.78
0.001	RMSProp	0.0001	0.994652	0.963652	0.985768	0.994771	0.990249	1777.75
0.001	RMSProp	0.0005	0.994812	0.965751	0.99786	0.98679	0.992294	1780.56
0.001	RMSProp	0.001	0.995	0.960497	0.998629	1	0.999314	1781.9
0.0001	RMSProp	0.0001	0.995	0.966918	0.995789	0.997036	0.996412	1786.84
0.0001	RMSProp	0.0005	0.995	0.97285	0.969771	0.980836	0.975272	1797.31
0.0001	RMSProp	0.001	0.995	0.97156	0.987294	0.989134	0.988213	1812.64
0.01	SGD	0.0001	0.995	0.9803	0.999534	0.999875	0.999704	1763.6
0.01	SGD	0.0005	0.995	0.974868	0.990041	0.999259	0.994629	1758.41
0.01	SGD	0.001	0.994973	0.976764	0.993859	0.999523	0.996683	1771.41
0.001	SGD	0.0001	0.995	0.96964	0.998326	1	0.999162	1776.25
0.001	SGD	0.0005	0.995	0.967239	0.998637	1	0.999318	1780.34
0.001	SGD	0.001	0.995	0.970472	0.99918	1	0.99959	1771.53
0.0001	SGD	0.0001	0.994973	0.970029	0.99391	0.999025	0.996461	1775.48
0.0001	SGD	0.0005	0.995	0.966689	0.998682	1	0.99934	1788.91
0.0001	SGD	0.001	0.995	0.968145	0.996667	0.999244	0.997953	1793.48

Model convergence is influenced by the learning rate. In general, higher rates, such as 0.01, with proper regularization, produce better results. Smaller learning rates, such as 0.001 and 0.0001, result in stable but slightly lower mAP50-95 values, suggesting that YOLOv11 benefits from a more aggressive rate in terms of faster learning without overshooting. This trend is consistent with previous work indicating that the fine-tuning of the learning rate improves object detection performance. For example, authors in [24] showed how optimizing the learning rate range between 10^{-3} and 10^{-5} improved the precision and recall of YOLOv4 during vehicle detection tasks.

Performance highly depends on the optimizer choice. Both SGD and Adam produced strong results in this study, but SGD provides slightly better performance on fine-grained localization. This contrasts with [25], which reported Adam outperforming alternatives by about 5–10%. The/This difference likely stems from interactions between the optimizer and many other design and training choices rather than from the optimizer alone. RMSProp, in turn, yields poor performance for larger learning rates, dropping below 0.3 mAP50-95, indicating unstable training. Lowering the learning rate brought a rise for RMSProp, but still below the level of SGD and Adam.

Weight decay also impacts the results; though, somewhat less dramatically than the other two parameters. The values 0.0001 –0.001 make for a good balance between regularization and generalization. A bit too small a weight decay may lead to slight overfitting. Larger values may result in an unnecessary constraining of learning. The best results appear for the lowest tested decay, 0.0001, combined with a higher learning rate and SGD.

B. Comparison of the Baseline and Optimized Model

The performance gains of the tuned YOLOv11 over the baseline are summarized in Table V. The mAP50 increased from 0.9907 to 0.995, Fertile and Infertile class performance remained constant at 0.995, gains were much higher across IoU thresholds, mAP50–95 went from 0.9698 to 0.9777, and the Fertile class improved by 0.31%, with the Infertile class improving by 1.29%.

After tuning, precision, recall, and F1-score showed improvement, as demonstrated in the performance curves in Figure 2. The precision increased from 0.9614 to 0.9961, with most of this change contributed by the Infertile class. Recall increased from 0.9679 to 0.9995. F1-score increased from 0.9646 to 0.9978. These shifts indicate fewer false positives, fewer missed detections, and greater balance in both precision and recall. The tuned model is also quicker. Inference time reduced from 9.717 ms to 9.472 ms, and total training time reduced by about 12%.

Figure 3 shows the confusion matrices for the baseline and tuned models. The baseline model correctly classified all 141 infertile eggs but mislabeled 13 fertile eggs as Infertile. The tuned model classified both classes perfectly and had fewer errors related to the background than the baseline model. This demonstrates that parameter tuning significantly improved the performance of the model in distinguishing fertile versus infertile eggs and also its robustness.

TABLE V. METRIC PERFORMANCE COMPARISON FOR BASELINE AND TUNED YOLO11 MODEL

Metrics		Baseline	Tuned
mAP50	All	0.9907	0.9950
	Fertile	0.9950	0.9950
	Infertile	0.9863	0.9950
mAP50-95	All	0.9698	0.9777
	Fertile	0.9661	0.9691
	Infertile	0.9736	0.9862
Precision	All	0.9614	0.9961
	Fertile	1.000	1.000
	Infertile	0.9229	0.9921
Recall	All	0.9679	0.9995
	Fertile	0.9358	0.999
	Infertile	1.000	1.000
F1	All	0.9646	0.9978
	Fertile	0.9668	0.9995
	Infertile	0.9599	0.9961
Inference time per image (ms)		9.717	9.472
Training time (s)		2003.55	1763.6

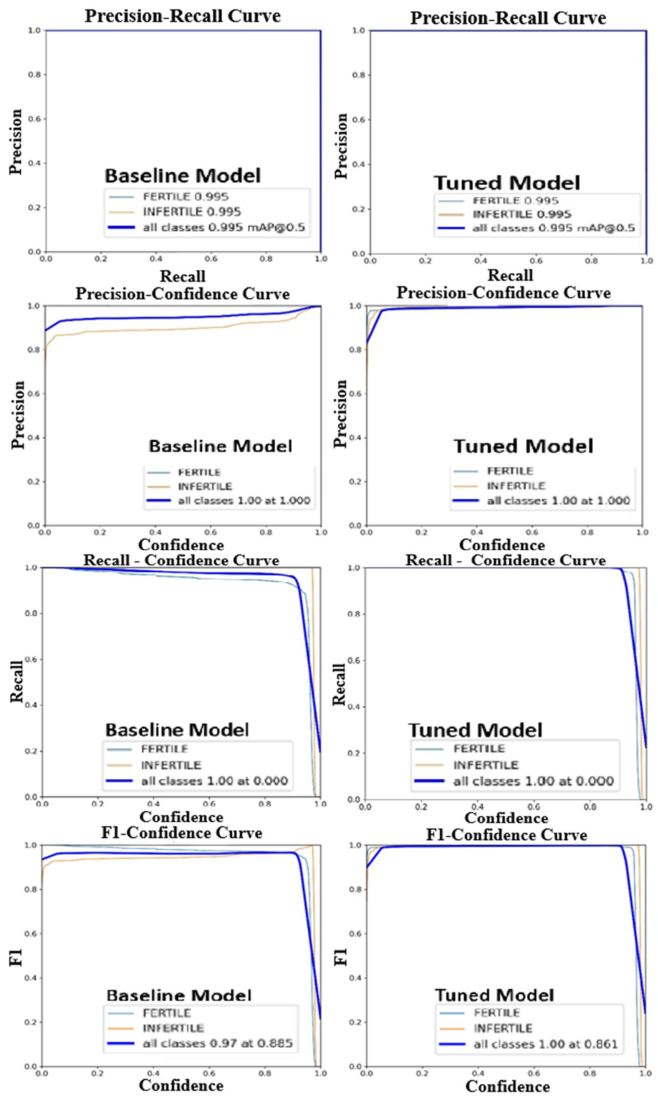

Fig. 2. Performance curves for baseline and tuned model.

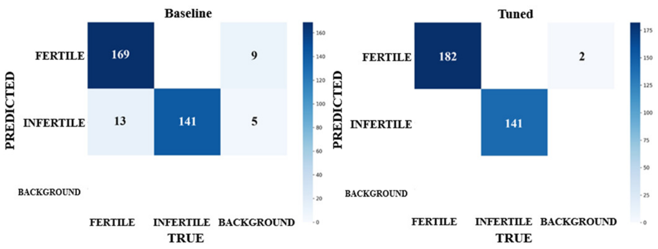

Fig. 3. Confusion matrix.

C. Comparison with Previous Studies

The results presented above established YOLOv11 as a superior approach in avian embryo classification. The outcomes align with, and in some cases surpass, those reported in comparable studies employing the egg fertility detection task outlined in Table VI.

TABLE VI. SUMMARY OF MODEL PERFORMANCE METRIC COMPARISON

Reference	Dataset	Model	Performance
Proposed model	1300 images (1253 chicken eggs, 47 background)	YOLOv11	97.88% mAP
		YOLOv11 Optimized	98.03% mAP
[9]	Duck egg	MobileOne-YOLOv7	97.79% mAP
[10]	Quail eggs (thermal images)	YOLOv5	99.5% mAP
		MobileNetV2	91.8% mAP
[12]	Chicken eggs	SSD MobileNet-V2 FPNLite	100% Accuracy
[26]	Duck eggs	AFF-YOLOX	97.55% mAP

V. CONCLUSION

This study presents a YOLOv11 model for the fertility of eggs from native Philippine chickens. This research uses a dataset of native chickens, with 1300 images (1253 chicken eggs, 47 background), and improves model performance using grid search optimization of the learning rate, optimizer, and weight decay. The best optimized parameters were the use of SGD, with a learning rate of 0.01, and a weight decay of 0.0001. This configuration produced the highest mAP50-95 of 0.9803 and near-perfect precision, recall, and F1-score. These gains, coupled with faster inference and reduced training time, demonstrate that hyperparameter tuning strengthens both the accuracy and efficiency. With the Philippine poultry sector requiring reliable and automated assessment of fertility, the proposed optimized YOLOv11 model presents a practical, easily scalable alternative to manual candling and facilitates more efficient hatchery operations.

ACKNOWLEDGMENT

No external funding was received for this study. The author expresses sincere appreciation to Maria Victoria Castillo for providing the equipment used in the development and testing of this study.

REFERENCES

[1] *Second Quarter 2025 Value of Production in Philippine Agriculture and Fisheries*, Quezon City 1101, Philippines: Philippine Statistics Authority, 2025.

[2] J. G. C. Rancapan, E. R. Arboleda, J. L. Dioses Jr, and R. M. Dellosa, "Egg Fertility Detection using Image Processing and Fuzzy Logic," *International Journal of Scientific & Technology Research*, vol. 8, no. 10, pp. 3228–3230, Nov. 2019.

[3] S. Saifullah, "Comparative Analysis of Long Short-Term Memory and Gated Recurrent Unit Models for Chicken Egg Fertility Classification using Deep Learning," in *International Electronic Conference on Applied Sciences*, online, Feb. 2025, Art. no. 7, https://doi.org/10.3390/engproc2025087007.

[4] K. K. Çevik, H. E. Koçer, and M. Boğa, "Deep Learning Based Egg Fertility Detection," *Veterinary Sciences*, vol. 9, no. 10, Oct. 2022, Art. no. 574, https://doi.org/10.3390/vetsci9100574.

[5] J. J. Maglasang *et al.*, "Duck Egg Embryonic Development Classification using Transfer Learning and CNN," *Smart Agricultural Technology*, vol. 11, Aug. 2025, Art. no. 100932, https://doi.org/10.1016/j.atech.2025.100932.

[6] L. Geng, H. Wang, Z. Xiao, F. Zhang, J. Wu, and Y. Liu, "Fully Convolutional Network with Gated Recurrent Unit for Hatching Egg Activity Classification," *IEEE Access*, vol. 7, pp. 92378–92387, 2019, https://doi.org/10.1109/ACCESS.2019.2925508.

[7] T. Saidani, "Deep Learning Approach: YOLOv5-based Custom Object Detection," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12158–12163, Dec. 2023, https://doi.org/10.48084/etasr.6397.

[8] D. Wu, D. Cui, M. Zhou, Y. Wang, J. Pan, and Y. Ying, "Using YOLOv5-DSE for Egg Counting in Conventional Scale Layer Farms," *IEEE Transactions on Industrial Informatics*, vol. 21, no. 1, pp. 405–414, Jan. 2025, https://doi.org/10.1109/TII.2024.3452270.

[9] Q. Li, Z. Shao, W. Zhou, Q. Su, and Q. Wang, "MobileOne-YOLO: Improving the YOLOv7 Network for the Detection of Unfertilized Duck Eggs and Early Duck Embryo Development - A Novel Approach," *Computers and Electronics in Agriculture*, vol. 214, Nov. 2023, Art. no. 108316, https://doi.org/10.1016/j.compag.2023.108316.

[10] V. M. Nakaguchi and T. Ahamed, "Development of an Early Embryo Detection Methodology for Quail Eggs Using a Thermal Micro Camera and the YOLO Deep Learning Algorithm," *Sensors*, vol. 22, no. 15, Aug. 2022, Art. no. 5820, https://doi.org/10.3390/s22155820.

[11] A. O. Adegbenjo, L. Liu, and M. O. Ngadi, "Non-Destructive Assessment of Chicken Egg Fertility," *Sensors*, vol. 20, no. 19, Sept. 2020, Art. no. 5546, https://doi.org/10.3390/s20195546.

[12] H. A. Muslim, H. Oktavianto, R. T. Widodo, E. Purwantini, and W. T. Sesulihatien, "Implementation of Chicken Eggs Fertility Detection Device Using SSD MobileNet-V2 FPNLite," in *International Electronics Symposium*, Denpasar, Indonesia, Aug. 2024, pp. 171–177, https://doi.org/10.1109/IES63037.2024.10665784.

[13] D. Musara, B. Sarema, D. Mashava, K. Chinguwo, and T. M. Muhla, "Design of an AI-based Egg Fertility Detection System for Incubators," *Open Access Research Journal of Science and Technology*, vol. 12, no. 1, pp. 001–009, Sept. 2024, https://doi.org/10.53022/oarjst.2024.12.1.0109.

[14] L. Huang, A. He, M. Zhai, Y. Wang, R. Bai, and X. Nie, "A Multi-Feature Fusion Based on Transfer Learning for Chicken Embryo Eggs Classification," *Symmetry*, vol. 11, no. 5, May 2019, Art. no. 606, https://doi.org/10.3390/sym11050606.

[15] V. K. Pooja, C. George, N. Mohamed, M. M. Shinu, and A. Raj V, "A Comprehensive Review of CNN Based Image Classification for Egg Fertility Detection," in *International Conference on Innovations in Mechanical Robotics Computing and Biomedical Engineering*, Kalady, Ernakulam, Kerala, India, Sept. 2025, pp. 44–52, https://doi.org/10.21467/proceedings.7.5.7.

[16] L. Geng, Q. Guo, Z. Xiao, J. Tong, and Y. Li, "Photoplethysmographic Waveform Detection for Determining Hatching Egg activity via Deep Neural Network," *Signal, Image and Video Processing*, vol. 16, no. 4, pp. 955–963, Jun. 2022, https://doi.org/10.1007/s11760-021-02040-y.

[17] L. Geng, Y. Xu, Z. Xiao, and J. Tong, "DPSA: Dense Pixelwise Spatial Attention Network for Hatching Egg Fertility Detection," *Journal of Electronic Imaging*, vol. 29, no. 02, Mar. 2020, Art. no. 023011, https://doi.org/10.1117/1.JEI.29.2.023011.

[18] S. Saifullah *et al.*, "Nondestructive Chicken Egg Fertility Detection using CNN-Transfer Learning Algorithms," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 9, no. 3, pp. 854–871, Sept. 2023, https://doi.org/10.26555/jiteki.v9i3.26722.

[19] E. J. Flores, "Chicken Egg Fertility." Kaggle, 2025, [Online]. Available: https://www.kaggle.com/datasets/ejfloresdmmmsu/chicken-egg-fertility.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.

[21] J. Du, "Understanding of Object Detection Based on CNN Family and YOLO," *Journal of Physics: Conference Series*, vol. 1004, Apr. 2018, Art. no. 012029, https://doi.org/10.1088/1742-6596/1004/1/012029.

[22] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO." Github, Jan. 2023, [Online]. Available: https://github.com/ultralytics/ultralytics.

[23] R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements." arXiv, Oct. 23, 2024, https://doi.org/10.48550/arXiv.2410.17725.

[24] G. D. Deepak and S. K. Bhat, "Optimizing YOLOv4 Hyperparameters for Enhanced Vehicle Detection in Intelligent Transportation Systems," *International Journal of Intelligent Transportation Systems Research*, Jul. 2025, https://doi.org/10.1007/s13177-025-00519-3.

[25] G. D. Deepak and S. K. Bhat, "Maximizing YOLOv2 Efficiency: A Study on Multiclass Detection of Indoor Objects," *Results in Engineering*, vol. 26, Jun. 2025, Art. no. 105405, https://doi.org/10.1016/j.rineng.2025.105405.

[26] Y. Liu, D. Xiao, J. Zhou, and S. Zhao, "AFF-YOLOX: An Improved Lightweight YOLOX Network to Detect Early Hatching Information of Duck Eggs," *Computers and Electronics in Agriculture*, vol. 210, Jul. 2023, Art. no. 107893, https://doi.org/10.1016/j.compag.2023.107893.