



Article

Study on Poultry Pose Estimation Based on Multi-Parts Detection

Cheng Fang ¹, Haikun Zheng ¹, Jikang Yang ¹, Hongfeng Deng ¹ and Tiemin Zhang ^{1,2,3,*}

¹ College of Engineering, South China Agricultural University, 483 Wushan Road, Guangzhou 510642, China; gu5457111@gmail.com (C.F.); haikun0619@gmail.com (H.Z.); yjkscau@gmail.com (J.Y.); redmaple-deng@outlook.com (H.D.)

² National Engineering Research Center for Breeding Swine Industry, Guangzhou 510642, China

³ Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou 510642, China

* Correspondence: tm-zhang@163.com

Simple Summary: Poultry farming is an important part of China's agriculture system. The automatic estimation of poultry posture can help to analyze the movement, behavior, and even health of poultry. In this study, a poultry pose-estimation system was designed, which realized the automatic pose estimation of a single broiler chicken using a multi-part detection method. The experimental results show that this method can obtain better pose-estimation results for a single broiler chicken with respect to precision, recall, and *F1* score. The pose-estimation system designed in this study provides a new means to provide help for poultry pose/behavior researchers in the future.

Abstract: Poultry pose estimation is a prerequisite for evaluating abnormal behavior and disease prediction in poultry. Accurate pose-estimation enables poultry producers to better manage their poultry. Because chickens are group-fed, how to achieve automatic poultry pose recognition has become a problematic point for accurate monitoring in large-scale farms. To this end, based on computer vision technology, this paper uses a deep neural network (DNN) technique to estimate the posture of a single broiler chicken. This method compared the pose detection results with the Single Shot MultiBox Detector (SSD) algorithm, You Only Look Once (YOLOV3) algorithm, RetinaNet algorithm, and Faster_R-CNN algorithm. Preliminary tests show that the method proposed in this paper achieves a 0.0128 standard deviation of precision and 0.9218 ± 0.0048 of confidence (95%) and a 0.0266 standard deviation of recall and 0.8996 ± 0.0099 of confidence (95%). By successfully estimating the pose of broiler chickens, it is possible to facilitate the detection of abnormal behavior of poultry. Furthermore, the method can be further improved to increase the overall success rate of verification.

Keywords: pose estimation; broiler chicken; deep learning; object detection; precision agriculture



Citation: Fang, C.; Zheng, H.; Yang, J.; Deng, H.; Zhang, T. Study on Poultry Pose Estimation Based on Multi-Parts Detection. *Animals* **2022**, *12*, 1322. <https://doi.org/10.3390/ani12101322>

Academic Editor: Alessandro Dal Bosco

Received: 3 April 2022

Accepted: 19 May 2022

Published: 22 May 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

Poultry farming is an important part of China's agriculture. With agricultural researchers' growing interest in precision agriculture and intelligent agriculture, the application of computer vision technology in the agricultural production management system is increasing [1–4]. In recent years, computer vision technology was proven by scholars to be an efficient method of posture estimation and behavior analysis [5,6]. Indeed, the automatic application of computer vision technology has provided a noticeable improvement in agricultural production management [7,8].

Zhuang et al. used image-processing methods to identify the skeleton posture of yellow-feathered broiler chickens [9]. Khan et al. used a dense stacked hourglass network to estimate the posture of pigs through RGB images acquired by a head-mounted Kinect camera. The algorithm adopted a bottom-up approach, labeled nine key points, and analyzed the pose behavior in an actual farm environment [10]. Fuentes et al. used the cattle behavior recognition framework based on a deep neural network to conduct behavior

recognition of the context-time information of the cattle in the video, which can identify up to 15 activities at different levels of the cattle [11]. Riekert et al. used deep learning to automatically detect the posture and position of pigs using a 2D camera. The algorithm's detection accuracy of posture and position reached 80.2% [12].

In multi-part detection of livestock, Huang et al. used the improved SSD algorithm to detect multiple parts of the rear-view image of cows and realized automatic cow body condition scoring (BCS) with 98.46% classification accuracy and 89.63% positioning accuracy of the algorithm [13]. Hu et al. classified cows by the fusion of features of multiple parts (head, trunk, and legs), and the recognition accuracy reached 98.36% [14]. Marsot et al. identified pig faces by detecting the eyes, nose, and other parts of 10 pigs, and the detection accuracy for a total of 320 test pictures reached 83% [15]. Salau et al. used k-nearest neighbor and neural network to classify the head, rump, back, legs, and udders of cattle, where the Hamming loss of the k-nearest neighbor classification was between 0.007 and 0.027, and the Hamming loss of the neural network was between 0.045 and 0.079 [16]. Wutke et al. automatically tracked and analyzed abnormal behaviors between pigs, such as tail-biting and ear-biting, using a combination of a pose-estimation algorithm to detect key points in multiple body parts of pigs and a Kalman filter, achieving 94.2% sensitivity, 95.4% accuracy, 95.1% F1 score, and 94.4% MOTA score [17].

With the development of deep-learning technology, there is a growing body of research using deep learning to estimate the posture of animals. For example, Mathis et al. successfully used CNN to develop a Deeplabcut framework that can analyze human and animal posture [18]. Pereira et al. developed the LEAP pose-estimation software to analyze animal postures and validated its performance by using fruit fly images [19]. Raman et al. conducted feature localization and spatio-temporal analysis of dog movement and posture through sequence CNN [20].

Based on the powerful tool of deep learning, this paper proposes a pose-estimation algorithm based on deep neural networks for broiler chickens. This paper aims to realize an automatic pose estimation of broiler chickens and precise monitoring of large-scale poultry farms, and the potential application of this method in poultry behavior analysis is further analyzed by comparing the ability of this method and four other commonly used pose-estimation algorithms to estimate the posture of individual chickens in the flock. The automatic estimation of poultry posture can help subsequent poultry researchers to analyze movement, behavior, and even the health of poultry.

2. Materials and Methods

2.1. Experimental Environment

The experimental environment is shown in Figure 1. The experiment in this paper was conducted in a poultry farm in Gaoming District, Foshan City, Guangdong Province, PR China. This poultry farm has two kinds of broilers, K90 and white recessive rock chickens (WRRC), both of which are common in Guangdong, so we studied their posture estimation in the research. Both the K90s and WRRCs were between 40 and 70 weeks old and were breeding birds, not for consumption. The image-acquisition system consisted of an HD camera (Logitech C922 Charge-Coupled Device camera) and a computer. The camera had a resolution of 1920×1080 pixels and took pictures of broiler chickens from multiple angles. Data were collected indoors and outdoors, with each collection time ranging from several seconds to several minutes, from 09:00 to 17:00 h. The indoor pen ($4 \text{ m} \times 3 \text{ m}$) used both the natural and artificial photoperiod, while the outdoor one used only the natural photoperiod. The outdoor pen ($6 \text{ m} \times 6 \text{ m}$) had enough space to allow the broilers free movement. The images collected by the camera were transmitted to the computer through a USB port for further processing. The experiment used a computer with a six-core processor, 2.4 GHz per core, 16 GB of RAM, a Windows 10 operating system, and a GTX 1060 6 G graphics card.



Figure 1. Experimental environment and test object.

The schematic is shown in Figure 2. The camera was 3–6 m away from the chicken at angles of 10 degrees to 80 degrees.

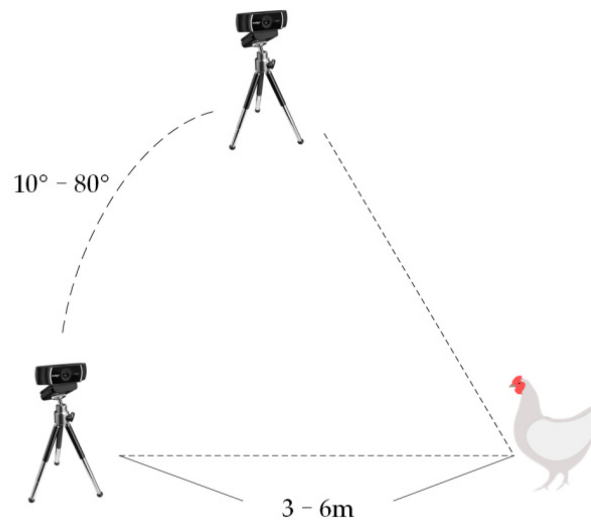


Figure 2. Photograph schematic.

2.2. Data Processing and Labelling

The collected data were screened, and any abnormal data caused by unexpected vibrations were eliminated. To reduce the memory consumption of GPU during training, all images collected were preprocessed by OpenCV (ver. 3.6.0) and the resolution adjusted to 512×512 pixels. The photos in the data set were manually labeled using EasyDL software to be used for subsequent pose estimation of a single chicken. The processed data set is shown in Figure 3.

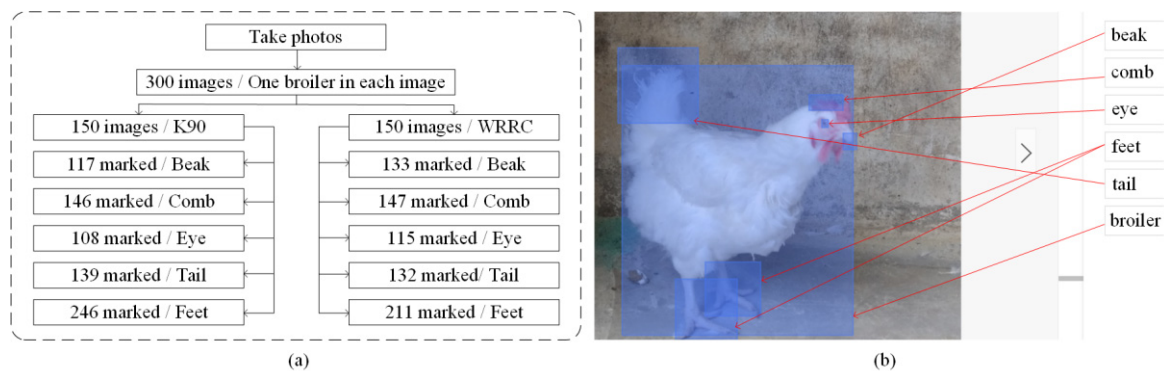


Figure 3. Datasets and marked examples: (a) Data partitioning; (b) Marked image as ground truth.

The chicken dataset includes 300 images of broilers: 150 K90 broilers and 150 WRRC broilers. The K90 set contains 117 marked beaks, 146 marked combs, 108 marked eyes, 139 marked tails, and 246 marked feet. The WRRC set contains 133 marked beaks, 147 marked combs, 115 marked eyes, 132 marked tails, and 211 marked feet.

2.3. Algorithm Framework and Implementation Steps

In this study, the algorithms were written in standard Python language. Figure 4 shows the flow of the BroilerPose pose-estimation algorithm, which references the Retinanet algorithm and consists of two steps [21]. The first step was to locate the target and the second step was to categorize the goals. A 50-layer residual network (ResNet-50) and a feature pyramid network (FPN) were used to construct the backbone to extract features from the image. The ResNet-50 was a bottom-up convolution network [22]. With the higher stage of convolution, the size of the resultant maps became smaller, and a higher level of semantics was retained. The FPN was a top-down convolution network. The lower-level feature layer in FPN was the combination of the higher-level feature layer and a corresponding ResNet-50 feature layer. The ResNet-50-FPN structure facilitated the extraction of both higher and lower-level relations [23]. Finally, the candidate frame was located and extracted through the Region Proposal Network (RPN), and the key points of the candidate frame were obtained and connected. Finally, the posture of the chicken was obtained.

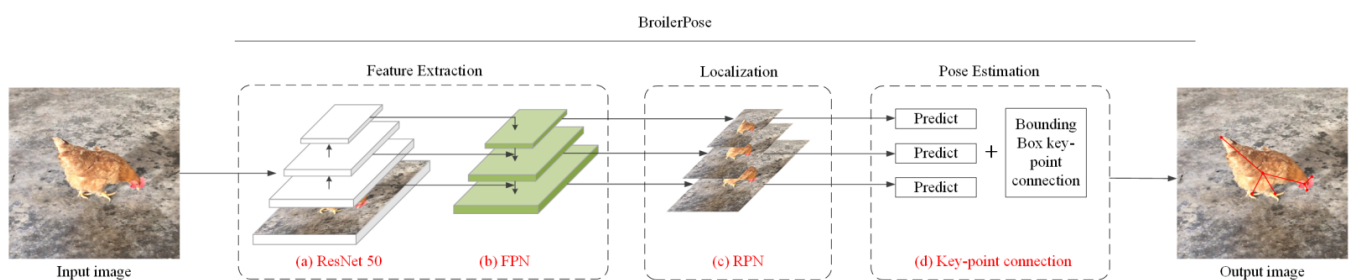


Figure 4. BroilerPose pose-estimation algorithm architecture: (a) The pre-trained ResNet-50; (b) FPN architecture, which includes a classify and a regression network; (c) RPN; (d) Key-point connection, output the broiler's posture.

After the broiler chicken pictures passed through the BroilerPose network structure, the results of six different categories were output. These were the bounding box (Bbox) of the broiler, beak, comb, eye, tail, and feet as $B_{\alpha}(x_{\alpha}, y_{\alpha}, x_{\alpha} + w_{\alpha}, h_{\alpha}), \alpha \in [1, 6]$. (x_{α}, y_{α}) was the coordinate of the upper left corner of the Bbox, while $(x_{\alpha} + w_{\alpha}, y_{\alpha} + h_{\alpha})$ was the coordinate point of the lower right corner of the Bbox. w_{α} and h_{α} were the width and height of the Bbox, respectively.

After obtaining the six Bbox categories, we output the central point of each Bbox as the key-point of the broiler chicken body part. The key-point was $K_i(X_i, Y_i), i \in [1, 8]$. X_i and Y_i are shown in Equation (1):

$$\begin{cases} X_i = x_\alpha + \frac{1}{2}w_\alpha \\ Y_i = y_\alpha + \frac{1}{2}h_\alpha \end{cases} \quad (1)$$

We then built the broiler chicken key-point connection algorithm, as shown in Table 1.

Table 1. Key-point connection combination.

Part	Key-Point	Combination
Broiler	K_1	(K_1, K_4)
Beak	K_2	(K_2, K_4)
Comb	K_3	(K_3, K_4)
Eye_left	K_4	(K_4, K_4)
Eye_right	K_5	(K_5, K_1)
Tail	K_6	(K_6, K_1)
Foot_left	K_7	(K_7, K_1)
Foot_right	K_8	(K_8, K_1)

As broiler chickens exist in various postures, there may be situations where the Bbox cannot be detected. When the Bbox was not recognized, we did not connect the key-point.

2.4. Algorithm Training

The training code was completed under the Keras deep-learning framework. From the collected data, we established the data set of broiler chickens described in Section 2.2 and randomly mixed the video of the data set. The size of the image used for algorithm input was 512×512 pixels. The ratio of the training set to the test set was 9:1. The whole training was iterated 1000 times using Stochastic Gradient Descent (SGD) as the network optimizer. SGD updated the parameters through each iteration to speed up the training [24]. The initial learning rate was set at 0.02.

2.5. Evaluation Metrics

After the detectors were trained, the testing set was used for evaluating. To determine whether every part had been correctly recognized, the intersection over union for each predicted part was computed using the area of overlap and union (Equation (2)):

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} \quad (2)$$

An IoU greater than 0.5 means the detectors detected the part of broiler chicken correctly.

Precision, recall, mean average precision (mAP), and $F1$ -score for detecting each part of the broiler chickens in the images were calculated using Equations (3)–(5). Precision is the ratio of true detection in all cases. Recall is the ratio of true detection in all manually labeled cases. $F1$ -score is the harmonic mean of precision and recall and a balancing metric for comprehensively evaluating false and missing cases:

$$P = \frac{TP}{TP + FP} \quad (3)$$

$$R = \frac{TP}{TP + FN} \quad (4)$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R} = \frac{2TP}{2TP + FP + FN} \quad (5)$$

where TP is true positive; FP is false positive; FN is a false negative; and TN is a true negative.

Average precision (AP) is defined as the area under the precision–recall curve and expressed as the mean precision at a set of 11 equally spaced recall levels $[0, 0.1, \dots, 1]$ [25]. The precision–recall curve is produced according to the predicted confidence level. The calculation of the AP is shown in Equation (6):

$$AP_{point=11} = \frac{1}{11} \sum_{\tilde{k} \in \{0, 0.1, \dots, 1\}} \max_{k \geq \tilde{k}} P(\tilde{k}) \quad (6)$$

where \tilde{k} is eleven equally spaced recall levels from 0 to 1. The maximum measured precision within a wiggle piece of the precision–recall curve was also used.

The mean average precision (mAP) is the average value of AP obtained for six different categories.

3. Results

3.1. Effects of Different Detection Methods

In the paper, five different methods were used to test the broiler chicken set, namely, BroilerPose, SSD, YOLOV3, RetinaNet, and Faster_R-CNN. Figure 5 shows the $F1$ -Score of five different algorithms with different thresholds.

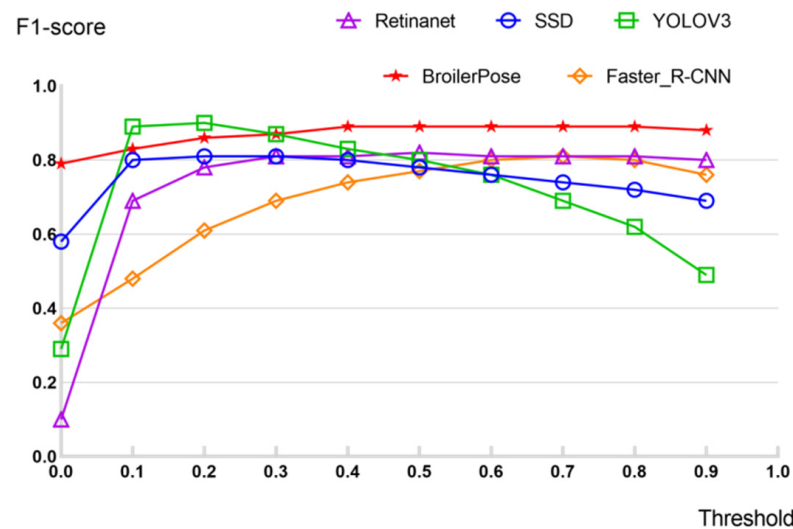


Figure 5. $F1$ -score for different algorithms, including BroilerPose algorithm, YOLOV3 algorithm, RetinaNet algorithm, SSD algorithm, and Faster_R-CNN algorithm. The higher the $F1$ -score, the better the detection effect.

By comparing the test results of five pose-estimation algorithms, the BroilerPose algorithm proposed in the paper reaches 0.89 in $F1$ -score when the threshold value is equal to 0.5. The RetinaNet algorithm achieves an $F1$ -score of 0.82. It is followed by the YOLOV3 algorithm ($F1$ -score = 0.80), the SSD algorithm ($F1$ -score = 0.78), and Faster_R-CNN ($F1$ -score = 0.77). We then calculated the overall AP of each algorithm and the mAP of individual parts.

Figure 6 shows the scores for different algorithms on the mAP , which can show the degree of the test model in all categories.

The mAP of the BroilerPose algorithm is 0.8652, the YOLOV3 algorithm is 0.8500, the Faster_R-CNN algorithm is 0.7928, the RetinaNet algorithm is 0.7540, and the SSD algorithm is 0.7375. Meanwhile, the training effects of different was shown in Table 2.

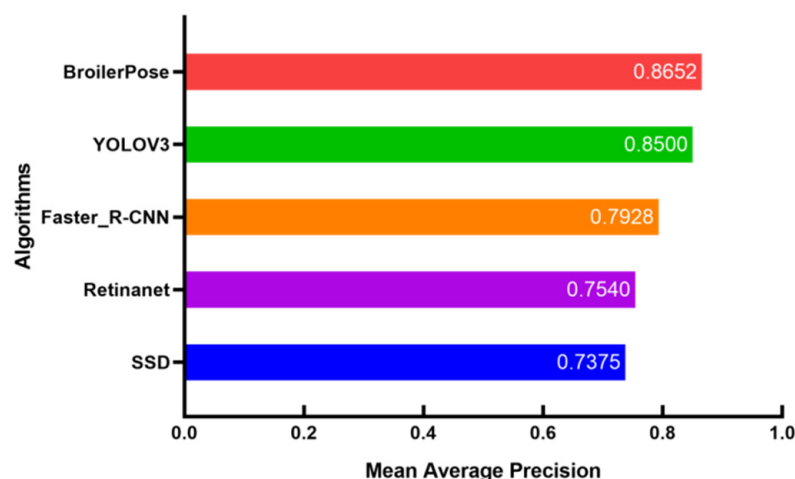


Figure 6. The performance of different algorithms in mAP.

Table 2. Comparison of training effects of different algorithms.

Bbox	Algorithms				
	BroilerPose	YOLOV3	Faster_R-CNN	RetinaNet	SSD
Broiler	0.997	0.998	0.994	0.998	0.999 ¹
Beak	0.772	0.774 ¹	0.65	0.641	0.563
Comb	0.837 ¹	0.756	0.651	0.785	0.772
Eye	0.790 ¹	0.768	0.74	0.728	0.734
Tail	0.893	0.904 ¹	0.873	0.891	0.901
Feet	0.902 ¹	0.9	0.849	0.897	0.816

¹ This means that this value has the highest score in the corresponding Bbox.

The results show that in the broiler Bbox, the recognition performance of the five algorithms reached more than 99%, among which the SSD algorithm was the highest, reaching 99.9%. In the beak and tail detection frame, YOLOV3 achieved the best results, reaching 77.4% and 90.4%, respectively. The BroilerPose algorithm proposed in this paper achieved the best results in the comb, eye, and feet (83.7%, 79.0%, and 90.2%, respectively). The precision and recall of various algorithms was shown in Table 3.

Table 3. Precision and recall of various algorithms.

	BroilerPose	YOLOV3	Faster_R-CNN	RetinaNet	SSD
Precision	0.919	0.933 ¹	0.840	0.881	0.838
Recall	0.865 ¹	0.850	0.793	0.754	0.737

¹ This means that this value has the highest score in the corresponding index.

The precision index from high to low is 93.3% (YOLOV3), 91.9% (BroilerPose), 88.1% (RetinaNet), 84.0% (Faster_R-CNN), and 83.8% (SSD).

The recall index from high to low is 86.5% (BroilerPose), 85.0% (YOLOV3), 79.3% (Faster_R-CNN), 75.4% (RetinaNet), and 73.7% (SSD).

3.2. Comparison of Posture of Different Models

To verify the pose-estimation ability of the algorithm, we selected pictures of broiler chickens from different angles for pose comparison. Figure 7 shows the partial results of the posture comparison of some broiler chickens.

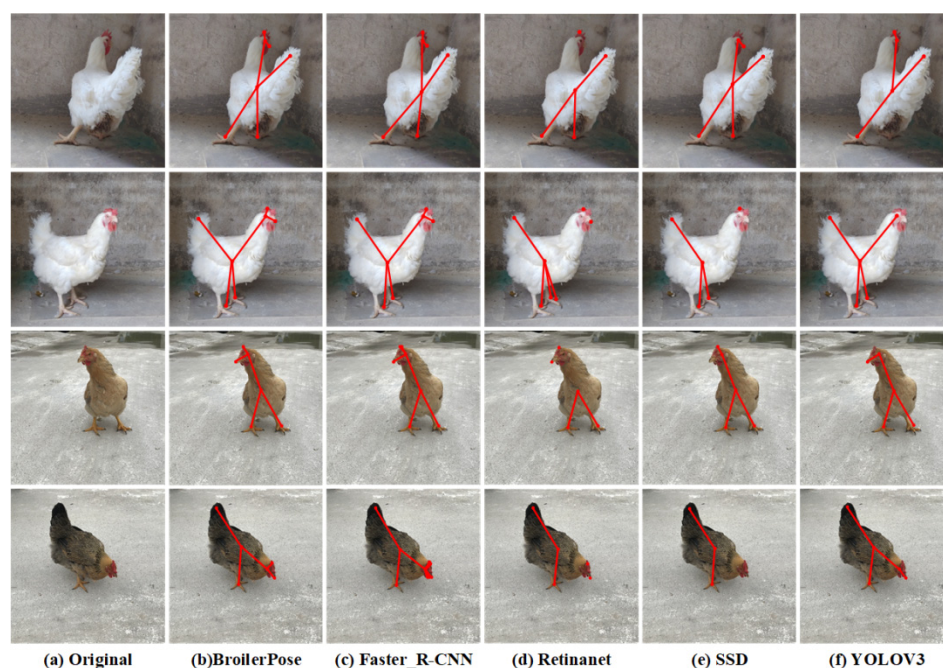


Figure 7. Partial results of posture comparison of broiler chickens.

The results of the test set were statistically analyzed. For the entire test set, the standard deviation of precision was 0.0128, the value of confidence (95%) was 0.9218 ± 0.0048 , the standard deviation of recall was 0.0266, and the value of confidence (95%) was 0.8996 ± 0.0099 . For K90, the standard deviation of precision was 0.0096, the value of confidence (95%) was 0.9255 ± 0.0053 , the standard deviation of recall was 0.0267, and the value of confidence (95%) was 0.8888 ± 0.0148 . For WRRCs, the standard deviation of precision was 0.0147, the value of confidence (95%) was 0.9181 ± 0.0081 , the standard deviation of recall was 0.0225, and the value of confidence (95%) was 0.9105 ± 0.0124 . Table 4 shows the results for two types of broilers indoors and outdoors.

Table 4. Precision and recall of various situations.

		K90	K90 (Indoor)	K90 (Outdoor)	WRRC	WRRC (Indoor)	WRRC (Outdoor)	All
Standard deviation	Precision	0.0096	0.0106	0.0092	0.0147	0.0081	0.011	0.0128
	Recall	0.0267	0.0375	0.0183	0.0225	0.0173	0.0226	0.0266
Confidence (95%)	Precision	0.9255 ± 0.0053	-	-	0.9181 ± 0.0081	-	-	0.9218 ± 0.0048
	Recall	0.8888 ± 0.0148	-	-	0.9105 ± 0.0124	-	-	0.8996 ± 0.0099

4. Discussion

The comparison between the BroilerPose pose-estimation algorithm and the other four algorithms shows that the BroilerPose pose-estimation algorithm demonstrates better pose-estimation performance.

Faster_R-CNN: The pyramid model can be used to solve the problem of the RCNN clipping scale change [26]. The attention mechanism in Natural Language Processing (NLP) is used for reference. The classification of regions of interest improves the speed of candidate box collection and has better detection for small objects; see Figure 7c.

YOLOV3 completes object positioning and classification together and returns to the bounding box's position and category at an output level [27]. Because there is no regional sampling, it has a good performance in global information, but it has a poor performance in small-scale information such as detecting the eye and comb; see Figure 7f.

SSD is an algorithm that uses a DNN to detect and classify objects in an image simultaneously. The algorithm generates a set of default boxes with different aspect ratios

and sizes and matches the default boxes with the real boxes to predict the confidence of object identification and position offset [28]. Like YOLOV3, SSD has poor performance in small-scale detection; see Figure 7e.

RetinaNet proposed a new loss function, Focal Loss, which can solve the problem of unbalanced positive and negative samples in target detection. However, in some cases where the detection target is relatively small, the recognition effect is not good; see Figure 7d.

However, by setting different IoU thresholds in the R-CNN of the network, BroilerPose performs well at small target detection; see Figure 7b.

For the test set, the standard deviation of precision between K90 and WRRC was 0.0096 and 0.0147; both test results show stability in precision. Meanwhile, both broilers had confidences (95%) of precision of more than 0.9, so the algorithm can properly work in both cases.

The BroilerPose pose-estimation algorithm performs well in mAP and F1-Score. After the accurate pose estimation, we can lay a foundation for follow-up behavior analysis. Through the analysis of the test data, the algorithm proposed in this paper has a good effect on the accuracy and recall rate of two different breeders. Furthermore, we can combine the tracking algorithm and BroilerPose pose-estimation algorithm to carry out continuous pose estimation and even behavioral analysis for single broiler chickens [29,30] to judge the movement state, health state, and welfare of broiler chickens.

5. Conclusions

In this paper, a pose-estimation algorithm based on DNN is proposed to estimate the pose of a single broiler chicken. By comparing this algorithm with other pose-estimation algorithms, the results shows that the precision and recall of this algorithm for a single broiler chicken pose estimation is 91.9% and 86.5%, respectively. The test set shows stability in the precision between K90 and WRRC, and both broilers had confidences (95%) of more than 0.9. In conclusion, the proposed method can recognize the posture of individual chickens, which is helpful for poultry researchers and accurate detection in large-scale farms.

Our method can estimate the pose of a single broiler chicken from multiple angles. In the case of multiple broiler chickens, however, there are all sorts of problems. Therefore, for future work, we hope to conduct more studies on poultry pose estimation through in-depth study.

Author Contributions: Conceptualization, C.F. and T.Z.; methodology, T.Z. and C.F.; software, C.F., H.Z. and J.Y.; validation, C.F., H.Z. and H.D.; data curation, J.Y. and H.D.; writing—original draft preparation, C.F.; writing—review and editing, T.Z. and C.F.; visualization, C.F.; supervision, T.Z.; project administration, T.Z.; funding acquisition, T.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Plan (grant No. 2021YFD1300101), Guangdong Province Special Fund for Modern Agricultural Industry Common Key Technology R&D Innovation Team (grant No. 2021KJ129) and Lingnan Modern Agricultural Science and Technology Guangdong Provincial Laboratory Maoming Laboratory independent scientific research project (grant No. 2021ZZ003), China.

Institutional Review Board Statement: The experiment was performed in accordance with the guidelines approved by the experimental animal administration and ethics committee of South China Agriculture University (SYXK-2019-0136).

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: The authors appreciate the support and assistance provided by the staff of the poultry farm of Gaoming District.

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

References

1. Benjamin, M.; Yik, S. Precision Livestock Farming in Swine Welfare: A Review for Swine Practitioners. *Animals* **2019**, *9*, 133. [[CrossRef](#)]
2. Astill, J.; Dara, R.A.; Fraser, E.D.G.; Roberts, B.; Sharif, S. Smart poultry management: Smart sensors, big data, and the internet of things. *Comput. Electron. Agric.* **2020**, *170*, 105291. [[CrossRef](#)]
3. Zheng, H.; Zhang, T.; Fang, C.; Zeng, J.; Yang, X. Design and Implementation of Poultry Farming Information Management System Based on Cloud Database. *Animals* **2021**, *11*, 900. [[CrossRef](#)]
4. Zhuang, X.; Zhang, T. Detection of sick broilers by digital image processing and deep learning. *Biosyst. Eng.* **2019**, *179*, 106–116. [[CrossRef](#)]
5. Yang, A.; Huang, H.; Zheng, B.; Li, S.; Gan, H.; Chen, C.; Yang, X.; Xue, Y. An automatic recognition framework for sow daily behaviours based on motion and image analyses. *Biosyst. Eng.* **2020**, *192*, 56–71. [[CrossRef](#)]
6. Zheng, C.; Yang, X.; Zhu, X.; Chen, C.; Wang, L.; Tu, S.; Yang, A.; Xue, Y. Automatic posture change analysis of lactating sows by action localisation and tube optimisation from untrimmed depth videos. *Biosyst. Eng.* **2020**, *194*, 227–250. [[CrossRef](#)]
7. Liu, D.; Oczak, M.; Maschat, K.; Baumgartner, J.; Pletzer, B.; He, D.; Norton, T. A computer vision-based method for spatial-temporal action recognition of tail-biting behaviour in group-housed pigs. *Biosyst. Eng.* **2020**, *195*, 27–41. [[CrossRef](#)]
8. Li, G.; Ji, B.; Li, B.; Shi, Z.; Zhao, Y.; Dou, Y.; Brocato, J. Assessment of layer pullet drinking behaviors under selectable light colors using convolutional neural network. *Comput. Electron. Agric.* **2020**, *172*, 105333. [[CrossRef](#)]
9. Zhuang, X.; Bi, M.; Guo, J.; Wu, S.; Zhang, T. Development of an early warning algorithm to detect sick broilers. *Comput. Electron. Agric.* **2018**, *144*, 102–113. [[CrossRef](#)]
10. Khan, A.Q.; Khan, S.; Ullah, M.; Cheikh, F.A. A Bottom-Up Approach for Pig Skeleton Extraction Using RGB Data. In *Lecture Notes in Computer Science, Proceedings of the 2020 International Conference on Image and Signal Processing, Marrakesh, Morocco, 4–6 June 2020*; Springer: Berlin/Heidelberg, Germany, 2020.
11. Fuentes, A.; Yoon, S.; Park, J.; Park, D.S. Deep learning-based hierarchical cattle behavior recognition with spatio-temporal information. *Comput. Electron. Agric.* **2020**, *177*, 105627. [[CrossRef](#)]
12. Riekert, M.; Klein, A.; Adrion, F.; Hoffmann, C.; Gallmann, E. Automatically detecting pig position and posture by 2D camera imaging and deep learning. *Comput. Electron. Agric.* **2020**, *174*, 105391. [[CrossRef](#)]
13. Huang, X.; Hu, Z.; Wang, X.; Yang, X.; Zhang, J.; Shi, D. An Improved Single Shot Multibox Detector Method Applied in Body Condition Score for Dairy Cows. *Animals* **2019**, *9*, 470. [[CrossRef](#)]
14. Hu, H.; Dai, B.; Shen, W.; Wei, X.; Sun, J.; Li, R.; Zhang, Y. Cow identification based on fusion of deep parts features. *Biosyst. Eng.* **2020**, *192*, 245–256. [[CrossRef](#)]
15. Marsot, M.; Mei, J.; Shan, X.; Ye, L.; Feng, P.; Yan, X.; Li, C.; Zhao, Y. An adaptive pig face recognition approach using Convolutional Neural Networks. *Comput. Electron. Agric.* **2020**, *173*, 105386. [[CrossRef](#)]
16. Salau, J.; Haas, J.H.; Junge, W.; Thaller, G. Determination of Body Parts in Holstein Friesian Cows Comparing Neural Networks and k Nearest Neighbour Classification. *Animals* **2021**, *11*, 50. [[CrossRef](#)]
17. Wutke, M.; Heinrich, F.; Das, P.P.; Lange, A.; Gentz, M.; Traulsen, I.; Warns, F.K.; Schmitt, A.O.; Gültas, M. Detecting Animal Contacts—A Deep Learning-Based Pig Detection and Tracking Approach for the Quantification of Social Contacts. *Sensors* **2021**, *21*, 7512. [[CrossRef](#)]
18. Mathis, A.; Mamidanna, P.; Cury, K.M.; Abe, T.; Murthy, V.N.; Mathis, M.W.; Bethge, M. DeepLabCut: Markerless pose estimation of user-defined body parts with deep learning. *Nat. Neurosci.* **2018**, *21*, 1281–1289. [[CrossRef](#)]
19. Pereira, T.D.; Aldarondo, D.E.; Willmore, L.; Kislin, M.; Wang, S.S.H.; Murthy, M.; Shaevitz, J.W. Fast animal pose estimation using deep neural networks. *Nat. Methods* **2019**, *16*, 117–125. [[CrossRef](#)]
20. Raman, S.; Maskeliūnas, R.; Damaševičius, R. Markerless Dog Pose Recognition in the Wild Using ResNet Deep Learning Model. *Computers* **2022**, *11*, 2. [[CrossRef](#)]
21. Vecvanags, A.; Aktas, K.; Pavlovs, I.; Avots, E.; Filipovs, J.; Brauns, A.; Done, G.; Jakovels, D.; Anbarjafari, G. Ungulate Detection and Species Classification from Camera Trap Images Using RetinaNet and Faster R-CNN. *Entropy* **2022**, *24*, 353. [[CrossRef](#)]
22. Li, G.; Hui, X.; Lin, F.; Zhao, Y. Developing and Evaluating Poultry Preening Behavior Detectors via Mask Region-Based Convolutional Neural Network. *Animals* **2020**, *10*, 1762. [[CrossRef](#)] [[PubMed](#)]
23. Lee, D.-S.; Kim, J.-S.; Jeong, S.C.; Kwon, S.-K. Human Height Estimation by Color Deep Learning and Depth 3D Conversion. *Appl. Sci.* **2020**, *10*, 5531. [[CrossRef](#)]
24. Jia, L.; Tian, Y.; Zhang, J. Domain-Aware Neural Architecture Search for Classifying Animals in Camera Trap Images. *Animals* **2022**, *12*, 437. [[CrossRef](#)] [[PubMed](#)]
25. Zuerl, M.; Stoll, P.; Brehm, I.; Raab, R.; Zanca, D.; Kabri, S.; Happold, J.; Nille, H.; Prechtel, K.; Wuensch, S.; et al. Automated Video-Based Analysis Framework for Behavior Monitoring of Individual Animals in Zoos Using Deep Learning—A Study on Polar Bears. *Animals* **2022**, *12*, 692. [[CrossRef](#)]
26. Tang, L.; Tang, W.; Qu, X.; Han, Y.; Wang, W.; Zhao, B. A Scale-Aware Pyramid Network for Multi-Scale Object Detection in SAR Images. *Remote Sens.* **2022**, *14*, 973. [[CrossRef](#)]
27. Kim, J.; Moon, N. Dog Behavior Recognition Based on Multimodal Data from a Camera and Wearable Device. *Appl. Sci.* **2022**, *12*, 3199. [[CrossRef](#)]

28. Akçay, H.G.; Kabasakal, B.; Aksu, D.; Demir, N.; Öz, M.; Erdoğan, A. Automated Bird Counting with Deep Learning for Regional Bird Distribution Mapping. *Animals* **2020**, *10*, 1207. [[CrossRef](#)]
29. Fang, C.; Huang, J.; Cuan, K.; Zhuang, X.; Zhang, T. Comparative study on poultry target tracking algorithms based on a deep regression network. *Biosyst. Eng.* **2020**, *190*, 176–183. [[CrossRef](#)]
30. 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](#)]