Journal of Applied Informatics and Computing (JAIC)
Vol.10, No.4, August 2026, pp. 3148~3155
e-ISSN: 2548-6861

Convolutional Neural Network-Based Approach For In-Ovo Embryo Classification

Khoironi [1]*, Ibram Maulana Akhsanul Qasasi [2]*, Ahmad Khairul Umam [3]*, Much Chafid [4]*, Ahmad Walid Hujairi [5]*, I Wayan Rangga Pinastawa [6], Musthofa Galih Pradana [7]****
* Informatics Engineering, Department of Informatics and Computer Engineering, Surabaya State Polytechnic of Electronics
** Informatics, Computer Science Faculty, University Pembangunan Nasional Veteran Jakarta
khoironi@pens.ac.id [1], ibram@it.pens.student.pens.ac.id [2], akumam@pens.ac.id [3], chafid@pens.ac.id [4], walid@pens.ac.id [5], rangga@upnvj.ac.id [6], musthofagalihpradana@upnvj.ac.id [7]

Article Info

Article history:

Received 2026-06-28
Revised 2026-08-02
Accepted 2026-08-07

Keyword:

Candling,
Convolutional Neural Network,
Chicken Embryo,
Non-Invasive Incubation.

ABSTRACT

The egg hatching process is a crucial stage in chicken breeding, as the quality of the resulting embryos will greatly determine the productivity and success of the cultivation. However, monitoring the internal condition of eggs during incubation is still limited and is generally done manually using the candling method. Therefore, a non-invasive approach is needed to support the process of monitoring embryo development. This study aims to develop a chicken embryo classification system based on in-ovo images using a Convolutional Neural Network (CNN). This system utilizes internal egg images taken with a camera integrated in the incubator, so that the condition of the eggs can be automatically classified into categories of fertile, infertile, and developmental failure. The research stages include collecting in-ovo image data, preprocessing in the form of removing the background using the rembg library, and resizing the images to 224 × 224 pixels. The dataset was then divided into training data and test data with a ratio of 90%:10%. The CNN model was built with several layers, namely convolution, max pooling, dropout, flattening, and fully connected, then trained for 28 epochs with a batch size of 32. The results showed that the developed CNN model was able to achieve an accuracy of 90.5% in the 24th epoch and experienced a decrease in the 26th to 28th epochs. Testing also used images from different incubators than those used from the dataset showed that the model could classify egg conditions well.

I. Introduction

Egg hatching is a crucial stage in the chicken breeding system because it directly determines the quality and quantity of production [1]. In modern livestock practices, this process is generally carried out using artificial incubators designed to maintain temperature, humidity, and egg rotation to ensure stable incubation conditions [2]. The use of incubators offers advantages in terms of efficiency and consistency, especially on a large production scale, thereby increasing productivity and reducing dependence on external environmental factors [3], [4].

However, one of the main challenges in the incubation process is the limited information regarding the internal condition of the eggs during the hatching period [5]. Farmers generally do not have adequate access to determine whether the incubated eggs are fertile containing embryos that have the potential to develop or infertile not experiencing fertilization [6]. In addition, difficulties also arise in identifying embryos that have died inside the egg, because this condition does not always show clear signs outside the shell. This limitation also includes a lack of information regarding the stages of embryo development, so the monitoring process still relies on manual inspection [7].

Manual inspection is usually carried out by opening the incubator and directly observing the eggs, either through candling or physical contact [8]. This practice has the potential to disrupt the stability of the incubation environment, such as changes in temperature and humidity, and increases the risk of shell damage or disturbances to the developing embryo [9]. The lack of accurate information regarding embryo development also makes it difficult for farmers to determine the appropriate hatching time. This condition can cause delays in handling newly hatched chicks, potentially increasing the risk of mortality or reducing the quality of DOC (Day Old Chick) [10].

To overcome these limitations, this study used an in-ovo imaging approach to non-invasively obtain internal information about eggs during the incubation process [11]. This approach allows monitoring of fertility, embryo development, and hatching indications without opening the incubator or making direct contact with the eggs, thus maintaining temperature and humidity stability [12]. Images obtained periodically through an integrated camera were then analyzed using a Convolutional Neural Network (CNN) due to its ability to extract and learn visual patterns related to changes in internal egg characteristics [13].

This research is a development of previous studies that focused on the field of poultry farming, specifically on the process of selecting DOD (Day Old Duck) after hatching [14]. Currently, the research is directed at the pre-hatching stage using chicken eggs as the main object. The selection of chicken eggs is based on a shorter incubation period compared to duck eggs, thus allowing pre-hatching research to be carried out more quickly [15], [16]. Thus, this research is expected to provide a real contribution to the development of image-based and artificial intelligence-based poultry embryo monitoring technology.

II. RESEARCH METHOD

This research was conducted through stages starting from the collection of in-ovo image datasets during the incubation period of chicken eggs, where image capture was carried out from day 1 to day 10 of the incubation period, during the incubation period, undeveloped or infertile eggs were still imaged as well as eggs that failed to develop or dead embryos. Then continued with the preprocessing process in the form of background removal and image size adjustment to 224 x 224. The dataset was then divided into training data and test data with a ratio of 90% training data and 10% testing data. Next, the design of the Convolutional Neural Network (CNN) model was built with several main components, namely the convolution layer, max pooling, dropout, flattening, and fully connected layer [17]. The designed model was then trained using the number of epochs 2 to 32 and a batch size of 32 before being tested on the test data. Performance evaluation was carried out using a confusion matrix performance report that includes precision, recall, f1-score, and support metrics, as well as confusion matrix analysis [18], [19].

The iterative process is carried out until the model reaches a predetermined level of accuracy or until the prediction error is minimal, allowing the system to be used optimally for in-ovo embryo classification. The research flow is shown in Figure 1.

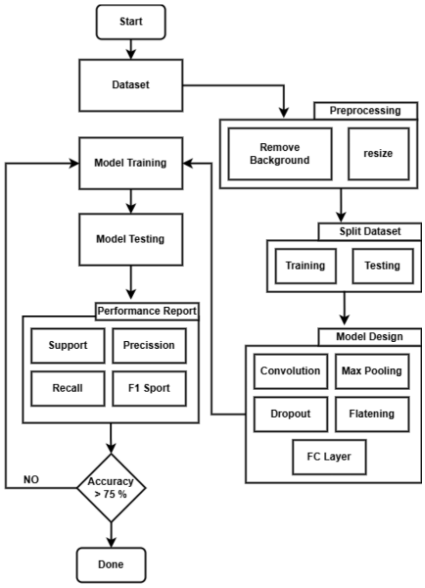

Figure 1. Research Flow

A. Dataset

The dataset used in this study was obtained from the process of observing chicken eggs from the first day of incubation to the 10th day using a homemade incubator equipped with a Logitech B525 webcam connected to a laptop with a resolution of 1280 x 720 pixels and a shooting distance of 35 cm above the incubator rack. Photos were taken periodically during the 10-day incubation period. then the obtained images were labeled in collaboration with farmers and resulted in 4,261 images of fertile eggs, 4,261 images of infertile eggs, and 4,260 images of undeveloped eggs or dead embryos with a total dataset of 12,782 images collected, which were then used as the main material in training and testing the Convolutional Neural Network (CNN) model. during the dataset collection process the researchers did not perform data augmentation, but took images using eggs directly with various egg positions.

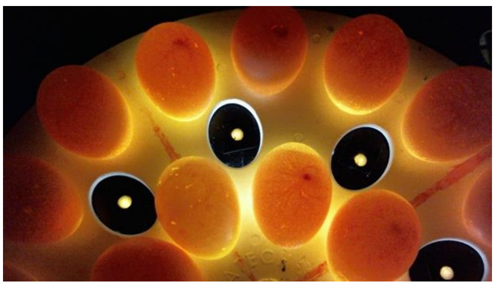

Figure 2. Candling Image

B. Preprocessing

A preprocessing stage was performed to prepare the images for the Convolutional Neural Network (CNN). First, the telescope image was cropped to obtain a uniform region of interest and resized to 224 × 224 pixels. Background removal was then performed using the Python-based rembg library to isolate the egg object from the surrounding background elements [20]. The use of rembg was intended to remove irrelevant background information, thereby reducing image noise and allowing the CNN to focus on the visual characteristics of the egg and embryo. Because the embryo and its internal structures are located entirely within the egg area, the background removal process only affected the external background around the egg and did not change the visual features of the embryo, such as blood vessels, embryo shape, or internal texture. Then, the processed image was resized using the OpenCV library to fit the CNN input size of 224 × 224 pixels.

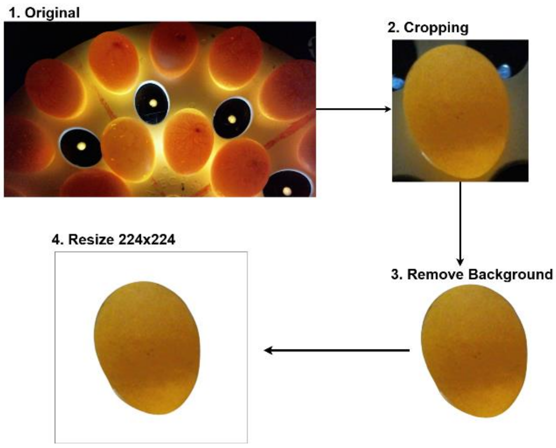

Figure 3. Preprocessing Flow

C. Split Dataset

After going through the preprocessing stage, the prepared image dataset was then divided into two parts: training data and testing data. Of the 12,782 available images, 11,800 were used as training data to build and adjust the parameters of the Convolutional Neural Network (CNN) model. Meanwhile, the remaining 982 images were allocated as test data to evaluate model performance [21]. This division was carried out to maintain a balance between model training and testing needs [22].

D. Model Design

The proposed CNN architecture was intentionally designed to be lightweight because the final system is intended for deployment on an ESP32-based embedded platform. Compared with deeper architectures, a lightweight CNN requires lower computational resources, memory usage, and inference time, making it more suitable for real-time embedded implementation. The Convolutional Neural Network (CNN) model used in this study is designed with multiple layers to optimally extract image features [23]. The initial stage begins with a rescaling layer that normalizes pixel values to match the input scale. Next, several convolutional layers are used with a gradually increasing number of filters (32, 64, up to 128) to detect increasingly complex visual patterns. Each convolutional layer is followed by batch normalization to stabilize data distribution and speed up the training process, and max pooling to reduce image dimensionality without losing important information. To prevent overfitting, dropout layers are applied at several stages after pooling [24]. After feature extraction, the image is converted into a one-dimensional vector through a flatten layer, then processed by a fully connected (dense) layer with 128 neurons. The final stage uses a dense output layer with two neurons representing the two main classes: fertile and infertile. The total parameters used in the model reach over 13 million, with the majority coming from the dense layer [25].

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 224, 224, 3)	0
conv2d (Conv2D)	(None, 224, 224, 32)	896
batch_normalization (BatchNormalization)	(None, 224, 224, 32)	128
conv2d_1 (Conv2D)	(None, 224, 224, 32)	9,248
max_pooling2d (MaxPooling2D)	(None, 112, 112, 32)	0
dropout (Dropout)	(None, 112, 112, 32)	0
conv2d_2 (Conv2D)	(None, 112, 112, 64)	18,496
batch_normalization_1 (BatchNormalization)	(None, 112, 112, 64)	256
conv2d_3 (Conv2D)	(None, 112, 112, 64)	36,928
max_pooling2d_1 (MaxPooling2D)	(None, 56, 56, 64)	0
dropout_1 (Dropout)	(None, 56, 56, 64)	0
conv2d_4 (Conv2D)	(None, 56, 56, 128)	73,856
batch_normalization_2 (BatchNormalization)	(None, 56, 56, 128)	512
conv2d_5 (Conv2D)	(None, 56, 56, 128)	147,584
max_pooling2d_2 (MaxPooling2D)	(None, 28, 28, 128)	0
dropout_2 (Dropout)	(None, 28, 28, 128)	0
flatten (Flatten)	(None, 100352)	0
dense (Dense)	(None, 128)	12,845,184
dropout_3 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

Figure 4. CNN Model Visualisation

E. Training

The model training process was conducted using even multiples of batch sizes, ranging from 2 to 32, to evaluate the effect of batch size on model performance. The validation loss graph shows that the training loss and validation loss curves both gradually and steadily decrease as the number of epochs increases. This condition indicates that the model's prediction error is decreasing for both training and validation data, and the distance between the two curves is relatively small, indicating stable generalization ability. At epoch 24, the model is in an optimal condition with a balance between decreasing training loss and stable validation loss. However, at epoch 26, a divergence between the training loss and validation loss curves begins to appear, with validation loss fluctuating even though training loss is still decreasing. This indicates the emergence of early symptoms of overfitting, where the model adapts more to the training data and generalization ability begins to decline [26]. This condition becomes more evident at epoch 28, when the fluctuation in validation loss becomes more significant and the distance between the two curves becomes larger [27]. Training loss continues to decrease, but validation loss tends to be unstable and even increases at some points. This pattern indicates that the model is overfitting, so that performance on the test data decreases even though accuracy on the training data remains high [28].

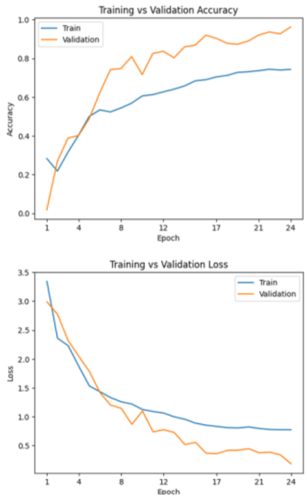

Figure 5. Visualization of Model Training Results.

F. *Testing*

The testing phase was conducted using a confusion matrix-based evaluation to assess model performance in more detail [29]. The evaluation was conducted to identify the level of prediction error on the test data to determine the model's ability to generalize patterns from the training data to new data. The test results showed that the model demonstrated the lowest level of prediction error in epoch 24 compared to other epochs. Conversely, in subsequent epochs, prediction errors began to increase, indicating the emergence of overfitting symptoms [30].

III. RESULTS AND DISCUSSION

A. *Epoch Analysis*

Model training experiments were conducted with even epoch variations starting from epoch 2 to find the optimal point that produces high accuracy while maintaining generalization ability. Based on the test results, it can be seen that in the 24th epoch the model achieved the best performance with an accuracy of 96.16%, a macro average F1-score of 0.87, and a weighted average F1-score of 0.95. In the 26th epoch, accuracy decreased to 94.55% with a macro average F1-score of 0.85 and a weighted average F1-score of 0.94. Although performance was still relatively high, a divergence between the training loss and validation loss curves began to appear, indicating early signs of overfitting. The decline in performance became more apparent in the 28th epoch, where accuracy dropped significantly to 87.68%, with a macro average F1-score of 0.67 and a weighted average F1-score of 0.84. Fluctuations in validation loss increased, while training loss continued to decrease. This pattern indicates that the model is too focused on the training data that it fails to recognize patterns in the validation data, which is a hallmark of overfitting [31].

TABLE I
EPOCH EXPERIMENT

Epoch	Accuracy	Macro AVG (F1)	Weighted AVG (F1)
24	**96.16%**	**0.87**	**0.95**
26	94.55%	0.85	0.94
28	87.68%	0.67	0.84

B. *Model Evaluation*

Analysis of the test results using a confusion matrix showed that most of the predicted values were concentrated on the main diagonal, indicating the model's ability to classify data well across almost all classes. In the 24th epoch, the model generated 952 correct predictions out of a total of 990 test data sets, with relatively small errors. Errors generally occurred in classes with similar visual characteristics, such as between adjacent observation days or similar embryonic conditions.

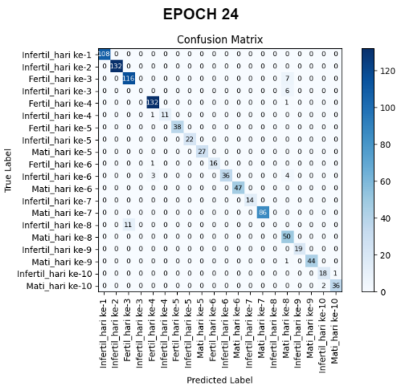

Figure 6. Confusion Matrix Evaluation at Epoch 24

In the 26 epoch, the number of prediction errors began to increase. The model generated 936 correct predictions from 990 test data sets, with 54 errors. Although most values remained on the main diagonal, the off-diagonal distribution of errors indicated a decline in the model's consistency in recognizing some classes [32]. Prediction errors were still dominated by classes with similar visual characteristics, but this trend indicated that the model was beginning to show early signs of overfitting [33].

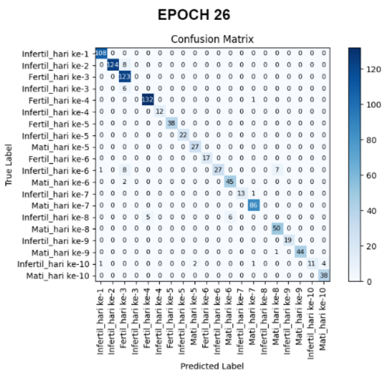

Figure 7. Confusion Matrix Evaluation at Epoch 26

The performance decline became more pronounced in the 28th epoch. The model generated only 868 correct predictions from 990 test data sets, with errors increasing to 122. The spread of values outside the main diagonal widened, indicating a decline in the model's ability to distinguish patterns between classes. Classification errors occurred not only in classes with similar characteristics but also began to appear in classes with more distinct conditions, such as fertile, infertile, and dead.

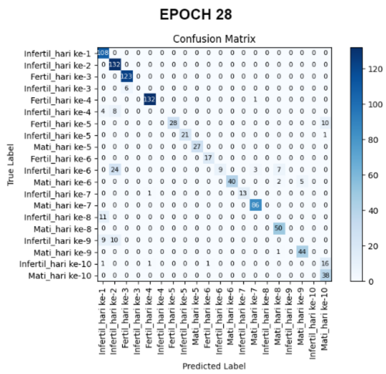

Figure 8. Confusion Matrix Evaluation at Epoch 28

This confusion matrix evaluation was conducted using images obtained from the candling process during the same incubation period, ensuring that the test data accurately represented the actual condition of the observed eggs. Overall, the confusion matrix evaluation results confirmed the finding that epoch 24 was the optimal point for the CNN model in in-ovo embryo classification.

C. Symptoms of Overfitting

Symptoms of overfitting begin to appear when the model is trained past the optimal epoch. At epoch 26, although the training loss continues to show a downward trend, the validation loss begins to fluctuate, and the gap between the two curves widens. This indicates that the model is increasingly adapting to the training data, but its generalization ability to the test data is starting to decline [34].

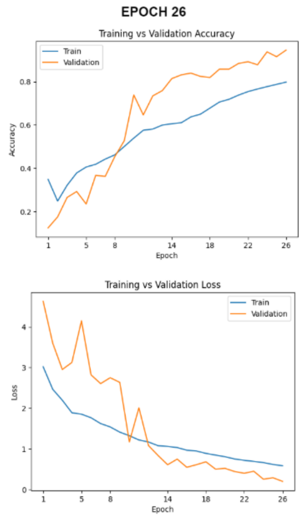

Figure 9. Visualization of Model Training Epoch 26

This phenomenon became even more apparent in epoch 28, where the validation loss was no longer stable and tended to increase, while the training loss continued to decrease. The increasing discrepancy between the training loss and validation loss curves indicates that the model was focusing too much on patterns in the training data, failing to recognize variations in the validation data [35]. Evaluation using a confusion matrix also reinforced this finding, with a more significant increase in the number of prediction errors compared to previous epochs.

Figure 10. Visualization of Model Training Epoch 28

Overall, this pattern indicates that the model is overfitting at higher epochs. This confirms that epoch 24 is the optimal point, as under these conditions, the model is still able to maintain a balance between high accuracy and low prediction error.

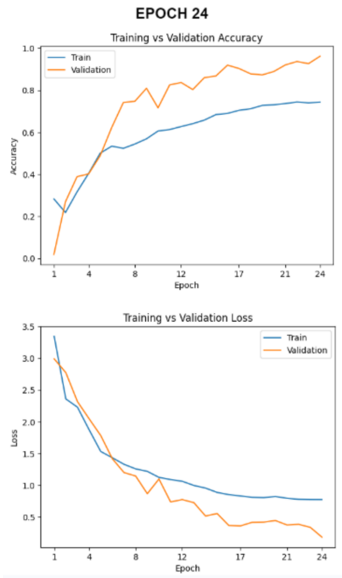

Figure 11. Visualization of Model Training Epoch 24

D. Comparison of Image Test Results

In addition to using the test data from the dataset split, additional testing was conducted using images from other incubators, as shown in Figure 13, which were not included in the training or test data. The purpose of this testing was to evaluate the model's generalization ability to new data with similar visual characteristics but originating from different sources.

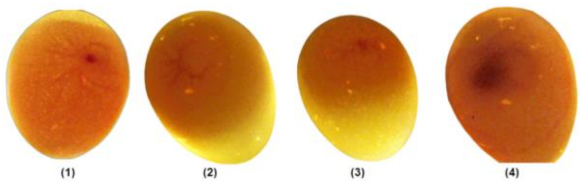

Figure 12. Candling Images from the same dataset

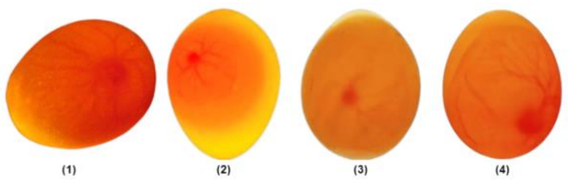

Figure 13. Candling image from another incubator

Based on visual comparison, the images from other incubators in Figure 13 show similar characteristics to the images from the personal incubator. For example, image (1) from another incubator has a match with image (1) from the personal incubator in Figure 12 which belongs to the fertile class on day 4, characterized by a blood vessel pattern that is starting to spread but is not too thick and the embryo size is still relatively small. Furthermore, images (2) and (3) from other incubators in Figure 13 show a similarity with images (2) and (3) from the personal incubator in Figure 12 which belongs to the fertile class on day 3, where the blood vessel network is just starting to develop and the embryo size is still very small. Meanwhile, image (4) from another incubator has a match with image (4) from the personal incubator which belongs to the dead class on day 6, characterized by the presence of a blood ring that is not too thick.

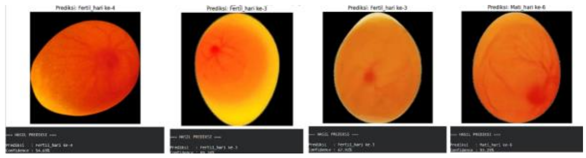
Figure 14. Classification Test Results using images from different incubators

Based on the model classification results in Figure 14, image (1) from another incubator is predicted as fertile on day 4 with a confidence of 54.69%. Images (2) and (3) are predicted as fertile on day 3 with a confidence of 89.38% and 67.92%, respectively. Meanwhile, image (4) is predicted as dead on day 6 with a confidence of 93.29%.

TABLE II
EVALUATION USING EXTERNAL IMAGES FROM A DIFFERENT INCUBATOR

Class	Number of Samples	Correct	Incorrect
Fertile	36	34	2
Infertile	32	32	0
Developmental Failure	39	35	4
Total	**107**	**101**	**6**

Testing with external incubator images was conducted using 107 candling images that had not been previously studied by the model. The external dataset consisted of 36 fertile images, 32 infertile images, and 39 images of embryos with failed development. The model was able to correctly classify 101 of the 107 images. The infertile embryo class achieved the highest performance with 100% accuracy, while the fertile and failed development classes still had prediction errors. Based on manual observations, these failures were due to embryos in the early stages of failed development often showing a blood ring pattern that closely resembled the blood vessel structure observed in normally developing fertile embryos at the beginning.

IV. CONCLUSION

This research successfully developed a Convolutional Neural Network (CNN) model for in-ovo embryo classification based on candling images. The dataset, obtained from a private incubator during 10 days of incubation, underwent preprocessing to produce uniform images measuring 224x224 pixels. The designed CNN model demonstrated the best performance at the 24th epoch, with an accuracy of 96.16%, a macro average F1-score of 0.87, and a weighted average F1-score of 0.95. Evaluation using a confusion matrix reinforced these findings, with the lowest number of prediction errors occurring at the 24th epoch.

At the 26th and 28th epochs, model performance began to decline with increasing prediction errors and fluctuations in validation loss, indicating the emergence of overfitting. Additional testing using images from other incubators demonstrated that the model remained capable of classifying images effectively according to the visual characteristics of each class, confirming its stable generalization ability to new data.

Overall, this study demonstrates that the developed CNN model is effective in classifying in-ovo embryos, with the 24th epoch representing the optimal training point. These findings contribute to the development of automated monitoring systems for the incubation process and provide opportunities for broader applications of artificial intelligence in poultry farming. However, the performance of the proposed system may still be affected by variations in image acquisition conditions, including lighting intensity during egg candling, camera quality, and egg position. Therefore, future studies should investigate the robustness of the proposed model under more diverse imaging conditions and compare its performance with other lightweight CNN architectures suitable for implementation on embedded platforms.

ACKNOWLEDGMENT

The author would like to thank the Directorate of Research, Technology, and Community Service (DRTPM), Directorate General of Higher Education, Research, and Technology, Ministry of Higher Education, Science, and Technology, for funding this research through the Novice Lecturer Research Program (PDP) for Fiscal Year 2026 with Contract Number [173/C3/DT/DT.05.00/PL-BARU/2026].

REFERENCES

[1] G. Irwan Ragut and I. Wayan Sujana, "Jurnal Mesin Material Manufaktur dan Energi Karateristik Perpindahan Panas pada Penetas Telur dengan Menggunakan Inkubator Kapasitas 30 Butir Telur," 2026.

[2] M. Hidayat, "JIP (Jurnal Informatika Polinema) Sistem Inkubator Penetasan Telur Cerdas Berbasis Iot Menggunakan Flutter Mutiplatform," 2026.

[3] M. Ghummah and M. Amrullah, "Implementasi Internet of Things (IoT) pada Sistem Penetasan Telur Otomatis untuk Meningkatkan Tingkat Keberhasilan Penetasan di Pamekasan," *Karapan Network Journal*, vol. 02, no. 02, 2026, doi: 10.20473/KNJ.X.X.pp-pp.

[4] W. Bilyaro, D. Lestari, and dan Ayu Sri Endayani, "Identifikasi Kualitas Internal Telur Dan Faktor Penurunan Kualitas Selama Penyimpanan Identification Of The Internal Quality Of Eggs And Factors Of Decreasing Quality During Storage," 2021.

[5] I. R. Juliarti, M. Latif, S. Wahyuni, A. I. Sudianto, Ach. Dafid, and H. Budiarto, "Deteksi Fertilitas Telur Ayam Menggunakan Metode YOLO untuk Sistem Sortir Otomatis," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 11, no. 2, pp. 305–314, Apr. 2026, doi: 10.30591/jpit.v11i2.10166.

[6] F. Noviani, I. Salamah, and L. Lindawati, "Rancang Bangun Sistem Pemisah Telur Fertil Dan In-Fertil Otomatis Dengan Metode Convolutional Neural Network (CNN)," *JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)*, vol. 9, no. 3, pp. 1547–1555, Aug. 2024, doi: 10.29100/jipi.v9i3.6101.

[7] Rafly Hadi Pangestu, S. Ch, and Supriono, "Rancang Bangun Purwarupa Alat Penyortir Kualitas Telur Ayam Berbasis Arduino Uno," *Bulletin of Computer Science Research*, vol. 5, no. 1, pp. 52–60, Dec. 2024, doi: 10.47065/bulletincsr.v5i1.428.

[8] N. F. Arini, A. Ubaidillah, K. A. Wibisono, and M. Ulum, "Identifikasi embrio dalam telur berbasis image processing," *Jurnal Teknik Elektro dan Komputasi (ELKOM)*, vol. 2, no. 1, pp. 11–19, Mar. 2020, doi: 10.32528/elkom.v2i1.3137.

[9] A. Diantoro and I. B. Santoso, "Eggs Fertilities Detection System on the Image of Kampung Chicken Egg Using Naive Bayes Classifier Algorithm," *MATICS*, vol. 9, no. 2, p. 53, Dec. 2017, doi: 10.18860/mat.v9i2.4198.

[10] Vincent, H. H. S. Pasaribu, W. Audrey, Jefanya Alexander Meidi Bangun, and Deryck Ethan Hong, "Detection of Chicken Egg Quality with Digital Image using EfficientNet-B7," *Journal Of Informatics And Telecommunication Engineering*, vol. 9, no. 1, pp. 176–188, Jul. 2025, doi: 10.31289/jite.v9i1.15233.

[11] B. Botta, S. S. R. Gattam, and A. K. Datta, "Eggshell crack detection using deep convolutional neural networks," *J. Food Eng.*, vol. 315, p. 110798, Feb. 2022, doi: 10.1016/J.JFOODENG.2021.110798.

[12] B. Guanjun, J. Mimi, X. Yi, C. Shibo, and Y. Qinghua, "Cracked egg recognition based on machine vision," *Comput. Electron. Agric.*, vol. 158, pp. 159–166, Mar. 2019, doi: 10.1016/J.COMPAG.2019.01.005.

[13] L. Mohammadpour, T. C. Ling, C. S. Liew, and A. Aryanfar, "A Survey of CNN-Based Network Intrusion Detection," Aug. 01, 2022, *MDPI*. doi: 10.3390/app12168162.

[14] Khoironi and I. W. R. Pinastawa, "Classification of Hybrid and Peking Duck DOD Varieties Based on Feather Images Using CNN," *Teknika*, vol. 14, no. 2, pp. 290–296, Jul. 2025, doi: 10.34148/teknika.v14i2.1270.

[15] K. A. Prasetia, Y. Sumaryana, and A. Sudiarjo, "Sistem Monitoring Temperatur Pada Inkubator Penetas Telur Bebek Menggunakan Modul Nodemcu 8266 Yang Terintegrasi Dengan Aplikasi BLYNK," *Jurnal Informatika dan Teknik Elektro Terapan*, vol. 12, no. 2, Apr. 2024, doi: 10.23960/jitet.v12i2.4146.

[16] K. Yao *et al.*, "Non-destructive detection of egg qualities based on hyperspectral imaging," *J. Food Eng.*, vol. 325, Jul. 2022, doi: 10.1016/j.jfoodeng.2022.111024.

[17] 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, Sep. 2023, doi: 10.26555/jiteki.v9i3.26722.

[18] E. Sehirli and K. Arslan, "An application for the classification of egg quality and haugh unit based on characteristic egg features using machine learning models," *Expert Syst. Appl.*, vol. 205, p. 117692, Nov. 2022, doi: 10.1016/J.ESWA.2022.117692.

[19] W. Zhang, L. Pan, S. Tu, G. Zhan, and K. Tu, "Non-destructive internal quality assessment of eggs using a synthesis of hyperspectral imaging and multivariate analysis," *J. Food Eng.*, vol. 157, pp. 41–48, Jul. 2015, doi: 10.1016/J.JFOODENG.2015.02.013.

[20] 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, Sep. 2023, doi: 10.26555/jiteki.v9i3.26722.

[21] H. Babaei, M. Zamani, and S. Mohammadi, "The impact of data splitting methods on machine learning models: A case study for predicting concrete workability," *Machine Learning for Computational Science and Engineering*, vol. 1, no. 1, p. 21, 2025, doi: 10.1007/s44379-025-00021-3.

[22] J. J. Salazar, L. Garland, J. Ochoa, and M. J. Pyrcz, "Fair train-test split in machine learning: Mitigating spatial autocorrelation for

improved prediction accuracy," *J. Pet. Sci. Eng.*, vol. 209, p. 109885, Feb. 2022, doi: 10.1016/J.PETROL.2021.109885.

[23] D. A. Varughese and S. Sridevi, "Optimization Strategies and Algorithms for Accelerating CNN on FPGA: A Comprehensive Review," *Archives of Computational Methods in Engineering*, vol. 33, no. 1, pp. 1205–1226, 2026, doi: 10.1007/s11831-025-10348-y.

[24] Q. T. Lam, M. H. N. Le, F. Y. Fan, N. Q. K. Le, and I. T. Lee, "Deep Learning-Based Dental Caries Diagnosis: A Modality-Stratified Systematic Review and Meta-Analysis of Faster R-CNN and Mask R-CNN," Mar. 01, 2026, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/diagnostics16050731.

[25] H. Yagnik, R. K. Gupta, and A. Pandya, "Deep Learning for Air Pollution Prediction: A Systematic Review of CNN, LSTM, GNN, and Transformer Architectures," *Archives of Computational Methods in Engineering*, 2026, doi: 10.1007/s11831-026-10639-y.

[26] A. P. Sheppert, "Techniques for mitigating overfitting in machine learning: a comprehensive review, taxonomy, and practical guide," 2026, *Frontiers Media SA*. doi: 10.3389/frai.2026.1794271.

[27] R. Diallo, C. Edalo, and O. O. Awe, "Machine Learning Evaluation of Imbalanced Health Data: A Comparative Analysis of Balanced Accuracy, MCC, and F1 Score," in *Practical Statistical Learning and Data Science Methods: Case Studies from LISA 2020 Global Network, USA*, O. O. Awe and E. A. Vance, Eds., Cham: Springer Nature Switzerland, 2025, pp. 283–312. doi: 10.1007/978-3-031-72215-8_12.

[28] K. Takahashi, K. Yamamoto, A. Kuchiba, and T. Koyama, "Confidence interval for micro-averaged F1 and macro-averaged F1 scores," *Applied Intelligence*, vol. 52, no. 5, pp. 4961–4972, 2022, doi: 10.1007/s10489-021-02635-5.

[29] K. Khoironi, J. Prasetyo, and A. W. Hujairi, "Penerapan Metode Dempster Shafer Pada Sistem Pakar Prediksi Jenis Kelamin," *Jurnal Teknik Elektro dan Komputasi (ELKOM)*, vol. 6, no. 1, pp. 77–85, May 2024, doi: 10.32528/elkom.v6i1.22404.

[30] M. Pan *et al.*, "A Hybrid CNN-Transformer Model for Soil Texture Estimation from Microscopic Images," *Agronomy*, vol. 16, no. 3, Feb. 2026, doi: 10.3390/agronomy16030333.

[31] S. Matharaarachchi, M. Turgeon, M. Domaratzki, and S. Muthukumarana, "Sequential Bayesian estimation of the F1 score using the Dirichlet-multinomial model," *Int. J. Data Sci. Anal.*, vol. 21, no. 1, p. 44, 2025, doi: 10.1007/s41060-025-00885-x.

[32] S. Rajvanshi, G. Kaur, A. Dhatwalia, Arunima, A. Singla, and A. Bhasin, "Research on Problems and Solutions of Overfitting in Machine Learning," in *Advances in Artificial-Business Analytics and Quantum Machine Learning*, K. C. Santosh, S. K. Sood, H. M. Pandey, and C. Virmani, Eds., Singapore: Springer Nature Singapore, 2024, pp. 637–651.

[33] J. Opitz, "From Bias and Prevalence to Macro F1, Kappa, and MCC: A structured overview of metrics for multi-class evaluation," 2022.

[34] N. Risse, "Detecting Overfitting of Machine Learning Techniques for Automatic Vulnerability Detection," in *ESEC/FSE 2023 - Proceedings of the 31st ACM Joint Meeting European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, Association for Computing Machinery, Inc, Nov. 2023, pp. 2189–2191. doi: 10.1145/3611643.3617845.

[35] C. Aliferis and G. Simon, "Overfitting, Underfitting and General Model Overconfidence and Under-Performance Pitfalls and Best Practices in Machine Learning and AI," in *Artificial Intelligence and Machine Learning in Health Care and Medical Sciences: Best Practices and Pitfalls*, G. J. Simon and C. Aliferis, Eds., Cham: Springer International Publishing, 2024, pp. 477–524. doi: 10.1007/978-3-031-39355-6_10.