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)	Automatic detection of trapping events of postnatal piglets in loose housing pen: comparison of YOLO versions 4, 5, and 8
Running Title (within 10 words)	Piglet trapping detection using YOLO versions 4, 5, and 8
Author	Taeyong Yun1, Jinsul Kim2, Jinhyeon Yun1,3, Tai-Won Um1
Affiliation	1 Department of Data Science, Chonnam National University, Gwangju 61186, Korea 2 School of Electronics and Computer Engineering, Chonnam National University, Gwangju 61186, Korea 3 Department of Animal Science, Chonnam National University, Gwangju 61186, Korea
ORCID (for more information, please visit https://orcid.org)	Taeyong Yun (https://orcid.org/0009-0003-7669-9073) Jinsul Kim (https://orcid.org/0000-0003-2294-3969) Jinhyeon Yun (https://orcid.org/0000-0002-0697-0679) Tai-Won Um (https://orcid.org/0000-0002-4922-1774)
Competing interests	No potential conflicts of interest relevant to this article were reported.
Funding sources State funding sources (grants, funding sources, equipment, and supplies). Include name and number of grant if available.	This work was partly supported by an Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2020-0-00833, A study on 5G based Intelligent IoT Trust Enabler and IITP-2022-RS-2022-00156287, Innovative Human Resource Development for Local Intellectualization support program).
Acknowledgements	This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2024-RS-2024-00437718), supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP). Additional funding was provided by a grant from the National Research Foundation (NRF) of Korea (RS-2024-00352491).
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: Yun TY, Um TW Data curation: Yun TY, Yun J Formal analysis: Yun TY, Kim J, Yun J Methodology: Yun TY, Kim J, Yun J Software: Yun TY, Kim J Validation: Yun J, Um TW Investigation: Kim J, Yun J Writing - original draft: Yun TY, Um TW Writing - review & editing: Yun J, Um TW
Ethics approval and consent to participate	All experimental animal protocols were reviewed and approved by Chonnam National University Institutional Animal Care and Use Committee under the reference CNU IACUC-YB-2021-167.

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	Jinhyeon Yun Tai-Won Um
Email address – this is where your proofs will be sent	- Jinhyeon Yun: pilot9939@jnu.ac.kr - Tai-Won Um: stwum@jnu.ac.kr
Secondary Email address	
Address	- Jinhyeon Yun: Department of Animal Science, Chonnam National University, Gwangju 61186, Korea - Tai-Won Um: Department of Data Science, Chonnam National University, Gwangju 61186, Korea
Cell phone number	- Tai-Won Um: +82-10-9124-7250
Office phone number	- Jinhyeon Yun: +82-62-530-2124 - Tai-Won Um: +82-62-530-5794
Fax number	

(Unstructured) Abstract (up to 350 words)

In recent years, the pig industry has experienced an alarming surge in piglet mortality shortly after farrowing due to crushing by the sow. This issue has been exacerbated by the adoption of hyperprolific sows and the transition to loose housing pens, adversely affecting both animal welfare and productivity. In response to these challenges, researchers have progressively turned to artificial intelligence of things (AIoT) to address various issues within the livestock sector. The primary objective of this study was to conduct a comparative analysis of different versions of object detection algorithms, aiming to identify the optimal AIoT system for monitoring piglet crushing events based on performance and practicality. The methodology involved extracting relevant footage depicting instances of piglet crushing from recorded farrowing pen videos, which were subsequently condensed into 2-3 min edited clips. These clips were categorized into three classes: no trapping, trapping, and crushing. Data augmentation techniques, including rotation, flipping, and adjustments to saturation and contrast, were applied to enhance the dataset. This study employed three deep learning object recognition algorithms—YOLOv4-Tiny, YOLOv5s and YOLOv8s—followed by a performance analysis. The average precision (AP) for trapping detection across the models yielded values of 0.963 for YOLOv4-Tiny, and 0.995 for both YOLOv5s, and YOLOv8s. Notably, trapping detection performance was similar between YOLOv5s and YOLOv8s. However, YOLOv5s proved to be the best choice considering its model size of 13.6 MB compared to YOLOv4-Tiny's 22.4 MB and YOLOv8's 21.4 MB. Considering both performance metrics and model size, YOLOv5s emerges as the most suitable model for detecting trapping within an AIoT framework. Future endeavors may leverage this research to refine and expand the scope of AIoT applications in addressing challenges within the pig industry, ultimately contributing to advancements in both animal husbandry practices and technological solutions.

Keywords (3 to 6): Piglet Crushing, Deep learning object-detection algorithm, YOLO, Trapping, AIoT

Introduction

In recent years, the widespread use of hyperprolific sows to boost productivity in the pig industry has also resulted in a notable rise in piglet mortality. This increase is primarily attributed to instances in which the piglets are crushed by the sows shortly after the farrowing process [1]. Meanwhile, there is a global shift toward emphasizing animal welfare in livestock farms. This is evident in the transition from closed farrowing crates, which restrict maternal mobility, to loose or free housing systems that provide sows with increased freedom of movement. However, this transition has raised concerns about an increase in piglet crush rates, which are commonly attributed to risky behaviors such as rolling and sudden transitions from standing to sternal lying [2, 3]. Piglet crushing constitutes a significant cause of death among pre-weaned piglets, contributing to over 50% of pre-weaning losses in pig farming [4]. Notably, most of these piglet fatalities occur within the first three days after farrowing [5].

Factors affecting crushing incidents can be categorized into genetics, environmental factors, parity and litter size, pig weight and health, housing system, and management [4]. Previous research aimed at mitigating crushing issues, without the use of artificial intelligence (AI) technology, has primarily focused on identifying and reducing these factors. To address the root cause of crushing events, it is crucial to identify these incidents, especially in the absence of farm staff.

Recent research has integrated AI to identify crushing incidents with minimal human intervention, primarily focusing on two main types: sound-based and video-based approaches. In sound-based research, a platform was developed using voice data to detect crushing through piglets' screams [6]. This study proposes an audio clip transform approach for preprocessing raw audio data and employs min-max scaling for machine learning to detect piglet screams. Despite technological advancements, these tools encounter challenges in scenarios where piglets cannot vocalize distress, such as head or full-body crushing incidents. In addition, pinpointing the precise location of the crushing event in cases involving multiple pens also poses a challenge. Furthermore, the barn environment's diverse noises, including piglet scuffles, running fans, and other ambient sounds, can cause device malfunctions, hindering precise recognition of piglet crushing incidents. Conversely, in a video-based AI study on crushing, the emphasis shifted to assessing the risk of crushing by recognizing the sow's behavior rather than directly identifying crushing events [7]. This study assessed sow behavior using a three-axis accelerometer and video data. Following the recognition of sow behaviors, maternal care was evaluated by scoring the risk and number of behavioral patterns associated with increased trapping events.

To maximize the utility of these technologies, a recent development involves the introduction of artificial intelligence of things (AIoT). AIoT represents the convergence of AI and internet of things (IoT), offering the capability to use networks and cloud services for real-time problem solving with minimal human intervention. Researchers have recently employed AIoT technologies to develop a pig tracking and monitoring system [8, 9]. The pig farming industry is increasingly incorporating and using these technologies. However, for the optimal use of AIoT, it must function within the constraints of the IoT environment. Therefore, in choosing the most suitable AIoT model, it is essential to strike a balance between efficiency and functionality in a resource-constrained environment. The selection process should prioritize both performance and model size.

This study aimed to identify an object detection algorithm within an AIoT framework capable of efficiently detecting piglet trapping and subsequently implementing it in practical applications. Object detection algorithms are broadly categorized into two-stage and one-stage models. The two-stage model involves a local proposal followed by a classification stage, offering high accuracy, albeit at a slower pace [10]. Conversely, single-stage models simultaneously perform classification and localization, offering higher speed and making them particularly suitable for IoT and mobile device applications [10]. Among the prominent single-stage object detection techniques, YOLO was introduced in 2015 by Joseph et al. [11]. The YOLO model encompasses Darknet-based versions such as YOLOv3 and YOLOv4, PyTorch-based models such as YOLOv5, and their successor models [12]. In this study, we implement and compare three representative models: YOLOv4, a modern Darknet-based model; YOLOv5, a popular PyTorch-based model; and YOLOv8, the latest model. The aim is to scrutinize these models, seeking the most effective in detecting piglet strangulation. Furthermore, the study aims to evaluate the feasibility of implementing, optimizing, and operating the selected model within an AIoT environment.

Materials and Methods

Video Material and Editing

Five sows were housed in loose pen conditions (2.4 × 2.3 m), with each farrowing pen equipped with a slatted concrete floor and a heat lamp. AIoT was installed seven days before the expected farrowing date to observe piglet birth, crushing, trapping, sow posture, and piglet tracking. Internet protocol cameras (HN0-E60; Hanwha Techwin, South Korea) were positioned 1.5 m above the sow's head with 1920 × 1080 pixels display resolution and 30 FPS frame rate.

After recording the video, the footage was reviewed to identify the section where the trapping incidents occurred. These sections were then extracted and collected for 24 h following the onset of parturition. Once identified, these specific scenes were extracted, and images were obtained for each frame. In this study, the YOLO bounding box program was used to generate bounding boxes and corresponding labels for individual images. As shown in Fig. 1, a data augmentation technique was applied to enhance the diversity of the training dataset. This technique involved variations in saturation and contrast, along with rotations (90°, 180°, 270°) and horizontal and vertical flips. As a result of this augmentation process, the total number of images increased significantly from 544 to 9792, creating a more comprehensive training dataset. The dataset was then divided into training, validation and test sets using a 6:2:2 ratio, resulting in 5875 images for training, 1958 images for validation and 1959 images for testing. This setup was based on a previous study that used YOLO to detect tomatoes in real time, which also used a 6:2:2 data split [13]. The systematic application of data augmentation and dataset separation aimed to increase the model's resilience in adapting to different learning environments. In the original dataset configuration prior to augmentation, there were 4570 classes for no trapping, 267 classes for trapping, and 129 classes for crushing.

Model training and object detection

The annotated dataset, without further conversions, served as the input for training three object detection algorithms: YOLOv4-Tiny, YOLOv5s, and YOLOv8s. YOLOv4-Tiny, a model based on Darknet, YOLOv5s, the most popular PyTorch-based model, and YOLOv8s, the most recently published model, were all trained in the Google Colab environment. We applied transfer learning to our dataset using the pre-trained model weights from the ImageNet dataset. In the experiments summarized in Table 1, the YOLO model was trained for 50 epochs with a batch size of 64 and a fixed learning rate of 0.01. This setup was used to fine-tune the model weights for optimal training performance. For models like YOLOv4-Tiny that do not use an epoch-based system, the training process is controlled by a hyperparameter called max-batch. The value of max-batch is calculated using the formula:

$$Max\,batch = \frac{Number\,of\,epochs \times Number\,of\,images\,to\,train}{Number\,of\,batches} \quad (1)$$

Given that the number of images per epoch was set to 5875 and the batch size to 64, the resulting max-batch value was approximately 4,589.84. This value was rounded to 4590, which was used as the max-batch parameter during training. During training, the input image was resized to 416 × 416 to for feature map extraction.

Each version of YOLO introduces architectural innovations aimed at improving the model's performance in object detection. YOLOv4 employs CSPDarknet53 for efficient feature extraction, coupled with SPP and PAN for multi-scale feature integration [14]. YOLOv5 enhances this with a focus structure and CSP backbone, paired with FPN and PAN for refined feature aggregation [15]. YOLOv8 further advances the architecture by incorporating C2f modules, optimizing both the backbone and neck for superior detection capabilities [16, 17].

Fig. 2 illustrates the results of applying YOLOv5s to a piglet crushing site after the learning process. Three classes were identified in this study: no trapping, trapping, and crushing. "No trapping" denotes that the piglet is fully visible on the screen without any part of its body being covered or crushed. Conversely, "trapping" indicates that the piglet has been compressed by its mother, resulting in part or all of its body being obscured. The term "crushing" is used when the piglet stops moving after being caught, indicating that it has succumbed to compression and has died. While detecting crushing from a single image is challenging due to data limitations and the visual similarity between sleeping and crushed piglets, this study represents a significant step forward. The foundation laid by this research will inform the development of more advanced detection systems. Future efforts will focus on incorporating tracking to improve detection accuracy.

Model evaluation metrics

The evaluation of a classification model involves several metrics, such as precision, recall, average precision, mean average precision, and F1 score. These metrics provide insight into different aspects of model performance.

In Equation (2), precision represents the percentage of instances that the model correctly classified as true among all instances it classified as true. Specifically, precision is calculated as:

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

where True Positives (TP) are the cases where the model correctly identifies a positive instance, and False Positives (FP) are the cases where the model incorrectly classifies a negative instance as positive. Equation (3) defines recall as the percentage of true instances that the model correctly identifies as positive out of the total number of actual positive instances. Recall is calculated as:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

where False Negatives (FN) are the cases where the model fails to identify an actual positive instance, incorrectly classifying it as negative. Precision measures the accuracy of positive predictions, while recall assesses the model's ability to detect all positive instances. Average Precision (AP) measures the precision value averaged over different confidence levels for a given class. It provides a comprehensive view of a model's performance by evaluating the precision at different confidence levels. The Mean Average Precision (mAP), as defined in Equation (4), represents the average of the APs computed across multiple classes or instances, providing an aggregate measure of performance across all classes.

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP_i \tag{4}$$

mAP@0.50 is a metric that evaluates the performance of an object detection algorithm by averaging the precision scores across all classes, assuming that predictions with an Intersection over Union (IoU) of 0.50 or higher are considered correct. The IoU in Equation (5) is a metric used to evaluate the accuracy of predictions made by an object detection algorithm. It is defined as the ratio of the area of overlap between the ground truth bounding box and the predicted bounding box to the area of their union.

$$IoU = \frac{TP}{TP + FP + FN} \tag{5}$$

Specifically, IoU measures how well the predicted bounding box aligns with the ground truth bounding box, providing a quantifiable measure of prediction quality.

The F1 score is a model evaluation metric used in classification models. Another widely used evaluation metric is accuracy, which is defined as the proportion of true values among all predictions. However, accuracy has a limitation, particularly in the context of unbalanced data, where it can be misleading. In scenarios where, for instance, the probability of cancer is 1%, the model can achieve 99% by classifying all patients as non-cancerous, presenting a potential vulnerability. Therefore, the F1 score is frequently employed for assessing unbalanced data. Equation (6) defines the F1 score as the harmonic mean of the precision and recall values. These metrics collectively provide a comprehensive evaluation of the classification model.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6}$$

The precision-recall curve is a graph of the change in precision and recall values as the confidence threshold changes. The graph has recall on the x-axis and precision on the y-axis. The AP signifies the average of the precision across different recall values. In the context of a precision recall curve, AP corresponds to the area under the curve.

Results and Discussion

Comparison of the average precision of different YOLO versions

Table 2 provides a comprehensive comparison of the AP, mAP, and F1 score derived from the training of YOLOv4-Tiny, YOLOv5s, and YOLOv8s. The performance across these models remains consistently robust, with a marginal difference observed. YOLOv5s (0.994) and YOLOv8s (0.994) achieved higher mAP compared to YOLOv4-Tiny (0.958), as shown in Table 2. A study comparing YOLOv4-Tiny and YOLOv5s found that YOLOv5s performed better, with a 0.133 higher mAP value than YOLOv4-Tiny [18]. In addition, a comparison between YOLOv5 and YOLOv8 showed a very small difference in mAP values of 0.006 [19]. This suggests that the performance difference between YOLOv5 and YOLOv8 is negligible. These findings are consistent with our results, which also show that YOLOv5 and YOLOv8 have similar performance metrics, while YOLOv4-Tiny lags behind.

When we analyzed the performance metrics by class, YOLOv4-Tiny performed poorly overall. However, it performed best in the Crushing class. YOLOv5s and YOLOv8s showed similar performance, probably due to their similar structures. Notably, YOLOv8s achieved relatively higher APs in the No trapping scenario compared to YOLOv5s, reflecting its structural improvements. The No trapping class of YOLOv8s (0.993) has an AP that is 0.002 higher than that of YOLOv5s (0.991). However, due to rounding to the fourth decimal place, the mAP for both models are almost identical: 0.9943 for YOLOv8s and 0.9937 for YOLOv5s, indicating a slight difference.

Model size is a critical factor in the practicality of IoT deployments, especially in small-scale computing environments. With a compact model size of 13.6 MB, YOLOv5s stands out as the most suitable choice for AIoT applications. This is in stark contrast to the larger sizes of YOLOv4-Tiny (22.4 MB) and YOLOv8s (21.4 MB), as shown in Table 2. Consequently, YOLOv5s proves to be the optimal model for AIoT applications, balancing high performance with a compact model size.

Detection results using YOLOv5s

Fig. 3 illustrates the precision-recall curve for all classes of YOLOv5s, the model considered most suitable for AIoT applications. YOLOv5s exhibits an AP of 0.991 for no trapping, 0.995 for trapping, and 0.995 for crushing, yielding an overall mAP of 0.994 (Fig. 3).

Although the AP for no trapping is slightly lower than that for the other classes, the recognition for trapping, which is the relevant class in this study, is 0.995 (Fig. 3), indicating a high level of performance.

The confusion matrix represents the ratio of the actual true value to the predicted true value for each class. Out of the total 18,281 detected individuals, 16,796 individuals were in no trapping, 1,049 individuals were in trapping, and 436 individuals were in crushing. Due to the data imbalance with the overwhelming number of individuals in no trapping, Fig. 5 presents the confusion matrix, depicting the performance as a percentage for each class. In Fig. 4, the confusion matrix for YOLOv5s provides a detailed breakdown of the predictions across all classes. In the no trapping class, the model achieves accurate predictions 98.9% of the time, with background recognition errors (failures to recognize no trapping) occurring only 1 % of the time (Fig. 4). For the trapping class, the model predicts trapping with 97.1% accuracy but occasionally misclassifies it as no trapping (2.7 %) (Fig. 4). Similarly, in the crushing class, the model achieves accurate predictions 97.9 % of the time but may misidentify it as no trapping (2.1 %) (Fig. 5). In particular, the background has a high probability (96.8%) of correctly predicting no trapping class when no trapping event is present (Fig. 4).

In the confusion matrix, the misidentification rate of no trapping in the background with objects is 0.968. Other studies showing confusion matrices for YOLOv5 also reveal a notable misidentification rate for other classes in the background with no objects [20]. However, this is a feature of confusion matrices, which are presented as percentages due to unbalanced data. Although 0.968 (Fig. 4) seems quite high, it represents a small percentage of the total misidentifications. To address this confusion, it is more intuitive to evaluate performance in terms of accuracy or F1 score.

Fig. 5 illustrates the optimal confidence hyperparameter values for class differentiation. A confidence value of 0.608 achieves the best F1-score of 0.97 for no trapping, whereas trapping is best detected with a confidence hyperparameter of 0.638, resulting in a perfect F1 score of 1.00 (Fig. 5). Similarly, for crushing, an optimal F1 score of 1.00 was achieved with a confidence hyperparameter value of 0.740 (Fig. 5). Attaining balanced performance across all classes, a confidence hyperparameter value of 0.621 achieves the highest F1-score of 0.99 (Fig. 5), demonstrating the model’s effective recognition of all classes.

The F1 confidence curve graph reveals a clear pattern with a rapid increase in the F1 score in the 0.0-0.2 confidence range. Performance is generally maintained or slightly improved up to 0.2-0.7. However, the F1 score shows a notable decrease when the confidence level exceeds 0.8. This aligns with similar findings in other studies where the variation in the F1 score with confidence showed a sharp increase up to 0.2 and a modest increase up to 0.7 [21]. Furthermore, in other studies, the confidence interval with the highest F1 score is usually in the range of 0.5-0.7, and the graphs in this study show the highest F1 score in this range for no trapping, trapping, and all classes, which is consistent with this result. However, for the crushing class, the best performance is in the high confidence interval (0.740) (Fig. 5), which seems to be a temporary phenomenon due to the lack of data for the crushing class.

Limitations and future research

While YOLO demonstrates robust performance in detecting trapping based on images, it has inherent limitations. Notably, the system can only detect trapping when a portion of the piglet's body is visible within the camera's field of view. This leaves it incapable of identifying situations where the entire body is trapped or events that occur outside the camera's field of view due to obstructions. In addition, the system is susceptible to false positives, particularly when certain parts of the sow's body, such as the ears, are mistakenly identified as trapping points, leading to inaccuracies in detection.

To address these challenges, future research will explore the integration of optical flow technology. Optical flow, a method for tracking objects by analyzing the temporal flow of video and detecting pixel movement between frames, has the potential to enhance trapping prediction [22]. This study aims to implement video-based trapping prediction technology using optical flow to overcome the limitations associated with image-based detection. This innovative approach aims to improve accuracy, particularly in distinguishing between similar objects such as the ears of a sow. By predicting object movement and controlling pixel flow, this research expects accurate identification of trapped piglets or sow body parts. This methodology will extend detection capabilities to scenarios in which the entire body is trapped, a subtlety overlooked by conventional image-based trapping detection models.

Conclusions

In this study, our objective is to apply an AIoT system that minimizes human intervention to address the critical issue of piglet crushing by sows, a leading cause of mortality in pig farms. Given the constrained AIoT environment, our model selection criteria extend beyond performance, encompassing model size as a pivotal factor for efficient deployment within AIoT frameworks. YOLOv4-Tiny did not demonstrate significantly superior performance compared with the other models. Moreover, its considerable model size makes it unsuitable for deployment in small-scale computing environments such as the IoT. Despite YOLOv8s being the latest version, it introduces potential uncertainties in stability when compared to the other models. In addition, the AP performance, especially for trapping, is comparable to YOLOv5s, even though YOLOv8s has a model size about 7.8 MB larger. These shortcomings render the model less suitable than YOLOv5s for certain AIoT applications based on specific metrics. Notably, YOLOv5s stands out for its exceptional performance in the trapping class and remarkably small model size. These qualities position it as an ideal choice for AIoT applications, particularly for tracking piglet crushing challenges in pig farms.

Acknowledgments

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2024-RS-2024-00437718), supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP). Additional funding was provided by a grant from the National Research Foundation (NRF) of Korea (RS-2024-00352491).

References

1. Andersen IL, Nævdal E, Bøe KE. Maternal investment, sibling competition, and offspring survival with increasing litter size and parity in pigs (Sus scrofa). Behav Ecol Sociobiol. 2011;65:1159-67. https://doi.org/10.1007/s00265-010-1128-4

2. Kobek-Kjeldager C, Pedersen LJ, Larsen MLV. Behavioural characteristics of fatal piglet crushing events under outdoor conditions. Livest Sci. 2023;268:105164. https://doi.org/10.1016/j.livsci.2023.105164

3. Yun J, Han T, Björkman S, Nystén M, Hasan S, Valros A, et al. Factors affecting piglet mortality during the first 24 h after the onset of parturition in large litters: effects of farrowing housing on behaviour of postpartum sows. Animal. 2019;13(5):1045-53. https://doi.org/10.1017/S1751731118002549

4. Liu T, Kong N, Liu Z, Xi L, Hui X, Ma W, et al. New insights into factors affecting piglet crushing and anti-crushing techniques. Livest Sci. 2022:105080. https://doi.org/10.1016/j.livsci.2022.105080

5. Manteuffel C, Hartung E, Schmidt M, Hoffmann G, Schön PC. Online detection and localisation of piglet crushing using vocalisation analysis and context data. Comput Electron Agric. 2017;135:108-14. https://doi.org/10.1016/j.compag.2016.12.017

6. Chen W-E, Lin Y-B, Chen L-X. PigTalk: An AI-based IoT platform for piglet crushing mitigation. IEEE Trans Industr Inform. 2020;17(6):4345-55. https://doi.org/10.1109/TII.2020.3012496

7. Sun G, Shi C, Liu J, Ma P, Ma J. Behavior recognition and maternal ability evaluation for Sows based on triaxial acceleration and video sensors. IEEE Access. 2021;9:65346-60. https://doi.org/10.1109/ACCESS.2021.3075272

8. Jung W, Kim S-H, Hong S-P, Seo J. An AIoT Monitoring System for Multi-Object Tracking and Alerting. Comput Mater Contin. 2021;67(1). https://doi.org/10.32604/cmc.2021.014561

9. Su W-T, Jiang L-Y, Tang-Hsuan O, Lin Y-C, Hung M-H, Chen C-C. AIoT-cloud-integrated smart livestock surveillance via assembling deep networks with considering robustness and semantics availability. IEEE Robot Autom Lett. 2021;6(4):6140-7. https://doi.org/10.1109/LRA.2021.3090453

10. Lee S, Lee G, Ko J, Lee S, Yoo W. Recent Trends of Object and Scene Recognition Technologies for Mobile/Embedded Devices. Electron Telecommun Trends. 2019;34(6):133-44. https://doi.org/10.22648/ETRI.2019.J.340612

11. Zou Z, Chen K, Shi Z, Guo Y, Ye J. Object detection in 20 years: A survey. Proc IEEE. 2023. https://doi.org/10.1109/JPROC.2023.3238524

12. Ultralytics YOLOv8 Docs. YOLO: A Brief History [Internet]. 2022 [cited 2023 Aug 14]. https://docs.ultralytics.com/

13. Zeng T, Li S, Song Q, Zhong F, Wei X. Lightweight tomato real-time detection method based on improved YOLO and mobile deployment. Computers and electronics in agriculture. 2023;205:107625. https://doi.org/10.1016/j.compag.2023.107625

14. 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

15. Zhang Y, Guo Z, Wu J, Tian Y, Tang H, Guo X. Real-time vehicle detection based on improved yolo v5. Sustainability. 2022;14(19):12274. https://doi.org/10.3390/su141912274

16. Terven J, Córdova-Esparza D-M, Romero-González J-A. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Machine Learning and Knowledge Extraction. 2023;5(4):1680-716. https://doi.org/10.3390/make5040083

17. Ju R-Y, Cai W. Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm. Scientific Reports. 2023;13(1):20077. https://doi.org/10.1038/s41598-023-47460-7

18. Kıvrak O, Gürbüz MZ. Performance comparison of yolov3, yolov4 and yolov5 algorithms: A case study for poultry recognition. Avrupa Bilim ve Teknoloji Dergisi. 2022(38):392-7.

19. Afonso MH, Teixeira EH, Cruz MR, Aquino GP, Boas ECV. Vehicle and Plate Detection for Intelligent Transport Systems: Performance Evaluation of Models YOLOv5 and YOLOv8.

20. Yu L, Qian M, Chen Q, Sun F, Pan J. An Improved YOLOv5 Model: Application to Mixed Impurities Detection for Walnut Kernels. Foods. 2023;12(3):624. https://doi.org/10.3390/foods12030624

21. Brüngel R, Friedrich CM, editors. DETR and YOLOv5: exploring performance and self-training for diabetic foot ulcer detection. 2021 IEEE 34th International Symposium on Computer-Based Medical Systems (CBMS); 2021: IEEE. https://doi.org/10.1109/CBMS52027.2021.00063.

22. Horn BK, Schunck BG. Determining optical flow. Artif intell. 1981;17(1-3):185-203. https://doi.org/10.1016/0004-3702(81)90024-2

Tables and Figures

Table 1. Parameters of YOLO models

Parameters	YOLOv4-Tiny	YOLOv5s	YOLOv8s
Number of iterations	Max-batch:4590	Epoch: 50	Epoch: 50
Batch	64	64	64
Learning Rate	0.01	0.01	0.01

Table 2. Comparison of different YOLO versions with respect to performance

Model	Size	F1-score	No trapping AP	Trapping AP	Crushing AP	mAP@0.50
YOLOv4-Tiny	22.4MB	0.92	0.933	0.949	0.993	0.958
YOLOv5s	13.6MB	0.99	0.991	0.995	0.995	0.994
YOLOv8s	21.4MB	0.99	0.993	0.995	0.995	0.994

1) F1-score: Harmonic mean of the precision and recall scores
2) AP: Average of precision for each class
3) mAP@0.50: Mean of AP for all classes when IoU threshold is 0.5

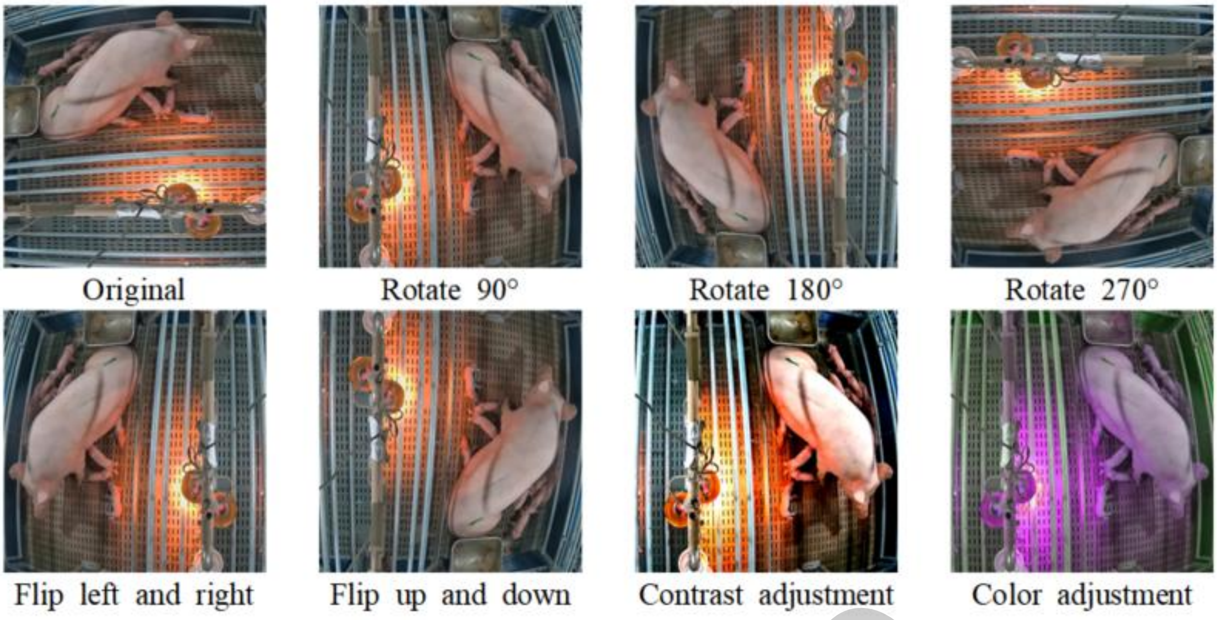

Fig. 1. Piglet crushing field image data augmentation

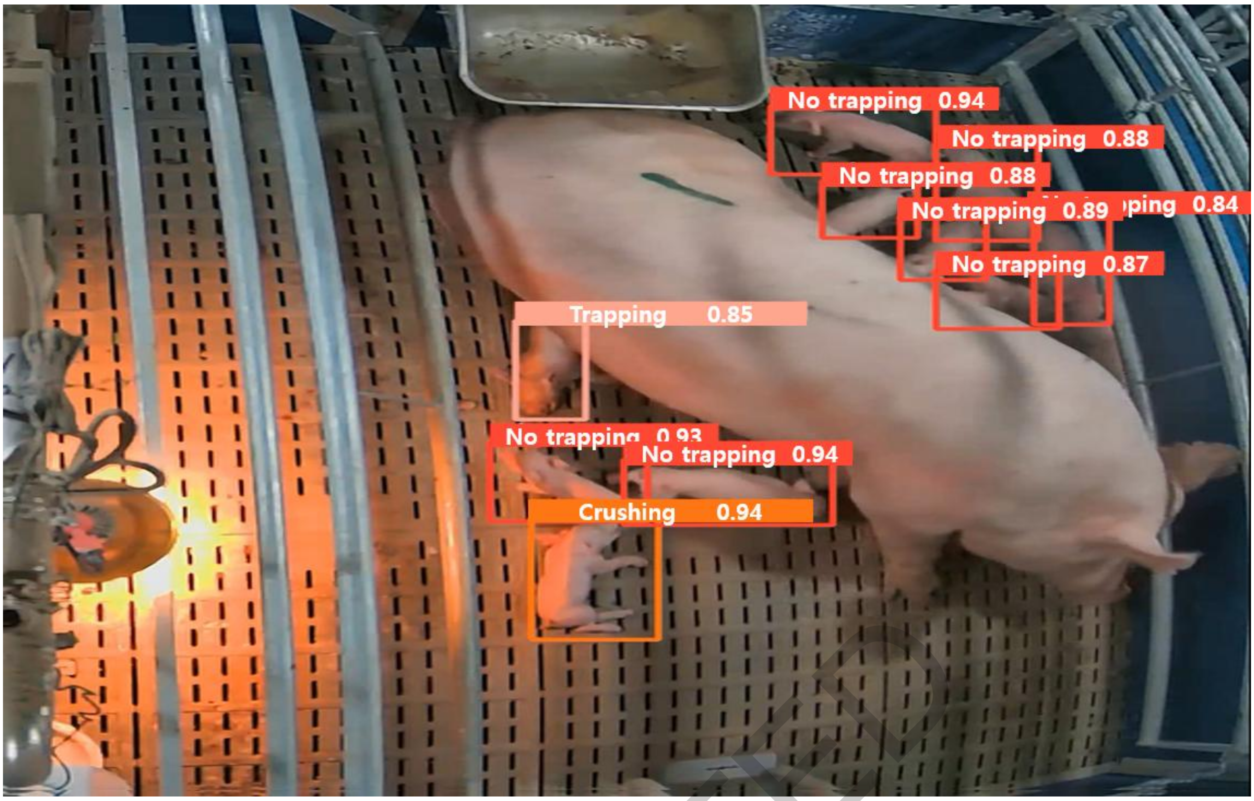

Fig. 2. Classification (no trapping, trapping, and crushing)

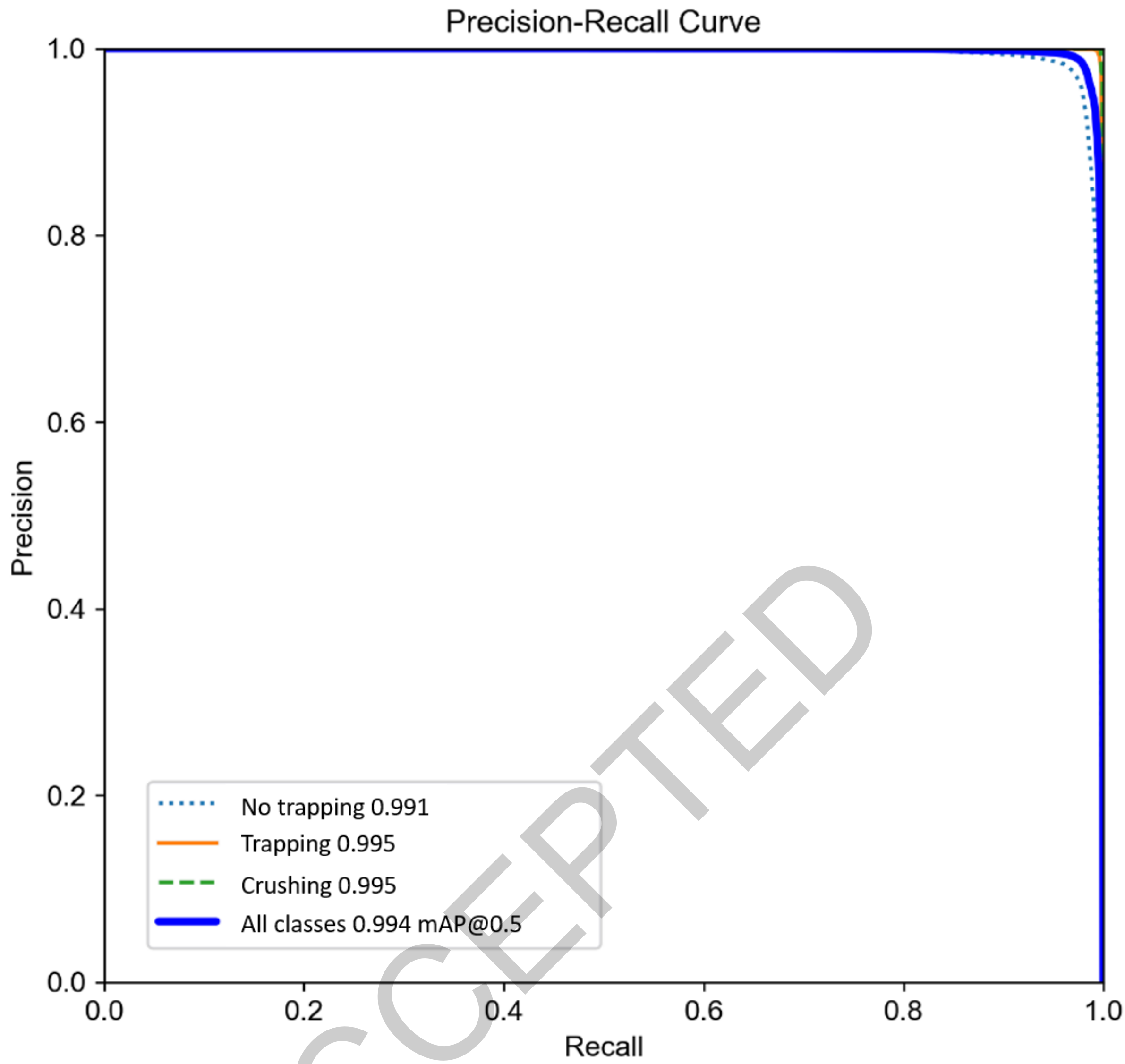

Fig. 3. Precision-recall curve for detecting piglet trapping events using YOLOv5s

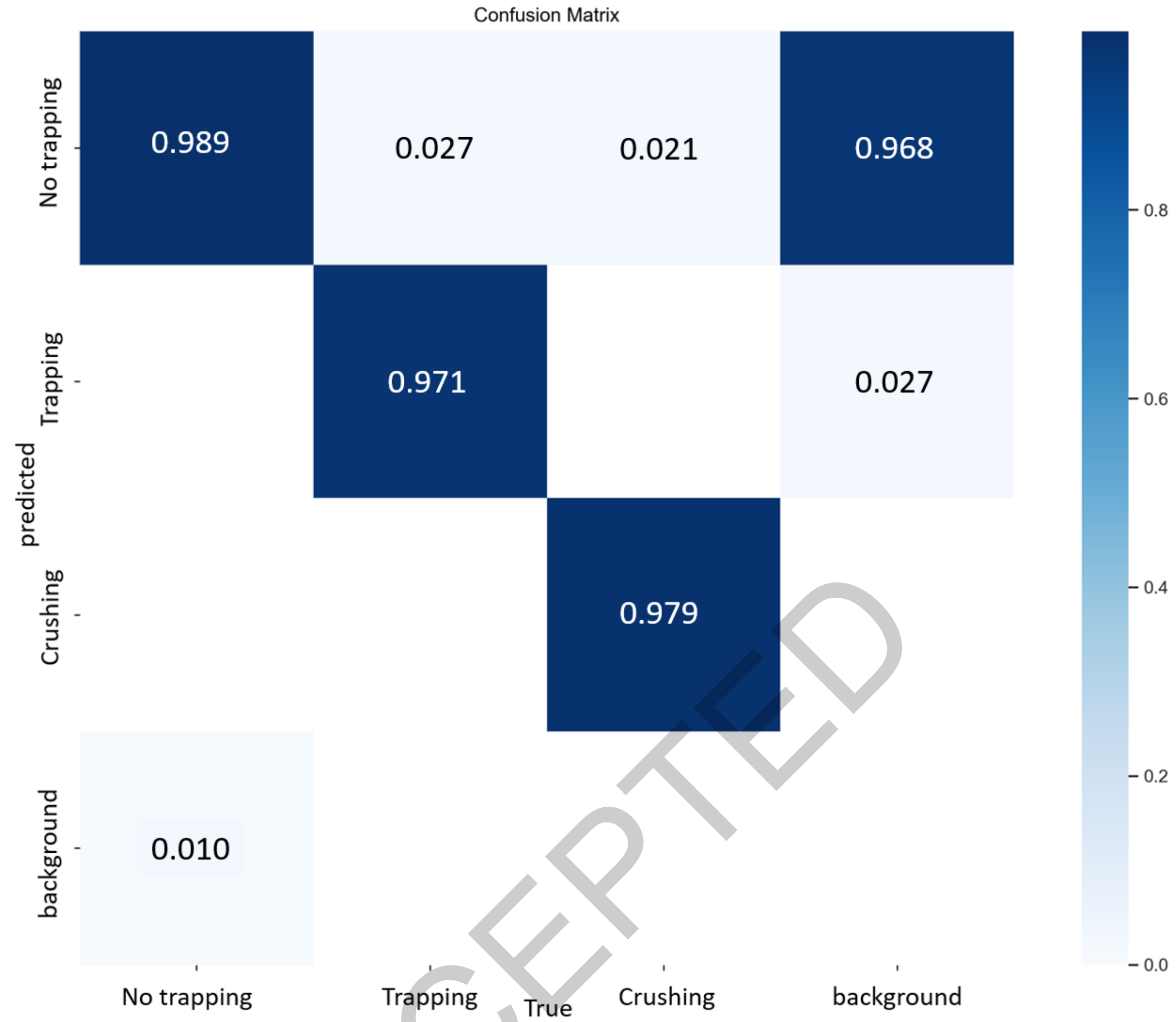

Fig. 4. Confusion matrix for detecting piglet trapping events using YOLOv5s

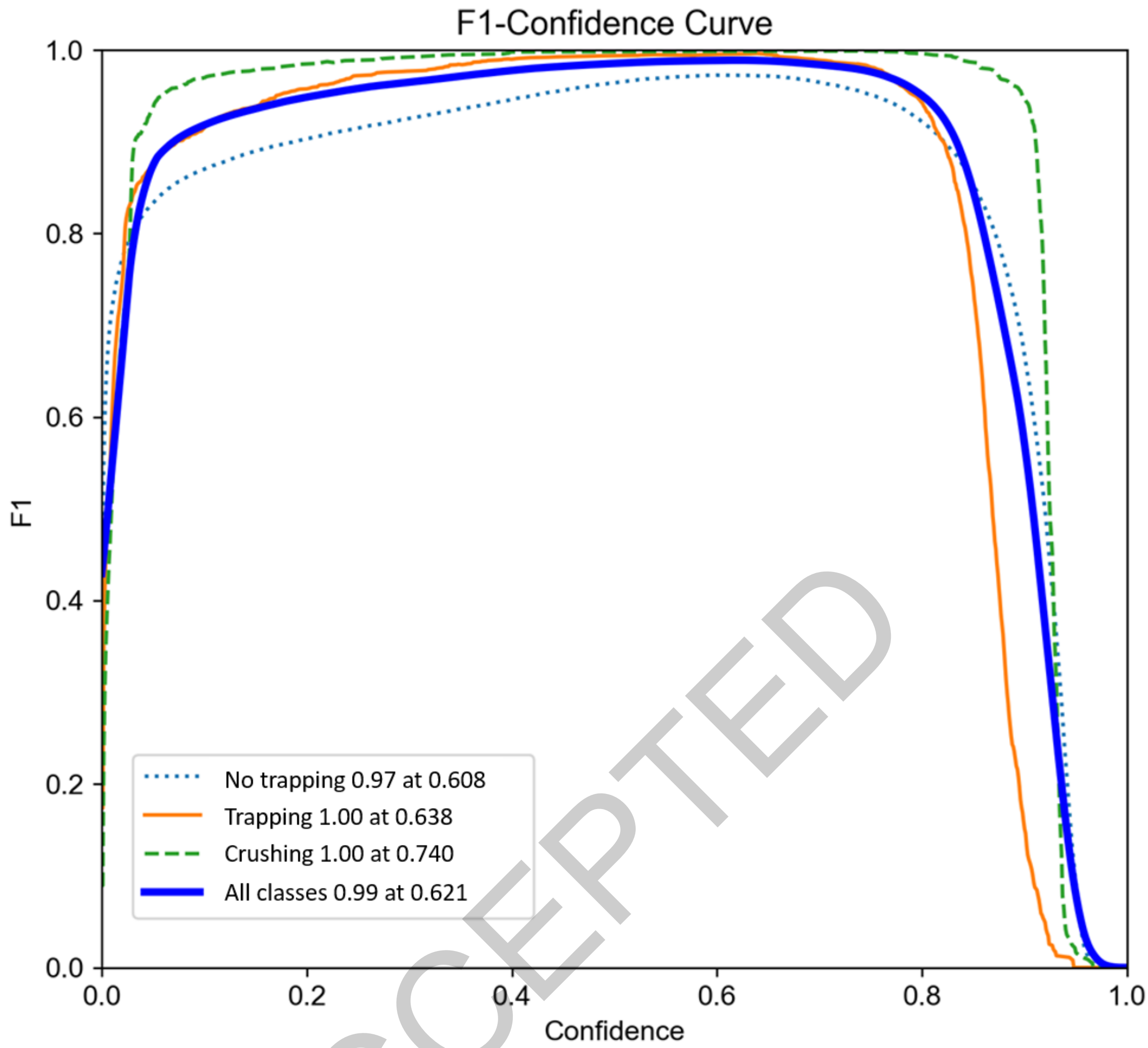

Fig. 5. F1-Confidence curve for detecting piglet trapping events using YOLOv5s