# 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 Duck Detection using Deep Neural Network based on RetinaNet Model in Smart Farming |
| **Running Title (within 10 words)** | A Study of 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 )<br>Hochul Kang (https://orcid.org/0000-0002-7733-2287 ) |
| **Competing interests** | No potential conflict of interest relevant to this article was reported. |
| **Funding sources**<br>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**<br>Please specify the authors' role using this form. | Conceptualization: Lee JY, Kang HC.<br>Data curation: Lee JY, Kang HC.<br>Formal analysis: Lee JY, Kang HC<br>Methodology: Lee JY, Kang HC<br>Software: Lee JY, Kang HC.<br>Validation: Lee JY, Kang HC.<br>Investigation: Lee JY, Kang HC<br>Writing - original draft: Lee JY, Kang HC<br>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 | |

## Abstract

In a duck cage, ducks are placed in various states. In particular, if a duck is overturned and falls or dies, it will adversely affect the growing environment. In order to prevent the foregoing, it was necessary to continuously manage the cage for duck growth. This study proposes a method using an object detection algorithm to improve the foregoing. Object detection refers to the work to perform classification and localization of all objects present in the image when an input image is given. To use an object detection algorithm in a duck cage, data to be used for learning should be made and the data should be augmented to secure enough data to learn from. In addition, the time required for object detection and the accuracy of object detection are important. The study collected, processed, and augmented image data for a total of two years in 2021 and 2022 from the duck cage. Based on the objects that must be detected, the data collected as such were divided at a ratio of 9 : 1, and learning and verification were performed. The final results were visually confirmed using images different from the images used for learning. The proposed method is expected to be used for minimizing human resources in the growing process in duck cages and making the duck cages into smart farms.



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

# Introduction

In a duck cage, ducks are placed in various states. In particular, if a duck is overturned and falls or a duck is dead during growth, a person must make the duck stand up or collect the duck. To that end, it was necessary for humans to continuously manage the cage during the growing process of ducks. In order to improve the foregoing, this study proposes a method to use an object detection algorithm to utilize a robot in a duck cage to observe ducks to check if any duck fell or died and make any duck fell stand up and collect any duck dead. According to Zaidi, Syed Sahil Abbas, et al. [24], object detection means the work to classifying and localize all objects present in the image when an input image is given. Object detection algorithms can be largely divided into one-stage methods and two-stage methods, and each method has advantages and disadvantages. The one-stage method is faster but less accurate. Data are necessary to train AI algorithms. In particular, a lot of processed data is required to use an object detection algorithm. However, there is no processed public data about the state of ducks in a duck cage environment. Therefore, in order to detect objects in the duck cage, it was necessary to firsthand collect, process, and augment data. This study collected, processed, and augmented image data from a duck cage for a total of two years of 2021 and 2022. The data collected as such will be discussed again in Materials and Methods. Finally, among the one-stage algorithms, RetinaNet [9] was used for learning and experiment. Unlike published data, data collected firsthand have many limitations. In particular, problems of the limited number of data and the imbalance of the correct answer to the data often occur. RetinaNet [9] is the most common algorithm that enables solving the imbalance problem of correct answers in collected data. By utilizing RetinaNet, it is possible to solve the bias of learning models created by the problems of imbalance of correct answers in data caused by relatively insufficient data collection.

This study is closely related to object detection in smart farms. Gikunda, Patrick Kinyua, and Nicolas Jouandeau [13] and Dhanya, V. G., et al. [22] collected and investigated cases where artificial intelligence was used in relation to smart farms. Dhanya, V. G., et al. [22] state that the agricultural industry is going through a process of rapid digital transformation and that technology is being made more powerful by state-of-the-art approaches such as artificial intelligence technology. Sa, Inkyu, et al. [5] proposes a DeepFruits model that finds about five kinds of fruits, such as sweet pepper and rockmelon, in a greenhouse using Faster R-CNN [2]. Bargoti, Suchet, and James Underwood [6] propose a method for finding apples, mangos, and almonds in an orchard by applying the DeepFruits [5] network. Sørensen, René A., et al. [10] propose a method for finding thistles that cause loss in crop yield using DenseNet [11] based on aerial photographs of crops. Albuquerque, Caio KG, et al. [15] studies a method for identifying water in a watering machine based on Mask R-CNN [7] in image frames captured by an unmanned aerial vehicle (UAV). Osorio, Kavir, et al. [16] compared and analyzed Mask R-CNN [7], SVMs [1], and YOLOv3 [12] for methods to detect weeds in lettuce crops. Riekert, Martin, et al. [17] conducted a study on a method to find a pig's position using Faster R-CNN [2]. Tedesco-Oliveira, Danilo, et al. [18] applied Faster R-CNN [2] and SSD [4] to study the development of an automated system for predicting cotton yields from color images acquired with a simple mobile device. Zhou, Zhongxian, et al. [19] compared various back-bone networks of SSD [4] to conduct a study on a method to find kiwi fruit in real time. Tang, Jiwen, et al. [21] propose a method of applying object detection to detect the distribution and precise shape of center pivot irrigation systems. Shojaeipour, Ali, et al. [20] applied two-stage YOLOv3 [12]-ResNet50 [3] to study a method for detecting the mouth region of a cow from a cow face image dataset for livestock welfare and management. Syed-Ab-Rahman et al. [23] propose an end-to-end anchor-based model to detect and classify citrus disease states.

Based on this, our paper analyzes the method of directly collecting, processing, and augmenting data for object detection on the state of ducks in a duck cage, and the application and the results of application of object detection algorithms. In order to check whether learning is successfully carried out using the collected data, the data are divided at a ratio of 9:1 based on the objects that must be detected and are learned and verified. As for the evaluation, the average precision is measured using the separated data for evaluation, and the final result is visually checked using images different from the images used for learning. The proposed method is expected to be used for minimizing human resources in the growing process in duck cages and making smart farms.

# Materials and Methods

## Data Collection

Data collection and generation is one of the most important and time-consuming tasks in any field of artificial intelligence. In this study, the data necessary for object detection are largely the video data of ducks in the duck cage, the bounding boxes that specify the locations of ducks by image frame, and the state class labels. However, there are no studies similar to this or it is not a common situation. That is, there is no public data. Therefore, this study proceeds from the data collection stage. When raising ducks in duck cages, ducks are not raised from eggs. Generally, baby ducks hatched from eggs are brought to a duck cage and raised, and all are delivered after a certain age. This is a characteristic of broiler ducks, and because of this characteristic, it is difficult to secure a large amount of data. However, deep learning requires a large amount of data in various types. To solve this problem, this study received image data directly from the duck cage over two years, 2021 and 2022, and uses techniques such as data augmentation. When receiving video data, the main point of view is whether the video has an appropriate height that can be used in real situations and whether the duck states are sufficiently diverse. An example of the video data provided is as shown in fig2.

## Data Labeling

The training images are extracted from the video as frames at the duck farm in 2021, and the bounding boxing and class labeling are carried out directly by human hands. There are three states where ducks can exist in the image: normal, fallen, and dead. In this case, as the length of the video increases, the number of frames becomes too large. As a result, the differences between the images between the frames of the video are not large, and as the video moves, frames where it is difficult to recognize the shapes of the ducks occur. In addition, when humans firsthand create labels, as the number of images increases, the problem of taking longer time also occurs. That is, taking and using all image frames is not good for learning and only increases the data generation time. In order to solve this problem, this study selected only one image per 5 to 10 frames, and labeled the 1285 images selected as such first. Duck cages raise large numbers of ducks. Therefore, when labeling an image for object detection, there is a problem that the number of ducks is excessively large, and ducks are dense. To solve this problem, it is necessary to clarify criteria when creating labels and to establish common rules. In this study, labels are created based on the duck in the frontmost of the image. In addition, only those ducks whose face, body, tail, and feet are clearly identified are identified in the normal state. The characteristics of the dataset created are examined with the labels and images created with the rule. Some problems were found due to the labeling results of the 2021 data. The ratios of dead ducks and fallen ducks in the data are overwhelmingly insufficient. This study solves this problem in three methods. First, we added more data which is provided in 2022 for improving the performance of the detection, and apply it to train. Second, we solved the problem by augmenting insufficient data using a data augmentation technique. Finally, the focal loss proposed in RetinaNet [9] is used. Focal loss was proposed to solve the class imbalance problem. The problem that humans firsthand carry out labeling one by one occurs. If labeling is carried out by humans, there is the problem that a long time is taken, and the stability of the label cannot be guaranteed. To solve the foregoing problems, the object detection model was first trained using the 2021 data. Thereafter, using the model, an automatic labeling program was created. Based on the program, the 2022 duck cage image data provided later were extracted by image frame, and thereafter, labeling was carried out first using an automatic labeling program. Finally, the labeling was inspected and corrected by humans to save time and improve stability. As such, 2852 images and labels were finally created. An example of a label created as such is shown in fig 4.

## Dataset

The number of data sets finally created is 2852. The average size of the image is 1748.30 and 999.94 for the width and height, respectively, and the total numbers of normal ducks, fallen ducks, and dead ducks in all images are 10461, 1208, and 381, respectively. The maximum number of normal ducks, fallen ducks, and dead ducks in one image is 24, 1, and 1, respectively.  Ducks in all states may or may not exist. Also, ducks in various states may appear simultaneously. The ratios of one duck object to image are 0.056, 0.053, and 0.082, respectively. Ducks in most states appear evenly throughout the image, but dead ducks always appear below the halfway of the image. [table. 1]

## RetinaNet Training

The purpose of this study is to find duck objects in the duck cage in real time. There are many similar object detection algorithms. However, as a characteristic of the collected datasets, the ratio of fallen ducks and dead ducks is overwhelmingly lower than that of normal ducks. This problem is called the state imbalance problem. To solve this problem, this study uses RetinaNet [9]. RetinaNet [9] has the advantage that the backbone model and the region proposal network can be freely changed. In addition, it is easy to apply new datasets because many studies have been conducted. Furthermore, the introduction of the focal loss solves the problem of state imbalance to some extent. The focal loss is an extended version of the cross entropy loss that reduces the weights of easy examples and focuses learning on difficult examples. Finally, real-time object detection is possible because it is a one-stage model. Therefore, RetinaNet [9] is used as the basic model of this study. A figure of the learning pipeline using RetinaNet [9] is as shown in fig5.

## Data Augmentation

The more the data used in deep learning, the better the deep learning. However, the total number of data used in this study is 2852. Many studies try to obtain more data for learning. However, when it is difficult to secure additional data, data are increased through data augmentation. This study augments data before using the data for learning. The techniques used in that case are brightness conversion, contrast conversion, saturation conversion, rotation, random resize, and flip. For brightness, contrast, and saturation conversions, values between 0.9 and 1.1 are randomly applied based on the image value. In the case of rotation, values between -20 degrees and 20 degrees are applied according to the characteristics of the image. Flip is applied left and right, and the application probability is 0.5. For random resize, a length of one of 640, 672, 704, 736, 768, and 800 is selected based on the length of the shortest side, and the length of the longest side is up to 1333. Finally, each technique is applied independently of the other. That is, several techniques may be applied at the same time, or none may be applied. Fig6 is an example of an image to which augmentation was applied.

### Fine Tuning

Fine-tuning is a method used to train one's own model based on an existing model that has been trained. Many deep learning approaches use fine-tuning to achieve a task. In this study too, the RetinaNet [9] model pretrained using the COCO dataset is fined-tuned and learned. There are two models prepared for fine-tuning, 1x model and 3x model, which will be used depending on the schedule. He, Kaiming, Ross Girshick, and Piotr Dollár [14] questioned fine-tuning and studied a new way of learning. They introduce training scheduling techniques, batch normalization, and methods that do not use fine-tuning. According to them, a learning schedule to search the COCO Dataset once based on the COCO Dataset is defined as a 1x schedule. That is, the prepared 1x pretraining model means a model that searches the COCO dataset once, carries out 90000 iterations, and has learning rates reduced to 1/10 at 60k and 80k. The 3x pretraining model is a model that searches the COCO dataset twice, caries out 270000 iterations, and has the learning rate reduced to 1/10 every 210k and 250k. In this study, both models are used for learning and the results are compared thereafter.

### 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. In addition, a total of three models are learned: a model to which data augmentation was not applied, a model to which data augmentation was partially applied, and a model to which data augmentation was fully applied. As for the model to which data augmentation was partially applied, it was found that the model to which only random resize and random flip were applied as elements found during learning performed better. Details can be found in Result Section. The basic RetinaNet [9] used in learning is a combination of ResNet50 [3] and FPN [8]. In addition, two models trained on the COCO dataset were prepared. We fine-tune from the two prepared models. In this case, focal loss is used as the loss and SGD is used as the optimizer. The basic learning rate is 1e-3, and the warm-up scheduler and the step scheduler are used as the learning schedulers. Therefore, the learning rate is first warmed up to 1000 iterations. The step scheduler reduces the basic learning rate by 1e-1 each at the last iterations, 5000 and 6000 iterations. The batch size is 16 and the iteration is 7000. One RTX 3090 was used for learning, and the time taken for the learning was about 2 hours.

## Results

The most commonly used value to measure performance in object detection is average precision (AP). In short, AP means the percentage of correct answers in the predicted boxes. AP is again divided into AP50, AP75, etc. according to the ratio of intersection over union (IoU) according to the degree of overlap between the predicted box and the correct answer box. AP means the average accuracy measurement method for all ratios of IoU, which increases by 0.05 from 0.5 to 0.95, AP50 means when IoU is greater than 0.5, and AP75 means when IoU is greater than 0.75. In this study, how accurate the combination of basic ResNet50 [3] and FPN [8] is checked for each AP according to the pretraining model and whether augmentation is carried out. Table 2. is a table of measurement of AP for 270 pieces of validation data. Table 3 is the result of measurement of AP by class for the same validation data.

According to Table 2 and Table 3, it can be seen that the performance of the 3x model is basically higher than that of the 1x model. In addition, the performance of the model to which only random resize and flipping were applied is superior to that of the model to which full augmentation was applied for validation data. It can be seen that excessive augmentation does not help the validation performance because the number of validation data is small, and the images are mainly those images with angles and shapes similar to those of the learning images. However, this is far from generalization, which is the goal of learning. Therefore, the validation data are augmented through flipping and rotation to generate 2770 validation data, and more general performance is measured thereafter. The results are in Table 4 and Table 5 below.

Through the results in Table 4 and Table 5, it can be seen that the generalization performance of the model to which full augmentation was applied is better. Therefore, in this study, the test is conducted using a model to which full augmentation is applied. In addition, between the 1x model and the 3x model, the 3x model generally has better performance. However, in the present evaluation, the average AP performance of the 1x model was shown to be better. Since the AP75 performance of the 3x model was better, the 3x model was used and applied to images different from the images used for learning and evaluation. Because the images to which the models were applied as such have no information of the actual objects, it was checked with eyes whether the images were searched well. The results checked with the eyes are as shown in fig 7, fig 8, and fig 9.

In addition, the average inference time per one image for all models is within 0.003 seconds. This shows that the inference time of this model is short and effective. Therefore, the model can be used for real-time detection.

# Discussion

This study collected and defined anomalous object detection datasets for making a smart farm for anomalous duck detection in a duck cage environment. Thereafter, using the datasets, learning and evaluation were caried out utilizing RetinaNet, a one-stage network. Finally, for good results, image augmentation, warm-up scheduler, etc. were used for comparison to explore the best algorithm between basic ResNet50 and FPN models. The datasets defined through the foregoing were shown to be usable and basic model guidelines were established. However, there are some limitations. First, the backbone network was not changed. In the case of object detection, the performance varies greatly depending on the size of the backbone network and the method of the region of interest network. If the size of the backbone model is increased, the accuracy will increase. However, due to the definition of the problem that objects should be detected in real time, a search process to find a network of an appropriate size is necessary. Second, a method that uses an object detection model other than RetinaNet is necessary. RetinaNet is a network that has been studied a lot and has characteristics suitable for solving our problems, but it is also an old model. This means that experiments should be caried out on other models that advanced RetinaNet while retaining the features. Finally, research on the improvement of a new network tailored to the datasets is needed. Currently, we applied our datasets based on a famous model and focused on exploring how well it performs. A study like this is also a study, and through this, we showed that our problem definition is solvable and that our datasets can be used well in a general model. However, this does not mean that general models published well fit our datasets. Research on new models that fit the characteristics of our datasets is also needed. All of these limitations will be addressed in the future based on this study by utilizing and developing the insights found in this study.



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

2. 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.

3. 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.

4. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, et al., editors. Ssd: Single shot multibox detector. European conference on computer vision; 2016: Springer.

5. Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCool C. Deepfruits: A fruit detection system using deep neural networks. sensors. 2016;16(8):1222. doi: https://doi.org/10.3390/s16081222.

6. Bargoti S, Underwood J, editors. Deep fruit detection in orchards. 2017 IEEE international conference on robotics and automation (ICRA); 2017: IEEE.

7. He K, Gkioxari G, Dollár P, Girshick R, editors. Mask r-cnn. Proceedings of the IEEE international conference on computer vision; 2017.

8. Lin T-Y, Dollár P, Girshick R, He K, Hariharan B, Belongie S, editors. Feature pyramid networks for object detection. Proceedings of the IEEE conference on computer vision and pattern recognition; 2017.

9. 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.

10. Sørensen RA, Rasmussen J, Nielsen J, Jørgensen RN, editors. Thistle detection using convolutional neural networks. EFITA WCCA 2017 Conference, Montpellier Supagro, Montpellier, France; 2017.

11. Zhu Y, Newsam S, editors. Densenet for dense flow. 2017 IEEE international conference on image processing (ICIP); 2017: IEEE.

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

13. Gikunda PK, Jouandeau N, editors. State-of-the-art convolutional neural networks for smart farms: A review. Intelligent computing-proceedings of the computing conference; 2019: Springer.

14. He K, Girshick R, Dollár P, editors. Rethinking imagenet pre-training. Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019.

15. Albuquerque CK, Polimante S, Torre-Neto A, Prati RC, editors. Water spray detection for smart irrigation systems with mask r-cnn and uav footage. 2020 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor); 2020: IEEE.

16. 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. doi: https://doi.org/10.3390/agriengineering2030032.

17. Riekert M, Klein A, Adrion F, Hoffmann C, Gallmann E. Automatically detecting pig position and posture by 2D camera imaging and deep learning. Computers and Electronics in Agriculture. 2020;174:105391. doi: https://doi.org/10.1016/j.compag.2020.105391.

18. Tedesco-Oliveira D, da Silva RP, Maldonado Jr W, Zerbato C. Convolutional neural networks in predicting cotton yield from images of commercial fields. Computers and electronics in agriculture. 2020;171:105307. doi: https://doi.org/10.1016/j.compag.2020.105307.

19. Zhou Z, Song Z, Fu L, Gao F, Li R, Cui Y. Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation. Computers and Electronics in Agriculture. 2020;179:105856. doi: https://doi.org/10.1016/j.compag.2020.105856.

20. 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. doi: https://doi.org/10.3390/agronomy11112365.

21. Tang J, Arvor D, Corpetti T, Tang P. Mapping center pivot irrigation systems in the southern Amazon from Sentinel-2 images. Water. 2021;13(3):298. doi: https://doi.org/10.3390/w13030298.

22. Dhanya V, Subeesh A, Kushwaha N, Vishwakarma D, Kumar TN, Ritika G, et al. Deep learning based computer vision approaches for smart agricultural applications. Artificial Intelligence in Agriculture. 2022. doi: https://doi.org/10.1016/j.aiia.2022.09.007.

23. Syed-Ab-Rahman SF, Hesamian MH, Prasad M. Citrus disease detection and classification using end-to-end anchor-based deep learning model. Applied Intelligence. 2022;52(1):927-38. doi: https://doi.org/10.1007/s10489-021-02452-w.

24. Zaidi SSA, Ansari MS, Aslam A, Kanwal N, Asghar M, Lee B. A survey of modern deep learning based object detection models. Digital Signal Processing. 2022:103514. doi: https://doi.org/10.1016/j.dsp.2022.103514.

# Tables and Figures

Table 1. Dataset Information

| | Total Number | Max Number | Min Number | Avg region rate | min top left x | min top left y | max top left x | max top left y |
|---|---|---|---|---|---|---|---|---|
| Duck | 10461 | 24 | 0 | 0.0563 | 0.00 | 0.00 | 1818.65 | 896.53 |
| Slap | 1208 | 1 | 0 | 0.0531 | 0.00 | 0.00 | 1380.64 | 850.54 |
| Dead | 381 | 1 | 0 | 0.0825 | 0.00 | 97.48 | 1611.73 | 832.36 |

Table 2. Duck detection RetinaNet result

| **backbone** | **scheduler** | **augmentation** | **AP** | **AP50** | **AP75** |
|---|---|---|---|---|---|
| Resnet50-FPN | 1x | none | 73.969 | 97.035 | 87.633 |
| Resnet50-FPN | 3x | none | 74.630 | 97.046 | 88.686 |
| Resnet50-FPN | 1x | part | 79.599 | 98.060 | 91.569 |
| Resnet50-FPN | 3x | part | **79.797** | **98.023** | **91.569** |
| Resnet50-FPN | 1x | all | 66.286 | 97.788 | 81.559 |
| Resnet50-FPN | 3x | all | 67.101 | 97.711 | 84.954 |

Table 3. Duck detection RetinaNet result by class

| **backbone** | **scheduler** | **augmentation** | **Duck** | **Slap** | **Dead** |
|---|---|---|---|---|---|
| Resnet50-FPN | 1x | none | 62.291 | 76.794 | 82.821 |
| Resnet50-FPN | 3x | none | 61.985 | 79.549 | 82.357 |
| Resnet50-FPN | 1x | part | 68.187 | 84.082 | **86.527** |
| Resnet50-FPN | 3x | part | **68.467** | **85.362** | 85.563 |
| Resnet50-FPN | 1x | all | 58.852 | 72.208 | 67.797 |
| Resnet50-FPN | 3x | all | 59.518 | 72.910 | 68.876 |

Table 4. Duck detection augmentation validation data RetinaNet result

| **backbone** | **scheduler** | **augmentation** | **AP** | **AP50** | **AP75** |
|---|---|---|---|---|---|
| Resnet50-FPN | 1x | none | 34.413 | 86.847 | 16.997 |
| Resnet50-FPN | 3x | none | 33.917 | 86.609 | 16.562 |
| Resnet50-FPN | 1x | part | 37.432 | 91.314 | 20.255 |
| Resnet50-FPN | 3x | part | 37.340 | 90.682 | 19.787 |
| Resnet50-FPN | 1x | all | **70.984** | 97.182 | 88.584 |
| Resnet50-FPN | 3x | all | 70.784 | **97.361** | **89.745** |

Table 5. Duck detection augmentation validation data RetinaNet result by class

| backbone | scheduler | augmentation | Duck | Slap | Dead |
|---|---|---|---|---|---|
| Resnet50-FPN | 1x | none | 32.513 | 38.990 | 31.737 |
| Resnet50-FPN | 3x | none | 32.061 | 37.264 | 32.426 |
| Resnet50-FPN | 1x | part | 37.408 | 41.936 | 32.953 |
| Resnet50-FPN | 3x | part | 37.499 | 41.278 | 33.244 |
| Resnet50-FPN | 1x | all | **62.786** | 76.781 | **73.386** |
| Resnet50-FPN | 3x | all | 64.474 | **76.871** | 71.007 |

(a) (b) (c)

fig1. (a) Original Image, (b) Ground Truth Image, (c) Predict Image

(a) (b) (c)

fig2. original data example (a) slap example, (b) dead example, (c) normal example

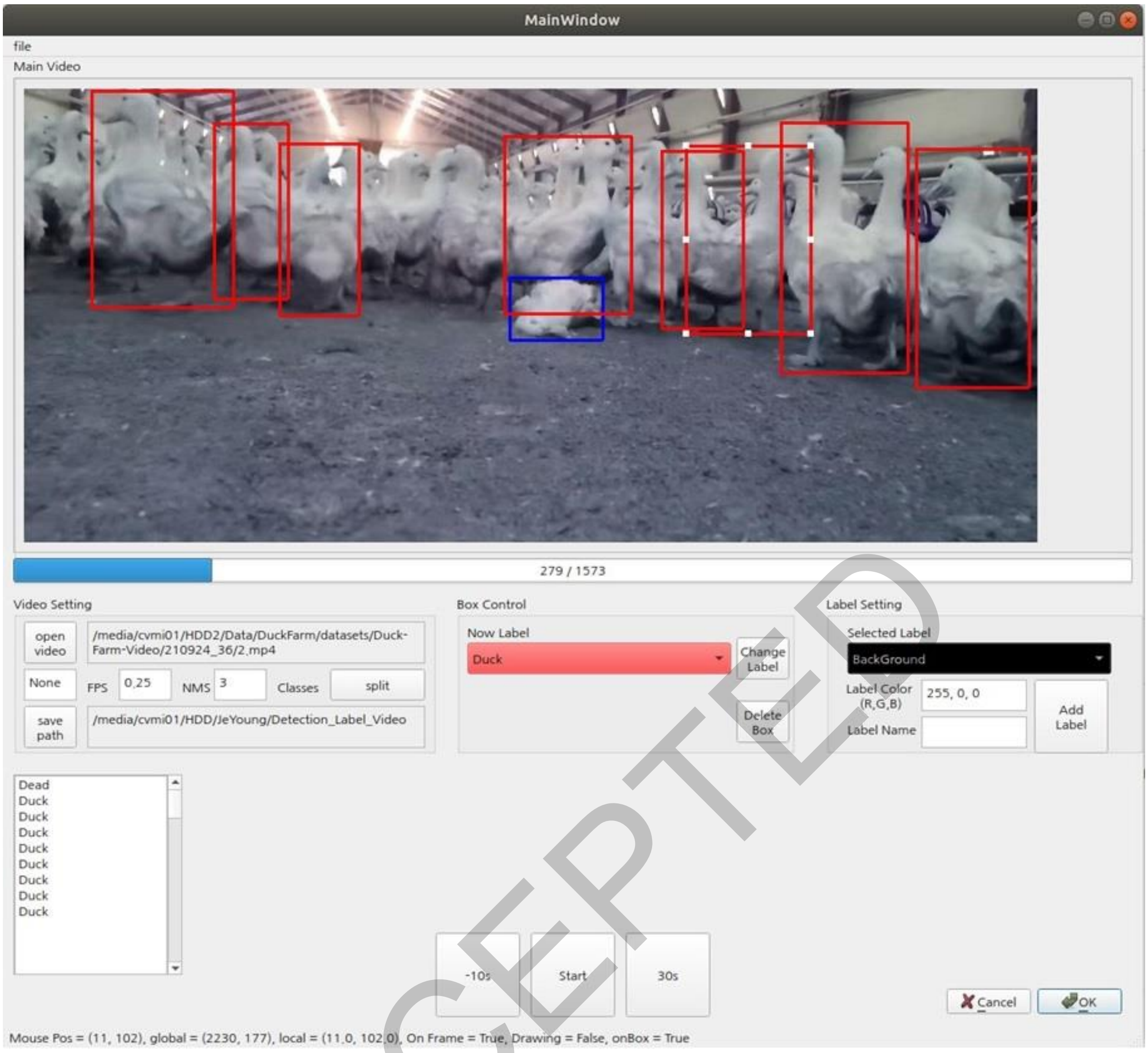
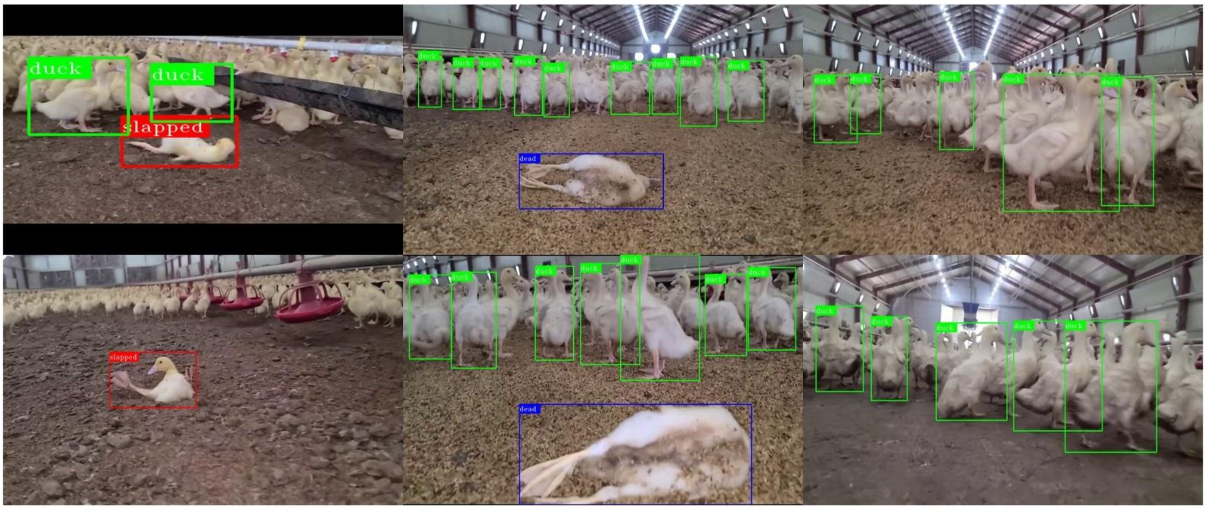



fig3. auto labeling program




(a) (b) (c)

fig4. labeling example (a) slap image, (b) dead image, (c) normal image

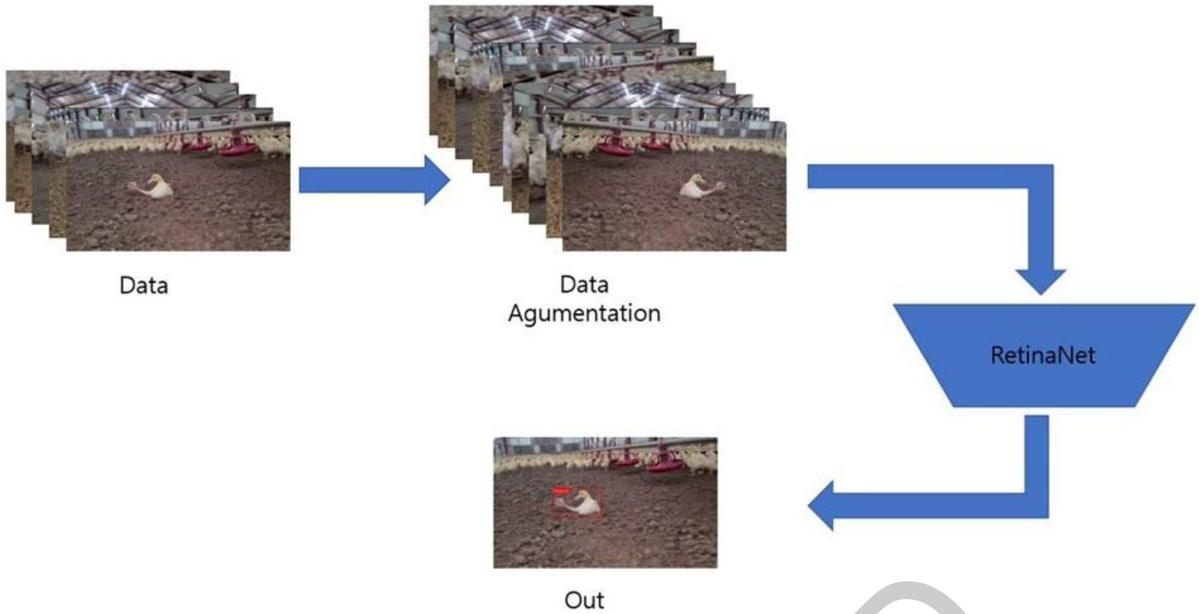



fig5. Duck Detection Training Pipeline

(a)

(b)

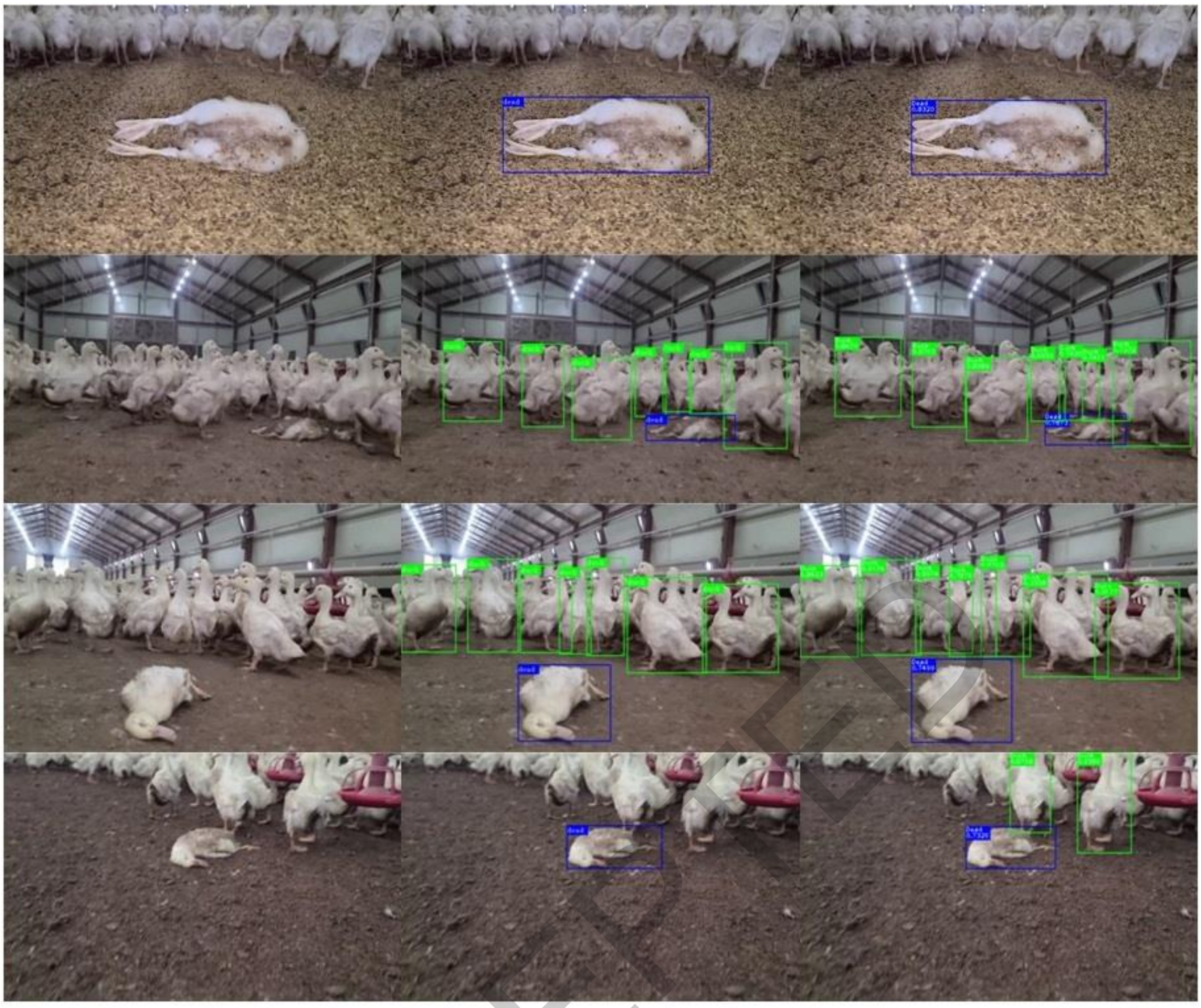

fig6. Data Augmentation Example (a)Original, (b)Augmentation

(a) (b) (c)

fig 7. normal duck result (a) original, (b) ground truth, (c) our result


(a) (b) (c)

fig 8. slap duck result (a) original, (b) ground truth, (c) our result



(a) (b) (c)

fig 9. dead duck result (a) original, (b) ground truth, (c) our result