Journal of Research in Artificial Intelligence for Systems and Applications (RAISA)
2025, Vol. 01, No. 01, E-ISSN : XXXX-XXXX

Original Research

Detection of Infectious Diseases in Broiler Chickens Based on Chicken Feces Images using A Modified YOLOv5 Algorithm

Anggi Wahyu Utomo[1], Dwiretno Istiyadi Swasono[2], Muhammad Arief Hidayat[3]

[1,2,3]Department of Computer Science, University of Jember

Abstract

This research focuses on detecting infectious diseases in broiler chickens through feces images using a modified YOLOv5 algorithm. Early disease detection is crucial for maintaining product quality and protecting consumer health. YOLOv5's high detection speed and accuracy make it suitable for identifying multiple objects, enabling farmers to anticipate and mitigate disease risks. The study used a dataset of 8,067 images across four categories: coccidiosis, healthy, NCD, and salmonella, split into 90:10 and 80:20 training-validation ratios. Six modified YOLOv5 models were tested. Key modifications included adding bottleneck layers, replacing SiLU activation with ELU, and incorporating CBAM in various configurations. The best result, with a 90.0% mAP, came from the sixth modification (C3CBAM layer) with a 90:10 split. Testing via a mobile application showed the second (80:20 split) and fourth (90:10 split) models excelled in detecting both large and small objects.

Keywords

Broiler Chicken; Chicken Feces; Chicken Diseases; Modification; YOLOv5.

1. Introduction

Chickens have a crucial role in fulfilling the high-protein and low-fat food needs through their meat and eggs throughout the world [1]. One of the breeds of broiler chickens farmed in Indonesia is broiler chicken. Broiler chickens have been specifically bred for marketing at a young age, characterized by relatively fast growth, broad chests, and abundant meat accumulation [2]. The population of broiler chickens in Indonesia in 2022 was 3,168,325,176. This data was obtained from 34 provinces in Indonesia in 2022 [3].

The increasing demand for chicken meat has driven heightened attention to chicken health. In order to maintain product quality and safeguard consumer health, several measures can be taken to reduce the spread of diseases that can threaten chickens. One such measure is early disease detection through chicken feces. Several infectious diseases can be detected through chicken feces, such as Newcastle disease caused by the Newcastle Disease Virus (NDV), pullorum disease caused by the bacterium Salmonella pullorum, and coccidiosis caused by the parasite coccidia [4]. If these diseases are too late to be detected, it can delay treatment, potentially leading to chicken mortality and significant losses for chicken farmers [5].

Through the rapid advancement of technology, detecting diseases that affect broiler chickens becomes easier, such as using computer division to examine feces images. Detecting chicken diseases based on feces images can provide accurate results regarding the type of disease affecting the chickens, as chicken feces sometimes contain various viruses or bacteria that cause infections. This can be observed from the characteristics of feces based on the disease [6]. Deep learning methods are among the options for detecting chicken diseases based on feces images [6].

*Corresponding author: Anggi Wahyu Utomo

Email addresses:

192410103049@mail.unej.ac.id (Anggi Wahyu Utomo), istiyadi@unej.ac.id (Dwiretno Istiyadi Swasono, ST), arief.hidayat@unej.ac.id (Muhammad Arief Hidayat)

Received: 12 12 2024; **Accepted:** 27 12 2024; **Published:** 15 01 2025

YOLO (You Only Look Once) is a deep learning model that can be used for object detection. The algorithm of YOLO has high detection speed and accuracy, making it highly effective at detecting multiple objects in images, videos, or real-time scenarios [7]. One study using YOLOv5 successfully detected healthy and diseased chickens based on feces images, achieving a highest average accuracy of 89.2% [8]. Another study classified infectious chicken diseases based on feces images using CNN (Convolutional Neural Network) and found that 95.40% were predicted to have coccidiosis, 90.21% were predicted to have Newcastle disease, 96.50% were predicted to have pullorum disease, and 94.97% were predicted to be healthy [4]. Additionally, another study developed a mobile application to detect chicken diseases based on feces images by implementing YOLOv3 for object detection and ResNet50 for classification tasks [9]. Through deep learning and its advancements, chicken farmers have the potential to a better diagnose of the diseases affecting their chickens and improve livestock health, thus enhancing production [6].

This research will discuss a disease detection system for broiler chickens based on feces images produced by broiler chickens using the YOLOv5 algorithm, as this study will focus on object detection. Object detection can identify and locate objects in images or videos, producing bounding boxes that indicate the location of the objects and their scores. Additionally, YOLOv5 offers flexibility and ease in modifying each of its layers according to needs. However, YOLOv5 also has a weakness in accurately determining the location of smaller objects or objects with more complex shapes within an image [10]. Therefore, modifications to its architecture are necessary to address these issues. Modifications are carried out by adding new layers or replacing existing ones with other modules within its architecture. Since this research will be conducted on mobile devices, the model used is YOLOv5s. YOLOv5s requires lower device specifications, making it suitable for mobile application development. This system can be used to help predict potential diseases affecting broiler chickens and increase the awareness of broiler chicken farmers about diseases that can infect their chickens at any time. To determine the actual disease, farmers can have the feces examined through laboratory tests conducted by a veterinarian or related expert.

2. Research Method

2.1. Type of Research

This research employs applied research. This type of research is aimed to find solutions to specific problems, ensuring that the research findings can be beneficial for individuals or group [11].

2.2. Location and Duration of Research

This research will be conducted at Mr. Doni's chicken farm, located in Wonokromo Village, Gondang District, Tulungagung Regency. The research will take approximately 4 months, from January to April 2024.

2.3. Research Stages

2.3.1. Literature Review

In the stage of literature review, the researcher collects relevant literature and information related to the research to be conducted. This involves searching for and exploring reliable sources that align with the research needs.

2.3.2. Data Collection

Data collection is conducted by obtaining a dataset from previous research available on the Kaggle website [12]. The dataset comprises 8,067 images of chicken feces categorized into four classes: 2,476 coccidiosis, 2,404 healthy, 562 Newcastle disease (NCD), and 2,625 salmonella. Chicken feces do not differ significantly across various chicken breeds, thus this dataset can be used to detect diseases in other chicken breeds based on feces images.

2.3.3. Data Anotation

In this process, each image will be annotated with bounding boxes and labels corresponding to the required classes around the objects in the image. We will use some classes, such as: "ncd" for objects detected with Newcastle disease, "coccidiosis" for objects detected with coccidiosis, "salmonella" for objects detected with Salmonella infection (pullorum disease), and "healthy" for objects detected as healthy. This process will be conducted using LabelImg. The results of the data annotation will be saved in .txt files containing the class and coordinates of the bounding boxes, which will be used as input for the model training process.

2.3.4. Data Pre-Processing

The images that initially formatted as jpg will be converted to png to maintain the image quality effectively. However, some images in the dataset are blurry. To address this issue, image sharpening will be applied using the image sharpen function in the PIL (Python Imaging Library) in Python. Image sharpening can improve the blurry images in the dataset. Next, the data will be divided into training data and validation data with ratios of 90:10 and 80:20 to compare the two. This step is conducted to evaluate how well the trained model detects infectious diseases in broiler chickens based on feces images with these data division. The subsequent step is to augment the training data to increase data variety and achieve better results during the training process. The augmentations include horizontal and vertical flips, as well as clockwise, counter-clockwise, and upside-down rotations.

Data augmentation can also address detection errors in YOLOv5 caused by a limited and less varied dataset [10].

2.3.5. Model Training

Before the model training process, pre-trained weights are used as a starting point to enhance the model's object detection capabilities with the provided data. The model to be implemented in the training process is YOLOv5s, chosen because it has fewer parameters and a smaller size, making it suitable for resource-constrained environments such as mobile devices. Then, the model will be modified with several scenarios to achieve the best results.

a. Modification 1

The modification focuses on the architecture of C3 module, which functions as a feature extractor from the input image. C3 module includes convolutional layers, bottleneck layers, and concatenation layers.

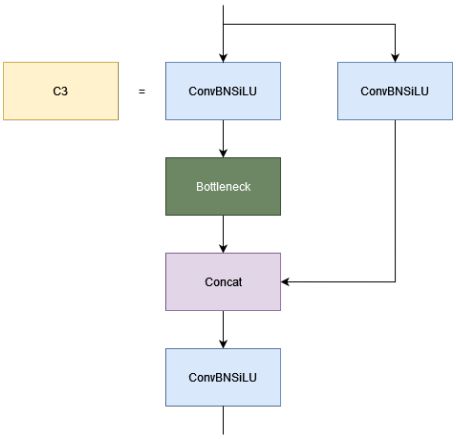


Figure 1. C3 Layer

In the existing setup, two convolutional layers are processed by the bottleneck layer and then passed to the concatenation layer, while another path bypasses the bottleneck layer and goes directly to the concatenation layer. The bottleneck layer reduces the number of parameters and accelerates computation without sacrificing model quality or performance. The modification involves adding a new bottleneck layer to the convolutional layer path that bypasses the bottleneck and goes directly to the concatenation layer. Then, the modified C3 module is integrated into the backbone of YOLOv5, replacing the previous C3 layer.

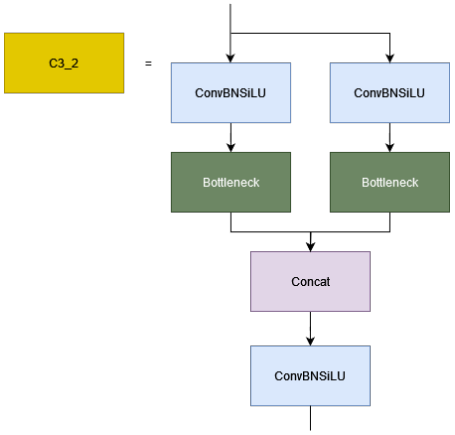


Figure 2. Modified C3 Layer

b. Modification 2

The modification targets the architecture of SPPF (Spatial Pyramid Pooling-Fast) module within the backbone of YOLOv5. The SPPF is designed to enhance the receptive field and separate important features using max pooling. The convolutional layers within the SPPF use SiLU (Sigmoid Linear Unit) as their activation function. SiLU provides smooth output, which can help to improve gradient flow during backpropagation and reduce vanishing gradient problems, enabling more expressive representations and smoother optimization landscapes.

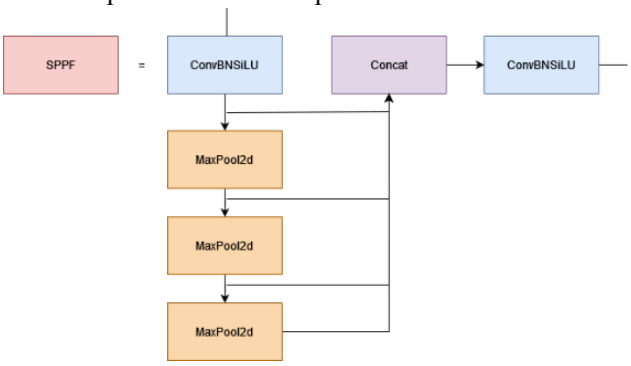


Figure 3. SPPF Layer

The modification is carried out by replacing SiLU activation in the convolutional layers with ELU (Exponential Linear Unit) activation. ELU reduces vanishing gradient issues by using identity for positive values and improves learning characteristics. ELU's negative values drive the average activation of units closer to zero, reducing computational complexity and accelerating the learning process.

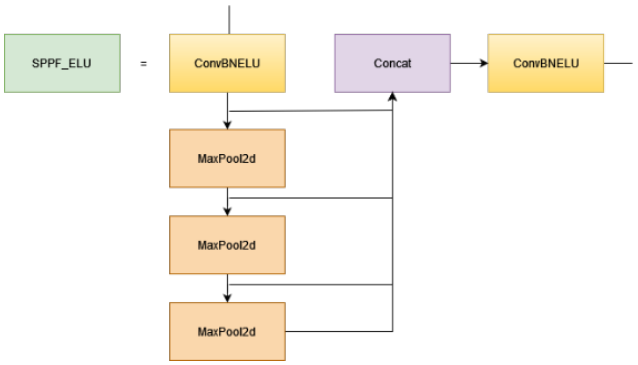


Figure 4. Modified SPPF Layer

c. Modification 3

In this modification, a CBAM (Convolutional Block Attention Module) layer is added to YOLOv5 architecture to replace C3 layer in backbone. CBAM is a module developed to enhance the ability of CNNs to extract more meaningful features by applying attention mechanisms in two main dimensions: channel and spatial. CBAM consists of two main components: CAM (Channel Attention Module) and SAM (Spatial Attention Module). CAM emphasizes key features, while SAM emphasizes the spatial location of those key features [13].

Feature extraction occurs through separate average pooling and max pooling operations. These features are processed separately through MLP networks, summed, and ultimately produce a feature vector. This operation is represented by the equation below:

$$Mc(F) = \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F)))$$

where σ is a sigmoid function used to map inputs to outputs continuously with values between 0 and 1. FFF represents the input feature map, and the MLP consists of two linear layers with ReLU as the activation function.

SAM produces a spatial attention map by connecting features generated by average pooling and max pooling within the channel dimension. The operation of SAM is represented by the equation below:

$$Ms(F) = \sigma(f7x7(Concat(AvgPool(F), MaxPool(F))))$$

where σ is the sigmoid function, f^7x7 represents a convolutional kernel size of 7 x 7, and concat is an operation that connects the processes of the average pooling and the max pooling.

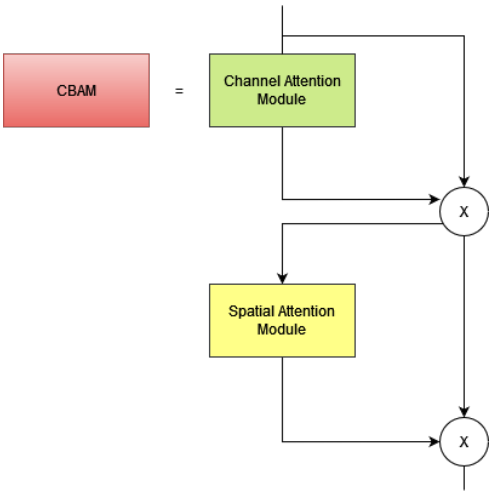


Figure 5. CBAM Module

d. Modification 4

In this modification, a CBAM (Convolutional Block Attention Module) layer is also applied within YOLOv5. Unlike Modification 3, where CBAM is placed in the backbone, this modification locates CBAM layer in the neck, immediately after the C3 layer. CBAM layer replaces the C3 layer to connect to the detection head with three different image scales. Adding CBAM layer after the C3 layer increases the number of modules in YOLOv5 from the previous 25 modules to 28 modules. This positioning aims to enhance the feature extraction process by leveraging the attention mechanism at a critical point before the final detection stages, potentially improving the model's accuracy and robustness in detecting objects at various scales.

e. Modification 5

This modification almost similar to modification 4 by adding a CBAM (Convolutional Block Attention Module) layer. However, in addition to place CBAM layer in the neck after the C3 layer, another CBAM layer is added to the backbone of YOLOv5, specifically after the C3 layer in the 8th module and before the SPPF layer. With this additional CBAM layer, the total number of modules in YOLOv5 increases from 28 to 29. This dual placement of CBAM layers—both in the backbone and neck—aims to enhance feature extraction and attention mechanisms, potentially leading to better object detection performance by improving the model's ability to focus on relevant features at multiple stages of the detection process.

f. Modification 6

This modification combines CBAM (Convolutional Block Attention Module) with C3 layer, creating a new layer called C3CBAM to replace C3 layer in backbone. In this modification, CBAM is applied within the bottleneck of C3 layer. The bottleneck typically involves two convolutional layers. After the input is processed by these two convolutional layers, the result is added to the value from the input that

bypasses the convolutional process.

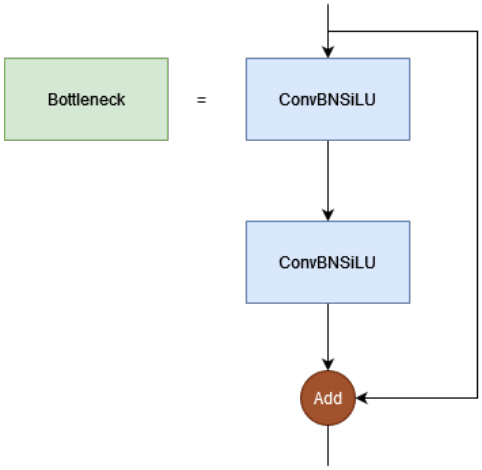


Figure 6. Bottleneck Module

In this new C3CBAM layer, after the input is processed by two convolutional layers, the resulting features are then processed by CBAM. After CBAM processes these features, the output is added to the value from the input that bypasses the convolutional and CBAM processes. This integration enhances the feature extraction process by leveraging the attention mechanism within the bottleneck, potentially leading to the better performance in object detection.

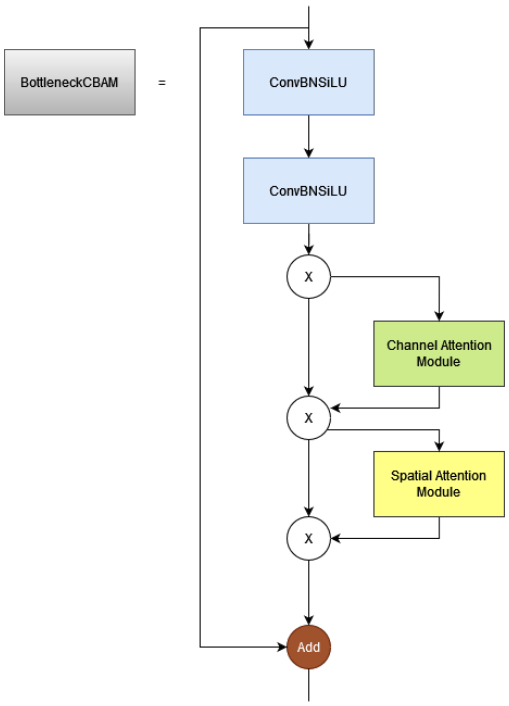


Figure 7. Modified Bottleneck Module

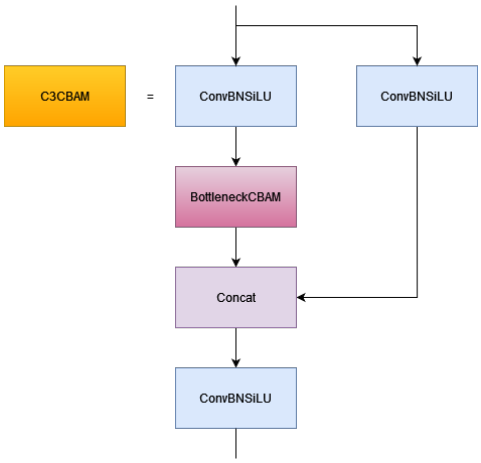


Figure 8. C3CBAM Layer

2.3.6. Model Evaluation

After training the model, the performance is evaluated using validation data that was previously divided into ratios of 90:10 and 80:20 for comparison. This is done to determine how well the trained model can directly detect infectious diseases in broiler chickens based on fecal images using these data divisions. The next step is measuring precision, recall, and mAP (mean average precision) based on the training results obtained.

Precision is the ratio of correctly predicted objects to the total predictions made by the model. The formula used to calculate precision is as follows.

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

Recall is the ratio of correctly predicted objects to the total actual results. The formula used to calculate recall is as follows.

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

Mean average precision (mAP) is used to measure how well the model can identify and position objects within images. The formula used to calculate mAP is as follows.

$$mAP = \frac{1}{n} \sum_{i=1}^{n} \quad AP_i \quad (3)$$

TP (true positive) is an object with actual positive data that the model correctly predicts as positive. TN (true negative) is an object with actual negative data that the model correctly predicts as negative. FP (false positive) is an object with actual negative data that the model incorrectly predicts as positive. FN (false negative) is an object with actual positive data that the model incorrectly predicts as negative. APi represents the average precision for each “i” object class. Average precision (AP) can be calculated by determining the precision and recall values at various threshold points and computing the area under the precision-recall curve.

3. Results and Analysis

3.1. Dataset Processing

The dataset that will be used for training must be processed first to achieve the better results.

3.1.1. Pre-Processing Data

The dataset that will be used in training is converted from JPG to PNG format. Then, image sharpening is performed using the image sharpen function in the Python Imaging Library (PIL). Image sharpening is used to enhance the contrast and lighting of an image. The image sharpen function uses a 3x3 kernel convolution matrix with the following values.

$$[-2\ -2\ -2\ -2\ 32\ -2\ -2\ -2\ -2\ ]$$

An example of an image before and after sharpening can be observed in Figure 2.

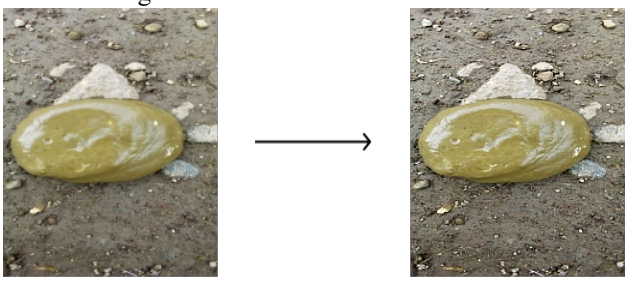


Figure 9. Before and After Sharpening Images

3.1.2. Split Dataset

The dataset is uploaded to the Roboflow website. Roboflow is a platform that facilitates the processing of datasets for classification or object detection. The dataset will be divided into training and validation data with ratios of 90:10 and 80:20.

3.1.3. Data Augmentation

After dividing the data, data augmentation will be performed. Data augmentation is used to increase the variety of dataset to achieve the better results. Data augmentation is also done using the Roboflow. The augmentations applied include horizontal and vertical flips, as well as rotations clockwise, counter-clockwise, and upside down. After augmentation, the number of training images increases from 6975 images to 20,555 images for the 90 : 10 division, and from 6152 images to 18,262 images for the 80 : 20 division.

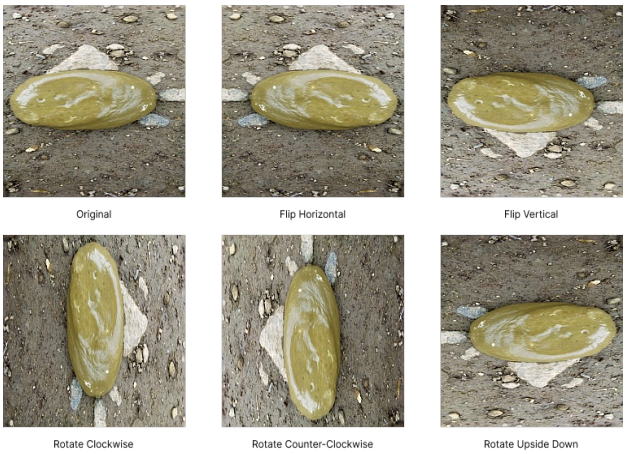


Figure 10. Data Augmentation

3.2. Model Training

Model training should be done using a device that is compatible with the running model. The GPU being used must be capable of handling heavy computational processes to accelerate training.

Table 1. Device Specifications

Hardware	Annotation
Operation System	Ubuntu 22.04 LTS
CPU	Intel(R) Core(TM) i7-9750H 2.60 GHz
GPU	NVIDIA GeForce GTX 1050
GPU Memory	4 GB
RAM	24 GB

Data training is performed by running the train.py file and configuring other settings such as the input dataset, the number of training iterations (epochs), pre-trained weights, the number of batches used during training (batch size), and model configurations. The entire model is trained using an image size of 224, 150 epochs, a batch size of 64, and Stochastic Gradient Descent (SGD) as the optimizer. The model used is YOLOv5s.

3.3. Model Evaluation

This evaluation model uses several dataset divisions to determine the best mAP value. The mAP value influences how accurate the model is in detecting a particular object in the image. The greater the map value obtained by the model, the more accurate and efficient the model is in detecting an object in the image, in this case it means that the model is very good for detecting infectious diseases suffered by broiler chickens through fecal images.

Meanwhile, the inference time value shows how long it takes the model to detect objects in an input image in milliseconds (ms). Testing was carried out on each model using sample data taken directly using a smartphone camera to obtain inference time values. The division of the dataset for training data and validation data respectively is in a ratio of

80:20 and 90:10.

The results obtained from dividing the dataset with a ratio of 80:20 can be observed in Table 4.3. There are 4 classes that can be detected by YOLOv5s model. In the coccidiosis class, YOLOv5s modification 5 model obtained the best results compared to other models by obtaining an mAP value of 90.9%. Meanwhile, the YOLOv5s modification 1 model obtained the lowest results in detecting coccidiosis by obtaining a mAP value of 79.9%. Based on these results, the modified YOLOv5s model 5 is very good for detecting feces infected with coccidiosis.

For the healthy class, YOLOv5s models with modifications 2 and 5 both achieved an mAP value of 92.3%. The lowest mAP value was obtained by YOLOv5s model with modification 1, at 67.6%. Based on these results, the YOLOv5s models with modifications 2 and 5 are equally effective at detecting healthy chicken feces, whereas the YOLOv5s model with modification 1 is very poor at detecting healthy chicken feces.

For the NCD class, YOLOv5s model with modification 2 achieved the highest mAP value of 86.4%, while the lowest mAP value was obtained by the YOLOv5s model with modification 1, at 50.6%. Based on these results, the YOLOv5s model with modification 2 is very effective at detecting Newcastle disease in chickens, whereas the YOLOv5s model with modification 1 is very poor at detecting Newcastle disease.

For the salmonella class, the YOLOv5s model with modification 5 achieved the highest mAP value of 90.2%, while the lowest mAP value was obtained by the YOLOv5s model with modification 1, at 57.8%. Based on these results, the YOLOv5s model with modification 5 is very effective at detecting pullorum disease in chickens, whereas the YOLOv5s model with modification 1 is very poor at detecting pullorum disease.

The mAP values for each class were then averaged to obtain the overall mAP value for all classes. The highest overall mAP value was obtained by YOLOv5s model with modification 2, at 89.6%, while the lowest overall mAP value was obtained by YOLOv5s model with modification 1, at 63.9%. The fastest model for object detection was the original YOLOv5s model, with an inference time of 16.5ms. Based on these mAP results, the YOLOv5s model with modification 2 is highly effective at predicting infectious diseases in chickens through fecal images.

Table 2. mAP values and inference time with 80:20 ratio of data division

Model (YOLOv5s)	mAP.50 Coccidiosis	Healthy	NCD	Salmonella	All Classes	Inference Time
Original	89.9%	91.8%	84.0%	89.5%	88.8%	16.5ms
Modification 1	79.9%	67.6%	50.6%	57.8%	63.9%	19.2ms
Modification 2	89.9%	92.3%	86.4%	89.8%	89.6%	16.6ms
Modification 3	89.6%	90.2%	81.8%	88.0%	87.4%	17.1ms
Modification 4	90.0%	91.5%	84.8%	89.4%	88.9%	20.2ms
Modification 5	90.9%	92.3%	84.7%	90.2%	89.5%	21.3ms
Modification 6	90.5%	91.2%	84.8%	90.1%	89.1%	21.6ms

The results obtained from the dataset division with a 90:10 ratio can be observed in Table 2. For the coccidiosis class, the original YOLOv5s model performed the best among the other models, achieving an mAP value of 92.2%. In contrast, YOLOv5s modification 1 achieved the lowest result in detecting coccidiosis with an mAP value of 82.4%. Based on these results, the original YOLOv5s model is very effective at detecting feces infected with coccidiosis.

For the healthy class, the original YOLOv5s model achieved the highest mAP value of 95.0%, while the lowest mAP value was obtained by YOLOv5s modification 1, with 69.1%. This indicates that the original YOLOv5s model is very good at detecting healthy chicken feces, whereas YOLOv5s modification 1 performs poorly in this task.

For the ncd class, YOLOv5s modification 6 achieved the highest mAP value of 83.8%, while the lowest mAP value was obtained by YOLOv5s modification 1, with 48.6%. Therefore, YOLOv5s modification 6 is very effective in detecting Newcastle disease in chickens, whereas YOLOv5s modification 1 is very poor at this detection.

For the salmonella class, the original YOLOv5s model achieved the highest mAP value of 91.1%, while the lowest mAP value was obtained by YOLOv5s modification 1, with 61.8%. This shows that the original YOLOv5s model is very good at detecting Pullorum disease in chickens, whereas YOLOv5s modification 1 is very poor at this detection.

Table 3. mAP values and inference time with 90:10 ratio of data division

Model (YOLOv5s)	mAP.50 Coccidiosis	Healthy	NCD	Salmonella	All Classess	Inference Time
Original	92.2%	95.0%	80.4%	91.1%	89.7%	16.6ms
Modification 1	82.4%	69.1%	48.6%	61.8%	65.5%	19.2ms
Modification 2	92.1%	94.8%	81.6%	90.2%	89.6%	16.6ms
Modification 3	91.1%	93.5%	76.9%	90.7%	88.1%	17.2ms
Modification 4	91.5%	94,4%	82.1%	90.8%	89.7%	20.4ms
Modification 5	91.5%	93.4%	79.8%	91.0%	88.9%	21.4ms
Modification 6	91.8%	94.7%	83.8%	89.8%	90.0%	21.6ms

The highest overall mAP was obtained by YOLOv5s modification 6, with 90.0%, while the lowest overall mAP was obtained by YOLOv5s modification 1, with 65.5%. The fastest models for object detection were the original YOLOv5s and modification 2, both achieving an inference time of 16.6ms.

3.4. The Implementation of YOLOv5 to Mobile Application

Model training should be done using a device that is compatible with the running model. The GPU being used must be capable of handling heavy computational processes to accelerate training.

3.4.1. Model Integration

For the implementation of YOLOv5 model into a mobile application, it is necessary to export the trained model file from the .pt format to the torchscript format. Torchscript optimizes the model for better performance on devices with limited resources, especially mobile devices. This facilitates the integration of YOLOv5 model into a mobile application locally without requiring an internet connection.

The system is built with several plugins to enable running the YOLOv5s model on a mobile application. These plugins include image_picker plugin to connect the mobile camera to the application and flutter_pytorch plugin for integrating the YOLOv5 model to perform object detection tasks on the mobile application.

3.4.2. System Testing

System testing was conducted directly at Mr. Doni's chicken farm during daylight conditions. The testing utilized a Samsung Galaxy A23 with 6GB RAM, 50MP camera resolution, and Snapdragon 695 chipset. The device was handheld, and images were captured as close as 30-40 cm to broiler chicken feces for detection. The results of the detection displayed images with bounding boxes indicating disease types and scores.

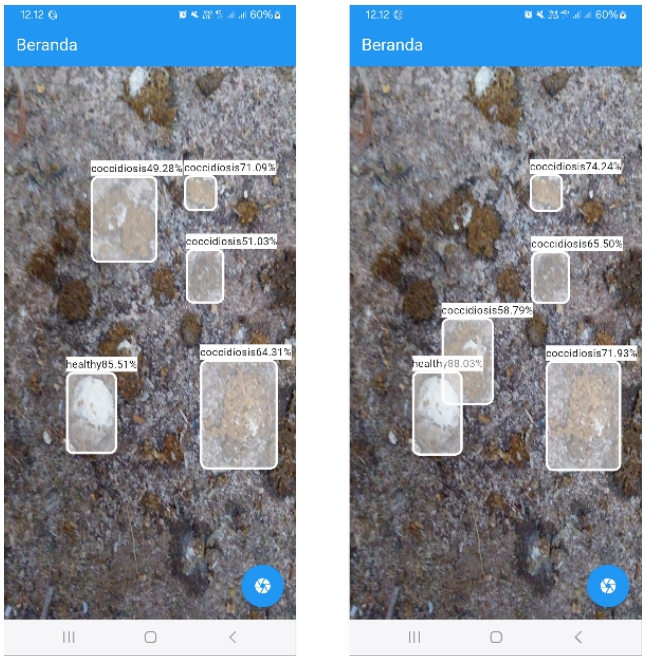
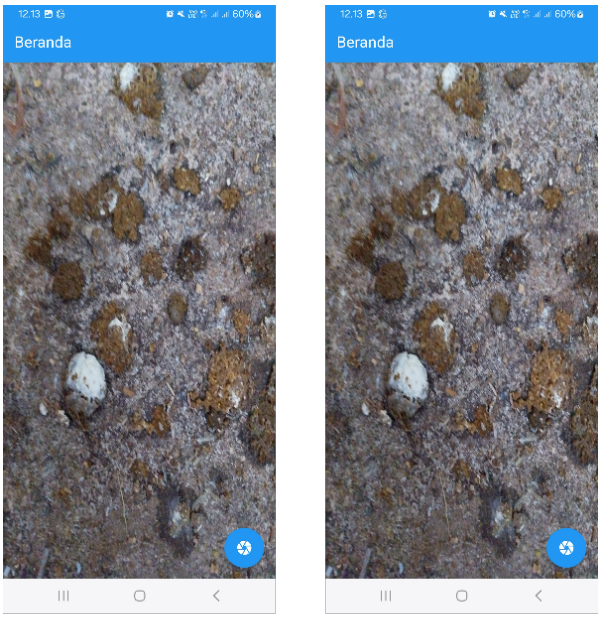


Figure 11. The testing of the original YOLOv5s

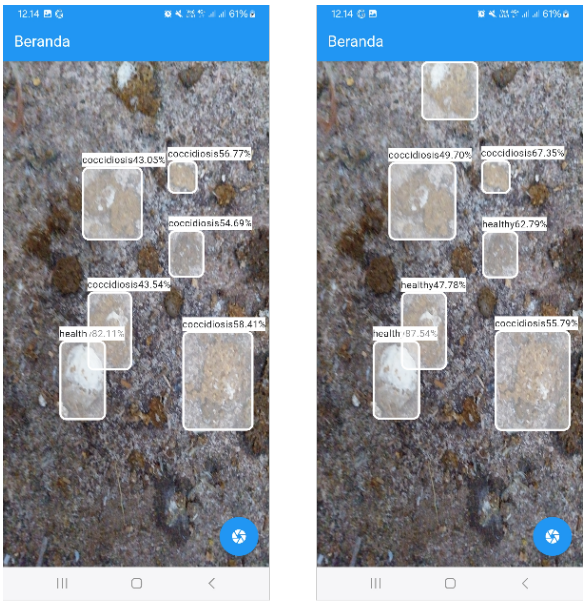
It can be observed in the testing of YOLOv5s original model in Figure 12 that it successfully detected objects suspected of being infected with disease. In the dataset division with a ratio of 90:10, it detected 5 objects indicating infection, both large and small. Meanwhile, in the dataset division with an 80:20 ratio, it also detected them well, although one object was located differently compared to the previous results.



Split 90 : 10 Split 80 : 20

Figure 12. The testing of YOLOv5s Modification 1

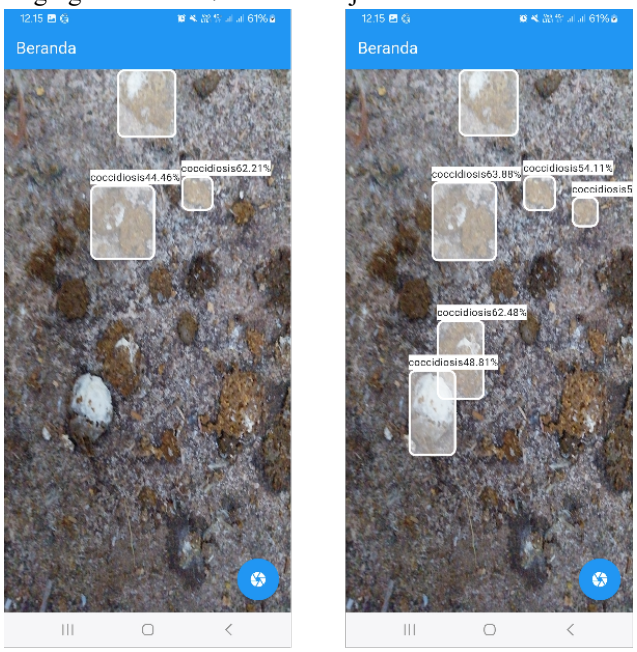
It can be observed in the testing of YOLOv5s modification 1 model in Figure 13. that it failed to detect objects suspected of being infected with disease in both dataset division with ratios of 90:10 and 80:20.



Split 90 : 10 Split 80 : 20

Figure 13. The testing of YOLOv5s Modification 2

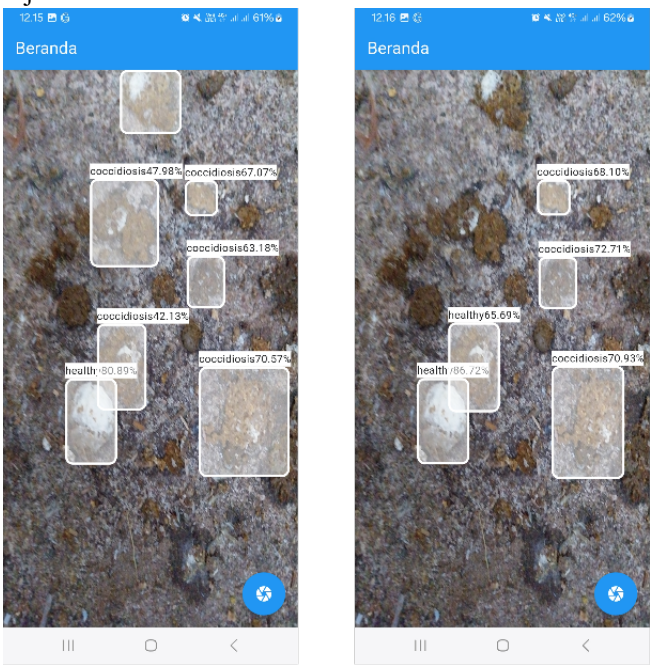
It can be observed in the testing of YOLOv5s modification 2 model in Figure 14. that it successfully detected objects suspected of being infected with disease. In the dataset division with a ratio of 90:10, it detected 6 objects that are suspected to be infected with disease, including both large and small objects. Similarly, in the dataset division with a ratio of 80:20, it also performed well, detecting an additional 1 object at a different location compared to the previous result, bringing the total to 7 detected objects.



Split 90 : 10 Split 80 : 20

Figure 14. The testing of YOLOv5s Modification 3

It can be observed in the testing of YOLOv5s modification 3 model in Figure 15 that it successfully detected objects suspected of being infected with disease. In the dataset divison with a ratio of 90:10, it detected 3 objects suspected to be infected with disease. Meanwhile, in the 80:20 dataset division, it detected 6 objects, including both large and small objects.



Split 90 : 10 Split 80 : 20

Figure 15. The testing of YOLOv5s Modification 4

It can be observed in the testing of YOLOv5s modification

4 model in Figure 16. that it successfully detected objects suspected of being infected with disease. In the dataset division with a ratio of 90:10, it detected 7 objects suspected to be infected with disease. Meanwhile, in the 80:20 dataset division, it detected 5 objects suspected to be infected with disease. Both results show detection of both small and large objects suspected of disease.

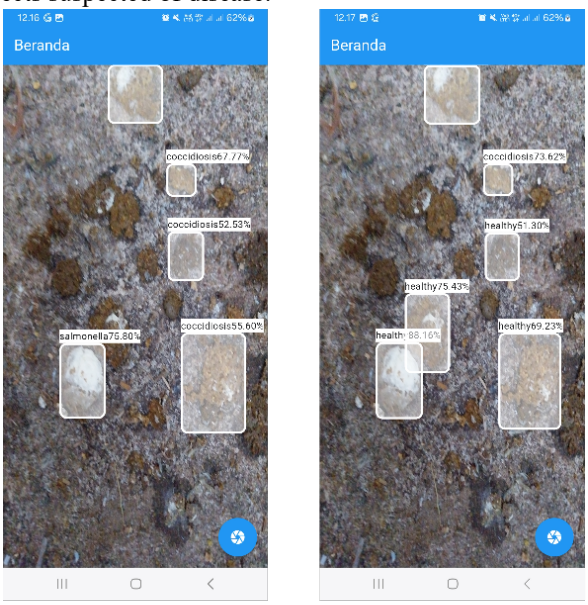


Figure 16. The testing of YOLOv5s Modification 5

It can be observed in the testing of the YOLOv5s modification 5 model in Figure 17 that it successfully detected objects suspected of being infected with disease. In the dataset division with a ratio of 90:10, it detected 5 objects suspected to be infected with disease. Meanwhile, in the 80:20 dataset division, it detected 6 objects suspected to be infected with disease. Both results show detection of both small and large objects suspected of disease.

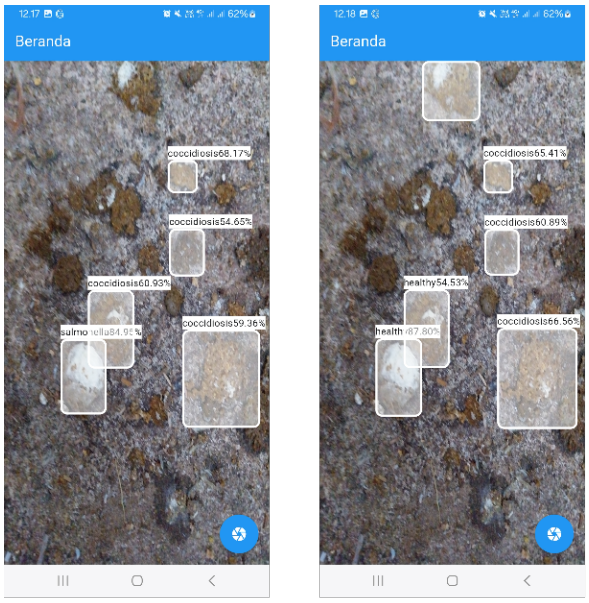


Figure 17. The testing of YOLOv5s Modification 6

It can be observed in the testing of YOLOv5s modification 6 model in Figure 18. that it successfully detected objects suspected of being infected with disease. In the dataset division with a ratio of 90:10, it detected 5 objects suspected to be infected with disease. Meanwhile, in the 80:20 dataset division, it detected 6 objects suspected to be infected with disease. Both results show detection of both small and large objects suspected of disease.

During the testing, two models were identified as capable of detecting the highest number of objects: the YOLOv5s modification 2 model with an 80:20 dataset division, and the YOLOv5s modification 4 model with a 90:10 dataset division. Both models were able to detect 7 objects suspected of being infected with disease, maintaining the same distances and objects, although there were differences in classes and scores between the two models.

4. Conclusion

Based on the previous research, the following conclusions can be drawn: (1) YOLOv5s modification 6 with a dataset division ratio of 90:10, integrating the Convolutional Block Attention Module (CBAM) with the C3 layer in the backbone, achieved the highest mAP value of 90.0% among other models. This indicates that the model has superior accuracy in detecting infectious diseases in broiler chickens through fecal images; (2) in direct testing on a mobile application, the YOLOv5s modification 2 model with an 80:20 dataset division and the YOLOv5s modification 4 model with a 90:10 dataset division were able to detect 7 objects suspected of disease at the same distance and objects, although there

were some differences in classes and scores between these two models; and (3) the YOLOv5s model exported to torchscript operated smoothly on Flutter-based mobile applications.

Recommendations for consideration and input in the improvement and development of this research include:

1. Using a larger dataset and ensuring balanced class distribution with higher-quality images.
2. For the future system development, consider using lighter models such as YOLOv5n for better performance on mobile platforms.
3. Explore other YOLO models like YOLOv8 or other CNN models for object detection in images.
4. Implement the model into IoT-based systems.

Acknowledgements

I would like to thank Dr. Dwiretno Istiyadi Swasono, ST., M.Kom. and Muhammad Arief Hidayat, S.Kom., M.Kom for their guidance, encouragement, and support throughout this research endeavor. I am grateful for the invaluable support provided by the Jember University throughout the research process. The facilities and resources offered were extremely helpful and have been instrumental in completing this research project successfully.

References

[1] Ahmed, G., Malick, R. A. S., Akhunzada, A., Zahid, S., Sagri, M. R., & Gani, A. (2021). An approach towards iot-based predictive service for early detection of diseases in poultry chickens. Sustainability (Switzerland), 13(23), 1–16. https://doi.org/10.3390/su132313396

[2] Rasyaf, M. (2006). Beternak Ayam Pedaging II. PT Penebar Swadaya. Jakarta

[3] Direktorat Jenderal Peternakan dan Kesehatan Hewan Kementerian Pertanian. (2022). Statistik Peternakan dan Kesehatan Hewan 2022/ Livestock and Animal Health Statistics 2022. Issn 2964-1047, 1, 1–276

[4] Kholil, M., Waspada, H. P., & Akhsani, R. (2022). Klasifikasi Penyakit Infeksi Pada Ayam Berdasarkan Gambar Feses Menggunakan Convolutional Neural Network. SINTECH (Science and Information Technology) Journal, 5(2), 198–204. https://doi.org/10.31598/sintechjournal.v5i2.1179

[5] Kim, H., Lee, W.-D., Jang, H.-K., Kang, M., & Kang, H.-K. (2022). The potential of non-movement behavior observation method for detection of sick broiler chickens. Journal of Animal Science and Technology, 65(2), 441–458. https://doi.org/10.5187/jast.2022.e105

[6] Machuve, D., Nwankwo, E., Mduma, N., & Mbelwa, J. (2022). Poultry diseases diagnostics models using deep learning. Frontiers in Artificial Intelligence, 5. https://doi.org/10.3389/frai.2022.733345

[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779–788. https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html

[8] Widyawati, W., & Gunawan, W. (2022). Early detection of sick chicken using artificial intelligence. TEKNIKA: JURNAL SAINS DAN TEKNOLOGI, 18(02), 136–141. http://jurnal.untirta.ac.id/index.php/ju-tek/

[9] Degu, M. Z., & Simegn, G. L. (2023). Smartphone based detection and classification of poultry diseases from chicken fecal images using deep learning techniques. Smart Agricultural Technology, 4(November 2022), 100221. https://doi.org/10.1016/j.atech.2023.100221

[10] Nafis Alfarizi, D., Agung Pangestu, R., Aditya, D., Adi Setiawan, M., & Rosyani, P. (2023). Penggunaan Metode YOLO Pada Deteksi Objek: Sebuah Tinjauan Literatur Sistematis. Jurnal Artificial Inteligent Dan Sistem Penunjang Keputusan, 1(1), 54–63. https://jurnalmahasiswa.com/index.php/aidanspk

[11] Irina, F. (2017). Metode Penelitian Terapan. Parama Ilmu. https://openlibrary.telkomuniversity.ac.id/pustaka/143081/metode-penelitian-terapan.html

[12] Allandclive. (2022). Chicken Disease Image Classification. Kaggle. https://www.kaggle.com/datasets/allandclive/chicken-disease-1

[13] Woo, S., Park, J., Lee, J. Y., & Kweon, I. S. (2018). CBAM: Convolutional block attention module. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11211 LNCS, 3–19. https://doi.org/10.1007/978-3-030-01234-2_1