Journal of Applied Data Sciences
Vol. 5, No. 3, September 2024, pp. 1052-1068

Improved Hybrid Machine and Deep Learning Model for Optimization of Smart Egg Incubator

Anita Febriani[1,*], Refni Wahyuni[2], Mardeni[3], Yuda Irawan[4], Rika Melyanti[5]

[1, 2, 3, 4]*Computer Science, Universitas Hang Tuah Pekanbaru, Pekanbaru, Indonesia*

[5]*Information System, Universitas Hang Tuah Pekanbaru, Pekanbaru, Indonesia*

(Received: June 2, 2024; Revised: June 24, 2024; Accepted: July 15, 2024; Available online: July 27, 2024)

Abstract

This research develops a Smart Egg Incubator that integrates IoT technology, fuzzy logic, and the YOLOv9-S Deep Learning model to enhance the efficiency and accuracy of hatching chicken eggs. The system automatically regulates temperature and humidity, maintaining temperature between 34.3°C and 39.5°C and humidity between 57% and 68% with a fuzzy logic success rate of 90%. The YOLOv9-S model enables real-time chick detection and classification with mAP50 of 93.7% and mAP50:95 of 71.3%. Efficiency improvements are measured through the success rate of fuzzy logic and improved detection and classification accuracy. This research also uses CNN for high-accuracy object classification, with model optimization performed using SGD to accelerate convergence and improve accuracy. The results indicate significant potential in improving the egg hatching process. The high accuracy and robustness of the YOLOv9-S model enhance real-time monitoring and decision-making in hatcheries, leading to higher hatching success rates, reduced chick mortality, and increased operational efficiency. Future designs can leverage these technologies to create more intelligent, automated systems requiring minimal human intervention, enhancing productivity and scalability. Additionally, IoT and deep learning integration can extend to other poultry farming areas, such as broiler production and disease monitoring, providing a comprehensive approach to farm management. Future research could focus on integrating the YOLOv10 model for even higher accuracy and efficiency, exploring diverse data augmentation techniques, optimizing fuzzy logic algorithms, and integrating additional sensors like CO2 and advanced humidity sensors to improve environmental regulation. These advancements would benefit not only smart incubator applications but also broader poultry farming areas.

Keywords: Smart Egg Incubator, IoT, Fuzzy Logic, YOLOv9-S, CNN, SGD

1. Introduction

Hatching machines or egg incubators have an important role in the world of livestock and poultry breeding, both for food production needs and for poultry enthusiasts [1], [2]. They are becoming increasingly popular due to their advantages over natural hatching methods that allow for significant improvements in the efficiency and effectiveness of the hatching process [3]. Egg Incubator have evolved from manual tools to semi-automatic machines and then to fully automatic versions [4], [5]. These developments not only speed up the breeding process of poultry, but also make it more effective and efficient [4], [6], [7].

The application of egg incubator in various scales of livestock enterprises shows an important shift from traditional hatching methods [8]. By integrating into Internet of Things (IoT) technology, modern hatcheries offer advantages for remote hatchery monitoring and management [9], [10]. IoT integration allows farmers to accurately regulate and monitor the internal conditions of the hatchery such as temperature and humidity through an app on a smartphone or other smart device [11], [12]. The benefits of using a hatching machine are its ability to increase the percentage of hatching success up to 80-90%, freeing poultry mothers from incubating duties so that they can immediately return to reproduction [13]. Egg Incubator overcome many other limitations of natural hatching such as the inability of the mother to incubate all the eggs or the risk of chicks dying due to trampling [14].

Addressing the previously identified problems, the proposed problem-solving approach for the development of the Smart Egg Incubator is the integration of IoT technology with Deep Learning and machine learning models. The implementation of an adaptive control system using fuzzy logic automatically adjusts the temperature and humidity in

***Corresponding author: Anita Febriani (nitasuheri@gmail.com)**
DOI: https://doi.org/10.47738/jads.v5i3.304

the incubator based on the collected sensor data to ensure optimal hatching environment conditions. The development and integration of Deep Learning YOLOv9 and CNN models aims to improve accuracy in real-time detection and monitoring of the hatching process, with the application of SGD as an optimizer for accuracy improvement. In addition, data augmentation techniques will be used to enrich the dataset, improve the model's ability to recognize more diverse patterns and overcome variations in image data.

The combination of IoT with machine learning and Deep Learning techniques will be developed to create a Smart Egg Incubator that is not only automatic in regulating the hatching environment, but also intelligent in monitoring and optimizing the hatching process. This approach is expected to increase the efficiency and effectiveness of egg hatching and provide solutions to the limitations that exist in previous egg hatching systems [15]. The data augmentation techniques used include rotation, rescaling, translation, and lighting variation to ensure the model can cope with various image conditions that may occur during the hatching process [16], [17]. The integration of these technologies not only ensures optimal conditions for the eggs but also enables detection and classification of hatched chicks with high accuracy, increasing the overall success of egg incubator.

Previous research used ESP32 DHT22 Sensor in building an egg incubator using IoT [18]. Subsequent research developed an automatic egg hatcher but faced issues when the machine failed without electricity [19]. Other researchers focused on optimal temperature using specific light bulbs as heating sources [20], rotating egg trays with DC motors without considering monitoring [21], and implementing solar panels without incubator status monitoring [22]. Further studies equipped egg incubators with LoRa networks for temperature management but lacked automatic egg rotation [23]. Research has also developed IoT-based egg incubators with subsystems such as embedded systems, web applications, and Telegram [12]. Other studies used DHT11 sensors and manual cameras for monitoring without Deep Learning integration [24].

To overcome the limitations of traditional hatching methods, such as the inability of the mother to incubate all the eggs or the risk of chicks dying due to trampling, the Smart Egg Incubator leverages IoT and Deep Learning technologies. IoT integration allows for precise control and monitoring of the hatching environment through remote access, while Deep Learning models enhance the accuracy of detecting and classifying hatched chicks in real-time, thereby increasing the overall success rate of the hatching process. The development of a Smart Egg Incubator is proposed by integrating IoT, Deep Learning, and machine learning technologies. An adaptive control system using fuzzy logic will automatically adjust temperature and humidity based on sensor data, ensuring optimal conditions for egg hatching. Deep Learning models YOLOv9 and CNN will be developed to improve detection accuracy and real-time monitoring of the hatching process. The integration of IoT with machine learning and Deep Learning techniques will create a Smart Egg Incubator that automatically and intelligently manages the environment and monitors the hatching process, expected to improve efficiency and effectiveness and provide solutions to previous limitations.

The novelty in developing the Hybrid Deep Learning - Machine Learning and IoT Model for Smart Egg Incubator lies in integrating innovative models, including fuzzy logic with IoT for adaptive temperature and humidity regulation according to the eggs needs during hatching. The YOLOv9 Deep Learning model combined with CNN [25] and optimized with Stochastic Gradient Descent (SGD) enables real-time monitoring and counting of hatched chicks with high accuracy [26]. Additional developments include an automatic egg container rotation system for temperature equalization and integrating visual technology and data analysis of the incubator environment to ensure optimal conditions. This hybrid model introduces advanced egg hatching methods and significantly contributes to the digital economy in poultry farming.

2. Research Methodology

The development stages of the Smart Egg Incubator will be carried out by integrating IoT and Machine learning, and creating a new hybrid model by developing a Deep Learning model approach to improve the accuracy of detecting chick objects. The proposed model development plan in this research can be found in figure 1.

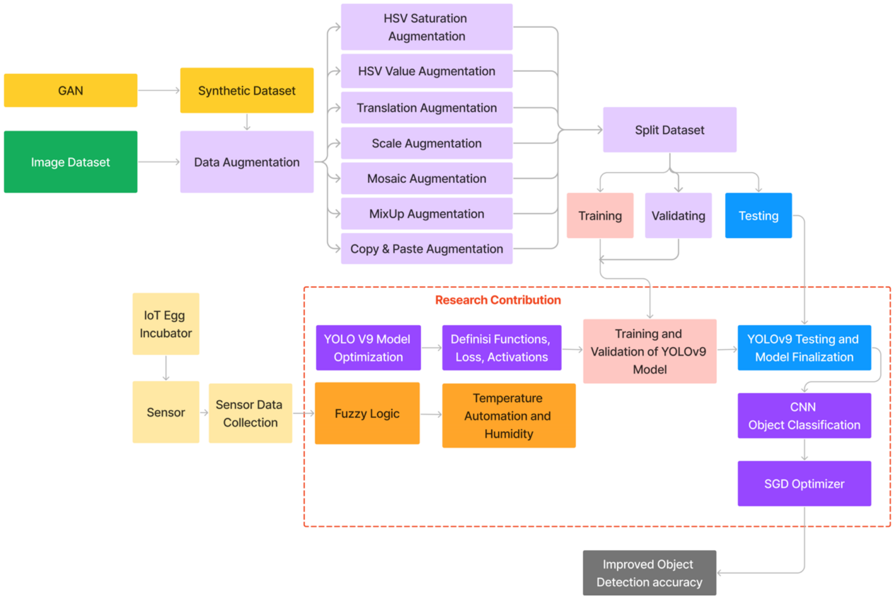

Figure 1. Flow of Model Development

In figure 1, the model development flow can explain the detailed methodological process for developing a Smart Egg Incubator that utilizes the latest technologies in Deep Learning, Machine learning, and IoT. The detailed steps within each stage of the methodology include:

2.1. Comprehensive Model Development Process

Sensors in the incubator collected real-time temperature and humidity data. The training dataset included video recordings from YouTube, images from Kaggle and GitHub, and synthetic data generated using GANs to ensure diversity. Collected images were sorted, cropped, and resized to 640x640 pixels. Data augmentation techniques such as rotation, rescaling, translation, lighting variation, HSV saturation, and value augmentation were applied to increase variation and prevent overfitting. The YOLOv9-S model was trained using the PyTorch framework with a batch size of 16, an initial learning rate of 0.01, and 35 epochs. MixUp and Mosaic Augmentation techniques were used to enhance robustness. The dataset was split into 70% training data, 20% validation data, and 10% test data for thorough evaluation. The SGD optimizer with a weight decay of 0.00005 adjusted model weights for fast convergence. Throughout training, loss values, precision, and recall metrics were monitored to fine-tune parameters, reducing overfitting and improving accuracy. This integrated approach ensured the development of an efficient and accurate Smart Egg Incubator system, enhancing its capability to monitor and regulate the hatching environment effectively.

2.2. Model Integration and Testing

The trained model is integrated with CNN for object classification and tested using real-time data from the incubator. Performance metrics such as mAP50 and mAP50:95 is used to evaluate the model's accuracy in detecting and classifying hatched chicks. The use of the Generative Adversarial Networks (GAN) model aims to create a synthetic dataset that enriches the training data due to the difficulty of obtaining a large number of chick image datasets [27], [28], [29]. This is important in helping the model to learn from a wider variety of images, enabling more accurate object detection. The use of GAN for dataset augmentation could use more details on their role and effectiveness in enriching training data. GAN are used to generate synthetic data that closely resembles real-world images, thus increasing the diversity of the training dataset. This augmentation technique is very effective in overcoming the limitations of small datasets by providing a larger and more varied set of training examples. By generating realistic images, GAN helps improve the robustness and generalization ability of deep learning models. The synthetic data generated by GAN ensures that the model is exposed to a wide range of variations, which is crucial for accurate object detection and classification in real-world scenarios.

Furthermore, the original image dataset augmentation technique involving image manipulation was used to increase the number of datasets for training [30], [31]. Augmentation is important to avoid overfitting and make the model more adaptable to small differences in new data in field trials. The division of the datasets into training, validation, and testing makes it possible to hone the model on the training data and align it with the validation data, as well as test its ability to generalize on the testing dataset. This is an optimization in ensuring that the model not only learns the training data but also understands the features essential for effective object detection.

The core of this research contribution is the development of a modified YOLO V9 model including more specific definitions of functions, losses, and activations [32], [33]. The training and validation of this model forms the basis for the automation of object detection, namely the detection of hatched chicks. After training the model is tested using a dataset set aside to ensure that the finalized model delivers high performance, fast real-time object detection and high accuracy. The model is then integrated with CNN for object classification [34] and optimized with SGD which is known for its efficiency in adjusting model weights to achieve fast convergence, so the entire system model will be tested to improve object detection accuracy [35]. The developed model will be tested using the objects of chicks, eggs, and newly hatched eggs using cameras attached to IoT devices in real-time.

YOLO V9 released in February 2024 introduces Programmable Gradient Information (PGI) to address information loss in neural networks by providing reliable gradients for weight updates [36], [37]. These innovations include a new network architecture Generalized Efficient Layer Aggregation Network (GELAN) that improves parameter utilization and achieves better results than previous versions of YOLO [38]. The following figure 2 describes the Pan Feature Maps of GELAN and YOLOv9:

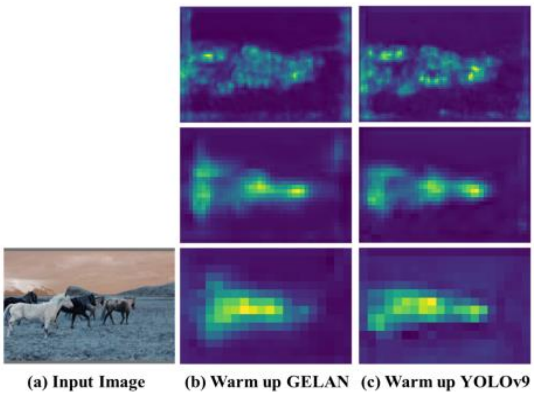

Figure 2. Pan Feature Maps of GELAN and YOLOv9 (GELAN + PGI)

GELAN is an architectural enhancement that enables better parameter usage and computational efficiency [39]. It combines the capabilities of CSPNet [40] to streamline gradients efficiently with ELAN architecture which is speed-oriented [41]. In this section, we present a comparison of the CSPNet, ELAN, and GELAN architectures, along with the development of the YOLOv9-S architecture. Figure 3 illustrates the comparison between the CSPNet, ELAN, and GELAN architectures, providing insights into the performance and structural differences among these three architectures and highlighting their respective advantages and limitations. Additionally, figure 4 demonstrates the development of the YOLOv9-S architecture, showcasing the progressive improvements and innovations incorporated into the YOLOv9-S design, leading to enhanced performance and accuracy.

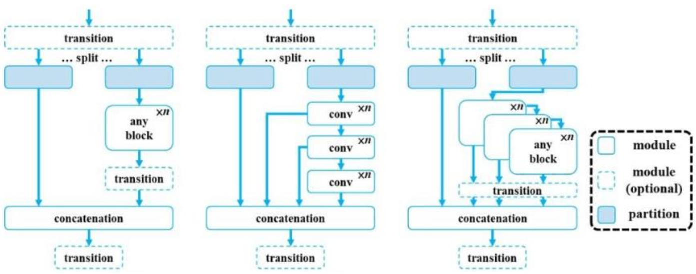

Figure 3. Comparison of CSPNet, ELAN, and GELAN architectures [38]

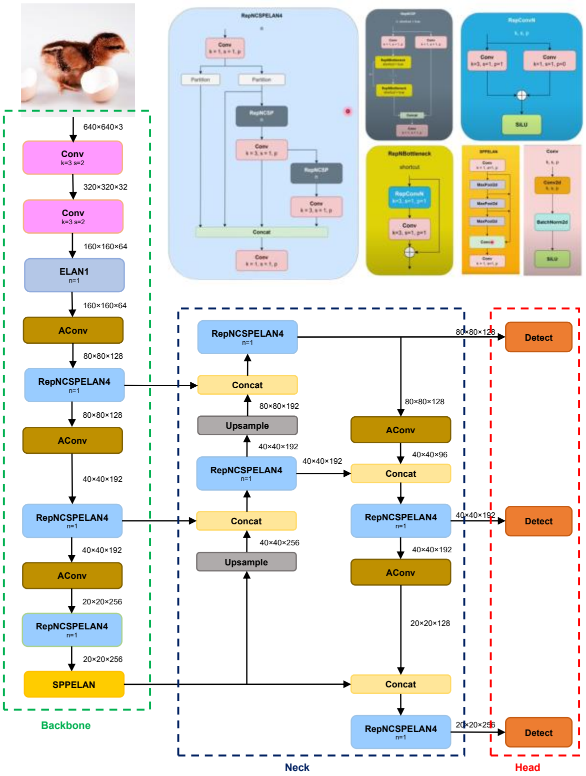

Figure 4. YOLOv9-S Architecture Development

Two important components of the Generalized Efficient Layer Aggregation Network (GELAN) in YOLOv9-S are SPPELAN and RepNCSPELAN4. SPPELAN incorporates Spatial Pyramid Pooling in the ELAN structure, starting with a convolution layer to adjust the channel dimensions, followed by spatial pooling to capture multi-scale information, and then consolidated through another convolution layer for detailed feature extraction. RepNCSPELAN4, an advanced version of CSP-ELAN, simplifies feature extraction by splitting the input from the initial convolution layer into two paths, processed through RepNCSP and convolution layers, then recombined. This dual-path strategy improves gradient flow, feature reuse, learning efficiency, and model inference speed.

YOLO is one of the most efficient and fast object detection architectures [42], capable of detecting objects in a single-stage compared to two-stage approaches such as R-CNN [43]. Figure 5 depicts the architecture of the object detector, which is divided into two stages: one-stage and two-stage detectors. This architecture consists of five main components: input, backbone, neck, dense prediction, and sparse prediction. The single-stage detector covers the process from input to dense prediction, while the two-stage detector proceeds from dense prediction to sparse prediction to improve object detection accuracy.

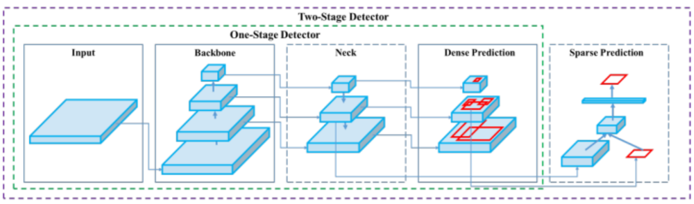

Figure 5. Object detection [44]

Reasons to choose YOLOv9-S [38] is due to the significant improvements it brings over previous versions, such as the YOLOv7 [45] and YOLOv8 [46] . The YOLOv9-S offers high speed that enables real-time image processing suitable for applications such as incubator monitoring. With improvements to the architecture and algorithms this model is capable of detecting objects with high accuracy. In addition, its efficiency in using computing resources makes it possible to run on more limited hardware. Figure 6 shows the comparison of real-time object detectors on the MS COCO dataset.

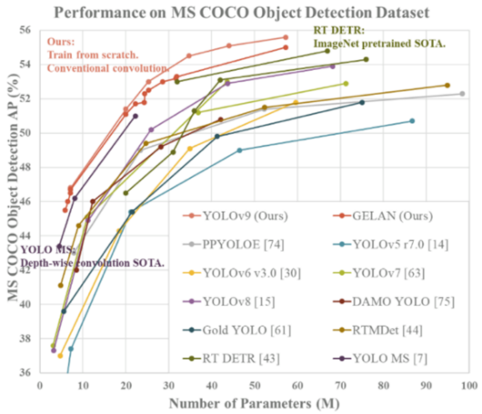

Figure 6. Comparison of real-time object detectors on the MS COCO dataset [38]

YOLOv9-S offers several key improvements over its predecessors, including enhanced detection accuracy and speed. One significant enhancement is the incorporation of the CSPDarknet53 backbone, which improves feature extraction and reduces computational complexity. Additionally, YOLOv9-S employs a new anchor-free detection head, which simplifies the detection process and reduces the number of hyperparameters that need tuning. These improvements result in higher mAP (mean Average Precision) scores and faster inference times. However, these enhancements come with trade-offs, such as increased memory usage due to the deeper network architecture and the need for more extensive training data to achieve optimal performance.

In this study, the YOLOv9-S architecture was chosen over more complex variants such as YOLOv9-C and YOLOv9-E due to several considerations. YOLOv9-S requires lower computation, uses fewer parameters, and is more efficient in resource usage, important for hardware with limited capacity. In addition, the simpler architecture allows the YOLOv9-S to process images very quickly, ensuring the monitoring system can operate in real-time without significant lag. Despite being simpler, the YOLOv9-S is powerful enough to detect hatched chicks with sufficient accuracy, making it a practical choice over more complex variant. Figure 6 shows the performance comparison of various real-time object detectors on the MS COCO dataset and table 1 presents a performance comparison of various YOLOv9-S architecture models on the MS COCO dataset.

Table 1. Performance comparison of YOLOv9-S architecture on MS COCO dataset

Model	Test Size	APval	P50val	AP75val	Param.	FLOPs
YOLOv9-T	640	38.3%	53.1%	41.3%	2.0M	7.7G
YOLOv9-S	640	46.8%	63.4%	50.7%	7.1M	26.4G
YOLOv9-M	640	51.4%	68.1%	56.1%	20.0M	76.3G
YOLOv9-C	640	53.0%	70.2%	57.8%	25.3M	102.1G
YOLOv9-E	640	55.6%	72.8%	60.6%	57.3M	189.0G

The table includes the models YOLOv9-T, YOLOv9-S, YOLOv9-M, YOLOv9-C, and YOLOv9-E, with test sizes set at 640. The performance metrics include APval (Average Precision), P50val (Precision at 50%), and AP75val (Average Precision at 75%), as well as the number of parameters and FLOPs (Floating Point Operations). YOLOv9-E achieves the highest performance across all metrics with an APval of 55.6%, P50val of 72.8%, and AP75val of 60.6%, but it also has the highest number of parameters (57.3M) and FLOPs (189.0G), indicating a trade-off between accuracy and computational complexity. The YOLOv9-S model, known for its high accuracy and speed in object detection, significantly enhances the real-time monitoring capabilities of the Smart Egg Incubator. This allows farmers to efficiently manage and monitor the hatching process remotely, reducing the need for constant manual supervision. The model's ability to accurately detect and classify hatched chicks in real-time ensures higher hatching success rates and reduces chick mortality. Additionally, the integration of YOLOv9-S with IoT technology enables proactive adjustments to the hatching environment, leading to optimal conditions for egg incubation. These technical advancements provide tangible benefits to users by increasing efficiency, improving hatching success rates, and offering greater convenience and control over the hatching process.

In the IoT layer for egg incubators, data is collected from sensors including temperature and humidity. Sensors attached to the incubator collect real-time data on conditions such as temperature and humidity, which play an important role in creating optimal hatching room conditions. The application of machine learning fuzzy logic algorithms on this sensor data for temperature and humidity adjustments adaptively and automatically according to the specific needs of the eggs to be hatched [47]. The hardware design of the Smart Egg Incubator can be seen in figure 7, and figure 8 shows the physical design of the Smart Egg Incubator:

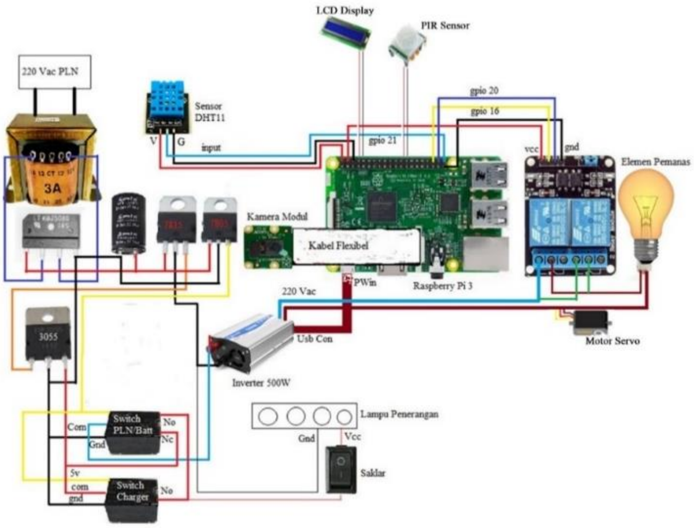

Figure 7. Smart Egg Incubator Tool Set

Figure 8. Design of the Smart Egg Incubator to be developed

The Smart Egg Incubator is built using several key components: the Raspberry Pi3 Model B microcontroller, which functions as the main processing unit; a webcam for monitoring the hatching process and assessing the quality of hatched eggs; and an egg incubator that serves as the primary hatching environment. Capacitors (33 PF) store electrical charges, while the DHT11 sensor monitors temperature and humidity, and the PIR sensor detects movement within the incubator, sending notifications to the user's smartphone via IoT technology when eggs hatch. An LCD display shows the incubator's temperature, and LED indicators signal when temperatures are too high, normal, or too low. A servo motor rotates the eggs for even exposure to light, and an incandescent lamp provides heat, turning on and off automatically to maintain optimal temperature. Additionally, a fan activates automatically to cool the incubator if the temperature exceeds the normal range.

3. Results and Discussions

3.1. IoT Implementation on Smart Egg Incubator

Figure 9 shows the display of the Smart Egg Incubator. From the test results that have been conducted, this incubator functions as expected and all components of this incubator are feasible to implement. The compact design and the use of advanced technology ensure that it can be effectively used in the egg hatching process with better control over temperature and humidity.

Figure 9. Smart Egg Incubator display

The benchmark for the success of the Smart Egg Incubator UHTP prototype development is seen from the success of the prototype in carrying out the egg hatching process and the success of the tool to maintain temperature and humidity. From the results of monitoring and evaluation of the Smart Egg Incubator, the temperature threshold data is 34.3 ° C to 39.5 ° C and humidity 57- 67% as shown in table 2.

Table 2. Temperature and Humidity Threshold Testing

Number of Testing	Temperature	Humidity	Description
1	34.3	57	Lights On, Fan Off
2	35.0	58	Lights On, Fan Off
3	36.0	58	Lights On, Fan Off
4	39.5	68	Lights Off, Fan On
5	37.2	63	Lights On, Fan Off
6	36.8	61	Lights On, Fan Off
7	39.3	67	Lights Off, Fan On

The results of testing the temperature and humidity thresholds of the Smart Egg Incubator in Table 2 how that at temperatures below the normal incubation temperature limit of 37.7 ° C the device will turn on the lights and turn off the fan to increase the temperature to the normal temperature limit of 37.7 ° C to 38.8 ° C. If the incubator has reached the maximum temperature of the normal temperature of 38.8 ° C then the incubator will turn off the lights and turn on the fan to the minimum limit of normal temperature. To regulate the temperature, it is necessary to control the lights periodically and for humidity, it is necessary to fill in enough water in the water tub until the temperature and humidity are appropriate. On the 18th day the signs of eggs began to appear with small cracks in the eggs and to be able to hatch completely it takes a day. From a trial sample of 20 eggs, in 19 days 3 eggs were successfully hatched. Figure 10 shows the conditions inside the incubator chamber.

Figure 10. Conditions in the Incubator Room

 ISSN 2723-6471

The following graph shows the results of the temperature and humidity threshold testing using fuzzy logic. This graph illustrates the changes in temperature and humidity over seven tests, with additional information about the condition of the lights and fans during the test. Figure 11 shows a graph of the test results using fuzzy logic.

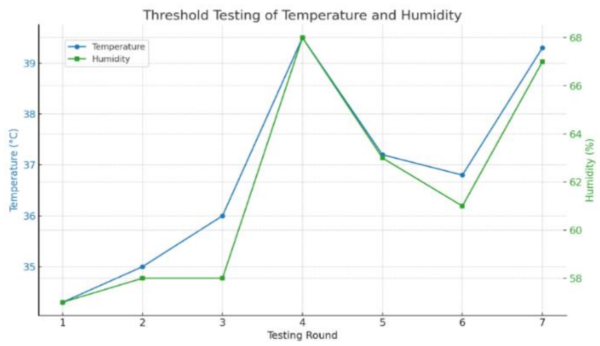

Figure 11. Graph of Test Results Using Fuzzy Logic

The application of fuzzy logic to the incubator system shows positive results in adaptively regulating temperature and humidity based on collected sensor data. Using fuzzy logic, the system can automatically adjust the environmental conditions of the incubator to ensure the temperature and humidity remain within the optimal range for hatching eggs. The test results show that the system successfully maintains the temperature between 34.3°C and 39.5°C and the humidity between 57% and 68%, with effective responses to changes in environmental conditions, such as turning off the lights and starting the fans when the temperature increases. The success rate of fuzzy logic in this test reached 90%.

2.2. Acquisition Data

Dataset collection is an important step in the development of a reliable object detection model. In this research, data was collected from various online sources to ensure the variety and completeness of the datasets. The following are some of the techniques and sources used to collect datasets: video footage from youtube, image sources from websites such as kaggle and github, and synthetic data using GAN.

2.3. Processing Data

Once the dataset is collected, the next step is to perform image preprocessing to train the YOLOv9-S model. Images from various online sources were sorted to select those that were appropriate and relevant, while those that were blurry, too dark, or unclear were removed. Each selected image was then cropped with a 1:1 ratio for aspect ratio uniformity, and resized to 640x640 pixels according to the standard input size of the YOLOv9-S model.

The processed dataset was then divided into two parts: 85% for training data and 15% for testing data. Out of a total of 439 images, 65 images were used for test data, and 374 images were used for the training process. Before training, rotational augmentation was performed on the training data so that the total data for training was 1,122 images. Figure 12 shows an example of the dataset used.

Figure 12. Sample datasets

ISSN 2723-6471

2.4. Training Model

The model training process involves several important steps and settings to ensure the YOLOv9-S model can detect objects with high accuracy. Data augmentation techniques are used to increase the variety of the dataset and help the model generalize better. The augmentation techniques used include HSV Saturation Augmentation to increase color variation [48], HSV Value Augmentation to make the model more robust to lighting changes [49], Translation Augmentation to make the model more resistant to shifting objects in the image, and Scale Augmentation to help the model recognize objects of various sizes [50]. This augmentation aims to enrich the dataset and improve the model's ability to deal with various conditions when detecting objects [51]. Figure 13 shows the HSV Saturation, HSV Values, Translation, and Scale augmentations, which were used to increase the variety in the dataset. These augmentation techniques help in enriching the training data, allowing the model to learn from different conditions and improve prediction accuracy.

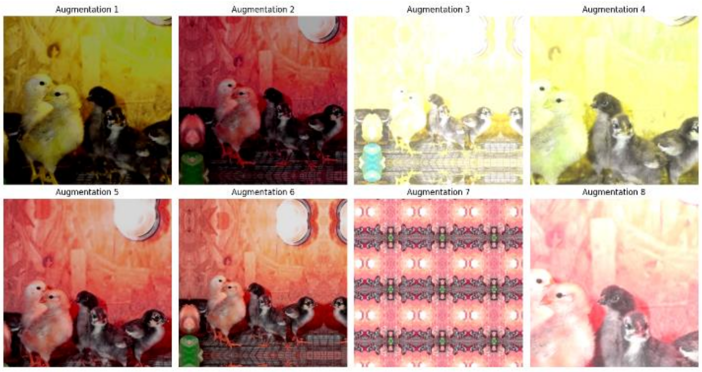

Figure 13. HSV Saturation, HSV Values, Translation, and Scale Augmentation

Mosaic Augmentation combines four different images into one to increase the diversity of context in the image [44], [52], [53]. Figure 14 shows the example of mosaic augmentation.

Figure 14. Example of mosaic augmentation

MixUp Augmentation mixes two images and their annotations to increase data variation and make the model more resistant to overfitting [54], [55]. Figure 15 shows the example of MixUp augmentation.

Figure 15. Example of MixUp augmentation

Copy & Paste Augmentation adds objects from one image to another to increase the number and variety of objects in the image [56], [57]. Figure 16 shows the example of copy and paste augmentation.

Figure 16. Example Of Copy & Paste Augmentation

The training process involved splitting the dataset into 70% training data, 20% validation data, and 10% test data. The model was trained using the PyTorch framework [58], with a number of epochs of 35 and a batch size of 16. The initial learning rate was set at 0.01, and the weight decay coefficient was 0.00005. Warm up training strategy is implemented in the first three epochs with a momentum value of 0.8. Furthermore, mosaic augmentation will be turned off in the last 5 epochs. The model training process involves setting hyperparameters and data augmentation techniques to achieve high object detection accuracy. The model was trained for 35 epochs with a batch size of 8, using the SGD optimizer. The initial learning rate was set at 0.01 with a momentum of 0.937 and weight decay of 0.00005. There is a warm-up for 3 epochs with a warm-up momentum of 0.8. The augmentation techniques used include HSV saturation augmentation with a value of 0.7, HSV value augmentation with a value of 0.4, translation augmentation of 0.1, and scale augmentation of 0.9. In addition, mosaic augmentation is applied with a value of 1.0, MixUp augmentation with 0.15, and copy & paste augmentation with 0.3. The training process is closed with close mosaic epochs of 5.

3.5. Performance Metrics

After the model was trained and validated on the Google Colab platform with a GPU platform, in the experiments conducted, the model can detect chicks both in a newly hatched state (wet) and in a dry state with a precision value at the time of training of 92%. Figure 17 shows the overall results of the proposed model, including loss, precision, and recall values.

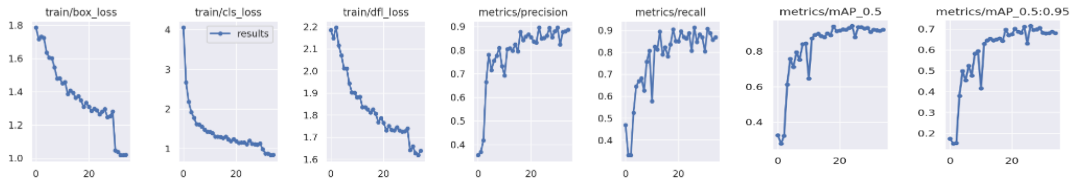

Figure 17. Training Results of The Proposed Model

The train/box_loss, train/cls_loss, and train/dfl_loss graphs show a consistent decrease in loss values during training, indicating the model is getting better at adjusting parameters for object detection. The metrics/precision and metrics/recall graphs show an increase in precision and recall, indicating the model is getting more accurate in detecting objects with fewer errors. Evaluation using test data showed high performance with mAP50 of 93.7%, indicating high accuracy in detecting and classifying objects at lower IoU thresholds. Meanwhile, the mAP50:95 of 71.3% demonstrates the model's ability to detect objects at various levels of difficulty. These results show that the model is able to achieve high object detection performance well under various conditions.

3.6. Result and Discussion

Figures 18 and figure 19 show the results of the model prediction in detecting the chick object.

Figure 18. Object Detection Results from The Trained Model

Figure 19. Object Detection Result on Egg Incubator

The figure above shows that the model can consistently detect and classify chicks under various lighting conditions and viewing angles even in complex environments such as smart egg incubators. The high parameter and computational efficiency with only 7.1 million parameters and 26.4 G FLOPs allow this model to be used in real-time applications such as egg incubator monitoring with high accuracy. To evaluate the performance of the proposed model, training and evaluation of other object detection models are conducted for comparison using the same datasets with 35 epochs and default parameter settings. The models used for comparison are YOLOv8 and YOLOv7 [45]. The following table 3 compares the evaluation results:

Table 3. Model Performance Comparison

Model	mAP50	mAP50:95	Param.	FLOPs
Our Model (YOLOv9-S)	93,7%	71,3%	7,1 M	26,4 G
YOLOv8-S	93,2%	72,7%	11,2 M	28,6 G
YOLOv7	87,6%	58,3%	37 M	103,2 G

The evaluation results show that the YOLOv9-S model has competitive performance compared to YOLOv8 and YOLOv7. Although YOLOv8 is slightly superior in mAP50:95, YOLOv9-S shows balanced performance with a lower number of parameters and FLOPs of 26.4 G, making it more computationally efficient, faster in inference, and more energy efficient. Designed for computational efficiency, YOLOv9-S enables real-time object detection with high accuracy. Advanced augmentation techniques such as mosaic and MixUp enhance model generalization capabilities. Optimal hyperparameter settings, including warm-up strategy and learning rate adjustment, contribute to the high performance of this model.

The achieved metrics, such as mAP50 of 93.7% and mAP50:95 of 71.3%, demonstrate the high accuracy and robustness of the YOLOv9-S model in detecting and classifying hatched chicks. For end-users, this high level of accuracy means more reliable monitoring of the hatching process, reducing the need for manual supervision and increasing the likelihood of identifying and addressing issues promptly. This can lead to higher hatching success rates and lower chick mortality. Additionally, the ability to accurately classify hatched chicks in real-time allows farmers to make immediate decisions regarding the care and management of the chicks, improving overall efficiency and productivity in the hatchery operations. These practical benefits highlight the value of the developed system in enhancing the effectiveness and reliability of egg incubation for end-users.

The integration of IoT technology significantly enhances the overall performance and scalability of the Smart Egg Incubator system. IoT enables real-time monitoring and control of the hatching environment, allowing for precise adjustments to temperature and humidity. This results in higher hatching success rates and reduced chick mortality. However, the implementation of IoT also presents challenges such as ensuring reliable network connectivity and data security. Scalability is achieved through the modular design of the system, allowing additional sensors and devices to be integrated as needed. Despite these advantages, limitations such as the dependency on stable internet connections and potential vulnerabilities to cyber-attacks need to be addressed.

4. Conclusion

The YOLOv9-S model has demonstrated high performance and reliability in object detection for smart incubator monitoring applications in chicken hatching. The model achieved an impressive mAP50 of 93.7% and mAP50:95 of 71.3%, with efficient parameter and computational usage (7.1 million parameters and 26.4 GFLOPs). Effective augmentation techniques and optimal hyperparameter settings contributed to these results. Additionally, fuzzy logic was used to adaptively regulate incubator temperature and humidity, maintaining optimal conditions with a 90% success rate. The system effectively controlled temperature between 34.3°C and 39.5°C and humidity between 57% and 68%. The integration of IoT in the smart incubator facilitated improved monitoring and control, resulting in higher hatching success rates, reduced chick mortality, and increased operational efficiency. The high accuracy and robustness of the YOLOv9-S model support its potential for enhancing real-time monitoring and decision-making in hatcheries. These technological advancements can lead to more intelligent, automated systems that require minimal human intervention, boosting productivity and scalability. Future research could focus on the integration of the upcoming YOLOv10 model, which promises even higher accuracy and efficiency. Additionally, exploring diverse data augmentation techniques could further enhance model robustness. Optimizing fuzzy logic algorithms and incorporating various sensors, such as CO2 and advanced humidity sensors, could improve environmental regulation in incubators. These advancements would benefit not only smart incubator applications but also other areas of poultry farming, such as broiler production and disease monitoring, providing a comprehensive approach to farm management and automation.

5. Declaration

5.1. Author Contributions

Conceptualization: A.F., R.W., M., Y.I., and R.M.; Methodology: R.W., M., Y.I., and R.M.; Software: A.F.; Validation: A.F., R.W., M., Y.I., and R.M.; Formal Analysis: A.F., R.W., M., Y.I., and R.M.; Investigation: A.F.; Resources: R.W.; Data Curation: R.W.; Writing - Original Draft Preparation: A.F., R.W., M., Y.I., and R.M.; Writing - Review & Editing: R.W., A.F., M., Y.I., and R.M.; Visualization: A.F.; All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

5.3. Funding

Acknowledgments are addressed to Direktorat Riset, Teknologi dan Pengabdian kepada Masyarakat Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi Indonesia for providing funding in the Fundamental Research Scheme - Regular Research Program 2024, as well as Universitas Hang Tuah Pekanbaru which has provided facilities for conducting research activities.

5.4. Institutional Review Board Statement

Not applicable.

5.5. Informed Consent Statement

Not applicable.

5.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] F. Peprah, S. Gyamfi, M. Amo-Boateng, E. Buadi, and M. Obeng, “Design and construction of smart solar powered egg incubator based on GSM/IoT,” *Sci. African,* vol. 17, no. Sept., p. e01326, pp. 1-10, 2022, doi: 10.1016/j.sciaf.2022.e01326.

[2] M. A. Mesquita et al., “Results of hatching and rearing broiler chickens in different incubation systems,” *Poult. Sci.*, vol. 100, no. 1, pp. 94–102, 2021, doi: 10.1016/j.psj.2020.09.028.

[3] M. D. Massey and J. A. Hutchings, “Thermal variability during ectotherm egg incubation: A synthesis and framework,” *J. Exp. Zool. Part A Ecol. Integr. Physiol.*, vol. 335, no. 1, pp. 59–71, 2021, doi: 10.1002/jez.2400.

[4] Y. Cheng et al., “Optimization of sterlet (Acipenser ruthenus) egg incubation,” *Anim. Reprod. Sci.*, vol. 215, no. Apr., pp. 1-23, 2020, doi: 10.1016/j.anireprosci.2020.106334.

[5] K. R. Cheepati and N. Balal, “Solar Powered Thermoelectric Air Conditioning for Temperature Control in Poultry Incubators,” *Sustain.*, vol. 16, no. 11, pp. 1-19, 2024, doi: 10.3390/su16114832.

[6] A. R. Gunderson, A. Fargevieille, and D. A. Warner, “Egg incubation temperature does not influence adult heat tolerance in the lizard Anolis sagrei,” *Biol. Lett.*, vol. 16, no. 1, pp. 1-5, 2020, doi: 10.1098/rsbl.2019.0716.

[7] S. A. Sampaio et al., “Influence of Monochromatic Light during Incubation on the Production and Metabolism of Low-Temperature Broiler Chicks,” *Animals*, vol. 14, no. 11, pp. 1-17, 2024, doi: 10.3390/ani14111620.

[8] M. E. Abd El-Hack, C. B. Hurtado, D. M. Toro, M. Alagawany, E. M. Abdelfattah, and S. S. Elnesr, “Impact of environmental and incubation factors on hatchability of duck eggs,” *Biol. Rhythm Res.*, vol. 53, no. 1, pp. 79–88, 2022, doi: 10.1080/09291016.2019.1628394.

[9] G. da S. Oliveira, V. M. dos Santos, J. C. Rodrigues, and S. T. Nascimento, “Effects of different egg turning frequencies on incubation efficiency parameters,” *Poult. Sci.*, vol. 99, no. 9, pp. 4417–4420, 2020, doi: 10.1016/j.psj.2020.05.045.

[10] A. Faroqi, M. R. Efendi, D. T. Ismail, and W. Darmalaksana, “Design of arduino uno based duck egg hatching machine with sensor DHT22 and PIR sensor,” *Proc. - 2020 6th Int. Conf. Wirel. Telemat. ICWT 2020*, vol. 6, no. Dec., pp. 14–17, 2020, doi: 10.1109/ICWT50448.2020.9243640.

[11] Y. Yanti, A. Rohman, S. Maesaroh, A. Mustopa, and R. M. Febrian, “The Implementation of The Internet of Things in The Duck Egg Incubator Monitoring System,” *TIERS Inf. Technol. J.*, vol. 3, no. 2, pp. 84–90, 2022, doi: 10.38043/tiers.v3i2.3891.

[12] S. B. Santoso, S. Adhy, N. Bahtiar, and I. Waspada, “Development of the Smart Chicken Eggs Incubator Based on Internet of Things Using the Object Oriented Analysis and Design Method,” *ICICoS 2020 - Proceeding 4th Int. Conf. Informatics Comput. Sci.*, vol. 4, no. November, pp. 0–5, 2020, doi: 10.1109/ICICoS51170.2020.9299000.

[13] A. H. Abdullah and R. Hamdan, “Smart Monitoring Temperature and Humidity Based on Incubator System using IoT,” *Evol. Electr. Electron. Eng.*, vol. 4, no. 2, pp. 859–866, 2023.

[14] K. Muttaqin, A. Ihsan, and H. Irawan, “Peningkatan Produktivitas Ternak Ayam Melalui Teknologi Inkubator Mesin Penetas Telur Berbasis Internet of Thing,” *JMM (Jurnal Masy. Mandiri)*, vol. 6, no. 5, pp. 3–10, 2022, doi: 10.31764/jmm.v6i5.10812.

[15] G. da S. Oliveira, C. McManus, I. R. R. Vale, and V. M. dos Santos, “Obtaining Microbiologically Safe Hatching Eggs from Hatcheries: Using Essential Oils for Integrated Sanitization Strategies in Hatching Eggs, Poultry Houses and Poultry,” *Pathogens*, vol. 13, no. 3, pp. 1-15, 2024, doi: 10.3390/pathogens13030260.

[16] F. Sufi, “Generative Pre-Trained Transformer (GPT) in Research: A Systematic Review on Data Augmentation,” *Inf.*, vol. 15, no. 2, pp. 1-27, 2024, doi: 10.3390/info15020099.

[17] C.-H. Lin, C. Kaushik, E. L. Dyer, and V. Muthukumar, “The good, the bad and the ugly sides of data augmentation: An implicit spectral regularization perspective,” J. *Mach. Learn.* Res., vol. 25, no. March, pp. 1–85, 2022.

[18] T. KONE, N. G. ANOH, B. G. YATANAN, Y. ange A. BROU, N. J. AHOUTOU, and J. D. A. ALLOU, “Design and Development of an Iot-Based Intelligent Incubator,” *Eng. Technol. J.*, vol. 9, no. 01, pp. 3396–3401, 2024, doi: 10.47191/etj/v9i01.21.

[19] K. B. Azahar, E. E. Sekudan, and A. M. Azhar, “Intelligent Egg Incubator,” *Int. J. Recent Technol. Appl. Sci.*, vol. 2, no. 2, pp. 91–102, 2020, doi: 10.36079/lamintang.ijortas-0202.129.

[20] D. Zakaria et al., “Egg Incubator Control System: A Review,” *J. Electr. Electron. Information, Commun. Technol.*, vol. 5, no. 1, p. 33-37, 2023, doi: 10.20961/jeeict.5.1.72718.

[21] B. K. Yadav, N. Pokhrel, D. Khatiwada, M. Khanal, T. Bajracharya, and R. Dhakal, “Design, Fabrication, and Performance Analysis of an Automatic Horizontal Egg Incubator,” *J. Inst. Eng.*, vol. 16, no. 1, pp. 77–85, 2021, doi: 10.3126/jie.v16i1.36557.

[22] T. L. Oo, H. Htay, and A. M. Oo, “Construction of Hen Egg Incubator Controlled Circuit Using Pic16F877a,” *J. Myanmar Acad. Arts Sci.*, vol. XVII, no. 2, pp. 461-482, 2019.

[23] M. A. Afandi, F. K. Purnomo, R. A. Rochmanto, and S. I. Purnama, “Monitoring and Controlling Temperature Egg Incubator Prototype Based LoRa Communication,” *Elinvo (Electronics, Informatics, Vocat. Educ.*, vol. 7, no. 2, pp. 119–126, 2023, doi: 10.21831/elinvo.v7i2.53664.

[24] S. Purwanti, A. Febriani, M. Mardeni, and Y. Irawan, “Temperature Monitoring System for Egg Incubators Using Raspberry Pi3 Based on Internet of Things (IoT),” *J. Robot. Control*, vol. 2, no. 5, pp. 349-352, 2021, doi: 10.18196/jrc.25105.

[25] M. Abdullah and A. F. Berhe, “A Hybrid Deep Learning CNN model for COVID-19 detection from chest X-rays,” *HELIYON*, vol. 10, no. 5, p. 101865, 2024, doi: 10.1016/j.heliyon.2024.e26938.

[26] S. Tomar, D. Dembla, and Y. Chaba, “Analysis and Enhancement of Prediction of Cardiovascular Disease Diagnosis using Machine Learning Models SVM, SGD, and XGBoost,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 4, pp. 469–479, 2024, doi: 10.14569/IJACSA.2024.0150449.

[27] D. J. Van Booven et al., “Synthetic Genitourinary Image Synthesis via Generative Adversarial Networks: Enhancing AI Diagnostic Precision,” *J. Pers. Med.*, vol. 14, no. 7, pp. 1-8, 2024.

[28] M. Shatnawi, “An Enhanced Model for Detecting and Classifying Emergency Vehicles Using a Generative Adversarial Network (GAN),” *Vehicles*, vol. 6, no. 3, pp. 1114–1139, 2024.

[29] M. Guo et al., “TDEGAN : A Texture-Detail-Enhanced Dense Generative Adversarial Network for Remote Sensing Image,” *Remote Sens.*, vol. 16, no. 13, pp. 1-23, 2024.

[30] P. Chlap, H. Min, N. Vandenberg, J. Dowling, L. Holloway, and A. Haworth, “A review of medical image data augmentation techniques for deep learning applications,” *J. Med. Imaging Radiat. Oncol.*, vol. 65, no. 5, pp. 545–563, 2021, doi: 10.1111/1754-9485.13261.

[31] C. Khosla and B. S. Saini, “Enhancing Performance of Deep Learning Models with different Data Augmentation Techniques: A Survey,” *IEEE Access*, vol. 9160048, no. Aug., pp. 79–85, 2022, doi: 10.1109/ICIEM48762.2020.9160048.

[32] H. T. Vo, K. C. Mui, N. N. Thien, and P. P. Tien, “Automating Tomato Ripeness Classification and Counting with YOLOv9,” *Int. J. Adv. Comput. Sci.* Appl., vol. 15, no. 4, pp. 1120–1128, 2024, doi: 10.14569/IJACSA.2024.01504113.

[33] Faruq Aziz and D. U. E. Saputri, “Efficient Skin Lesion Detection using YOLOv9 Network,” *J. Med. Informatics Technol.*, vol. 2, no. 1, pp. 11–15, 2024, doi: 10.37034/medinftech.v2i1.30.

[34] S. Saifullah and R. Drezewski, "Modified Histogram Equalization for Improved CNN Medical Image Segmentation," *Procedia Comput. Sci.*, vol. 225, no. June, pp. 3021–3030, 2023, doi: 10.1016/j.procs.2023.10.295.

[35] S. Suklabaidya and I. Das, "Performance Comparison of Multiple ANN Optimizer on IoT-enabled Sensor Fire Dataset," *Int. Arab J. Inf. Technol.*, vol. 20, no. 5, pp. 767–772, 2023, doi: 10.34028/iajit/20/5/9.

[36] J. Li, Y. Feng, Y. Shao, and F. Liu, "applied sciences IDP-YOLOV9 : Improvement of Object Detection Model in Severe Weather Scenarios from Drone Perspective," *Appl. Sci.*, vol. 14, no. 12, pp. 1-21, 2024.

[37] Y. Li, Z. Hu, Y. Zhang, J. Liu, W. Tu, and H. Yu, "DDEYOLOv9: Network for Detecting and Counting Abnormal Fish Behaviors in Complex Water Environments," *Fishes*, vol. 9, no. 6, pp. 1–21, 2024, doi: 10.3390/fishes9060242.

[38] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," *Cornell Univ.*, vol. 2402, no. Feb, pp. 1-4, 2024.

[39] T. Balakrishnan and S. S. Sengar, "RepVGG-GELAN: Enhanced GELAN with VGG-STYLE ConvNets for Brain Tumour Detection," *arXiv Prepr. arXiv2405.03541*, vol. 2405, no. May, pp. 1–12, 2024.

[40] C. Y. Wang, H. Y. Mark Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, and I. H. Yeh, "CSPNet: A new backbone that can enhance learning capability of CNN," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2020, no. Jun., pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203.

[41] C. Y. Wang, H. Y. M. Liao, and I. H. Yeh, "Designing Network Design Strategies Through Gradient Path Analysis," J. *Inf. Sci. Eng.*, vol. 39, no. 2, pp. 975–995, 2023, doi: 10.6688/JISE.202307_39(4).0016.

[42] A. Ahmed, A. S. Imran, A. Manaf, Z. Kastrati, and S. M. Daudpota, "Enhancing wrist abnormality detection with YOLO: Analysis of state-of-the-art single-stage detection models," *Biomed. Signal Process. Control*, vol. 93, no. Jan., pp. 1-13, 2024, doi: 10.1016/j.bspc.2024.106144.

[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[44] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020.

[45] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors," vol. 00721, no. Jul., pp. 7464–7475, 2023, doi: 10.1109/cvpr52729.2023.00721.

[46] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-Time Flying Object Detection with YOLOv8," 2023.

[47] F. Furizal, S. Sunardi, A. Yudhana, and R. Umar, "Energy Efficiency with Internet of Things Based Fuzzy Inference System for Room Temperature and Humidity Regulation," *Int. J. Eng. Trans. A Basics*, vol. 37, no. 1, pp. 187–200, 2024, doi: 10.5829/ije.2024.37.01a.17.

[48] Anisa Nur Azizah and C. Fatichah, "Tajweed-YOLO: Object Detection Method for Tajweed by Applying HSV Color Model Augmentation on Mushaf Images," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 7, no. 2, pp. 236–245, 2023, doi: 10.29207/resti.v7i2.4739.

[49] E. K. Kim, H. Lee, J. Y. Kim, and S. Kim, "Data augmentation method by applying color perturbation of inverse PSNR and geometric transformations for object recognition based on deep learning," *Appl. Sci.*, vol. 10, no. 11, pp. 1-20, 2020, doi: 10.3390/app10113755.

[50] R. Bai, Q. Xu, Z. Meng, L. Cao, K. Xing, and F. Fan, "Rolling bearing fault diagnosis based on multi-channel convolution neural network and multi-scale clipping fusion data augmentation," *Meas. J. Int. Meas. Confed.*, vol. 184, no. Feb., pp. 1-17, 2021, doi: 10.1016/j.measurement.2021.109885.

[51] D. S. Park, "Specaugment on Large Scale Datasets," *ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. - Proc.*, vol. 2020, no. May, pp. 6879–6883, 2020, doi: 10.1109/ICASSP40776.2020.9053205.

[52] Y. Li, R. Cheng, C. Zhang, M. Chen, H. Liang, and Z. Wang, "Dynamic Mosaic algorithm for data augmentation," *Math. Biosci. Eng.*, vol. 20, no. 4, pp. 7193–7216, 2023, doi: 10.3934/mbe.2023311.

[53] W. Hao and S. Zhili, "Improved mosaic: Algorithms for more complex images," *J. Phys. Conf. Ser.*, vol. 1684, no. 1, pp. 1-8, 2020, doi: 10.1088/1742-6596/1684/1/012094.

[54] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "MixUp: Beyond empirical risk minimization," *6th Int. Conf. Learn.*

Represent. ICLR 2018 - Conf. Track Proc., vol. 1710, no. Apr, pp. 1–13, 2018.

[55] B. Ahmmed et al., "Deep learning with mixup augmentation for improved pore detection during additive manufacturing," *Sci. Rep.*, vol. 14, no. 1, pp. 1–14, 2024, doi: 10.1038/s41598-024-63288-1.

[56] G. Ghiasi et al., "Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 9577055, no. Jun., pp. 2917–2927, 2021, doi: 10.1109/CVPR46437.2021.00294.

[57] J. S. Kang and K. Chung, "STAug: Copy-Paste Based Image Augmentation Technique Using Salient Target," *IEEE Access*, vol. 10, no. Aug., pp. 123605–123613, 2022, doi: 10.1109/ACCESS.2022.3224141.

[58] F. Laporte, J. Dambre, and P. Bienstman, "Highly parallel simulation and optimization of photonic circuits in time and frequency domain based on the deep-learning framework PyTorch," *Sci. Rep.*, vol. 9, no. 1, pp. 1–9, 2020, doi: 10.1038/s41598-019-42408-2.