JAST (Journal of Animal Science and Technology) TITLE PAGE

Upload this completed form to website with submission

ARTICLE INFORMATION	Fill in information in each box below
Article Type	Research article
Article Title (within 20 words without abbreviations)	A Study of Improved Duck Detection using YOLO-based Deep Neural Networks
Running Title (within 10 words)	A Study of Improved Duck Detection
Author	Jeyoung Lee1, Hochul Kang1
Affiliation	1 Department of Digital Media, The Catholic University of Korea, 43, Jibong-ro, Bucheon-si, Gyeonggi-do, Republic of Korea
ORCID (for more information, please visit https://orcid.org)	Jeyoung Lee (https://orcid.org/0000-0002-1464-7839) Hochul Kang (https://orcid.org/0000-0002-7733-2287)
Competing interests	No potential conflict of interest relevant to this article was reported.
Funding sources State funding sources (grants, funding sources, equipment, and supplies). Include name and number of grant if available.	Not applicable.
Acknowledgements	Not applicable.
Availability of data and material	Upon reasonable request, the datasets of this study can be available from the corresponding author.
Authors' contributions Please specify the authors' role using this form.	Conceptualization: Lee JY, Kang HC. Data curation: Lee JY, Kang HC. Formal analysis: Lee JY, Kang HC Methodology: Lee JY, Kang HC Software: Lee JY, Kang HC. Validation: Lee JY, Kang HC. Investigation: Lee JY, Kang HC Writing - original draft: Lee JY, Kang HC Writing - review & editing: Lee JY, Kang HC
Ethics approval and consent to participate	This article does not require IRB/IACUC approval because there are no human and animal participants.

CORRESPONDING AUTHOR CONTACT INFORMATION

For the corresponding author (responsible for correspondence, proofreading, and reprints)	Fill in information in each box below
First name, middle initial, last name	Hochul, , Kang
Email address – this is where your proofs will be sent	hckang19@catholic.ac.kr
Secondary Email address	
Address	43, Jibong-ro, Bucheon-si, Gyeonggi-do, Republic of Korea
Cell phone number	+82-10-8354-5863
Office phone number	
Fax number	

A Study of Improved Duck Detection using YOLO-based Deep Neural Networks

Abstract

In duck cages, ducks are placed in various states. Our prior research was able to successfully detect various duck states with RetinaNet algorithms. However, the shift towards real-world applications, specifically smart farm robots, necessitates a more rapid and efficient detection model. To that end, This study investigates the utilization of YOLO-based Deep Neural Networks using duck cage datasets. YOLO is one of the most famous one-stage models for Object Detection tasks. And have many different versions. The study compares accuracy from YOLOv3, YOLOv5 to YOLOv8. Also, using various data augmentation to improve performance. The final results were visually confirmed using images different from the images used for learning. In conclusion, YOLO-based methods demonstrate their potential for effectiveness in real-world applications.

Keywords: Duck detection; Duck farming; Smart farming; Object detection; Deep neural network; Computer vision; Yolo

Introduction

In our previous research[1], we explored various states of ducks in diverse duck farming environments and examined the reasons for detecting these states. Additionally, we researched the possibility of transforming duck farming into a smart farming system using Deep Learning-based algorithms with RetinaNet[2]. However, real-world applications demand faster and more accurate methods. While RetinaNet[2] showed promising performance, it had a limit to use in real-world applications. To address this, we turn our attention to YOLO-based methods[3, 4, 5, 6, 7, 8], the most widely used object detection algorithms in real-world applications. YOLO is the one-stage model among object detection algorithms and one of the most famous and traditional networks. Therefore, extensive research has been conducted, resulting in various versions of YOLO. Our research conducts a comparative analysis, focusing on the fundamental and widely adopted versions: YOLOv3[3], YOLOv5[5], YOLOv6[6], YOLOv7[7], and YOLOv8[8]. To achieve this, we utilize our previous dataset and compare and analyze which YOLO model and RetinaNet performs. In addition, it uses more diverse data augmentation techniques than previous methods to improve performance.

In our previous work[1], we explored various smart farming technologies and related object detection methods. While various object detection algorithms have been applied in smart farming, This study is closely related to models using YOLO. Osorio, Kavir, et al. [9] conducted a comparative study of three models, Mask R-CNN [10], SVMs [11], and YOLOv3 [3], for weed detection in lettuce crops. Shojaeipour, Ali, et al. [12] developed a 2-stage model using YOLOv3 [3]-ResNet50 [13] to detect the oral area of cattle from images of their faces for livestock welfare management. Hong, Suk-Ju, et al.[14] attempted various object detection models to find birds in unmanned aerial vehicle imagery, revealing that Faster R-CNN[15] was more accurate, while YOLO[5] was faster. Wang, Xuewen, et al.[16] developed LDS-YOLO, a lightweight variation of YOLO, for the task of finding dead trees in forests, and Jiang, Kailin, et al.[17] enhanced the performance of YOLOv7 using an attention mechanism to effectively detect ducks.

Based on this background, this paper utilizes previous data to compare and analyze which YOLO model performs better under the same conditions. In conclusion, we find that YOLO-based methods[3,4,5,6,7] outperform RetinaNet[2]. The evaluation follows the previous approach, dividing objects to be detected based on a 9:1 ratio, measuring mean average precision(mAP) using separate data, and visually confirming the final results. In conclusion, we anticipate that methods utilizing YOLO can be applied effectively in real-world duck farming environments for smart farming.

Materials and Methods

Dataset

The dataset used in this study is the same of the dataset employed in our previous research[1], consisting of 2852 images. The average size image is 1748.30 and 999.94 for the width and height, and annotated to

categorize ducks into three states: normal ducks, slap ducks, and dead ducks. In total, the dataset comprises 10461, 1208 and 381 for the normal, slap and dead. The maximum number of states in a single image is 24 normal ducks, 1 fallen duck, and 1 dead duck. Ducks in various states may or may not be present in each image, and multiple states of ducks can coexist within the same image. The distribution of duck objects within the images is characterized by ratios of 0.056, 0.053, and 0.082 for normal, slap, and dead ducks, respectively. Ducks in most states appear evenly throughout the image, but dead ducks always appear below the halfway point of the image. An example of the video data provided is as shown in Fig1.

YOLO Training

YOLO comes in various versions, and our research applies different YOLO models to our dataset for performance comparison. Unlike the previous RetinaNet[2] used in our research, YOLO is known for its lightweight and fast algorithms, making it widely applied in real-world applications. Previous studies related to livestock using YOLO have predominantly used YOLOv3[3]. However, YOLOv3[3] is an older model and did not focus extensively on duck farms. The study most similar to ours, Jiang, Kailin, et al.[17], used YOLOv7[7]. Therefore, we tested from YOLOv3[3] to YOLOv7[7]. Additionally, we trained with the new YOLO model, YOLOv8[8]. The tested algorithm as shown in Table 1.

We applied our dataset to various YOLO versions and trained them under the same conditions for comparison. Firstly, we fixed the input image size at 640x640, a standard size provided by YOLOv3[3] to YOLOv8[8]. Secondly, we fine-tuned using pre-trained models on the same dataset. The use of pre-trained models for transfer learning is a conventional practice that ensures robust performance even with limited data, such as in our dataset. Thirdly, we applied the same data augmentation techniques and training methods. YOLO, being an improved algorithm over time, supports different training techniques depending on the release, which significantly impacts training performance. Therefore, we used consistent training methods and data augmentation. Further details on data augmentation are provided in the following section, and information related to training methods is described in the final section of the Methods.

Data Augmentation

In deep learning, performance tends to improve with larger datasets. However, collecting extensive data has its limitations. To address this constraint, traditional practice involves data augmentation. In our prior study[1], we also employed data augmentation, yet the data of ducks in the farm was challenged in applying various data augmentation. In this study, we expand the use of data augmentation.

Basically, we apply the left-right flip by 50% probability to augment the dataset. Moreover, we randomly rotate images by angles ranging from -20 to 20 degrees. Approximately 20% of each image is subjected to translation, and a scaling factor of approximately 0.5 is applied. While these techniques worked effectively in our prior study, in this research, we employ additional methods. Firstly, we apply a shearing transformation of up to 10 degrees. Augmentation through HSV adjustments is also incorporated. Hue values are transformed by up to 0.02 relative to the original, saturation is adjusted up to 0.7, and value is altered by up to 0.4. The MixUp technique[19] is also adopted. MixUp is a well-known augmentation method that aids in creating diverse scenarios for object detection data augmentation. The perspective transformation of approximately 0.0001 is applied, introducing variations in perspective. Each technique can be applied independently, either alone or in combination with others. An example of the augmented data is as shown in Fig2.

Train Details

For learning and validation, the data are divided into train data and validation data at a ratio of 9:1. When dividing the data, the data are divided based on classes so that the data can be divided fairly by class. Our training procedure was meticulously designed to ensure consistent evaluation across all 17 YOLO models we examined. We use pretrained weights, using the MS-COCO dataset[20] as a foundational knowledge base. To optimize the training process, we adopted Mixed Precision Training[21], harnessing the computational capabilities of our two Nvidia RTX 3090 GPUs. Training spanned 30 epochs, with an initial learning rate of 1e-2 and a gentle warm-up over the first 3 epochs for stability. A decaying learning rate strategy with a decay factor of 5e-4 was employed to facilitate efficient weight updates. With a batch size of 32, we aimed for efficient parallel processing across our hardware setup. This consistent approach, from Pretraining to hardware

configuration, ensured that each YOLO model, from YOLOv3 to YOLOv8, received uniform treatment in our rigorous evaluation.

Results

In our study, we conducted a comparative analysis between our prior research[1] and YOLO, utilizing two separate evaluation datasets. The first dataset, consisting of 270 carefully curated images, was categorized into three classes: ducks (1130 instances), slapped ducks (121 instances), and dead ducks (39 instances). We calculated and presented the Average Precision (AP) scores, parameter counts(Params), and FLOPs (Floating Point Operations) for each model, as outlined in Table 1. The results from Table 1 reveal that YOLO requires fewer parameters and fewer computations than RetinaNet[2], while maintaining superior accuracy. Furthermore, among the YOLO versions, YOLOv8[8] exhibited better performance compared to other versions. YOLOv3[3] demonstrated strong performance but needed many parameters. In contrast, YOLOv5[5] had the lowest parameters but showed less impressive results. Based on these findings, we determined that training with YOLOv8[8] yielded the most favorable outcomes.

Next, we tested the dataset with applied data augmentation techniques, designed to validate the models' ability to generalize to diverse scenarios. For this assessment, we adhered to the same data augmentation methods used in our prior research, to enable a direct comparison. In contrast to our previous study[1], we observed instances where augmented images contained ducks partially outside the image frame. To ensure fairness, these images were excluded, resulting in a total of 2768 images for evaluation. We measured AP, class-specific AP, and mean inference time per image. The results as shown in Table 2, reveal that YOLO models exhibit lower AP values compared to RetinaNet[2]. YOLO excels at detecting live ducks, while RetinaNet[2] shows superior performance in detecting other states. However, considering our research's primary objective of real-world applicability, a faster and more accurate model becomes imperative. According to Table 2, the AP differences are marginal (within 0.03), but YOLO models outperform RetinaNet with an impressive maximum inference speedup of up to 25.4 times. Therefore, for real-world deployment, YOLO emerges as an excellent choice. An example of the result is as shown in Fig 3, Fig 4.

Discussion

Our comparative analysis of YOLO models and RetinaNet[2], as detailed in the results section, underscores several important considerations. To begin with, YOLO models, particularly YOLOv8[8], showcase their efficiency in object detection tasks. Despite having fewer parameters and requiring fewer computations, they exhibit sufficiently accurate and faster performance. This suggests the potential for YOLO models to be deployed in resource-constrained environments especially like a smart farm where computational efficiency is paramount. One crucial aspect revealed by our results is the trade-off between accuracy and speed. While RetinaNet[2] is a little bit superior in accuracy, YOLO models excel at rapidly detecting, making them more suitable for real-time applications. This balance between speed and accuracy is a crucial determinant for the choice of model in practical implementations. The inference speedup of up to 25.4 times is a remarkable advantage in real-world situations. Our research primarily focuses on real-world applicability in duck farming environments. Therefore, it's the rapid detection capabilities of YOLO models that make them a more favorable choice for our intended use case. However, our study reveals specific limitations that demand attention. Firstly, YOLO models' performance on augmented data leaves room for improvement. Addressing this limitation necessitates the acquisition of more extensive and diversified datasets, encompassing a broader spectrum of scenarios to enhance the models' generalization abilities. Secondly, the dataset exhibits class imbalance, highlighting the need for a more balanced dataset with equal class representation. Future research efforts should prioritize techniques such as oversampling, undersampling, or the use of specialized loss functions to mitigate this imbalance.

In conclusion, our research demonstrates the value of adopting YOLO models in real-world duck farming environments. The speed and efficiency they offer are well-suited for practical deployments. These findings contribute to the broader field of object detection, where the choice of model should be guided by the specific demands like smart farm of the application.

Acknowledgments

This work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry(IPET) and Korea Smart Farm R&D Foundation(KoSFarm) through Smart Farm Innovation Technology Development Program, funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA) and Ministry of Science and ICT(MSIT), Rural Development Administration(RDA) (grant number: 421024-04). And this work was supported by Korea Institute of Planning and Evaluation for Technology in Food, Agriculture and Forestry(IPET) funded by Ministry of Agriculture, Food and Rural Affairs(MAFRA) (grant number: 321092-03-1-HD030). And this research was supported by the Catholic University of Korea.

References

1. Lee J, Kang H. A Study of Duck Detection using Deep Neural Network based on RetinaNet Model in Smart Farming. Journal of Animal Science and Technology. 2023. https://doi.org/10.5187/jast.2023.e76

2. Lin T-Y, Goyal P, Girshick R, He K, Dollár P, editors. Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision; 2017. https://doi.org/10.1109/ICCV.2017.324

3. Redmon J, Farhadi A. Yolov3: An incremental improvement. arXiv preprint arXiv:180402767. 2018. https://doi.org/10.48550/arXiv.1804.02767

4. Bochkovskiy A, Wang C-Y, Liao H-YM. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:200410934. 2020. https://doi.org/10.48550/arXiv.2004.10934

5. Jocher G, Chaurasia A, Stoken A, Borovec J, Kwon Y, Fang J, et al. ultralytics/yolov5: v6. 1-TensorRT, TensorFlow edge TPU and OpenVINO export and inference. Zenodo. 2022. https://doi.org/10.5281/zenodo.6222936

6. Li C, Li L, Jiang H, Weng K, Geng Y, Li L, et al. YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:220902976. 2022. https://doi.org/10.48550/arXiv.2209.02976

7. Wang C-Y, Bochkovskiy A, Liao H-YM, editors. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2023.https://doi.org/10.48550/arXiv.2207.02696

8. Reis D, Kupec J, Hong J, Daoudi A. Real-Time Flying Object Detection with YOLOv8. arXiv preprint arXiv:230509972. 2023. https://doi.org/10.48550/arXiv.2305.09972

9. Osorio K, Puerto A, Pedraza C, Jamaica D, Rodríguez L. A deep learning approach for weed detection in lettuce crops using multispectral images. AgriEngineering. 2020;2(3):471-88. https://doi.org/10.3390/agriengineering2030032.

10. He K, Gkioxari G, Dollár P, Girshick R, editors. Mask r-cnn. Proceedings of the IEEE international conference on computer vision; 2017. https://doi.org/10.1109/ICCV.2017.322

11. Noble WS. What is a support vector machine? Nature biotechnology. 2006;24(12):1565-7. https://doi.org/10.1038/nbt1206-1565.

12. Shojaeipour A, Falzon G, Kwan P, Hadavi N, Cowley FC, Paul D. Automated muzzle detection and biometric identification via few-shot deep transfer learning of mixed breed cattle. Agronomy. 2021;11(11):2365. https://doi.org/10.3390/agronomy11112365.

13. He K, Zhang X, Ren S, Sun J, editors. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. https://doi.org/10.1109/CVPR.2016.90

14. Hong S-J, Han Y, Kim S-Y, Lee A-Y, Kim G. Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery. Sensors. 2019;19(7):1651. https://doi.org/10.3390/s19071651

15. Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems. 2015;28. https://doi.org/10.48550/arXiv.1506.01497

16. Wang X, Zhao Q, Jiang P, Zheng Y, Yuan L, Yuan P. LDS-YOLO: A lightweight small object detection method for dead trees from shelter forest. Computers and Electronics in Agriculture. 2022;198:107035. https://doi.org/10.1016/j.compag.2022.107035

17. Jiang K, Xie T, Yan R, Wen X, Li D, Jiang H, et al. An attention mechanism-improved YOLOv7 object detection algorithm for hemp duck count estimation. Agriculture. 2022;12(10):1659. https://doi.org/10.3390/agriculture12101659

18. Niu Z, Zhong G, Yu H. A review on the attention mechanism of deep learning. Neurocomputing. 2021;452:48-62. https://doi.org/10.1016/j.neucom.2021.03.091

19. Zhang H, Cisse M, Dauphin YN, Lopez-Paz D. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:171009412. 2017. https://doi.org/10.48550/arXiv.1710.09412

20. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al., editors. Microsoft coco: Common objects in context. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13; 2014: Springer. https://doi.org/10.1007/978-3-319-10602-1_48

21. Micikevicius P, Narang S, Alben J, Diamos G, Elsen E, Garcia D, et al. Mixed precision training. arXiv preprint arXiv:171003740. 2017. https://doi.org/10.48550/arXiv.1710.03740

Tables and Figures

Table 1. Results of validation image.

Model	Input Size (pixels)	AP50	AP50-95	params(M)	FLOPs(G)

RetinaNet-1x-Aug[1]	640	0.978	0.659	38	206
RetinaNet-3x-Aug[1]	640	0.978	0.683	38	206
YOLOv3[3]	640	0.982	0.800	61.9	156.4
YOLOv5-N[5]	640	0.982	0.753	**1.9**	**4.5**
YOLOv5-S[5]	640	0.963	0.608	7.2	16.5
YOLOv5-M[5]	640	0.965	0.623	21.4	49.0
YOLOv5-L[5]	640	0.967	0.558	47.0	109.1
YOLOv5-X[5]	640	0.964	0.581	87.7	205.7
YOLOv6-N[6]	640	0.974	0.756	4.7	11.4
YOLOv6-S[6]	640	0.977	0.791	18.5	45.3
YOLOv6-M[6]	640	0.976	0.810	34.9	85.8
YOLOv6-L[6]	640	0.979	0.812	59.6	150.7
YOLOv7[7]	640	0.954	0.480	36.9	104.7
YOLOv7-X[7]	640	0.960	0.444	71.3	189.9
YOLOv8-N[8]	640	0.982	0.771	3.2	8.7
YOLOv8-S[8]	640	0.983	0.803	11.2	28.6
YOLOv8-M[8]	640	0.983	**0.815**	25.9	78.9
YOLOv8-L[8]	640	0.983	0.812	43.7	165.2
YOLOv8-X[8]	640	**0.984**	0.813	68.2	257.8

Table 2. Augmentation validation image result.

Model	AP50	AP50-95	AP duck	AP slapped	AP dead	Average Inference Time(image/s)
RetinaNet-1x-Aug[1]	0.9687	**0.6808**	0.600	**0.745**	**0.697**	0.0325
RetinaNet-3x-Aug[1]	0.96857	0.6779	0.603	**0.745**	**0.684**	0.0381

YOLOv8-N[8]	0.978	0.649	0.634	0.682	0.630	**0.0015**
YOLOv8-S[8]	**0.980**	0.678	**0.669**	0.712	0.654	0.0027
YOLOv8-M[8]	0.979	0.665	0.658	0.705	0.632	0.0086
YOLOv8-L[8]	**0.980**	0.641	0.636	0.667	0.619	0.0080
YOLOv8-X[8]	0.978	0.625	0.626	0.662	0.586	0.0136

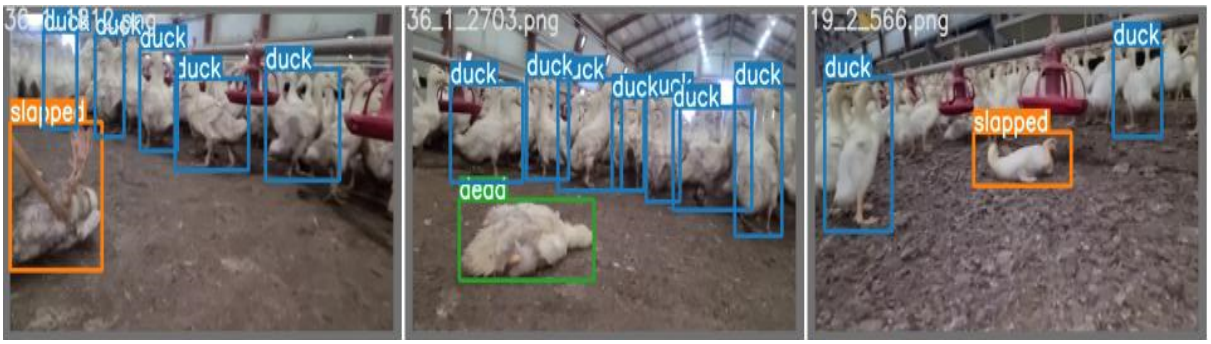

Fig1. duck farm dataset examples

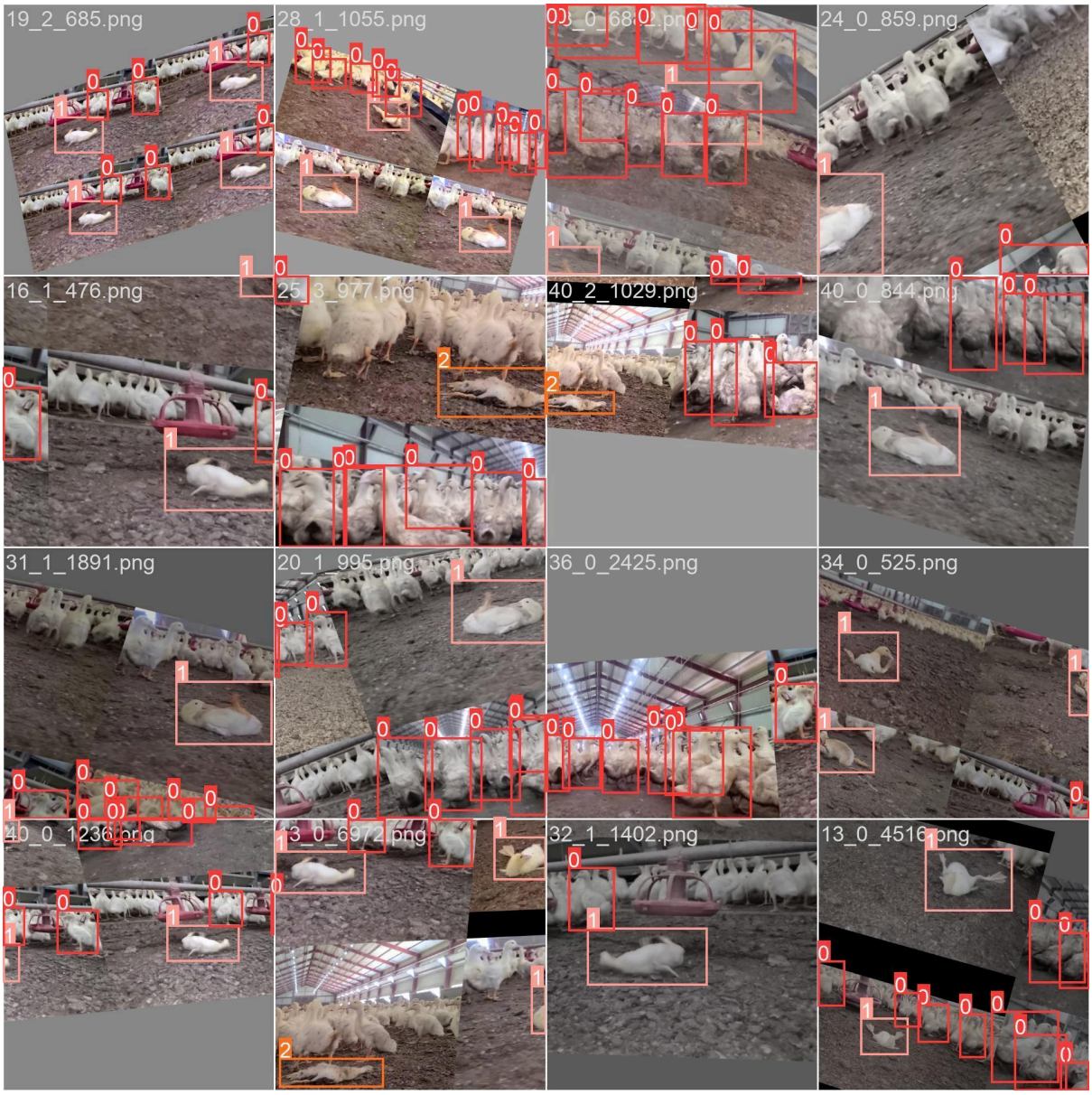

Fig2. Data Augmentation Example

Fig3. Result example. left graound truth, right pred

Fig4. Result example. left graound truth, right pred