Optimizing Livestock Productivity with Computer Vision-Based Cow Estrus Detection in Free Stall Barns using Various YOLOv8 Models

Teddy Surya Gunawan
Electrical and Computer Engineering Department
International Islamic University Malaysia
53100 Kuala Lumpur, Malaysia
tsgunawan@iium.edu.my

Mira Kartiwi
Information Systems Department
International Islamic University Malaysia
53100 Kuala Lumpur, Malaysia
mira@iium.edu.my

Ali Saifudin
Department of Veterinary Sciences
Airlangga University
Surabaya, Indonesia
ali.saifudi.38578-2022@fkh.unair.ac.id

Levy Olivia Nur
School of Electrical Engineering
Telkom University
Bandung, Indonesia
levyolivia@telkomuniversity.ac.id

Bambang Setia Nugroho
School of Electrical Engineering
Telkom University
Bandung, Indonesia
bambangsetianugroho@telkomuniversity.ac.id

2023 IEEE 9th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA) | 979-8-3503-4338-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSIMA59853.2023.10373431

***Abstract* — In the domain of livestock management, the precise detection of estrus in cows is crucial for reproductive efficiency and enhanced livestock production. Traditional methods, primarily based on human observation, are labor-intensive and can be error-prone. This study leverages YOLOv8, a cutting-edge computer vision technology, for cow estrus detection. Our evaluation reveals that YOLOv8 achieved a remarkable accuracy rate, outperforming conventional methods in speed and reliability. Specifically, the model demonstrated a precision of 96%, a recall of 96.1%, and a mean average precision (mAP) of 98.35% for the 50% intersection over union (IoU) threshold. By integrating YOLOv8, we highlight the potential for substantial improvements in reproductive efficiency, labor cost savings, and increased profitability in the cattle sector. This work emphasizes the transformative impact of advanced technology in agriculture and paves the way for future innovations in livestock management.**

Keywords—Livestock management, cow estrus detection, computer vision, deep learning, YOLOv8.

I. INTRODUCTION

The production of livestock, particularly cows, has increased significantly in recent years. As the primary source of milk and meat, cows significantly contribute to the global food supply. According to [1], the global cattle population has been steadily increasing, highlighting the significance of efficient livestock management.

The accurate detection of estrus in cows is a crucial component of livestock management. Estrus, the period during which a female bovine is fertile, is characterized by a variety of behaviors and physical signs. It is crucial to recognize these symptoms, as the estrus phase lasts only 8 to 30 hours. If this window is missed, the next estrus cycle will occur between 17 and 24 days later. This delay not only lengthens the calving interval but also reduces the reproductive efficiency of livestock, resulting in potential productivity and profit losses [2, 3].

Historically, estrus detection techniques have been classified as invasive or non-invasive. Invasive methods frequently involve internal examinations or procedures that may cause the animal discomfort, such as electronic noses [4], accelerometers [5], pedometers, and pressure sensors [6]. In contrast, non-invasive methods, such as herders' visual observations, rely on external signs and behaviors, such as infrared thermography [7], surveillance cameras [8], and audio [9]. However, these conventional methods are prone to human error and require continuous monitoring, making them laborious and less reliable.

Computer vision has emerged as a promising tool for automating the estrus detection process as a result of technological advancements [3, 10]. YOLOv8 (You Only Look Once version 8) stands out among computer vision technologies due to its rapid and accurate object detection capabilities. In the context of cattle management, YOLOv8 offers a revolutionary method for identifying the distinct behavioral patterns associated with cows' estrus. By utilizing computer vision techniques, we can not only reduce the labor-intensive nature of estrus detection but also improve its precision and timeliness. This transition to automated detection ensures that no estrus cycle is missed, thereby optimizing the breeding process and ensuring greater reproductive efficiency.

As the livestock industry continues to expand, the demand for effective and precise estrus detection methods grows. Computer vision, in particular YOLOv8, offers a viable solution to the difficulties inherent in traditional estrus detection methods, paving the way for a more sustainable and profitable cattle industry.

II. COW ESTRUS DETECTION

Estrus detection is crucial for the reproductive efficiency of dairy cattle, especially when using artificial insemination, as it influences the calving-to-conception interval, which has a direct effect on milk yield and profitability. While the primary indicator of estrus is a cow's willingness to be mounted, this behavior has diminished, making secondary signs, such as mounting other cows, more telling. This detection is complicated by factors such as increased milk production, management practices, housing conditions, floor types, and temperature, which are all governed by endocrine hormone regulation.

As shown in Figure 1, the timing of ovulation and the age of the egg during sperm penetration are crucial for conception. Ovarian steroid hormones and the maturation of the Graafian follicle affect estrus behavior, specifically the cow's standing still for mounting. Ovulation typically occurs 20 to 30 hours after the onset of this behavior, but hormonal levels and follicle development can cause variations [11]. It is important to note that optimal time to inseminate occurs between 6 to 12 hours after the onset of estrus, with sperm having a 24- to 34-

979-8-3503-4338-0/23/$31.00 ©2023 IEEE

hour lifespan in the reproductive tract and the ova's viability is limited to a mere 6 to 12 hours following its release (ovulation).

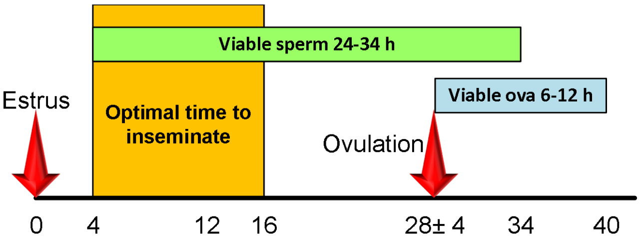


Fig. 1. Timing of onset estrus detection and artificial insemination

Estrus detection is crucial to reproductive performance, prompting the agricultural sector to develop and implement cutting-edge technologies. These developments are intended to enhance the detection of estrus by observing animal behavior, either in place of or in addition to conventional visual observations. Fig. 2 depicts an assortment of these cutting-edge technologies. They include a variety of tools and techniques, such as camera and infrared camera systems that employ computer vision to capture behavioral nuances and accelerometers that monitor movement patterns. In addition, microphones that pick up specific vocalizations, pressure sensors that detect mounting activity, and tail chalk that indicates estrus visually are utilized. More sophisticated methods include monitoring vaginal and body temperatures, analyzing specific substances in milk that indicate estrus, and using pedometers and accelerometers to measure activity. These technologies provide farmers with an extensive toolkit to improve the precision and efficacy of estrus detection in their herds.

Computer vision offers a revolutionary method for detecting estrus in free-stall barns, overcoming the difficulties of manual monitoring in large spaces with numerous cattle. In such environments, cows exhibit behavioral changes indicative of estrus, which can be captured and analyzed by computer vision. This technology not only detects subtle movements and interactions that are frequently missed by human observers, but it also ensures consistent, round-the-clock monitoring, thereby eliminating human errors, fatigue, and biases.

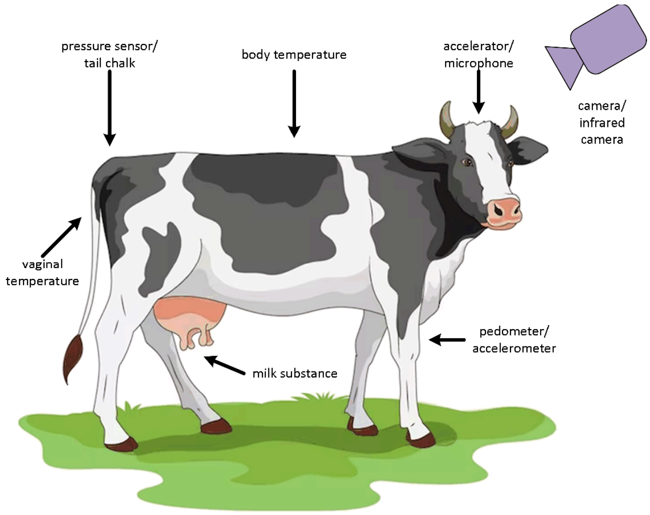
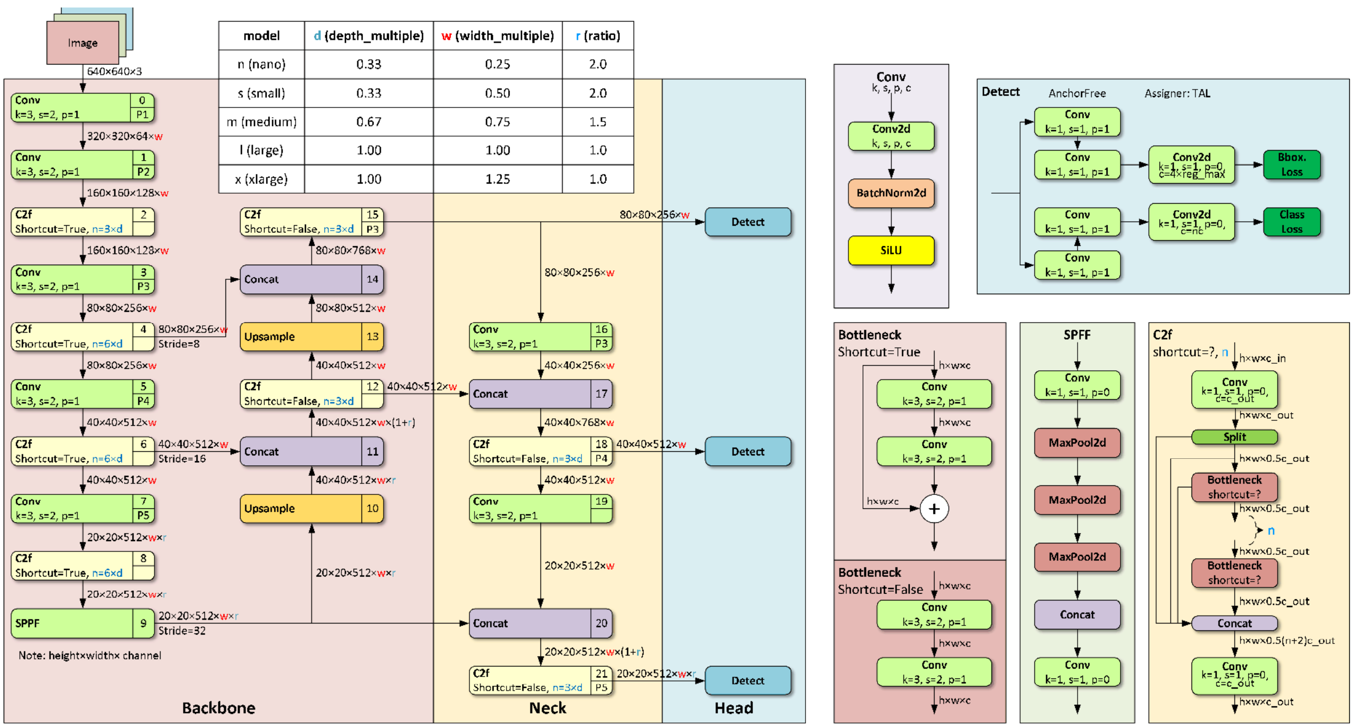


Fig. 2. Various methods for cow estrus detection

In addition, computer vision offers a data-driven strategy for herd management. Continuously collected data can be stored and analyzed for long-term trends, shedding light on breeding decisions, nutritional adjustments, and herd health as a whole. Computer vision transcends its status as a mere technological tool by positioning itself as a strategic asset for efficient and informed cattle management in free-stall barns.



model	d (depth_multiple)	w (width_multiple)	r (ratio)
n (nano)	0.33	0.25	2.0
s (small)	0.33	0.50	2.0
m (medium)	0.67	0.75	1.5
l (large)	1.00	1.00	1.0
x (xlarge)	1.00	1.25	1.0

Fig. 3. Various YOLOv8 Models

III. YOLOV8 ARCHITECTURES

YOLO (You Only Look Once) is a well-known object detection and image segmentation model developed in 2015 by Joseph Redmon and Ali Farhadi at the University of Washington. In 2016, YOLOv2 introduced batch normalization and anchor boxes, and in 2018, YOLOv3 was released with an enhanced backbone network and spatial pyramid pooling. The redesigned detection head and Mosaic data enhancement were introduced with YOLOv4 in 2020. After Meituan adopted YOLOv6 for their autonomous delivery robots, YOLOv5 featured hyperparameter optimization and export adaptability. YOLOv7 initiated pose evaluation. YOLOv8 by Ultralytics is a state-of-the-art model that includes detection, segmentation, pose estimation, tracking, and classification, demonstrating its adaptability for a vast array of vision AI applications [12]. Various YOLOv8 models is shown in Fig. 3.

The most recent cutting-edge model from Ultralytics, YOLOv8, was created for object detection, image classification, and instance segmentation. As successors to the influential YOLOv5, YOLOv8 introduces several architectural enhancements and developer-centric improvements. Although it is currently undergoing active development, with Ultralytics continuously refining and enhancing its features in response to community feedback, its credentials are already impressive. In particular, as indicated by COCO metrics, YOLOv8 is more accurate than YOLOv5 in Roboflow 100 dataset evaluations. Notable features of the model include a user-friendly command-line interface and a Python package designed to simplify the developer experience. Its extensive and expanding community ensures that developers can quickly find guidance and support for their computer vision projects.

IV. IMPLEMENTATION AND EXPERIMENTAL SETUP

To ensure the robustness and efficacy of the experiment, the software implementation made use of a variety of tools and libraries. Python version 3.9.16 forms the basis of our codebase. OpenCV version 4.6.0 was utilized for image processing and computer vision tasks. PyTorch, version 2.0.1, was the primary deep-learning framework used for this project. We utilized version 11.8 of the PyTorch-CUDA package for GPU acceleration and improved computational performance. In addition, we integrated torchvision version 0.15.2 for a variety of vision-related utilities and pretrained models. For the software stack to work effectively with the YOLO architecture and its variants, version 8.0.180 of the Ultralytics package was essential.

We utilized a high-performance hardware configuration for YOLOv8 experiments to ensure optimal analysis and execution. The processor was an Intel i9-13900K with 64 GB of DDR5 memory, allowing for efficient multitasking and data management. To meet storage requirements, a 1 TB NVME drive was utilized, enabling rapid data read and write operations. We used the NVIDIA GeForce RTX 4080 GPU with 16 GB of VRAM to ensure accelerated model training and inference processes for our deep learning tasks requiring high computational power. It is important to note that the training time mentioned in our results is relative to this particular hardware specification, and variations in hardware may result in different training durations.

Chokchai Farm in Khao Yai, Thailand, one of the largest dairy farms in Asia, collected the data [3]. Based on their behaviors, it was determined that three Holstein Friesian cows, each weighing approximately 450 kg and aged four years, were in estrus. These cows were then isolated in a designated pen for closer observation and examination for insemination readiness. A camera was installed within this enclosure to capture footage at a rate of 1.5 frames per second, producing 1280×960 pixel, 24-bit RGB images. Farm experts extracted and labeled 2,000 images from this footage to identify bounding boxes, body parts, and whether or not the cow was in estrus. Labelme and Label Studio were among the labeling tools used. The dataset generated by this procedure is available online at https://github.com/dsmlr/CowXNet [3]. Of the 2,000 images, 80% were allocated for training and the remaining 20% were used for validation.

V. RESULTS AND DISCUSSION

This section explores the dataset preparation, training outcomes with various YOLOv8 models and evaluates the testing performance of the optimal model.

A. Dataset Preparation

To convert a YOLOv4 image dataset with annotations stored in a pandas DataFrame to a YOLOv8 compatible format and partition it into training and validation sets, we begin by loading the annotations DataFrame from the 'annotations.pkl' file using pandas. Once loaded, we can inspect the DataFrame's columns and unique class labels. Next, we split the dataset into an 80% training set and a 20% validation set using the `train_test_split` function from `sklearn.model_selection`. We ensured that the split is stratified based on the class labels to maintain a consistent distribution of classes in both subsets.

After splitting, we converted the bounding box format to the YOLO format. For each image in the training and validation sets, we calculate the center coordinates, width, and height of the bounding box. We saved these annotations in separate text files corresponding to each image in the 'train/' and 'val/' directories respectively. Lastly, we organize the image files by moving them to their respective 'train/' and 'val/' directories. We ensure that the image filenames in the DataFrame match the actual image filenames in our directory, adjusting the file extension if necessary. By following these steps, we have a YOLOv8-compatible dataset ready for training and validation.

B. Training of Various YOLOv8 Models

Fig. 4 illustrates an example of Python code for YOLOv8 experiments with a fixed batch size, epoch, and initial learning rate. The complex relationship between model complexity, training time, and accuracy across YOLOv8 variants is examined in Table I.

```
from ultralytics import YOLO
# Load a pretrained model
model = YOLO('yolov8n.pt')
# Train the model
results = model.train(data='cow8.yaml',
epochs=100, lr0=0.01, batch=16)
```

Fig. 4. Code Snippet of Training Phase

Observing the progression of the number of parameters from YOLOv8n to YOLOv8x reveals a discernible pattern. This progression in model complexity correlates with the observed training durations; as expected, more complex models require longer training periods. However, it appears that this investment of time was worthwhile, as models with enhanced parameters, such as YOLOv8l and YOLOv8x, also demonstrate superior precision. While this suggests a direct relationship between model complexity and accuracy, it is important to weigh the benefits of this increased accuracy against the practical constraints of longer training times, especially in real-world applications where time and computational resources are frequently limited.

TABLE I. TRAINING RESULTS

Model	Parameters (millions)	Training Time (hours)	mAP50	mAP50-95
YOLOv8n	3.2	0.274	0.979	0.914
YOLOv8s	11.2	0.304	0.974	0.922
YOLOv8m	25.9	0.497	0.978	0.928
YOLOv8l	43.7	0.718	0.970	0.918
YOLOv8x	68.2	1.146	0.979	0.925

Table I details the training outcomes of various YOLOv8 models, including their parameters, durations of training, and mean average precision scores. Training time increases as the number of parameters (a measure of model complexity) increases. This trend illustrates the increased computational burden associated with the training of increasingly complex models. The mAP50, which represents the overlap precision, remains above 0.97 across all models. This suggests that all models detect objects that overlap the ground truth by at least 50 percent with near-perfect precision. Compared to the mAP50-95, the mAP average at various overlap thresholds reveals a greater degree of variation. Despite the increased complexity of YOLOv8l and YOLOv8x, YOLOv8m has the highest mAP50-95 value with a value of 0.928, which is marginally better than YOLOv8s but still superior to YOLOv8l and YOLOv8x.

YOLOv8m is a well-balanced option in terms of both performance and computational efficiency. It strikes a balance between model complexity and training time, with slightly superior mAP50-95 performance. Although YOLOv8l and YOLOv8x have more parameters and longer training times, their performance does not increase proportionally, making YOLOv8m the superior option. In light of these findings, YOLOv8m emerges as the best model for further testing, achieving the best balance between accuracy and computational resources.

C. Testing the Optimum Model

As shown in Fig. 5, the code snippet utilizes the ultralytics library to manage and validate a YOLO model. It then loads a model that has been pretrained using weights from a specific path after importing the YOLO class. It generates performance metrics by validating this model with the val() method, which internally references a dataset and training configurations that have been previously set. These metrics include mean average precision (mAP) values at different Intersections Over Union (IOU) thresholds (such as 0.50 and 0.75) and a list of mAP values for each dataset category. This code provides an overview of the object detection capabilities of the model across a variety of IOU thresholds and per category.

```
from ultralytics import YOLO
# Load the optimum model
model = YOLO('runs/detect/train3/weights/best.pt')
# Validate the model
metrics = model.val()
```

Fig. 5. Code Snippet of Testing Phase

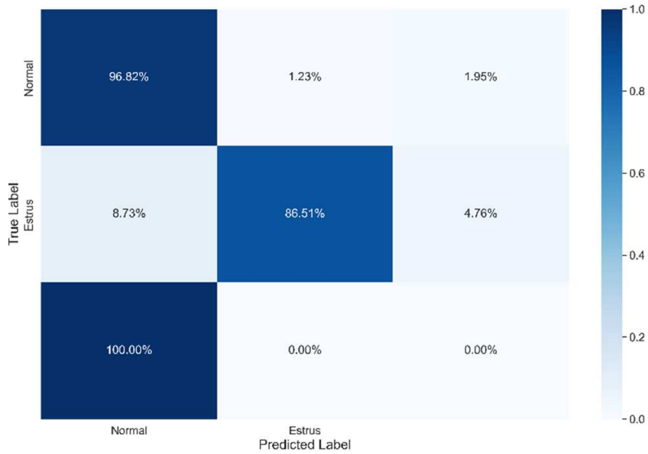


Fig. 6. Normalized confusion matrix for various classes

As depicted in Fig. 6, the confusion matrix provides a thorough evaluation of a model trained to differentiate between two classes: "Normal" and "Estrus." The precision of 0.96 indicates that 96 percent of all instances predicted by the model as belonging to a particular category were correctly classified. This level of accuracy suggests that the model generates few false positives. The recall of 0.9611 means that out of all the actual instances of a class in the dataset, the model correctly identified 96.11 percent of them, indicating a low rate of false negatives. The excellent mAP50 (mean average precision at 50 percent IoU) score of 0.9835 indicates that the model has a high overlap between the predicted and actual bounding boxes for both classes. The mAP50-95 is 0.9302, indicating consistent performance across multiple overlap criteria. The fitness score of 0.9355 reflects the model's overall performance by combining precision, recall, and mAP. In addition, the processing speed metrics reveal that preprocessing requires approximately 0.2545 seconds, inference requires approximately 3.7902 seconds, and post-processing requires approximately 0.6375 seconds, shedding light on the model's efficacy and optimization opportunities.

Fig. 7 is a comprehensive visualization of the training and validation processes' associated metrics and loss values. Specifically, it depicts the "train/box_loss" parameter, which represents the discrepancy between the predicted and actual bounding boxes during training. In addition, "train/cls_loss" indicates the classification error during the training phase, shedding light on the model's ability to distinguish between classes. The "train/dfl_loss" provides insight into the detection feature learning performance of the model. In addition, the figure displays "metrics/precision(B)" and "metrics/recall(B)," which represent the proportion of accurate positive predictions and the proportion of actual positives that were correctly predicted. Displaying "val/box_loss," "val/cls_loss," and "val/dfl_loss" for the validation set provides a comparative view of the model's performance on unseen data in terms of bounding box prediction, classification, and detection feature learning. The figure concludes with an explanation of the "metrics/mAP50(B)" and "metrics/mAP50-95(B)" metrics, which are mean average precision metrics at different Intersections over Union (IoU) thresholds and provide a holistic view of the model's accuracy across different overlap criteria.

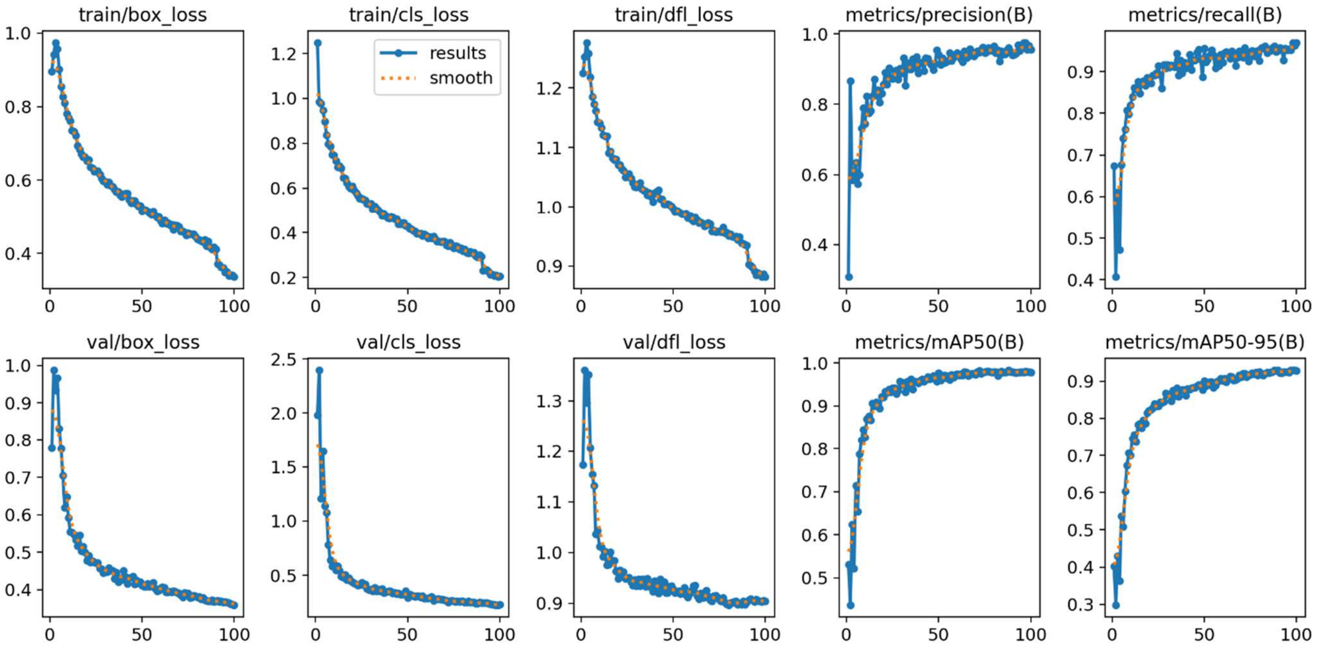


Fig. 7. Training and validation for YOLOv8m

Fig. 8. Samples of cow estrus detection in the free stall barn

In Fig. 8, we are presented with illustrative examples of the detection and classification outcomes for cows, highlighting their categorization into either "normal" or "estrus" conditions. This visualization offers a clear representation of how the model discerns and classifies cows based on their physiological states, providing a tangible understanding of its performance in real-world scenarios. Finally, YOLOv8m boasts advanced detection suitable for real-time applications. Easily convertible to TensorRT or ONNX, it ensures rapid performance across platforms. This adaptability, coupled with reduced latency and increased throughput, positions YOLOv8m as essential for varied real-time tasks.

VI. Conclusions and Future Works

In our exhaustive study, we investigated the potential of advanced object detection, with a particular emphasis on the YOLOv8m model. Our methodological approach was supported by exhaustive testing and validation, which ensured the model's adaptability to a wide variety of real-time applications. We observed a significant decrease in latency and a marked improvement in throughput, which were compelling results. Moreover, the model's precision in detecting and classifying conditions, as evidenced by metrics such as a 96 % precision rate and a 98.35 % mAP50 for cow detection, highlights its robustness and accuracy. In essence, our research demonstrates that the YOLOv8m model is a game-changer in real-time object detection by fusing technical excellence with tangible, consequential outcomes. Future endeavors will concentrate on the real-time implementation and validation of the model, particularly in the context of tie-stall barn environments.

Acknowledgment

The authors would like to acknowledge the support from the International Islamic University Malaysia, Airlangga University, and Telkom University for the research funding and facilities. The authors acknowledge that Petronas Research Sdn Bhd supported the hardware used.

References

[1] A. Monteiro, S. Santos, and P. Gonçalves, "Precision agriculture for crop and livestock farming—Brief review," *Animals,* vol. 11, no. 8, p. 2345, 2021.

[2] N.-A. Najm, L. Zimmermann, O. Dietrich, A. Rieger, R. Martin, and H. Zerbe, "Associations between motion activity, ketosis risk and estrus behavior in dairy cattle," *Preventive veterinary medicine,* vol. 175, p. 104857, 2020.

[3] T. Lodkaew, K. Pasupa, and C. K. Loo, "CowXNet: An automated cow estrus detection system," *Expert Systems with Applications,* vol. 211, p. 118550, 2023.

[4] A. S. Ali *et al.*, "Estrus Detection in a Dairy Herd Using an Electronic Nose by Direct Sampling on the Perineal Region," *Veterinary sciences,* vol. 9, no. 12, p. 688, 2022.

[5] S. Benaissa *et al.*, "Calving and estrus detection in dairy cattle using a combination of indoor localization and accelerometer sensors," *Computers and electronics in agriculture,* vol. 168, p. 105153, 2020.

[6] S. Reith and S. Hoy, "Behavioral signs of estrus and the potential of fully automated systems for detection of estrus in dairy cattle," *Animal,* vol. 12, no. 2, pp. 398-407, 2018.

[7] H. P. Marquez, D. Ambrose, A. Schaefer, N. Cook, and C. Bench, "Evaluation of infrared thermography combined with behavioral biometrics for estrus detection in naturally cycling dairy cows," *Animal,* vol. 15, no. 7, p. 100205, 2021.

[8] M. Saint - Dizier and S. Chastant - Maillard, "Towards an automated detection of oestrus in dairy cattle," *Reproduction in domestic animals,* vol. 47, no. 6, pp. 1056-1061, 2012.

[9] I. Devi *et al.*, "Vocal cues based Decision Support System for estrus detection in water buffaloes (Bubalus bubalis)," *Computers and Electronics in Agriculture,* vol. 162, pp. 183-188, 2019.

[10] P. Tassinari *et al.*, "A computer vision approach based on deep learning for the detection of dairy cows in free stall barn," *Computers and Electronics in Agriculture,* vol. 182, p. 106030, 2021.

[11] C. E. Cardoso Consentini, M. C. Wiltbank, and R. Sartori, "Factors that optimize reproductive efficiency in dairy herds with an emphasis on timed artificial insemination programs," *Animals,* vol. 11, no. 2, p. 301, 2021.

[12] J. Terven and D. Cordova-Esparza, "A comprehensive review of YOLO: From YOLOv1 and beyond. arXiv 2023," *arXiv preprint arXiv:2304.00501,* 2023.