


Article

Research on Laying Hens Feeding Behavior Detection and Model Visualization Based on Convolutional Neural Network

Hongyun Hao ¹, Peng Fang ², Wei Jiang ¹ , Xianqiu Sun ³, Liangju Wang ¹ and Hongying Wang ^{1,*}

¹ College of Engineering, China Agriculture University, Beijing 100083, China

² College of Engineering, Jiangxi Agriculture University, Nanchang 330045, China

³ Shandong Minhe Animal Husbandry Co., Ltd., Yantai 265600, China

* Correspondence: hongyingw@cau.edu.cn; Tel.: +86-13-6810-17695

Abstract: The feeding behavior of laying hens is closely related to their health and welfare status. In large-scale breeding farms, monitoring the feeding behavior of hens can effectively improve production management. However, manual monitoring is not only time-consuming but also reduces the welfare level of breeding staff. In order to realize automatic tracking of the feeding behavior of laying hens in the stacked cage laying houses, a feeding behavior detection network was constructed based on the Faster R-CNN network, which was characterized by the fusion of a 101 layers-deep residual network (ResNet101) and Path Aggregation Network (PAN) for feature extraction, and Intersection over Union (IoU) loss function for bounding box regression. The ablation experiments showed that the improved Faster R-CNN model enhanced precision, recall and F1-score from 84.40%, 72.67% and 0.781 to 90.12%, 79.14%, 0.843, respectively, which could enable the accurate detection of feeding behavior of laying hens. To understand the internal mechanism of the feeding behavior detection model, the convolutional kernel features and the feature maps output by the convolutional layers at each stage of the network were then visualized in an attempt to decipher the mechanisms within the Convolutional Neural Network(CNN) and provide a theoretical basis for optimizing the laying hens' behavior recognition network.

Keywords: laying hens; feeding behavior; Faster R-CNN; model visualization



Citation: Hao, H.; Fang, P.; Jiang, W.; Sun, X.; Wang, L.; Wang, H. Research on Laying Hens Feeding Behavior Detection and Model Visualization Based on Convolutional Neural Network. *Agriculture* **2022**, *12*, 2141. <https://doi.org/10.3390/agriculture12122141>

Academic Editors: Xiuguo Zou, Zheng Liu, Xiaochen Zhu, Wentian Zhang, Yan Qian and Yuhua Li

Received: 10 November 2022

Accepted: 9 December 2022

Published: 13 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, researchers have studied the health and welfare of animals by monitoring their individual behaviors [1,2]. A laying hen's behavioral activities can be divided into feeding, drinking, resting, fighting, etc. Feeding is one of the most important behaviors in the life of laying hens, and it accounts for more than 40% of total activity time [3]. In the large-scale poultry breeding farm, abnormal feeding behavior of laying hens could reflect a health and welfare problem in the long term. For example, the decline in feed frequency and feed intake of some hens may indicate the possibility of disease, while the large-scale deterioration of the feed frequency may indicate that timely feeding is needed. On the contrary, the simultaneous and unexpected occurrence of high feed intake and low egg production may also reflect a health problem of laying hens. Thus, monitoring the feeding behavior of laying hens is significant in the breeding farm.

Traditionally, image processing technology is used to identify or classify poultry behaviors. However, it has the disadvantages of poor model generality, robustness, and difficulty in feature extraction [4–7]. Deep learning technology can learn the characteristics of the data itself through a large number of samples and has the advantages of speed, accuracy, and robustness; it is widely used in image detection and segmentation of animals. Some researchers have utilized deep learning and machine vision methods to detect typical behaviors of livestock and poultry, such as feeding, climbing, drinking, and excretion [8–14]. Wang et al. [15] built a laying hens behavior detection model based on the YOLOv3 network, which could recognize the feeding, mating, standing, and fighting

behaviors of laying hens. To identify broilers' lameness, Nasiri et al. [16] used CNN to extract the key points of the broiler's body and Long Short-Term Memory (LSTM) to classify the lameness of broilers. Fang et al. [17] employed a similar method for pose estimation and behavior classification of broiler chickens, which could identify broiler behaviors such as eating, standing, walking, running, resting, and preening. Geffen et al. [18] detected and counted the laying hens in the battery cages with the Faster R-CNN network and achieved 89.6% accuracy at cage level. Fang et al. [19] constructed a laying hens behavior detection network based on the Faster R-CNN network and knowledge-distillation technology, which significantly improved model performance while reducing the model inference time.

Previous research has proved that CNN could realize the analysis and recognition of image content and effectively solve the problems related to animal behaviors. However, we lack an understanding of its internal implementation mechanism, and the outstanding recognition performance lacks explanation. Therefore, during the model development process, a model with better performance can only be obtained through continuous trial and error [20].

In this research, we developed a feeding behavior detection model for stacked cage hens based on an improved Faster R-CNN network [21]. To solve the problem of loss of low-level features in the network, a feature extraction network based on the path aggregation network was constructed, and the regression loss function was improved, which significantly improved the performance of the feeding behavior detection network. Following this, the convolutional kernel features and the feature maps output by the convolutional layers at each stage of the network were visualized in an attempt to interpret the mechanism within the convolutional neural network and provide a theoretical foundation for the continuous optimization of the hens' behavior detection network.

2. Materials and Methods

2.1. Experimental Setup

The experiment in this research was conducted in Deqingyuan Ecological Park, Yanqing, Beijing, China. Laying hens (Jinghong 1) were reared in a 4 layers-stacked cage breeding house. There was a total of 9200 cages; each cage was 45 cm wide, 60 cm deep, and 50 cm high. A nipple drinker was installed inside the cage, and a feed trough was seated outside, with a light source located directly above the passageway. Six laying hens were reared in a single cage, and usually, 2–4 laying hens were in the feeding position for feeding, and the rest were drinking or resting.

The image acquisition system in this experiment consisted of three digital cameras (XCG-CG240C, SONY, Shanghai, China) with a resolution of 1920×1200 pixels, three fixed focus lenses (Ricoh FLCC0614A 2M, RICOH, Philippines), and a mobile inspection platform. The cameras were mounted on the mobile inspection platform at an angle of 30 degrees downward horizontally and were controlled by a microcomputer (Dell OptiPlex 7080MFF, Dell Inc., Xiamen, China) to capture images of the laying hens. Figure 1 shows the image acquisition system and the housing condition of laying hens. The inspection platform traveled to the front of each cage to collect images of the hens. Images were collected without adding additional light to minimize stress on the hens.

2.2. Data Collection and Labeling

Images were collected from November to December 2021. We selected 100 cages of laying hens for image acquisition and finally selected 1000 images as the original dataset. The data collection followed the Experimental Animal Welfare and Animal Experiment Ethics Committee of China Agricultural University guidelines. As shown in Figure 2, due to the difference in light intensity between hen layers, images collected from the first and second layers were enhanced using the Retinex enhancement algorithm to improve the image readability. After that, the original image set was labeled with the free image label tool "Labelme", in which hens whose heads were near or in the feeding trough were labeled as "feeding" and the others were labeled as "resting". In the detection work, the CNN

does not have scale invariance and rotation invariance due to the fixed characters of the convolution itself. The adaptive ability of the CNN to target changes almost comes from the diversity of data itself. The more and more comprehensive the data, the higher the accuracy of the trained model [10]. Therefore, the dataset was expanded to 2000 images by 90° random rotation, adding Gaussian noise and randomly adjusting image contrast to improve the model's generalization ability. Finally, the dataset contained 4268 samples of hens labeled as “feeding” and 4836 samples of hens labeled as “resting”, and was randomly divided into a training set, validation set, and test set (7:2:1).

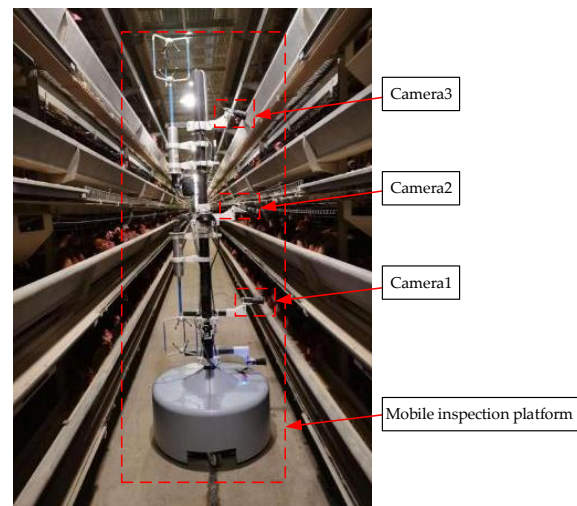


Figure 1. The image acquisition system and housing condition of laying hens.



Figure 2. Image sample of hens.

2.3. Faster R-CNN Network

The feeding behavior detection model was constructed based on the Faster R-CNN network in this research. As shown in Figure 3, the Faster R-CNN network can be divided into four parts: feature extraction network, Region Proposal Network (RPN), Region of interest (ROI) pooling network, bounding box regression and classification. The feature extraction network is used to extract the feature maps. The features maps are then shared with the region proposal network and the ROI pooling network, where the region proposal network extracts the candidate bounding boxes to the ROI pooling network, and through the ROI pooling layer, each ROI generates a fixed-size feature map; finally, regression and classification of the bounding boxes are performed.

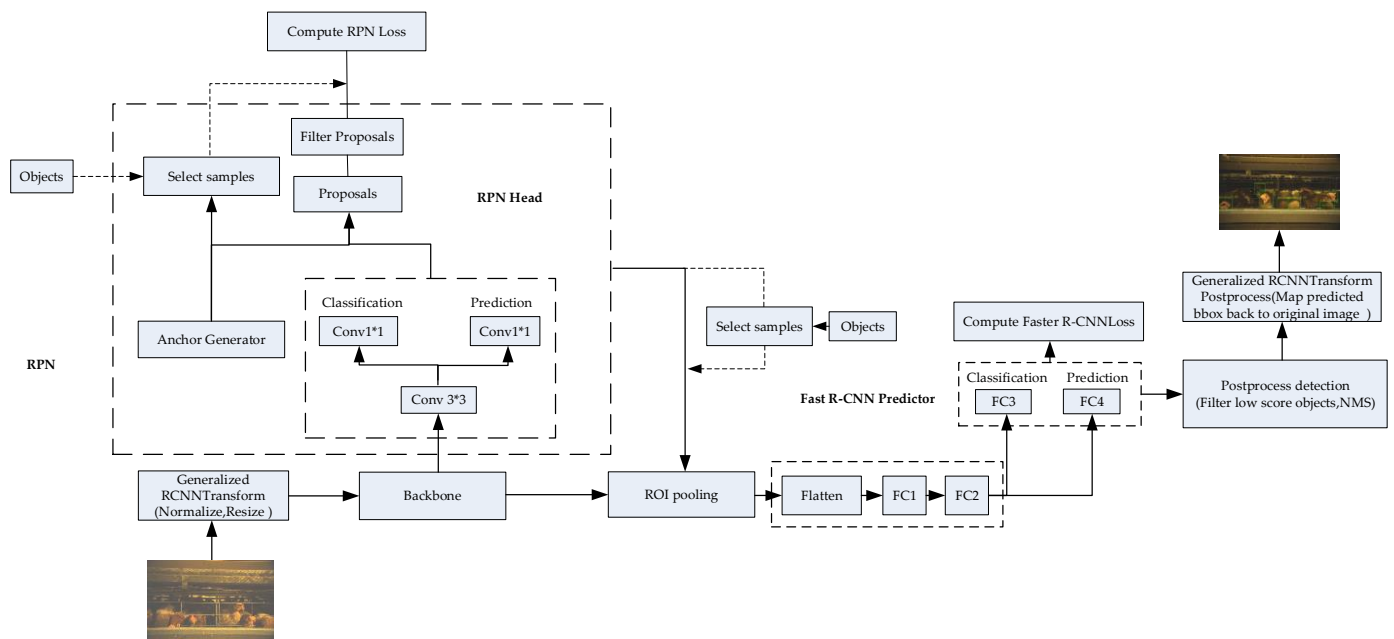


Figure 3. Structure of Faster R-CNN network. FC: fully connected layer, Conv: convolution.

2.4. Construction of Feature Extraction Network Based on Path Aggregation Network

In CNN, low-level layers focus on image details such as edge shape and object position, while deep layers will focus on strong semantic information. The object detection network needs to be concerned about the image's semantic information, position information, and pixel details. Therefore, it is necessary to fully use the features extracted by each level of the backbone network so that the input feature maps of the region proposal network get both semantically vital information and low-level localization information. The Faster R-CNN network achieves this through the Feature Pyramid Network (FPN), which significantly improves the detection ability of the Faster R-CNN network for small objects. However, in the “bottom-up” transmission architecture of FPN networks, the path from the shallow features to the top layer is too long. As shown by the red dotted path in Figure 4, the features extracted from the last convolutional layer of the second stage (stage 2) of the ResNet 101 network pass through hundreds of layers to the top layer (P5). The low-level feature information suffers severe losses through the transmission over long paths, which makes it difficult to preserve accurate target location information in the top-level feature map. Liu et al. [22] proposed a path aggregation network (PAN), for instance segmentation, which significantly improved the performance of an instance segmentation network by creating a bottom-up path augmentation, adaptive feature pooling structure and fully connected fusion method.

In this research, the bottom-up path augmentation of the PAN was introduced into the Faster R-CNN network. The four feature fusion layers were added after the FPN network by lateral connection, the architecture of which is shown in Figure 4b. With the addition of the bottom-up pathway augmentation, the low-level features extracted in the second stage of the ResNet 101 network were transmitted to feature map P2 by a lateral connection and subsequently passed through the feature map N2 to the top feature layer N5 (the path shown by the green dotted line in Figure 4). It took less than ten layers to transmit the low-level features to the top layer, which significantly shortened the information transmission path; the low-level feature information can be better retained in the top feature map, which is conducive to the accurate localization of the targets.

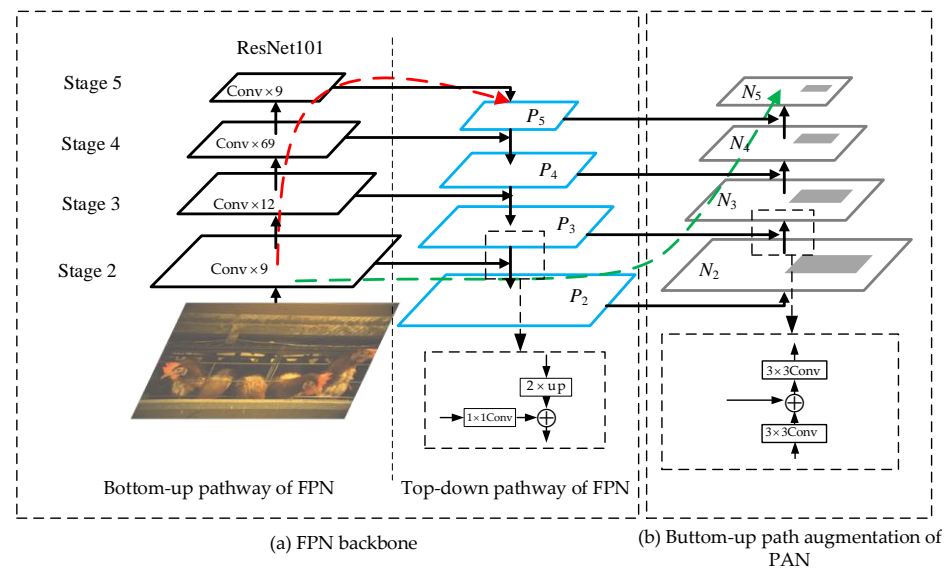


Figure 4. Structure of the improved feature extraction network. Conv: convolution, up: upsampling, \oplus : add.

2.5. Optimisation of the Loss Function

The regression loss and classification loss composed the loss of Faster R-CNN network. Among them, the Faster R-CNN network utilized the $smooth_{L1}$ loss as the regression loss, as shown in Equations (1) and (2).

$$L_{reg} = \lambda \frac{1}{N_{reg}} \sum_i p_i^* smooth_{L1}(t_i, t_i^*) \quad (1)$$

$$smooth_{L1}(t_i, t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2 & (|t_i - t_i^*| < 1) \\ |t_i - t_i^*| - 0.5 & (|t_i - t_i^*| \geq 1) \end{cases} \quad (2)$$

where L_{reg} is the regression loss of the Faster R-CNN, N_{reg} is the number of anchors, p_i^* is 1 if the anchor is positive and is 0 if the anchor is negative, t_i is a vector representing the 4 parameterized coordinates of the predicted bounding box, t_i^* is that of the ground-truth box associated with a positive anchor.

When calculating the regression loss of the network by the $smooth_{L1}$ function, the 4 points of the predicted bounding boxes are treated as independent of each other, and their respective loss values are calculated and then summed up to obtain the total regression loss. In fact, the four points are related to each other. IoU is usually used to evaluate the proximity between the predicted bounding boxes and the ground truth. When multiple predicted bounding boxes get the same $smooth_{L1}$ loss value, their IoU values may vary greatly. Thus, performing regression on the 4 points in isolation is inappropriate, and the predicted bounding boxes composed of the 4 points should be regarded as a whole for the regression. In this research, IoU loss [23], is used to replace the $smooth_{L1}$ loss in the Faster R-CNN network. The IoU loss function is defined as:

$$IoU_{loss} = -\ln(IoU) \quad (3)$$

$$IoU = \frac{I}{U} \quad (4)$$

where IoU is the intersection and union ratio of the predicted bounding boxes and the ground truth; I is the area of the intersection region of the predicted bounding boxes and the ground truth; U is the union region of the predicted bounding boxes and the ground truth.

2.6. Model Training

In this research, the training work was performed on a Dell computer with an Intel(R) Core (TM) i7—9700K, an NVIDIA GeForce GTX2080 GPU (11 GB), and 16 GB of memory. The operating environment was Ubuntu18.04, CUDA 10.2, cuDNN 8.0.1, and Python 3.7. The model was trained for 16,000 steps, with an initial learning rate of 0.001, a momentum of 0.9, Stochastic Gradient Descent (SGD) optimizer, and a weight decay of 0.0001. The learning rate increased to 0.002 after 8000 steps. In order to obtain the best model, weights were saved every 2000 steps.

3. Results

Different optimization methods of the feeding behavior detection network were tested in this experiment: ① Faster R-CNN network with Resnet101 and feature pyramid network as the feature extraction network, and $smooth_{L1}$ function as the regression loss function (ResNet_fpn_smooth). ② Faster R-CNN network with the Resnet101, path aggregation network, and feature pyramid network as the feature extraction network, and $smooth_{L1}$ function as the regression loss function (ResNet_pafpn_smooth). ③ Faster R-CNN network with the Resnet101 and feature pyramid network as the feature extraction network, and IoU loss as the regression loss function (ResNet_fpn_iou) ④ Faster R-CNN network with the Resnet101, path aggregation network and feature pyramid network as the feature extraction network, and IoU loss as the regression loss function (ResNet_pafpn_iou). The performance of the above four recognition models was tested with the test set, and the same image was input into each of the above four models to obtain the four sets of output results in Figure 5; all models could accurately identify the feeding and resting behaviors of hens.

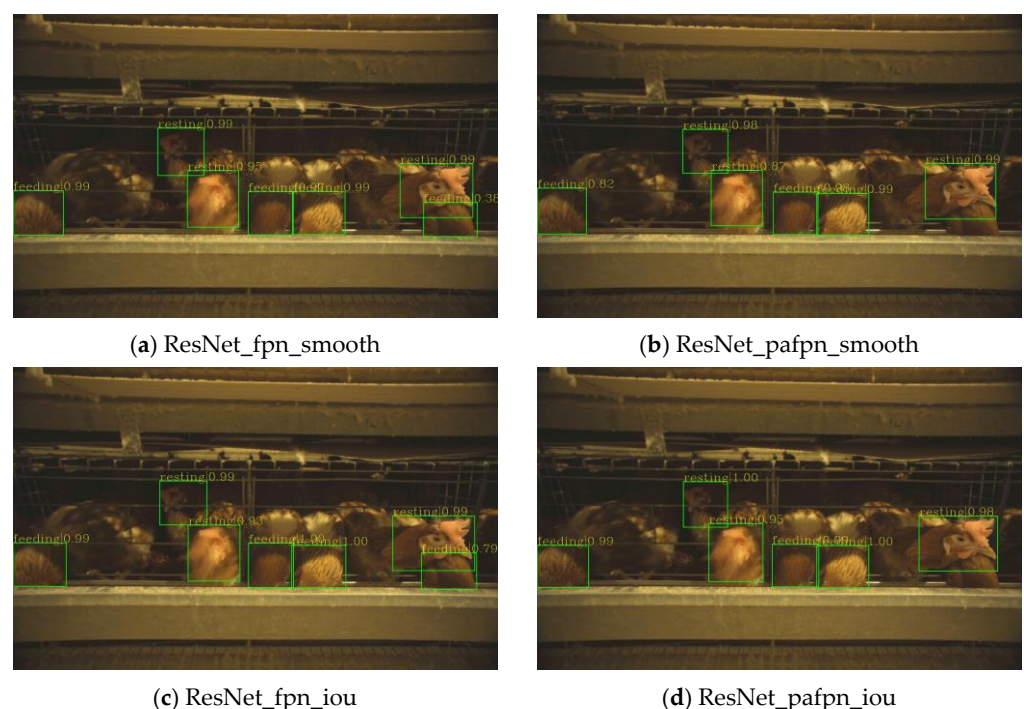


Figure 5. Detection results of different models.

The Precision (P), recall (R), and average inference time (t) were used in this experiment to evaluate the feeding behavior detection model performance. As shown in Table 1, the detection precision of all models was above 80%. The accuracy, recall and F1-score of the ResNet_fpn_smooth were 84.4%, 72.67% and 0.781, respectively, while the corresponding values were 87.2%, 71.3% and 0.785 for the ResNet_pafpn_smooth. There was a noticeable improvement in the precision index after adding the path aggregation network to the Faster

R-CNN network and a slight decrease in the recall index. In addition, the inference time of both models was similar, which indicated that the path aggregation network improved the retention rate of low-level feature information and improved the detection precision of the object without increasing the model complexity too much. The precision, recall and F1-score of the ResNet_fpn_iou were 88.73%, 73.49% and 0.804, respectively, higher than that of ResNet_fpn_smooth and ResNet_pafpn_smooth, which means that the IoU loss function could calculate the error between the predicted and true values of the bounding box more accurately, to obtain more accurate prediction results. Finally, the ResNet_pafpn_iou got a precision of 90.12%, a recall of 79.14% and a F1-score of 0.843, which was the best.

Table 1. Performance comparison of different models.

Models	Precision/%	Recall/%	F1-Score	Average Inference Time/s
ResNet_fpn_smooth	84.40	72.67	0.781	0.143
ResNet_pafpn_smooth	87.20	71.31	0.785	0.145
ResNet_fpn_iou	88.73	73.49	0.804	0.143
ResNet_pafpn_iou	90.12	79.14	0.843	0.144

Figure 6 shows the training loss curve of the ResNet_fpn_smooth, ResNet_pafpn_smooth, ResNet_fpn_iou, and ResNet_pafpn_iou. The training loss decreased to a low value within a short time after the training started, then slowly reduced with the training process. The training loss became flat when the iteration was about 14,000 times and no longer declined. When the number of iterations reached 16,000, the training process ended, and the model converged. Based on the training loss curves in Figure 6, ResNet_pafpn_iou achieved the lowest converged loss, which indicated the effectiveness of the optimization process.

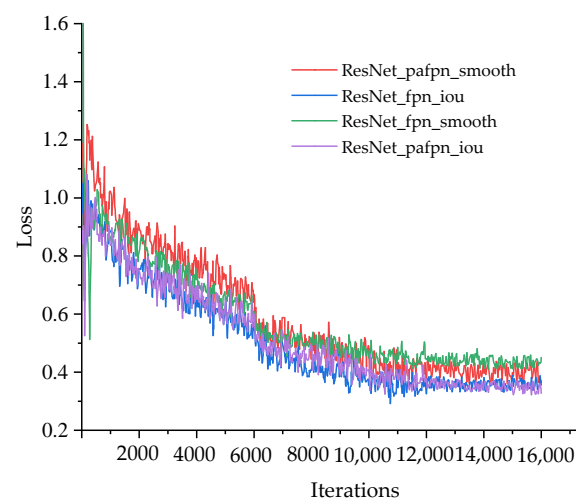


Figure 6. Training loss curves of different models.

4. Discussion

In the CNN, each layer of the network extracts different features through the convolution kernel, and the network will integrate the extracted features to realize the interpretation of the image content. Visualization of the CNN was first proposed by Zeiler et al. [20]. Subsequently, visualization techniques such as Class Activation Maps (CAM) [24] and Gradient-weighted Class Activation Maps (Grad-CAM) [25] were developed.

In this research, taking ResNet101 as an example, the features relating to the convolution kernel and the feature maps generated by the convolution layer of the feeding behavior detection network were visualized. The aim was to understand the internal mechanism of the convolutional neural network, providing a theoretical basis for the optimization of the

behavior recognition network of hens. The ResNet101 network consists of 101 convolution layers and can be divided into 5 stages. In the Faster R-CNN network, the feature maps extracted in the first stage of ResNet101 are not sent to the region proposal network. Therefore, we only visualized the feature maps of the last convolution layers in the second to fifth stages to analyze the differences between the extracted features of the low-level and the top ones.

4.1. Visual Analysis of the Feature Maps

The number of feature maps output in the second, third, fourth, and fifth stages of the ResNet101 network was 256, 512, 1024, and 2048, respectively, and all of the feature maps were single-channel images. In this section, all the single-channel feature maps of each stage were merged into a multi-channel image, and the 4 feature maps with the most significant activation features were output for visualization. Figure 7 shows the visualization results.

The training process of the CNN imitates the cognitive function of the human brain. The human visual system performs image recognition step by step, and people will first understand the color and brightness features in the image, then the simple geometric features such as points, lines, and edges, and after that, the slightly complex features (high-dimensional information) such as texture in the image, finally, forming the concept of the whole image. The CNN similarly processed the image. As shown in Figure 7, low-level layers in the second stage mainly extracted the image's low-level features, such as contour, edge, and color features. It focused more on the image's overall color and line information, not only the contour of the hen. With the deepening of the network, the third and fourth stages focused more on the texture of the image. In the third and fourth stages, the network gradually focused on the contour of hens, and some key features were extracted, including their head and cockscomb. As the network got more profound, the features extracted by the network began to be highly abstract, and the naked eye could no longer recognize the specific content of the extracted features. However, the convolutional neural network can extract essential information from it, and the area of concern of the network is basically focused on the hen's contour, ignoring the background. The subsequent fully connected convolution layers processed the features extracted from the high-level layer to complete the detection and classification of the hens.

4.2. Visual Analysis of the Convolution Kernels

The convolution kernel of the CNN is responsible for extracting features from the image. By visualizing the convolution kernel, we can more intuitively understand the features extracted by the convolution kernel of the image and clearly understand CNN's internal mechanism. The gradient lifting method is used to compute the input image when the convolution kernel of each layer in ResNet101 reaches the maximum activation state, and the input image is the feature extracted by the convolution kernel. This section visualized the first 36 convolution kernels of the last layer in stage 2–stage 5.

From the visualized results in Figure 8, in the second stage of the ResNet101, the convolution kernel extracted some low-level features such as color, line, and texture features. The combination of color and line features formed wavy and long strip textures. With the network getting deeper, the kernels in the fourth and fifth stages extracted more complicated texture features, spiral, circular, and various shape combinations of texture features. The convolution kernel became more and more complex, and the extracted features became more and more refined. A large number of complex and refined texture features gradually depicted the contour of the detection object (hens) as the network got deeper. In summary, the low-level layers of the network mainly extracted general features such as edges, lines, and some simple textures, while deeper layers could extract complex and semantically strong features (feather, eye), which were similar to the target characteristic to be detected.

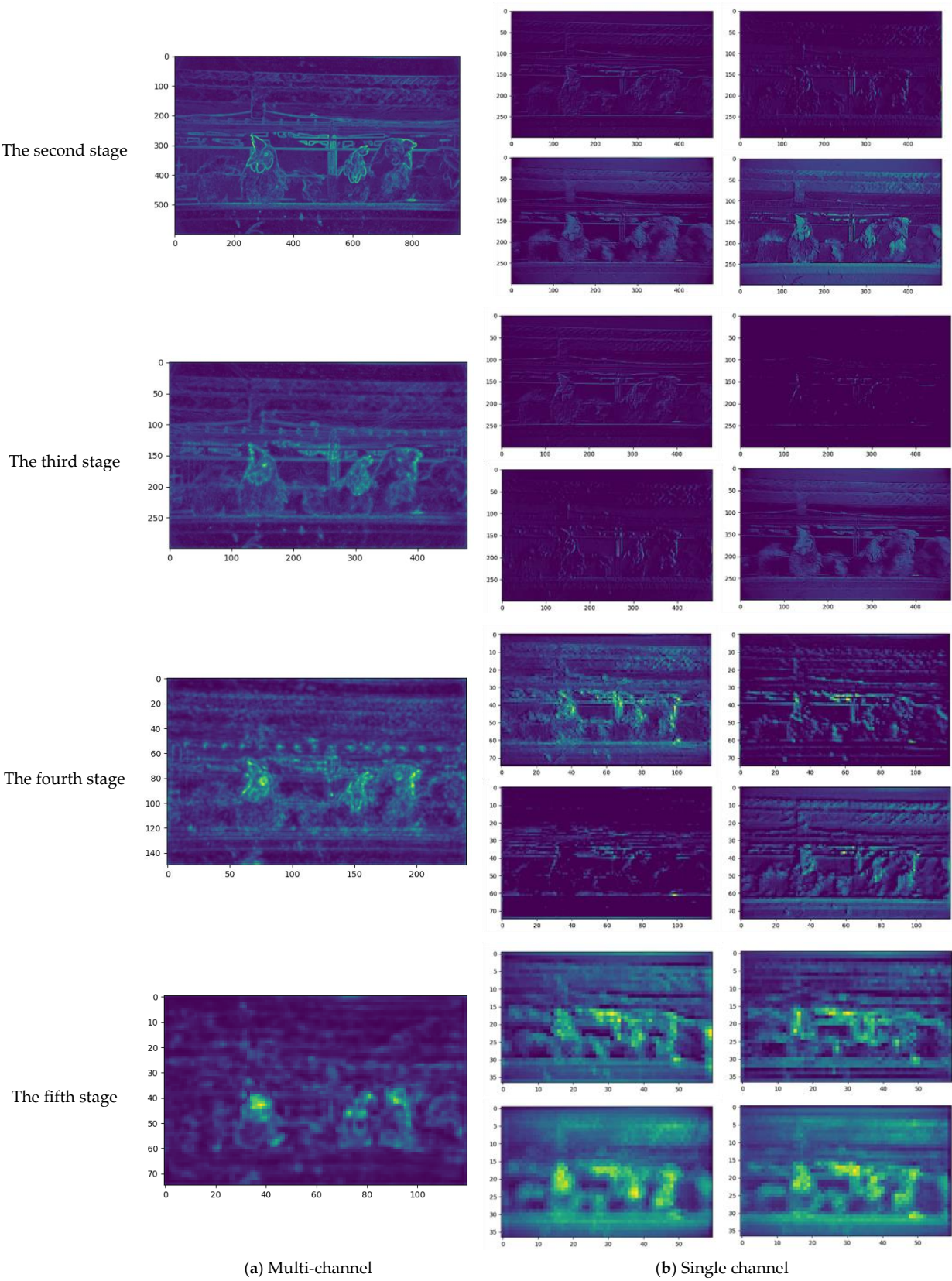


Figure 7. Feature maps from stage 2 to stage 5.

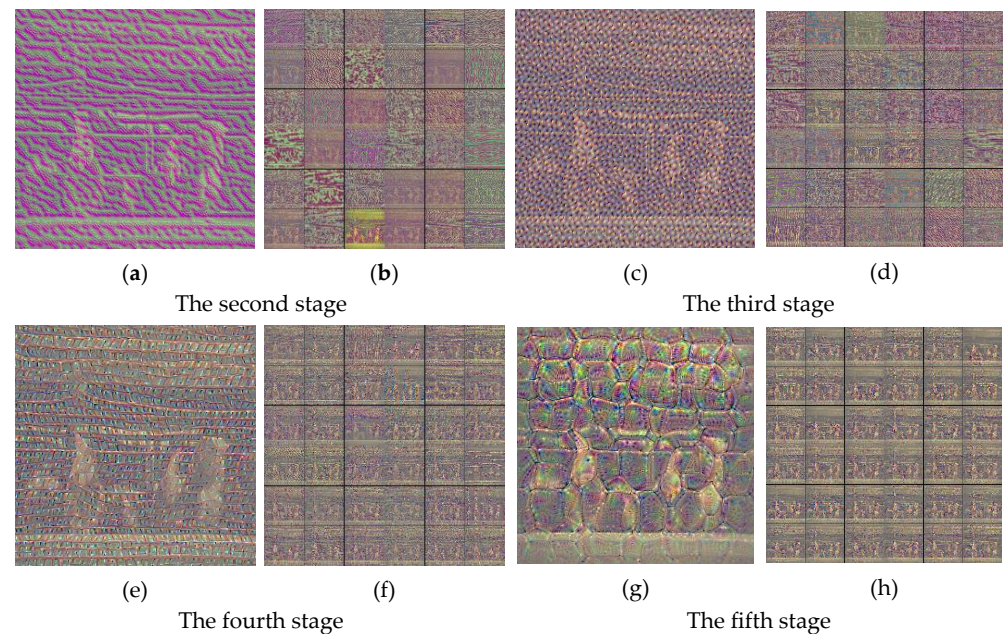


Figure 8. Convolution kernel visualization results. (a,c,e,g) are the visualization result of the first convolution kernels at each stage, respectively. (b,d,f,h) are the visualization results of the first 36 convolution kernels at each stage, respectively.

4.3. Limits and Future Work

It is worth noting that there were still some limitations to this study. In the detection of the feeding behavior of laying hens, only feeding behavior and resting behavior were taken into consideration; other behaviors, such as fighting, drinking, and egg laying, were not considered in this research. The small cage size and the lighting conditions of the stacked cage breeding house caused this. The drinking and laying behaviors of the hens always occurred inside the cage, while the feeding and resting laying hens would stay close to the front door, blocking the camera's view. Additionally, the low illumination of the house would result in almost no light inside the cage, which means that the camera cannot collect valid images for the detection work. Fighting behavior is often observed during feeding and can be obscured by the trough, making sample collection more complex. In future work, we will attempt to use an infrared camera to capture images and select a better angle.

Furthermore, the Faster R-CNN model was a two-stage object detection network, which was slower in detection speed than other networks studied [26,27]. Thus, a one-stage object detection network such as SSD [28], and YOLOv4 [29] should be considered to further improve the feeding behavior detection model. Lastly, the feeding behavior detection model has been developed for the stacked cage laying hens, but is not suitable for laying hens with other feeding methods. Therefore, the model can be further improved through the collection of more data from laying hens with different feeding patterns.

5. Conclusions

In this work, an improved Faster-RCNN model was constructed to recognize the feeding behavior of stacked caged hens based on a path aggregation network and IoU loss function. The precision, recall and F1-score of the model were improved from 84.40%, 72.67%, 0.781 to 90.12%, 79.14% and 0.843, respectively, and the average detection time was almost unchanged. After that, an ablation experiment was conducted to demonstrate the effectiveness of the improvement and visualize the output feature maps of the convolution layer and the convolution kernel features of the feeding behavior detection network, respectively. Based on the visualization results, the convolutional neural network's internal mechanism was analyzed to explain the CNN's performance and provide a theoretical basis for further optimization of the detection model. In general, the developed model and

visual analysis method in this research could provide technical support for the subsequent monitoring of the health status and welfare status of laying hens and could also provide a reference for the optimization of other animal detection models. In future work, we will consider using a one-stage object detection network to optimize the feeding behavior detection model further and detect more behaviors, such as drinking and egg laying, to provide further technical support for poultry farm management.

Author Contributions: Conceptualization, H.H. and P.F.; methodology, P.F.; software, P.F.; validation, H.H. and P.F.; formal analysis, H.H.; investigation, H.H. and P.F.; resources, X.S.; data curation, P.F.; writing—original draft preparation, H.H. and W.J.; writing—review and editing, H.H. and L.W.; visualization, P.F.; supervision, L.W. and H.W.; project administration, H.W.; funding acquisition, H.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by the Ministry of Science and Technology, China under grant number: 2017YFE0122200.

Institutional Review Board Statement: The experiment was conducted following the guidelines of Experimental Animal Welfare and Animal Experiment Ethics Committee of China Agricultural University (Approved number: AW12112202-5-1).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Aydin, A.; Bahr, C.; Berckmans, D. A Real-Time Monitoring Tool to Automatically Measure the Feed Intakes of Multiple Broiler Chickens by Sound Analysis. *Comput. Electron. Agric.* **2015**, *114*, 1–6. [\[CrossRef\]](#)
2. Yang, X.; Zhao, Y.; Street, G.M.; Huang, Y.; Filip To, S.D.; Purswell, J.L. Classification of Broiler Behaviours Using Triaxial Accelerometer and Machine Learning. *Animal* **2021**, *15*, 100269. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Hansen, I.; Braastad, B.O. Effect of Rearing Density on Pecking Behaviour and Plumage Condition of Laying Hens in Two Types of Aviary. *Appl. Anim. Behav. Sci.* **1994**, *40*, 263–272. [\[CrossRef\]](#)
4. Pereira, D.F.; Lopes, F.A.A.; Filho, L.R.A.G.; Salgado, D.D.; Neto, M.M. Cluster Index for Estimating Thermal Poultry Stress (Gallus Gallus Domesticus). *Comput. Electron. Agric.* **2020**, *177*, 105704. [\[CrossRef\]](#)
5. Neves, D.P.; Mehdizadeh, S.A.; Tschärke, M.; de Alencar Nääs, I.; Banhazi, T.M. Detection of Flock Movement and Behaviour of Broiler Chickens at Different Feeders Using Image Analysis. *Inf. Process. Agric.* **2015**, *2*, 177–182. [\[CrossRef\]](#)
6. de Alencar Nääs, I.; da Silva Lima, N.D.; Gonçalves, R.F.; Antonio de Lima, L.; Ungaro, H.; Minoru Abe, J. Lameness Prediction in Broiler Chicken Using a Machine Learning Technique. *Inf. Process. Agric.* **2021**, *8*, 409–418. [\[CrossRef\]](#)
7. Del Valle, J.E.; Pereira, D.F.; Mollo Neto, M.; Gabriel Filho, L.R.A.; Salgado, D.D. Unrest Index for Estimating Thermal Comfort of Poultry Birds (Gallus Gallus Domesticus) Using Computer Vision Techniques. *Biosyst. Eng.* **2021**, *206*, 123–134. [\[CrossRef\]](#)
8. Jia, N.; Kootstra, G.; Koerkamp, P.G.; Shi, Z.; Du, S. Segmentation of Body Parts of Cows in RGB-Depth Images Based on Template Matching. *Comput. Electron. Agric.* **2021**, *180*, 105897. [\[CrossRef\]](#)
9. Qiao, Y.; Truman, M.; Sukkarieh, S. Cattle Segmentation and Contour Extraction Based on Mask R-CNN for Precision Livestock Farming. *Comput. Electron. Agric.* **2019**, *165*, 104958. [\[CrossRef\]](#)
10. Lamping, C.; Derks, M.; Groot Koerkamp, P.; Kootstra, G. ChickenNet—An End-to-End Approach for Plumage Condition Assessment of Laying Hens in Commercial Farms Using Computer Vision. *Comput. Electron. Agric.* **2022**, *194*, 106695. [\[CrossRef\]](#)
11. Xiao, D.; Lin, S.; Liu, Y.; Yang, Q.; Wu, H. Group-Housed Pigs and Their Body Parts Detection with Cascade Faster R-CNN. *Int. J. Agric. Biol. Eng.* **2022**, *15*, 203–209. [\[CrossRef\]](#)
12. da Silva Santos, A.; de Medeiros, V.W.C.; Gonçalves, G.E. Monitoring and Classification of Cattle Behavior: A Survey. *Smart Agric. Technol.* **2023**, *3*, 100091. [\[CrossRef\]](#)
13. Liu, L.; Zhou, J.; Zhang, B.; Dai, S.; Shen, M. Visual Detection on Posture Transformation Characteristics of Sows in Late Gestation Based on Libra R-CNN. *Biosyst. Eng.* **2022**, *223*, 219–231. [\[CrossRef\]](#)
14. Cheng, M.; Yuan, H.; Wang, Q.; Cai, Z.; Liu, Y.; Zhang, Y. Application of Deep Learning in Sheep Behaviors Recognition and Influence Analysis of Training Data Characteristics on the Recognition Effect. *Comput. Electron. Agric.* **2022**, *198*, 107010. [\[CrossRef\]](#)
15. Wang, J.; Wang, N.; Li, L.; Ren, Z. Real-Time Behavior Detection and Judgment of Egg Breeders Based on YOLO V3. *Neural Comput. Appl.* **2020**, *32*, 5471–5481. [\[CrossRef\]](#)
16. Nasiri, A.; Yoder, J.; Zhao, Y.; Hawkins, S.; Prado, M.; Gan, H. Pose Estimation-Based Lameness Recognition in Broiler Using CNN-LSTM Network. *Comput. Electron. Agric.* **2022**, *197*, 106931. [\[CrossRef\]](#)
17. Fang, C.; Zhang, T.; Zheng, H.; Huang, J.; Cuan, K. Pose Estimation and Behavior Classification of Broiler Chickens Based on Deep Neural Networks. *Comput. Electron. Agric.* **2021**, *180*, 105863. [\[CrossRef\]](#)

18. Geffen, O.; Yitzhaky, Y.; Barchilon, N.; Druyan, S.; Halachmi, I. A Machine Vision System to Detect and Count Laying Hens in Battery Cages. *Animal* **2020**, *14*, 2628–2634. [[CrossRef](#)]
19. Fang, P.; Hao, H.; Wang, H. Behavior Recognition Model of Stacked-cage Layers Based on Knowledge Distillation. *Trans. Chin. Soc. Agric. Mach.* **2021**, *52*, 300–306. [[CrossRef](#)]
20. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2014; p. 8689. [[CrossRef](#)]
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
22. Liu, S.; Qi, L.; Qin, H.; Shi, J.; Jia, J. Path Aggregation Network for Instance Segmentation. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8759–8768. [[CrossRef](#)]
23. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T. UnitBox: An Advanced Object Detection Network. In *Proceedings of the 24th ACM international conference on Multimedia*, Amsterdam, The Netherlands, 15–19 October 2016; pp. 516–520. [[CrossRef](#)]
24. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2020**, *128*, 336–359. [[CrossRef](#)]
25. Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; Torralba, A. Learning Deep Features for Discriminative Localization. *arXiv* **2015**, arXiv:1512.04150. [[CrossRef](#)]
26. Jiang, K.; Xie, T.; Yan, R.; Wen, X.; Li, D.; Jiang, H.; Jiang, N.; Feng, L.; Duan, X.; Wang, J. An Attention Mechanism-Improved YOLOv7 Object Detection Algorithm for Hemp Duck Count Estimation. *Agriculture* **2022**, *12*, 1659. [[CrossRef](#)]
27. Yang, J.; Zhang, T.; Fang, C.; Zheng, H. A Defencing Algorithm Based on Deep Learning Improves the Detection Accuracy of Caged Chickens. *Comput. Electron. Agric.* **2023**, *204*, 107501. [[CrossRef](#)]
28. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; Volume 9905, pp. 21–37. [[CrossRef](#)]
29. Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLO v4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.