Automatic detection of broiler's feeding and aggressive behavior using you only look once algorithm

Sri Wahjuni[1], Wulandari[1], Rafael Tektano Grandiawan Eknanda[1], Iman Rahayu Hidayati Susanto[2], Auriza Rahmad Akbar[1]
[1]Department of Computer Science, Faculty of Mathematics and Natural Science, IPB University, Bogor, Indonesia
[2]Department of Animal Production and Technology, Faculty of Animal Science, IPB University, Bogor, Indonesia

Article Info

Article history:

Received Nov 11, 2022
Revised Jan 27, 2023
Accepted Mar 10, 2023

Keywords:

Critical-condition notification
Object detection
Real-time monitoring
Remote observation
Smart coop

ABSTRACT

The high market demand for broiler chickens requires that chicken farmers improve their production performance. Production cost and poultry welfare are important competitiveness aspects in the poultry industry. To optimize these aspects, chicken behavior such as feeding and aggression needs to be observed continuously. However, this is not practically done entirely by humans. Implementation of precision live stock farming with deep learning can provide continuous, real-time and automated decisions. In this study, the you only look once version 4 (YOLOv4) architecture is used to detect feeding and aggressive chicken behavior. The data used includes 1,045 feeding bounding boxes and 753 aggressive bounding boxes. The model training is performed using the k-fold cross validation method. The best mean average precision (mAP) values obtained were 99.98% for eating behavior and 99.4% for aggressive behavior.

Corresponding Author:

Sri Wahjuni
Department of Computer Science, Faculty of Mathematics and Natural Science, IPB University
Kampus IPB Dramaga, Jl. Meranti W20 L5, Bogor, Jawa Barat, Indonesia
Email: my_juni04@apps.ipb.ac.id

1. INTRODUCTION

Chicken is a livestock commodity that has a large enough demand so that the broiler industry has a very high potential for increasing economic growth. Indonesia's market demand for broiler meat in 2020 reached 3,442,558 tons, greater than beef and buffalo with a demand of 269,603.4 tons [1]. The high market demand for this commodity means that broiler breeders need to improve their production performance. The research of [2] aims to increase the excellence competitiveness in broiler chicken business by decreasing the production costs. According to this research, the main component on the production cost structure is feed, around 70%.

The frequency of feeding and the amount of feed are considered in calculating the production costs. However, the welfare of broilers also needs to be considered. One indicator of the welfare level can be seen from the level of broiler mortality. Productivity increases with broiler welfare [3]. There are several problems related to the welfare of broilers, and one of the main problems is aggressive behavior. Increased aggressiveness of male broilers causes female broilers to experience injury and stress. Hens that are stressed and injured tend to be susceptible to disease and infection and even death [4]. The separation of aggressive chickens is a way to overcome aggressiveness [5]. The decision to feed and select aggressive chickens is made through continuous observations by experienced breeders. However, this is not practical to do when considering the large dimensions of the farm, it reduces the attention given to the condition of each chicken and requires a larger number of workers. The existence of precision livestock farming can help farmers monitor and control livestock productivity as well as livestock health and welfare in a continuous, real-time and automated manner [6]. The precision livestock farming method utilizes various technologies, one of which is image-based. The first step necessary to determine the feeding decisions, and the selection of aggressive chickens is to detect their feeding and aggressive behaviors.

Image-based technology using the deep learning method to achieve precise feeding has been applied by Khairunissa *et al.* [7] with the intention of detecting the movement of the poultry, with the goal of recognizing particular poultry habits. The research to detect the movement of birds in a free-range system uses the single shot multibox detector (SSD) architecture with a pretrained model from the common object in context (COCO) dataset. The precision of the model to detect the movement of the birds reached 60.4%. The third version of the you only look once (YOLOv3) object detection architecture included in the deep learning method has been applied to improve livestock welfare by detecting chicken behavior such as eating (feeding), fighting, and drinking. The accuracy of the feeding, fighting and drinking detections is 93.10%, 88.67% and 86.88%, respectively, obtained by the model [8].

In 2020, YOLO produced its fourth version which had an improved performance. In addition, YOLOv4 was efficiently developed so that conventional graphics processing units (GPUs) could obtain real-time, high-quality, and promising object detection results [9]. The research of [10] used the YOLOv4 architecture to count pears in real time. A trained model is able to achieve a mean average precision (mAP) on the test data of above 96%, and the inference speed reaches 37.3 frames per second (fps). With the aforementioned advantages, YOLOv4 is the best candidate for implementation in this study. The purpose of this study is to build an effective and efficient deep learning model to detect feeding behavior and aggressiveness in broilers by applying YOLO v4. Furthermore, it is hoped that the resulting model can be implemented in a smart coop monitoring system that provides notifications if unexpected conditions occur in the coop. The rest of this paper is organized. The closely related background areas are presented in section 2; in section 3 we explain the methods used in this research; and the results obtained are presented in section 4. The last section wraps up the overall paper and its conclusions.

2. RESEARCH METHOD

The following subsection will provide an overview of the you only look once version 4 (YOLOv4) algorithm implemented in this research. To provide a brief information on the chicken behaviors that are related to this research, the next subsection contains a short description of the feeding and aggressive behaviors. The research steps that guide this work are explained in the last subsection.

2.1. Overview of you only look once object version 4 (YOLOv4)

The YOLOv4 algorithm is the result of the development of the first YOLO version. This algorithm claimed to be a faster and more accurate state-of-the-art detector but it can be implemented on conventional GPUs, so it is widely used. Modern detection systems generally have 2 parts, the backbone and the head. The backbone section extracts features to improve accuracy [11]. The head section is used to predict the final classification and refine the bounding box [12]. YOLOv4 consists of the backbone: cross-stage partial (CSP) CSPDarknet53 [13], neck: spatial pyramid pooling (SPP) [14] and path aggregation network (PAN) [15], and head: YOLOv3. The new terms used in YOLOv4 are bag of freebies (BoF) and bag of specials (BoS). The BoF in object detection can be interpreted as data augmentations and semantic data distributions that may have a bias. This method tends to replace the training strategy with only a small increase in inference costs. BoS is a collection of postprocessing methods and plugin modules that enlarge the receptive field, introduce attention mechanisms, and strengthen feature integration capabilities; they add a small inference cost but can significantly increase accuracy. On the backbone, YOLOv4 provides BoF: CutMix [16] and mosaic data augmentation, DropBlock regularization [17], class label smoothing [18], and BoS: Mish activation [19], CSP connections and multi input weighted residual connections (MIWRC). In the detector section, YOLOv4 provides BoF: CioU-loss, cross mini-batch normalization (CmBN), DropBlock regularization, mosaic data augmentation, self-adversarial training, elimination of grid sensitivity, the use of multiple anchors for a single ground truth, optimal hyperparameters, and random training shapes, as well as BoS: Mish activation, SPP block, spatial attention module (SAM) block [20], PAN path aggregation block and distance intersection over union- non-maximum suppression (DioU-NMS).

In optimizing data classifications, YOLOv4 applies class label smoothing, which converts hard labels into soft labels. Specifically, the one-hot encoded training data ($\delta_{k,y}$) are transformed into a combination of the parameter divided by the number of classes (K) by multiplying the one-hot encoded labeling score minus 1. The class label smoothing method can be seen in (1).

$$q'(k) = (1 - \alpha) * \delta_{k,y} + \frac{\alpha}{K} \tag{1}$$

This formula can overcome the overfitted model training cases so that the model is more adaptive. YOLOv4 also adds DropBlock to its model to address the overfitting issue. The DropBlock method removes continuous regions from the feature map of a layer to reduce neural networks that are too large. The activation function used in YOLOv4 is a mish activation. Mish is a nonmonotomic function that can be defined in (2).

$$f(x) = x.\tanh(\varsigma(x)) \tag{2}$$

Functions $\varsigma(x)$ equal to $\ln(1 + e^x)$ or a softplus activation. The Mish activation function can overcome the significantly slowed training caused by the near-zero gradient. On the neck, SPP is applied with the advantage of allowing the input of images of various sizes. The size of the input image will be transformed by the SPP into a fixed size so that it can be fed to the classifier. In SPP, multilevel pooling is applied so that the model remains robust on deformative objects.

2.2. Feeding and aggressive behaviors of broiler chicken

Behavior is derived from the offspring of its predecessors and the impact of the environment on its genotype. There are some behaviors that persist against environmental changes, which can be said to be a fixed pattern of action. An ethogram is a behavioral profile, a catalog of the major fixed action patterns of a species [21]. Chicken behavior is summarized in an ethogram based on [22] and can be seen in Table 1.

Table 1. Chicken feeding behavior ethogram and aggressive behavior [22], [23]

Behavior	Definition
Feeding	Seen as a chicken swallow food with its head extending to the place to eat
Fights	Two chickens face each other, with their necks and heads at the same height and kick
Pecks	One chicken raises its head and violently pecks the body of the other chicken
Wing-flapping	Chicken wings spread and flap on the side or above the body

Chicken feeding behavior occurs in a cycle, with a tendency to act at certain photoperiods called diurnal cycles according to [24]. Chickens usually eat at photoperiods of six to eight hours, and the peak usually occurs in the morning. However, in broiler chickens, the commercial rearing process is exposed to continuous light so that it has a photoperiod of 24 hours or 23 hours to maximize food intake and growth rates. These data are based on the statement of [25] referred to by [26]. The feeding intention of broilers is controlled by the satiety mechanism rather than the starvation mechanism. This causes broilers to spend more time eating than in other activities, as long as food is still available. Broiler chickens will stop eating when they reach their maximum physical capacity [27].

Aggression of broiler chickens can be driven by social rank, group size in the coop and hunger. In the research of [28], the precision feeding method was tested by adjusting the food needs of each broiler based on body weight. The approach involves building a feeding station in the coop that can only be entered by one chicken. When the chicken enters, the tool will provide a portion of food based on a certain meal time. When the feeding time is up, the chicken inside the station is automatically removed. This approach causes the broiler chicken feeding duration to be short, causing aggressiveness in the chickens. In addition, fluctuations in social ranking are rare. Social ranking fluctuations are indicated by the frequency of the appearance of aggressive behavior. When social ranking is stable, aggressive behavior will decrease and is replaced by threat behavior that tends to be less destructive. However, the larger the group size in a coop, the longer it takes to achieve social stability. Note that the group size reaches its saturation point at a certain group size. When the density and group size are too large, the broilers tend to choose to compete for resources, thereby reducing the frequency of the aggressive behavior [29].

2.3. Research steps

There are five steps that guide this research, they are, data collection, data preprocessing, data division, model training and model evaluation. These steps can be seen in the general block diagram in Figure 1. Detail of each step is explained in the following paragraphs.

2.3.1. Data acquisition

Data used in this research were recorded from flocks in the field section of block B poultry unit, Faculty of Animal Science, IPB University. Data acquisition was performed from 18 August 2021 until

13 September 2021. The video recording is in h264 format with a size of 1,280×720 pixels using a speed of 25 fps.

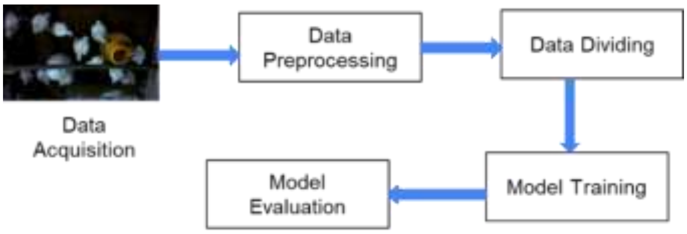


Figure 1. Research steps of broiler behavior detection

2.3.2. Data preprocessing

This stage produces the data that are used for model development. The data are obtained from videos that are converted into a JPG format image file for every single frame. The converted data are selected based on the object activity that is relevant to the object detection classes. The object detection classes are eaten and aggressive. The aggressive class is a combination of several behaviors that show aggressive habits, such as fights, pecks and wing-flapping, as listed in Table 1.

To balance the classes, random augmentation is performed. Data augmentation is an approach to overcome overfitting from the root of the problem, namely, the training dataset. This augmentation adds manipulated data to the training dataset to increase its numbers. The addition of these data can be called oversampling. The augmentation that can be applied is a geometric augmentation, because according to research by [30], there was a 3% increase in the performance of the CIFAR-10 classification using geometric augmentation. Random augmentation was performed with a rotation of 1° to 180°, and the result of random rotation was transformed to a reflection about the *y*-axis.

2.3.3. Data dividing

The model training method uses the cross-validation k–fold method. Cross validation k-fold is a method of evaluating and comparing algorithms by dividing K parts of the data in a balanced manner, where in the K-th iteration, the K data section will be used as validation data and the other data sections will be used as training data. The application of the k-fold cross validation method is generally for three contexts, performance estimation, model selection and model parameter setting [31]. The data are divided by a ratio of 80:20, which means that 80% of the dataset will be used for training data, and 20% will be used for test data.

2.3.4. Model training

The model will be trained with the configuration in Table 2. The configuration parameters used are width x height, subdivision and learning rate. The width × height parameter is the width and height that will be inputted to the model. The subdivision parameter is the number of fractions of the batch processed by the GPU in the model building process. The number of images processed in one iteration is referred to as a batch. The subdivision is the number of batch fractions. In this study, the batch parameter is set to 64. If the subdivision parameter is 32, the image or data to be processed by the GPU are two images multiplied by 32 parts, and this image is processed in parallel.

Table 2. Model training parameter configuration

Scheme	Width × Height	Subdivision	Learning rate
A	128×128	64	0.001
B	416×416	64	0.001
C	416×416	32	0.01
D*	416×416	32	0.001

*Research scheme D uses the standard configuration by Bochkovskiy *et al.* [9]

2.3.5. Model evaluation

The detection model in this study will be evaluated using the metrics intersection over union (IoU), precision, recall and mAP. The IoU metric is the result of dividing the overlap area between the detected ground-truth and bounding box with the union area of the two bounding boxes. The IoU formula can be seen in (3).

$$IoU = \frac{area\ of\ overlap}{area\ of\ union} \quad (3)$$

IoU values can be used to express the classification of true positives (TPs), false-positives (FPs), and false negatives (FNs). The IoU threshold value used for object detection is 50%. The threshold values will be used to determine TP, FP and FN. If the IoU is more than the threshold, then it is classified as TP. If the IoU is less than the threshold, it is classified as FP. Classification FN is if the IoU is equal to 0 [32].

This classification is useful for finding the precision and recall values. The precision value is a measure to show the model's ability to identify relevant objects according to the object detection results. The recall value is a measure to show the model's ability to find objects that match the ground truth. If defined in a formula, Precision (Pr) and Recall (Rc) can be calculated in (4) and (5). In (4) and (5), it is assumed that in the dataset, there are *G* ground truths, and the model produces *N* bounding box predictions. In the prediction results, there are *S* bounding boxes that successfully predict the ground truths.

$$Pr = \frac{\sum_{n=1}^{S} TP_n}{\sum_{n=1}^{S} TP_n + \sum_{n=1}^{N-S} FP_n} = \frac{\sum_{n=1}^{S} TP_n}{semua\ deteksi} \quad (4)$$

$$Rc = \frac{\sum_{n=1}^{S} TP_n}{\sum_{n=1}^{S} TP_n + \sum_{n=1}^{G-S} FN_n} = \frac{\sum_{n=1}^{S} TP_n}{semua\ ground\ truths} \quad (5)$$

The average precision (AP) metric is the value of the area under the curve between Pr and Rc. This curve summarizes the trade-off between precision and recall determined by the confidence level of the bounding box. The detection model can be said to be good when the level of confidence decreases, and the precision and recall remain high. The AP formula can be seen in (6).

$$AP = \frac{1}{N} \sum_{n=1}^{N} Pr_{interp}(R_r(n)) \quad (6)$$

In (6), $Pr_{interp}$is generated from the precision and recall graph that has been interpolated because the graph is still in monotonic form. The value $R_r(n)$is the recall value in the interpolation of N points. The formula $R_r(n)$can be seen in (7).

$$R_r(n) = \frac{N-n}{N-1}\ n = 1, 2, \dots, N \quad (7)$$

The *AP* scores are obtained in individual classes. To find the whole class, we need to calculate the *mAP*. The *mAP* formula can be seen in (8). The value $AP_i$is the average precision value in the *i*-th class, and C is the total existing class.

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i \quad (8)$$

3. RESULTS AND DISCUSSION

3.1. Data preprocessing

125,893 images are obtained from the conversion. The images are annotated in the form of a bounding box along with a class label. From the total converted images, selections need to be made because there are frames that do not show any relevance to the class to be detected. A total of 4,711 images were used for annotation. The images are annotated in the form of a bounding box along with a class label. Image annotation was performed using LabelImg version 1.8.1. In Figure 2, an example of the LabelImg display in the annotation process is shown. The aggressive class is a combination of behaviors derived from aggressive habits because the specific behavior of the aggressive habits shows a small frequency. Chickens are fed ad libitum, and the availability of feed in the coops is sufficient for all the chickens so that it does not cause a significant frequency of aggressive behavior due to lack of competition [28]. Even though the aggressive class has been combined with behavior, the number of bounding box annotations is still smaller than the eating behavior. The bounding box annotations before augmentation were 7,857 boxes with eating labels and 739 boxes with aggressive labels. The YOLO bounding box format used can be seen from the data pieces in Table 3. Figure 3 shows an example of the annotation results from each class. Figure 3(a) shows an example of annotation result for Feeding class, Figure 3(b) for Fight class, Figure 3(c) for Peck class, and Figure 3(d) for Wing-Flapping class. The classification follows [22] and [23]. In this research, due to limited data, class Fight, Peck and Wing-Flapping are grouped into one class, that is Aggressive class.

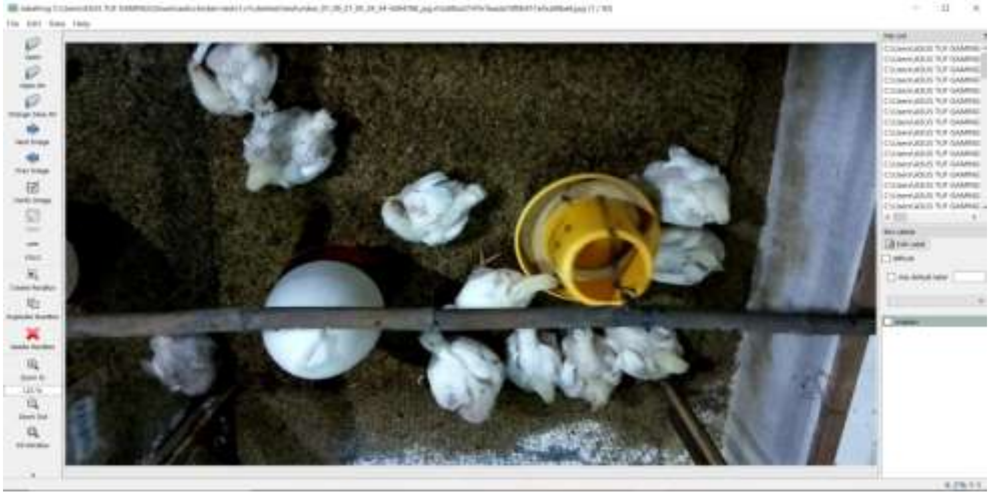


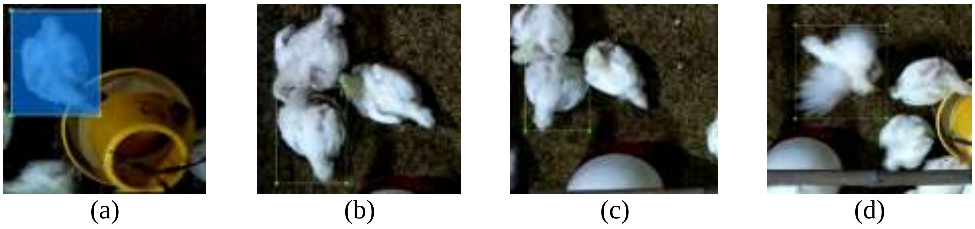
Figure 2. LabelImg window on the annotation process

Table 3. Sample data with YOLO bounding box format

Image name	c	X	Y	h	w
video_01_09_21….h2646	1	0.551172	0.284722	0.107031	0.258330
video_01_09_21….h26419	1	0.551953	0.282639	0.102344	0.237500
video_01_09_21….h26442	1	0.572656	0.645833	0.129688	0.175000
video_01_09_21….h26442	1	0.762109	0.554167	0.094531	0.211111

with:
Image name: Annotated file/image name
c: class of bounding box annotation with 1 as “eat” and 0 as “aggressive”
X: center coordinate of the bounding box on the X axis
Y: center coordinate of the bounding box on the Y. axis
h: width of the bounding box
w: height of the bounding box

The X, Y, h and w values are *floats* because they are the ratio of the width and height of the image. Therefore, they can practically change the size of the architecture of the input image for the network. This makes it easy to adjust to data availability.



(a) (b) (c) (d)

Figure 3. Feeding, (a) feeding, (b) fight, (c) pecks, and (d) wing-flapping dataset annotations

To balance between the classes, the classes with small occurrence frequencies will be augmented, and those with large occurrences will be deleted until the number is close to the other classes. Images that have the aggressive class will be augmented. For images that have two concurrent classes (aggressive class and eating class), the feeding class will be set aside before being augmented. The aggressive class is added as much as the image needs to balance according to the average class appearance across the image. Next, the training data and test data were distributed randomly, with the number of bounding boxes listed in Table 4.

Table 4. Number of bounding boxes of training and test data

Class name	Training data	Test data
Eat	823	222
Aggressive	686	167

3.2. Model training

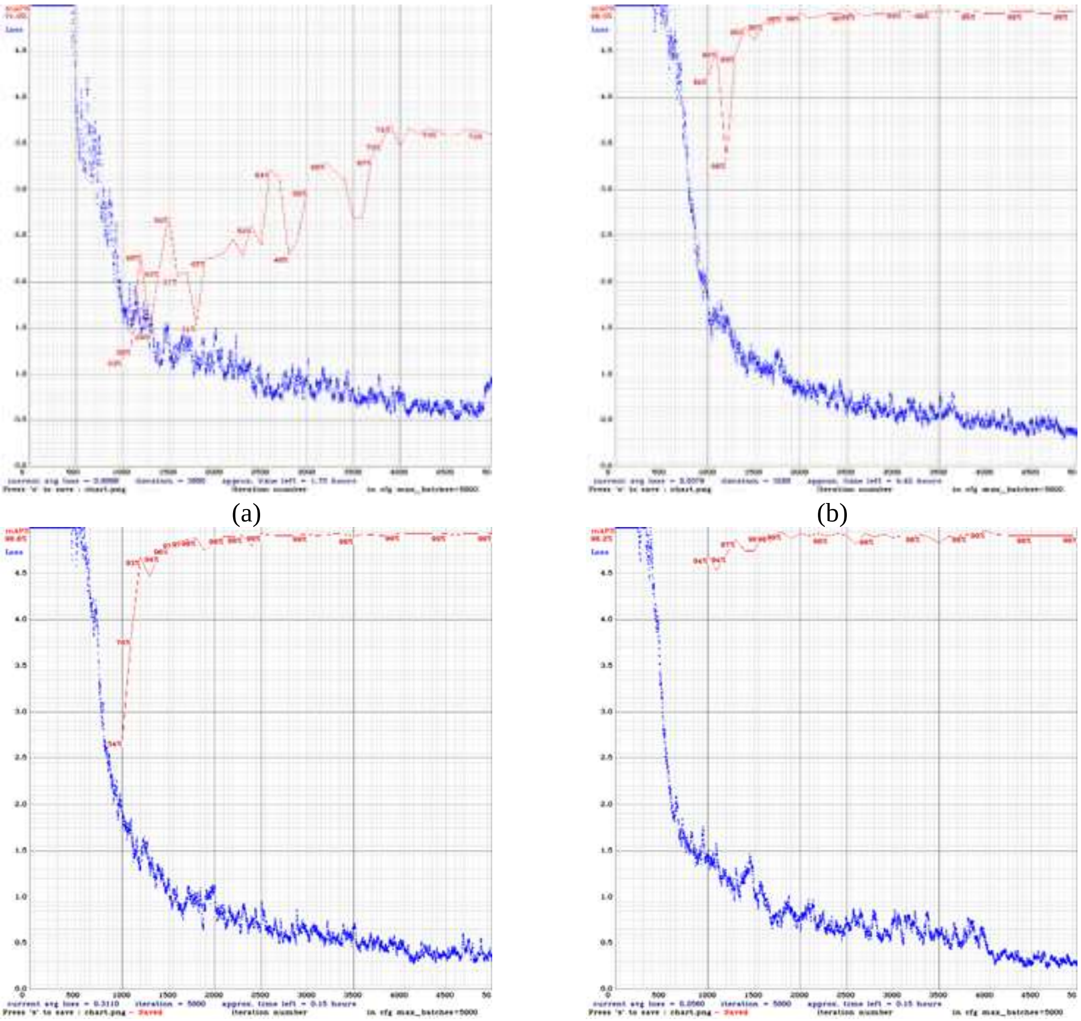
The model training was implemented using a laptop with an AMD Ryzen 5 3550H CPU with Radeon Vega Mobile Gfx (8 Core), 8 GB RAM, connected to Google Collaboratory using a 12 GB Nvidia K80 GPU on its backend server. Prior to training, the training data and validation data are uploaded to the Roboflow web application. The data already stored in the web application are exported and used in the Google Collaboratory notebook file.

Figure 4 shows a graph of each training model. The blue line (the lower line) shows the *loss* value for each iteration, and the red line (the upper line) shows the mAP value. Each subfigure using configuration as listed in Table 3. A graphic comparison between Scheme A (Figure 4(a)) and Scheme B (Figure 4(b)), with different input sizes of 128×128 and 416×416, shows that Scheme B, with a larger input size, supports the model for faster convergence. The loss value at size 128×128 experiences greater oscillation. In addition, Scheme A is overfitting in iterations above 4,500 because the *loss* value has increased from before. A comparison of the graph between Scheme B (Figure 4(b)) and Scheme D (Figure 4(c)) when wanting to compare the effect of subdivision on training, shows that the loss curve does not show a significant difference. This shows that the subdivision parameters do not affect the training performance. However, it should be noted that the subdivision is the number of data fragments that will be entered into the GPU process; the smaller the subdivision value is, the more GPU memory will be used. Changes in the subdivision parameters will be relevant if the research uses a conventional GPU. A comparison of Scheme D (Figure 4(c)) and Scheme C (Figure 4(d)) when looking at the effect of the learning rate on the training performance shows that Scheme D with a smaller learning rate can converge more quickly. The average loss value is below 0.5 in 3,500 iterations in Scheme D, while in Scheme C, the average loss value is below 0.5 in 4,000 iterations. During model training, a graph is generated between the loss value and the number of iterations and the mAP value after the 1,000th iteration. Model training is carried out until the 5,000th iteration. Samples were taken at the first fold in each model to compare the training performance.

3.3. Model evaluation

Table 5 shows the training results of each model and the five folds. Of the four training schemes, the largest mAP value reached 99.69% resulting from the input configuration of 416×416, subdivision of 32, and learning rate of 0.001 (Scheme D). The mAP value of this model is obtained from the results of the 4-fold average of the eating and aggressive mAP values in the scheme. While the smallest mAP value is generated from the 128×128 input configuration, the subdivision is 64, and the learning rate is 0.001 (Scheme A). In scheme A, the highest mAP model value is 73.95%. The size of the input is the most significant factor in the performance of the object detection. This is related to the purpose of YOLOv4 and the use of CSPDarknet53 as a backbone to increase the receptive area so that the architecture can detect more accurately. The larger the input image size is, the better the object detection, especially against small targets [33].

The subdivision parameter has no significant effect on the object detection performance. When compared, Scheme B and Scheme D, which have the same input size and learning rate but different subdivisions, give the results of the mAP model value with a difference of 0.02%. This is in accordance with the statement of [9] that the mini-batch size or the division between batches and subdivisions does not have a significant effect on the object detection performance and only affects the estimated training time. The results in Table 5 also show that the learning rate parameter does not provide a significant change in performance in this research. This is shown by schemes C and D, which only have a slight difference in the mAP model value. An example of the model detection results of the four schemes can be seen in Figure 5. Each subfigure shows the example of model detection result for each scheme as stated in Table 3. The green box indicates the bounding box of the ground truth, while the blue box indicates the bounding box of the prediction results. In Figure 5(a), which uses small size image (128×128), there is no prediction result bounding box. The prediction result bounding box appears in Figure 5(b), Figure 5(c) and Figure 5(d), which use image size of 416×416. Figure 5(a) is different from Figure 5(b) in the image size, while the Figure 5(c) is different from Figure 5(d) in the learning rate.



(a)

(b)

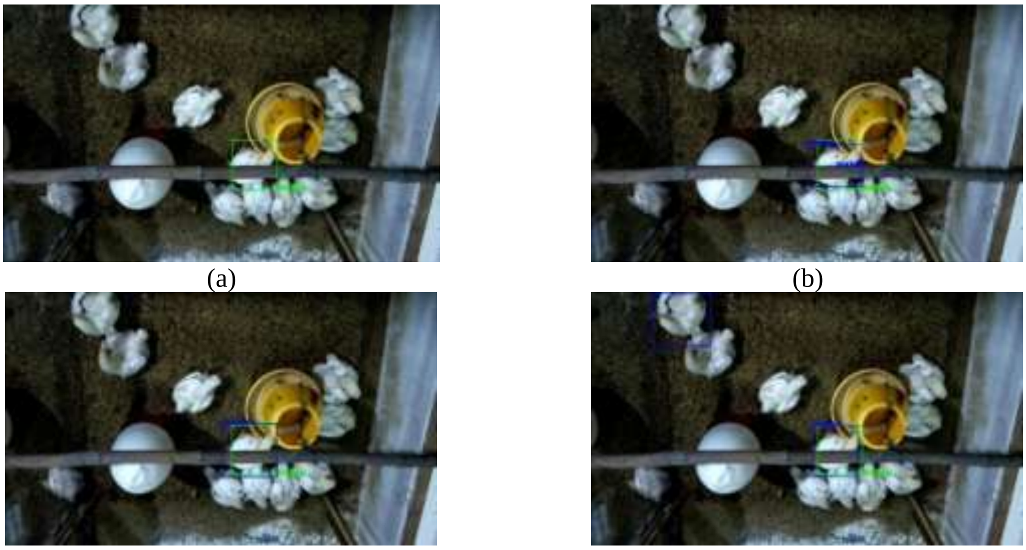
(c)

(d)

Figure 4. Loss and iteration graph of first fold model training for, (a) Scheme A: 128×128–64–0.001, (b) Scheme B: 416×416–64–0.001, (c) Scheme D: 416×416–32–0.001, and (d) Scheme C: 416×416–32–0.01

Table 5. Modeling results with 5-fold cross validation

Scheme	Fold	mAP of Behavior		mAPModel	A-IoU
		Aggressive	Feeding		
A	1	71.68%	76.23%	73.95%	31.83%
	2	65.92	72.83%	69.38%	36.16%
	3	62.14%	72.93%	67.54%	35.91%
	4	66.55%	75.11%	70.83%	36.24%
	5	65.49%	74.96%	70.23%	36.00%
B	1	98.09%	98.74%	98.41%	84.18%
	2	99.40%	99.92%	99.66%	84.90%
	3	99.40%	99.95%	99.67%	82.97%
	4	99.42%	97.83%	98.63%	83.94%
	5	96.83%	98.91%	97.87%	81.86%
C	1	99.16%	99.16%	99.16%	79.39%
	2	97.79%	99.46%	98.63%	80.05%
	3	99.40%	99.94%	99.67%	83.41%
	4	98.20%	99.96%	99.08%	86.32%
	5	99.40%	99.91%	99.66%	84.26%
D	1	98.52%	98.76%	98.64%	85.58%
	2	98.63%	98.79%	98.71%	79.93%
	3	99.44%	98.43%	98.93%	83.76%
	4	99.39%	99.98%	99.69%	81.52%
	5	99.40%	99.97%	99.69%	83.41%



(a) (b)

(c) (d)

Figure 5. Bounding box comparison between the ground truth and model prediction for, (a) Scheme A: 128×128–64–0.001, (b) Scheme B: 416×416–64–0.001, (c) Scheme C: 416×416–32–0.01 and (d) Scheme D: 416×416–32–0.001

4. CONCLUSION

This research was successful in developing a model for detecting the feeding and aggressive behavior of broiler chickens using YOLOv4. In model training with a 128×128–64 subdivision –0.001 learning rate scheme, the highest mAP was 71.68% and 76.23% for aggressive and feeding behavior, respectively. The model trained with a 416×416–64 subdivision –0.001 learning rate scheme obtained a higher mAP than the previous scheme. In this scheme, the highest mAP for the aggressive behavior was 99.40%, and the highest mAP for the feeding behavior was 99.95%. The third scheme trained the model using the 416×416–32 subdivisions –0.01 learning rate parameters. This scheme produced the highest mAP of 99,40% for aggressive behavior and 99.94 for feeding behavior. The last scheme used a parameter value of 416×416–32 subdivisions –a 0.001. The learning rate reached 99.39% as the highest mAP of the aggressive behavior and 99.98% for the feeding behavior. Scheme C produced the highest average IoU (A-IoU), with similar mAP values for both behaviors. This study shows that the size of the image has a significant effect on the mAP value. However, the effect of the subdivision parameter values and the learning rate still needs to be studied more deeply.

ACKNOWLEDGEMENTS

The authors would like to thank The Department of Computer Science, IPB University for facilitating the work. We also thank to The Faculty of Animal Science, IPB University for the field facility.

REFERENCES

[1] M. Yuwono, “Farm in numbers 2022,” *Central Bureau of Statistics*, pp. 1–121, 2022.

[2] S. Rahardjo, M. P. Hutagaol, and A. Sukmawati, “Strategy to growth an excellence competitiveness (A case study : Poultry company in Indonesia),” *International Journal of Scientific and Research Publications*, vol. 7, no. 11, pp. 593–598, 2017.

[3] S. Vetter, L. Vasa, and L. Ózsvári, “Economic aspects of animal welfare,” *Acta Polytechnica Hungarica*, vol. 11, no. 7, pp. 119–134, 2014, doi: 10.12700/aph.11.07.2014.07.8.

[4] S. T. Millman, “The animal welfare dilemma of broiler breeder aggressiveness,” *Poult. Perspect. Newsl. Coll. Agric. Nat. Resour.*, vol. 4, no. 1, pp. 1–6, 2002.

[5] S. A. Queiroz and V. U. Cromberg, “Aggressive behavior in the genus Gallus sp,” *Revista Brasileira de Ciencia Avicola*, vol. 8, no. 1, pp. 1–14, 2006, doi: 10.1590/S1516-635X2006000100001.

[6] J. Schillings, R. Bennett, and D. C. Rose, “Exploring the potential of precision livestock farming technologies to help address farm animal welfare,” *Frontiers in Animal Science*, vol. 2, 2021, doi: 10.3389/fanim.2021.639678.

[7] J. Khairunissa, S. Wahjuni, I. R. H. Soesanto, and W. Wulandari, “Detecting poultry movement for poultry behavioral analysis using the multi-object tracking (MOT) algorithm,” *Proceedings of the 8th International Conference on Computer and Communication Engineering, ICCCE 2021*, pp. 265–268, 2021, doi: 10.1109/ICCCE50029.2021.9467144.

[8] J. Wang, N. Wang, L. Li, and Z. Ren, “Real-time behavior detection and judgment of egg breeders based on YOLO v3,” *Neural Computing and Applications*, vol. 32, no. 10, pp. 5471–5481, 2020, doi: 10.1007/s00521-019-04645-4.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: optimal speed and accuracy of object detection,” *arXiv > Computer Science > Computer Vision and Pattern Recognition*, 2020, doi: 10.48550/arXiv.2004.10934.

[10] A. I. B. Parico and T. Ahamed, “Real time pear fruit detection and counting using yolov4 models and deep sort,” *Sensors*, vol. 21, no. 14, 2021, doi: 10.3390/s21144803.

[11] Y. Chen, T. Yang, X. Zhang, G. Meng, X. Xiao, and J. Sun, “DetNAS: Backbone search for object detection,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[12] J. Guo *et al.*, “Hit-detector: Hierarchical trinity architecture search for object detection,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 11402–11411, 2020, doi: 10.1109/CVPR42600.2020.01142.

[13] 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 Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, vol. 2020-June, pp. 1571–1580, 2020, doi: 10.1109/CVPRW50498.2020.00203.

[14] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8691 LNCS, no. PART 3, pp. 346–361, 2014, doi: 10.1007/978-3-319-10578-9_23.

[15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “PANet: path aggregation network for instance segmentation,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2019, pp. 8759–8768, doi: 10.1109/cvpr.2018.00913.

[16] S. Yun, D. Han, S. Chun, S. J. Oh, J. Choe, and Y. Yoo, “CutMix: Regularization strategy to train strong classifiers with localizable features,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2019-October, pp. 6022–6031, 2019, doi: 10.1109/ICCV.2019.00612.

[17] G. Ghiasi, T. Y. Lin, and Q. V. Le, “Dropblock: A regularization method for convolutional networks,” *Advances in Neural Information Processing Systems*, vol. 2018-December, pp. 10727–10737, 2018.

[18] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 2016, vol. 2016-Decem, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.

[19] D. Misra, “Mish: A self regularized non-monotonic activation function,” *arXiv > Computer Science > Machine Learning*, 2019, doi: 10.48550/arXiv.1908.08681.

[20] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: convolutional block attention module,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19, doi: 10.1007/978-3-030-01234-2_1.

[21] I. J. H. Duncan, “The ethogram of the domesticated hen,” *The Laying Hen and its Environment*, pp. 5–18, 1980, doi: 10.1007/978-94-009-8922-1_2.

[22] A. B. Webster and J. F. Hurnik, “An ethogram of white leghorn-type hens in battery cages,” *Canadian Journal of Animal Science*, vol. 70, no. 3, pp. 751–760, 1990, doi: 10.4141/cjas90-094.

[23] S. McGary, I. Estevez, and E. Russek-Cohen, “Reproductive and aggressive behavior in male broiler breeders with varying fertility levels,” *Applied Animal Behaviour Science*, vol. 82, no. 1, pp. 29–44, 2003, doi: 10.1016/S0168-1591(03)00038-8.

[24] C. J. Savory and B. O. Hughes, “Behaviour and welfare,” *British Poultry Science*, vol. 51, no. SUPPL. 1, pp. 13–22, 2010, doi: 10.1080/00071668.2010.506762.

[25] P. D. Lewis and R. M. Gous, “Broilers perform better on short or step-up photoperiods,” *South African Journal of Animal Science*, vol. 37, no. 2, pp. 90–96, 2007, doi: 10.4314/sajas.v37i2.4032.

[26] I. Rodrigues and M. Choct, “Feed intake pattern of broiler chickens under intermittent lighting: Do birds eat in the dark?,” *Animal Nutrition*, vol. 5, no. 2, pp. 174–178, 2019, doi: 10.1016/j.aninu.2018.12.002.

[27] E. A. M. Bokkers and P. Koene, “Eating behaviour, and preprandial and postprandial correlations in male broiler and layer chickens,” *British Poultry Science*, vol. 44, no. 4, pp. 538–544, 2003, doi: 10.1080/00071660310001616165.

[28] T. E. Girard, M. J. Zuidhof, and C. J. Bench, “Aggression and social rank fluctuations in precision-fed and skip-a-day-fed broiler breeder pullets,” *Applied Animal Behaviour Science*, vol. 187, pp. 38–44, 2017, doi: 10.1016/j.applanim.2016.12.005.

[29] I. Estevez, R. C. Newberry, and L. J. Keeling, “Dynamics of aggression in the domestic fowl,” *Applied Animal Behaviour Science*, vol. 76, no. 4, pp. 307–325, 2002, doi: 10.1016/S0168-1591(02)00013-8.

[30] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[31] P. Refaeilzadeh, L. Tang, and H. Liu, “Cross-validation,” *Encyclopedia of Database Systems*, pp. 532–538, 2009, doi: 10.1007/978-0-387-39940-9_565.

[32] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. Da Silva, “A comparative analysis of object detection metrics with a companion open-source toolkit,” *Electronics (Switzerland)*, vol. 10, no. 3, pp. 1–28, 2021, doi: 10.3390/electronics10030279.

[33] H. Feng, G. Mu, S. Zhong, P. Zhang, and T. Yuan, “Benchmark analysis of YOLO performance on edge intelligence devices,” *Cryptography*, vol. 6, no. 2, 2022, doi: 10.3390/cryptography6020016.

BIOGRAPHIES OF AUTHORS

Sri Wahjuni is a member of the Division of Computer Systems and Networks as well being on the teaching staff at the Department of Computer Science IPB University; she obtained a bachelor's degree from the Sepuluh Nopember Institute of Technology (ITS) as well as a master's and doctoral degree from the University of Indonesia. Her research fields include embedded systems, heterogeneous networks and the Internet of Things. She is a member of the Institute of Electrical and Electronics Engineers (IEEE) and the International Association of Engineers (IAENG). She can be contacted at email: my_juni04@apps.ipb.ac.id.

Wulandari received the M. Agr.Sc degree (Master of Agricultural Science) in the Division of Environmental Science and Engineering, Graduate School of Agriculture, Kyoto University, Japan. She has been a computer science lecturer in the Department of Computer Science, IPB University, Bogor, Indonesia, since 2017. In addition, she is serving as the head of the Internet of Things for Smart Urban Farming Laboratory (ISurf) (2019–present). Her research interests are in image processing, machine learning, the Internet of Things, and biosensing engineering. She is a member of the Japan Society of Agriculture Machinery (JSAM), International Association of Engineers (IAENG), and the Institute of Electrical and Electronics Engineers (IEEE). She can be contacted at email: wulandari.ilkom@apps.ipb.ac.id.

Rafael Tektano Grandiawan Eknanda holds a Bachelor of Computer Science Engineering (B.Comp.) from the Department of Computer Science, IPB University, Bogor, Indonesia. His research areas of interest include machine learning and image processing. He can be contacted at email: rafael_28@apps.ipb.ac.id.

Iman Rahayu Hidayati Susanto is a professor and head of the Poultry Production Division at the Department of Animal Production Science and Technology, IPB University; she obtained a bachelor's degree from IPB, a master's degree from Padjadjaran University and a doctoral degree from Universiti Putera Malaysia; Her research fields include production, nutrition and the processing of poultry products, poultry functional food based on patent acquisition (omega eggs and low cholesterol carcass), behavioral applications, poultry farming (breed and local) and the use of IoT in poultry. She can be contacted at email: imanso@apps.ipb.ac.id.

Auriza Rahmad Akbar holds a Bachelor of Computer Science (BCS), Master of Computer Science (MCS), in addition to several professional certificates and skills. He is currently lecturing with the Department of Computer Science at Institut Pertanian Bogor, Indonesia. His research areas of interest include Computer Networks, Parallel Processing, and the Internet of Things (IoT). He can be contacted at email: auriza@apps.ipb.ac.id.