International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 15, No. 4, December 2026, pp. 1861∼1870
ISSN: 2252-8776, DOI: 10.11591/ijict.v15i4.pp1861-1870

A muzzle-based self-enhancing cow recognition method using YOLO and LLM

Rifat Sarker Aoyon[1], Ismail Hossain[2], Jia Uddin[3]
[1]Department of Computer Science and Engineering, Brac University, Dhaka, Bangladesh
[2]Department of Computer Science and Engineering, George Mason University, Fairfax, USA
[3]Department of AI and Big Data, Endicott College, Woosong University, Daejeon, South Korea

Article Info

Article history:

Received Dec 30, 2024
Revised Jun 10, 2026
Accepted Jul 5, 2026

Keywords:

Cow muzzle
Image processing
LLM
Self-enhancing
YOLO

ABSTRACT

In this research, we have presented an approach to muzzle-based cow identification. To recognise muzzles, we have used you only look once (YOLO) and the structural similarity index (SSI). In the proposed method, a large amount of data is not required for the image processing task. Because here we included a self-enhancing feature, which helps the model to predict accurately and also helps to improve performance by increasing the amount of data. This certain program helps the model to improve its capability for delivering faster responses. large language model (LLM) is responsible for adding the self-enhancing feature. On the other hand, to find out the most suitable YOLO model for our research, we have trained data with three different versions of YOLO: YOLO v7, v8, and v9. We evaluated the performances of these three versions in terms of Precision, Recall, mAP50, and mAP50-95 and found YOLO v9 to be the best among them. It excels with a precision of 0.391, a recall of 0.916, a mAP50 of 0.559, and a mAP50-95 of 0.307. Here, data have been collected manually and were annotated by Roboflow.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Jia Uddin
Department of AI and Big Data, Endicott College, Woosong University
Daejeon, South Korea
Email: jia.uddin@wsu.ac.kr

1. INTRODUCTION

The cow is one of the most essential domestic animals in the entire world. People domesticate cows for various reasons, like the production of dairy, meat, leather accessories, and so on [1]. Moreover, cows can play crucial roles in the economy of a region [2]. So, people make cow farms so that they can take care of cows appropriately. To organize properly, sometimes cows need to take treatment. In this instance, cow recognition is highly needed. Besides, there are several regions of the world where the farmer can get insurance for the cow [3]. In this case, cow recognition is also required. Like these two sectors, there are also many more sectors available where cow recognition systems can play very important roles [4]. There are many AI-based cow recognition research projects that have been conducted till now. In AI-based solutions, data is pretty essential. On the other hand, it is time-consuming and pretty complicated to collect a sufficient amount of data. Additionally, the AI-based architectures are not suitable for non-computer scientists.

Moreover, when the model encounters data which is not familiar to the model, then it will be difficult for the model to provide appropriate output. That is why we have decided to make a novel framework of cow recognition that does not need a large amount of data and is easy for anyone to use, even without a background

in computer science. We used you only look once (YOLO) and structural similarity index (SSI) to recognize cows [5]. YOLO is pretty significant for image processing tasks like object detection, image segmentation, and so on. Besides, for real-time predictions, the performance of YOLO is quite phenomenal. As a consequence, we decided to use YOLO for detecting muzzles. Afterwards, we completed the task of muzzle comparison via SSI. The large language model (LLM) is trained on vast datasets to understand and respond to human language using transformer architecture [6]. YOLO has launched several versions till now. YOLO v7 is one of the amazing versions of YOLO for image processing and computer vision tasks [7]. YOLO v7 is comparatively better than its predecessors in terms of accuracy of real-time object detection and reduced 40% parameter than the earlier edition of the YOLO. We can divide the architecture of YOLO v7 into mainly three segments, which are the backbone, neck, and head. Here, the backbone is responsible for fetching essential features from the images and sending these to the Neck. The Neck combines all the features of the backbone and makes a pyramid. Finally, Head works for output prediction. YOLO v8 is one of the versions of YOLO that can outperform most other image processing algorithms [8]. To predict the bounding box or any class from the YOLO v8 model, it is required to follow several steps. In this case, the first one is dividing the image into a grid. The next task is to extract features using convolutional neural network (CNN) [9]. Then, the YOLO v8 model can predict the bounding box or class. And In YOLO v9, programmable gradient information (PGI) and the generalized efficient layer aggregation network (GELAN) enhanced its performance [10]. Like YOLO v7, the architecture of YOLO v9 can be divided into three parts: the backbone, neck, and head. Furthermore, the activities of these three parts are also similar. But YOLO v9 has an advanced layer, which enables YOLO v9 to perform better. In this research, we trained model with three different versions of YOLO and provided their performance. It will assist us in getting an idea of the appropriate version of YOLO for our proposed research.

Along with these, we have added self-enhancing features that enable our proposed system to consistently develop performance by interacting with the users. To develop this novel feature, we used LLM technology [11]. Our motive in this research is to launch a platform that will be capable of recognizing cows and improving the performance of the platform consistently by interacting with its users. This cow recognition system can be divided into two major segments, one of which is detecting muzzles from the cow image, and the other is comparing muzzles. We have used YOLO and SSI for the tasks of muzzle detection and muzzle comparison, respectively. Because, YOLO models are very popular for deployment because they are fast, easy to use, work well on low-cost devices, and have good support from the open-source community. So, YOLO is quite essential for non-computer scientists, who want to deploy object detection models in real-world applications. On the other hand, LLM technology is used here to improve performance by providing a faster response. Besides, LLM will help increase the amount of data consistently by interacting with the users. Also, we have tested the performance of three versions of YOLO, named YOLO v7, v8, and v9 to find the most suitable versions for our research.

2. RELATED WORKS

Zin *et al.* [12] have presented an approach to identify the cow based on the ear tag. Here, to fulfill this task appropriately, we used YOLO and CNN. They have collected data from the videos of cows on a dairy farm in Japan. Lu *et al.* [13] have disclosed a way of identifying cows by analyzing their iris. They analyzed the iris in three steps. Here, the last part is cow recognition, and transform (2D-CWT) is used by them to do this recognizing task. Gu *et al.* [14] have presented a way to recognize the behavior of the cows in terms of their activities. To accomplish this task, they collected 400 data of the cows. Yao *et al.* [15] have shared an idea of recognizing the cow face using feature extraction. In this research, the authors have used 50,000 data and got accuracy of 98.3% and 94.1% for the detection and cow face recognition, respectively. Li *et al.* [16] proposed an approach to recognise cows using CNN. Here, the authors have developed the CNN by including Alexnet.This improved model is able to secure the accuracy of 97.95%.

Zin *et al.* [17] have disclosed a unique way to recognize cows in terms of the black and white pattern of their bodies. Initially, they used inter-frame differencing and a horizontal histogram-based approach to fetch the body from the cow image and cropped this segment afterwards. Eventually, this certain region was utilized for training via deep learning algorithm. Kumar *et al.* [18] have shown a way to identify the cows based on the muzzle. They divided the entire research into mainly two segments, where the first is detecting muzzles and the last is comparing muzzles with the database. They used machine learning algorithm for detecting muzzles from the image of the cow face. On the other hand, one-shot-similarity

(OSS) matching and distance metric learning based techniques were used for comparing the muzzle structure. Bello *et al.* [19] have invented an incredible way to identify cows using their body patterns. They used CNN to train the model for cow recognition, while the amount of data was 1,000 images of the cows.

Zin *et al.* [20] have presented a method of cow tracking using the ear tag of the cow. At the beginning, they concentrated on detecting heads and did this via YOLO. Afterwards, they worked on ear recognition and have done so in three segments. To fulfill these three parts, they used image processing and CNN technologies.

Bello *et al.* [21] have shown a method to figure out the cow nose pattern . To fulfill this job, they used Stacked denoising auto-encoder and deep belief network. To train the model, they used data from 4,000 images. Kumar *et al.* [22] proposed an approach to recognise the faces of cattle using speeded up robust feature and local binary patterns. They used 1,200 images to train the model and got accuracy of 92.5%. Wang *et al.* [23] have shown a method for identifying estrus cows using YOLOv8n, where they used 1716 data. Besides, they have also highlighted the effectiveness of their proposed model. Li *et al.* [24] used a phenomenal dataset to identify cow behavior. They have collected data from 107 cows and divided their behavior into five classes. Weng *et al.* [25] introduced a unique CNN to recognize cows. All the data were collected from two cows, and images were collected from different angles to train properly. Kumar *et al.* [26] have presented a framework via YOLO to recognize the cow. Here, the propesed framwewok works to fetch only important feature of cow face and muzzle for ensuring both resource efficiency and accurate cattle identification. Bakhshayeshi *et al.* [27] have presented a method to recognise cows leveraging YOLOv5 and Siamese neural networks. With only 2,500 cow images, their proposed system secures significant accuracy. Additionally, this framework has the scope to integrate various sectors like farms, cargo areas, sale yards and so on.

3. METHOD

To complete our research, we have followed the sequence in Figure 1. Initially, we concentrated on collecting data and did the remaining tasks afterwards.

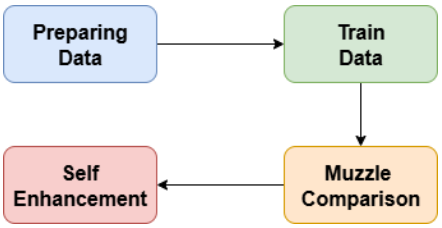


Figure 1. Process sequence of our research

3.1. Preparing data

We initially collected the images of the cow muzzle from online sources, some of them manually in real life. We have collected a total of 189 images of cow muzzles [28]. Afterwards, we used roboflow to create a bounding box in the region of the cow muzzle. Here, 142 data were allocated for training, whereas 31 and 16 images were used for validation and testing purposes. Each image was resized to a square dimension of 640x640 pixels. This resizing was performed by stretching the image to fit the specified dimensions, which may alter the original aspect ratio of the image.

3.2. Training and muzzle detection

To train the model for detecting muzzles from the cow images, we used the YOLO model. To find out the best YOLO model, we trained the data with three different versions of YOLO. These are YOLO v7, v8, and v9. We run each version for up to 25 epochs. In Table 1, the information for the training configuration of each model is given. We have also provided the three model summaries in Table 1. After training the models, the models were saved in the PyTorch format with the.pt extension. This file, which contains the trained model's weights and architecture, was subsequently utilized for detecting muzzles. We get the prediction of the coordinate of the muzzle from the trained model. Then, the muzzle part is cropped using the predicted coordinates. Afterwards, this cropped region will be moved forward to the comparison phase.

Table 1. Training configuration and model summary

Model name	Epochs	Image size	Batch size	Momentum	Weight decay	Layers	Parameters	Gradients	GFLOPs
YOLOv7	25	640×640	16	0.937	0.0005	415	37,201,950	37,201,950	105.1
YOLOv8	25	800×800	16	0.937	0.0005	225	11,136,374	11,136,358	28.6
YOLOv9	25	640×640	16	0.937	0.0005	621	25,438,614	25,438,598	103.2

3.3. Muzzle comparision

At first, the two muzzles will be converted to the same dimensions and converted to grayscale afterwards. Following that, the absolute difference between the grayscale images is computed. This difference in image highlights the regions where pixel values vary between the two images. The difference image is then thresholded to produce a binary image. Pixels with differences above a specified threshold are marked, emphasizing the areas of significant change. After that, the SSI is calculated [29]. The SSI measures how similar two grayscale images are by comparing their brightness, contrast, and structure. Grayscale images are used in SSI calculations to simplify the comparison by focusing solely on brightness and structural features without the added complexity of color differences. This makes it easier to assess and quantify similarities and differences in image content. If the SSI value is greater than 0.66, then it will be considered that two muzzles are the same. Otherwise, these are different.

3.3.1. Structural similarity index

The SSI is an incredible algorithm to measure the similarity between two images. To fetch the similarity via SSI, the pixel-wise comparison is not just between the two images; also, the overall structural, contrast, and luminance similarity between the images will be calculated. To extract the SSI, we first need to figure out the difference between two images of luminance, contrast, and structure. Luminance calculation is the process of calculating the average brightness of both images. On the other hand, contrast calculation means measuring the variations in brightness. The structure difference is the consistency of the patterns and textures between the two images. After finding these three values, it is needed to combine this three values.Then we will get the SSI [30]. The process of generating SSI is visually shown in Figure 2.

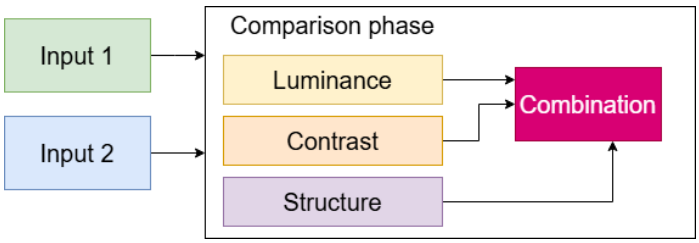


Figure 2. Process of generating SSI

The luminance comparison between two images x and y is given by (1), which measures how similar the average brightness levels of the images are. Here, μ_x and μ_y represent the mean luminance of images x and y, respectively, and C_1 is a constant to avoid division by zero. The constant C_1 is defined in (2).

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{1}$$

$$C_1 = (K_1 \cdot L)^2 \tag{2}$$

The contrast comparison is defined by (3), which evaluates the similarity in contrast between images x and y. Here, σ_x and σ_y denote the standard deviations of pixel values in images x and y, respectively, and C_2 is a constant to avoid division by zero. The constant C_2 is defined in (4).

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{3}$$

$$C_2 = (K_2 \cdot L)^2 \tag{4}$$

In (5) describes the structure comparison, which assesses how similar the structural patterns in images x and y are. The covariance σ_{xy}, along with the standard deviations σ_x and σ_y, and constant C_3, which is often defined as $C_3 = \frac{C_2}{2}$, are used in this calculation.

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x \sigma_y + C_3} \tag{5}$$

The final SSI is given by (6), which combines the luminance, contrast, and structure comparisons using weights α, β, and γ to determine the overall similarity between the images.

$$SSI(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma} \tag{6}$$

Finally, the covariance calculation is provided in (7). This equation calculates the covariance σ_{xy} between images x and y, which is crucial for the structure comparison component.

$$\sigma_{xy} = \frac{1}{N-1} \sum_{i=1}^{N} (x_i - \mu_x)(y_i - \mu_y) \tag{7}$$

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{8}$$

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{9}$$

$$\text{mAP50} = \frac{1}{C} \sum_{i=1}^{C} \text{AP}_i(\text{IoU} = 0.50) \tag{10}$$

$$\text{mAP50-95} = \frac{1}{C} \sum_{i=1}^{C} \frac{1}{T} \sum_{t=1}^{T} \text{AP}_i(\text{IoU}_t) \tag{11}$$

3.4. Self enhancement

After predicting the muzzle from the YOLO model, it will first be checked to see whether the predicted output is accurate or not. To do so, we send a query to LLM with the output image. If the result is positive, the process will be forwarded to the next step. Otherwise, the accurate coordinates of the muzzle bounding box will be predicted via LLM. Afterwards, the image and predicted coordinates will be saved in the database and will move on to the next phase. Thus, the amount of data will be consistently increasing by interacting with the users, which will improve the performance of providing faster responses. However, this platform will always be capable of providing the correct response. Here, we used GPT 4o Api for getting output [31]. To get an accurate prediction from GPT 4o LLM, we initially provided several cow images with their corresponding coordinates of the muzzle so that LLM can understand the pattern of recognition of the region of the muzzle. The process sequence of self-enhancement feature is visually given in Figure 3.

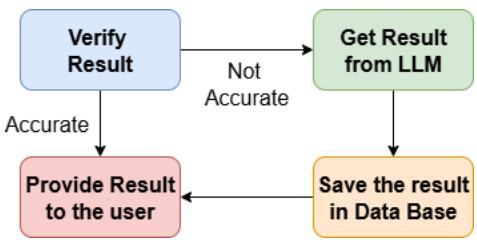


Figure 3. Process sequence of Self Enhancement

3.5. Ultimate output prediction

The overview of ultimate output prediction is visually presented in the Figure 4. Initially, the user needs to provide two inputs. Following that, the coordinates of the muzzles will be predicted from the trained model for two inputs separately. Whether the prediction is accurate or not will be checked by our proposed "Self Enhancement" feature. After going through the process of "Self Enhancement", the ultimate output of muzzle co-ordinate will be predicted. Then, muzzles will be cropped, and these cropped muzzles will be compared based on the SSI. If the SSI value is greater than 0.66, then it will be considered that two muzzles are the same. Otherwise, these are different.

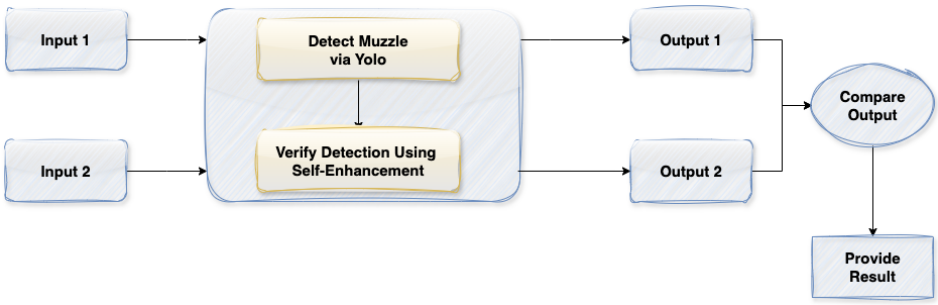


Figure 4. Overview of ultimate output prediction

4. RESULT ANALYSIS

In our research, we trained the data using three different versions of YOLO. The performance of YOLOv7, YOLOv8, and YOLOv9 was evaluated based on four metrics: precision, recall, mAP50, and mAP50-95 [32]-[35]. The results of these three versions in terms of the four metrics are provided in Table 2. Precision is the accuracy of positive predictions, while recall measures the ability to identify all relevant instances of a model. The formulas for precision and recall are provided in (8) and (9), respectively. On the other hand, mAP50 is the mean average precision calculated at an intersection over union (IoU) threshold of 0.50, while mAP50-95 is the mean average precision averaged over multiple IoU thresholds ranging from 0.50 to 0.95. The formulas for mAP50 and mAP50-95 are given in (10) and (11), respectively.

From Table 2, we can see that YOLOv9 outperforms the other versions in terms of eecall, mAP50, and mAP50-95. However, YOLOv7 provides the best result in terms of precision. Based on the four performance metrics in Table 2, YOLOv9 is selected as the best overall. Although YOLOv7 and YOLOv8 perform better than YOLOv9 in Precision, YOLOv9 is preferred due to its superior performance in the more critical metrics of mAP50 and mAP50-95.

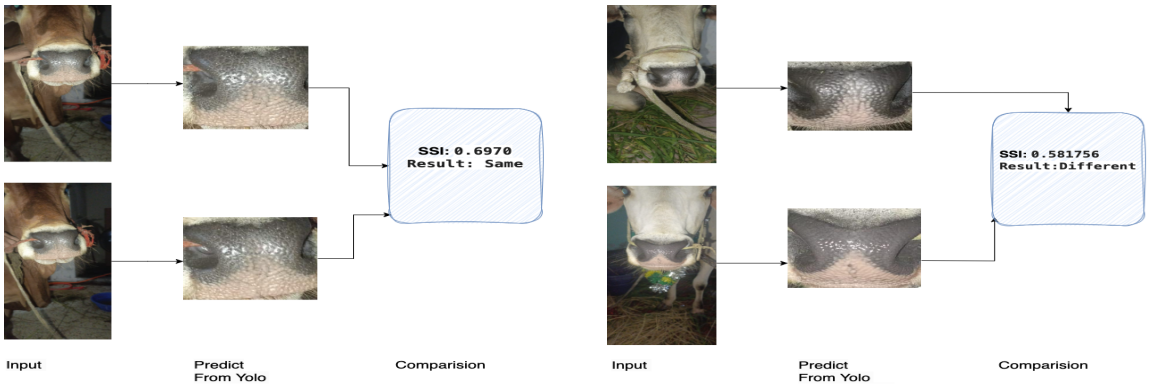
Figure 5 presents the final prediction results. Figures 5(a) and 5(b) provide two sample final predictions. From our proposed system, users can determine whether two cows are the same or not based on the muzzle. The SSI values of the inputs in Figures 5(a) and 5(b) are 0.6970 and 0.581756, respectively. If the SSI is greater than 0.66, the cows are considered to be the same. Therefore, the ultimate outputs for Figures 5(a) and 5(b) are the same and different, respectively. These results indicate that the proposed algorithm is capable of making accurate predictions.

Moreover, we have added a unique feature to the proposed system named "Self-Enhancing" using an LLM. Figure 6 shows three sample muzzle predictions generated by the LLM.

Finally, Figure 7 presents the performance comparison of the three YOLO versions. Figures 7(a)-7(c) show the performance of YOLOv7, YOLOv8, and YOLOv9, respectively, based on Precision, Recall, mAP50, and mAP50-95.

Table 2. Performance comparison

Model Name	Precision	Recall	mAP50	mAP50-95
YOLO v7	0.761	0.239	0.351	0.0816
YOLO v8	0.585	0.56	0.535	0.304
YOLO v9	0.391	0.916	0.559	0.307



(a) (b)

Figure 5. Sample ultimate prediction (a) same cows and (b) different cows

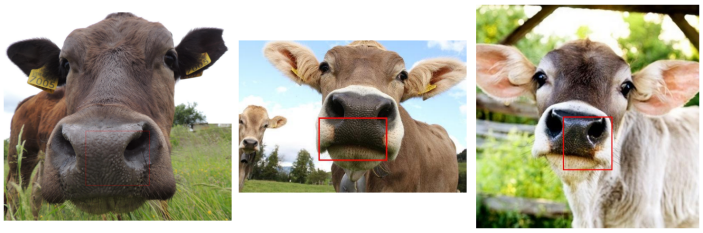
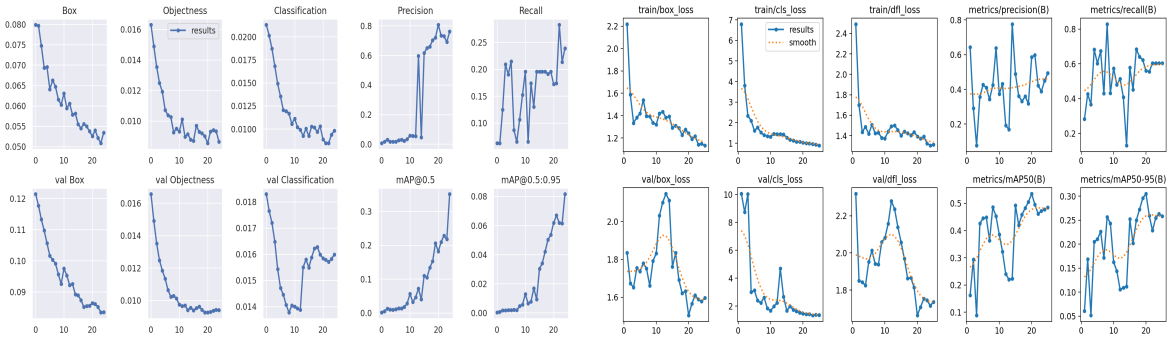
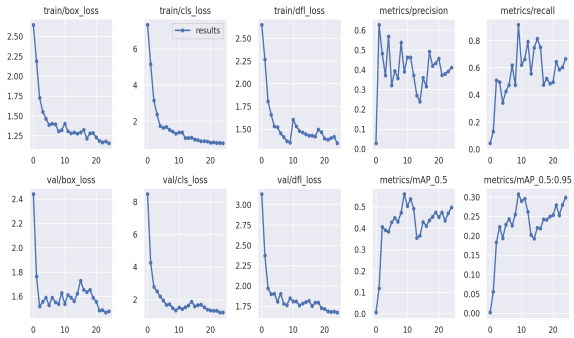


Figure 6. Sample prediction using LLM



(a) (b)



(c)

Figure 7. Training performance comparison of the evaluated YOLO models;
(a) performance of the trained YOLOv7 model, (b) performance of the trained YOLOv8 model, and
(c) performance of the trained YOLOv9 model

5. CONCLUSION

In this research, we have demonstrated an insane method and also presented the effectiveness of the performance of the proposed framework where YOLO and LLM work together to deliver appropriate output to the users. Moreover, we have also compared the performance of three types of YOLO versions named YOLOv7, YOLOv8, and YOLOv9. On the other hand, it is also proved that YOLO is well capable of learning with a little amount of Roboflow-annotated data. That has shown potential to implement the services of YOLO and Roboflow in many industry-friendly products. Along with this, the integration of LLM into the system has made the scope to implement the system instantly to the users without any data, and that becomes feasible because of having a feature in the system named self-enhancing. That is how we accomplished the research of making a user-friendly cow identification framework with low computational resources and auto-improving ability. In the future, we have the motive to develop this framework by ensuring the data security of the users by implementing distributed ledger technology like blockchain. This improvement will open more doors to use this framework where data security is highly essential. Also, we have another objective to add offline LLM like LLaVA in the future.

FUNDING INFORMATION

This research is funded by Woosong University Academic Research 2026.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Rifat Sarker Aoyon	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Ismail Hossain		✓				✓		✓	✓	✓	✓	✓		
Jia Uddin					✓		✓			✓		✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

The dataset is available at this link: https://github.com/JiaUddinPhD/Cow-Muzzle-Dataset.git.

REFERENCES

[1] T. Mihret, Mitku, Fentahun, and T. Guadu, "Dairy farming and its economic importance in Ethiopia: a review," *World Journal of Dairy & Food Sciences*, vol. 12, no. 1, pp. 42–51, 2017, doi: 10.5829/idosi.wjdfs.2017.42.51.

[2] R. Grant and W. H. Miner, "Economic benefits of improved cow comfort," *Novus International, Inc.*, no. April, pp. 1–26, 2015, [Online]. Available: https://www.dairychallenge.org/pdfs/2015_National/resources/Novus_Economic_Benefits_of_Improved_Cow_Comfort_April_2015.pdf.

[3] J. Trivedi and B. K. Soni, "A study on cattle insurance in villages of gandhinagar district," *Business Perspectives and Research*, vol. 2, no. 2, pp. 25–36, 2014, doi: 10.1177/2278533720140204.

[4] A. I. Awad, "From classical methods to animal biometrics: a review on cattle identification and tracking," *Computers and Electronics in Agriculture*, vol. 123, pp. 423–435, Apr. 2016, doi: 10.1016/j.compag.2016.03.014.

[5] M.-J. Chen and A. C. Bovik, "Fast structural similarity index algorithm," *Journal of Real-Time Image Processing*, vol. 6, no. 4, pp. 281–287, Dec. 2011, doi: 10.1007/s11554-010-0170-9.

[6] E. Hemberg, S. Moskal, and U.-M. O'Reilly, "Evolving code with a large language model," *Genetic Programming and Evolvable Machines*, vol. 25, no. 2, p. 21, Dec. 2024, doi: 10.1007/s10710-024-09494-2.

[7] S. Zhou *et al.*, "An accurate detection model of takifugu rubripes using an improved YOLO-V7 network," *Journal of Marine Science and Engineering*, vol. 11, no. 5, p. 1051, May 2023, doi: 10.3390/jmse11051051.

[8] F. M. Talaat and H. ZainEldin, "An improved fire detection approach based on YOLO-v8 for smart cities," *Neural Computing and Applications*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023, doi: 10.1007/s00521-023-08809-1.

[9] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, vol. 33, no. 12, pp. 6999–7019, 2021.

[10] M. A. R. Alif and M. Hussain, "YOLOv1 to YOLOv10: a comprehensive review of YOLO vari- ants and their application in the agricultural domain," *arXiv preprint arXiv:2406.10139*, 2024.

[11] Z. Qi *et al.*, "WebRL: training LLM web agents via self-evolving online curriculum reinforcement learning," in *13th International Conference on Learning Representations, ICLR 2025*, Jan. 2025, pp. 65879–65909. [Online]. Available: http://arxiv.org/abs/2411.02337

[12] T. T. Zin *et al.*, "Cow identification system using ear tag recognition," in *2020 IEEE 2nd Global Conference on Life Sciences and Technologies (LifeTech)*, IEEE, Mar. 2020, pp. 65–66, doi: 10.1109/LifeTech48969.2020.1570625232.

[13] Y. Lu, X. He, Y. Wen, and P. S. P. Wang, "A new cow identification system based on iris analysis and recognition," *International Journal of Biometrics*, vol. 6, no. 1, p. 18, 2014, doi: 10.1504/IJBM.2014.059639.

[14] J. Q. Gu, Z. H. Wang, R. H. Gao, and H. R. Wu, "Cow behavior recognition based on image analysis and activities," *International Journal of Agricultural and Biological Engineering*, vol. 10, no. 3, pp. 165–174, 2017, doi: 10.3965/j.ijabe.20171003.3080.

[15] L. Yao, Z. Hu, C. Liu, H. Liu, Y. Kuang, and Y. Gao, "Cow face detection and recognition based on automatic feature extraction algorithm," in *Proceedings of the ACM Turing Celebration Conference - China*, New York, NY, USA: ACM, May 2019, pp. 1–5, doi: 10.1145/3321408.3322628.

[16] S. Li *et al.*, "Individual dairy cow identification based on lightweight convolutional neural network," *PLOS ONE*, vol. 16, no. 11, p. e0260510, Nov. 2021, doi: 10.1371/journal.pone.0260510.

[17] T. T. Zin, C. N. Phyo, P. Tin, H. Hama, and I. Kobayashi, "Image technology based cow identification system using deep learning," in *Lecture Notes in Engineering and Computer Science*, 2018, pp. 320–323.

[18] S. Kumar, S. K. Singh, T. Dutta, and H. P. Gupta, "A fast cattle recognition system using smart devices," in *Proceedings of the 24th ACM international conference on Multimedia*, New York, NY, USA: ACM, Oct. 2016, pp. 742–743, doi: 10.1145/2964284.2973829.

[19] R.-W. Bello, A. Zawawi, A. Sufril, D. A., and F. Noah, "Image-based individual cow recognition using body patterns," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 3, pp. 92–98, 2020, doi: 10.14569/IJACSA.2020.0110311.

[20] T. T. Zin *et al.*, "Automatic cow location tracking system using ear tag visual analysis," *Sensors*, vol. 20, no. 12, p. 3564, Jun. 2020, doi: 10.3390/s20123564.

[21] R. Bello, A. Z. H. Talib, and A. S. A. Bin Mohamed, "Deep learning-based architectures for recognition of cow using cow nose image pattern," *Gazi University Journal of Science*, vol. 33, no. 3, pp. 831–844, Sep. 2020, doi: 10.35378/gujs.605631.

[22] S. Kumar, S. Tiwari, and S. K. Singh, "Face recognition for cattle," in *2015 Third International Conference on Image Information Processing (ICIIP)*, IEEE, Dec. 2015, pp. 65–72, doi: 10.1109/ICIIP.2015.7414742.

[23] Z. Wang, Z. Hua, Y. Wen, S. Zhang, X. Xu, and H. Song, "E-YOLO: recognition of estrus cow based on improved YOLOv8n model," *Expert Systems with Applications*, vol. 238, p. 122212, Mar. 2024, doi: 10.1016/j.eswa.2023.122212.

[24] K. Li, D. Fan, H. Wu, and A. Zhao, "A new dataset for video-based cow behavior recognition," *Scientific Reports*, vol. 14, no. 1, p. 18702, Aug. 2024, doi: 10.1038/s41598-024-65953-x.

[25] Z. Weng, F. Meng, S. Liu, Y. Zhang, Z. Zheng, and C. Gong, "Cattle face recognition based on a two-branch convolutional neural network," *Computers and Electronics in Agriculture*, vol. 196, p. 106871, May 2022, doi: 10.1016/j.compag.2022.106871.

[26] N. Kumar, A. Sharma, A. Kumar, R. Singh, and S. K. Singh, "Cattle verification with YOLO and cross-attention encoder-based pairwise triplet loss," *Computers and Electronics in Agriculture*, vol. 234, p. 110223, Jul. 2025, doi: 10.1016/j.compag.2025.110223.

[27] I. Bakhshayeshi, E. Erfani, F. R. Taghikhah, S. Elbourn, A. Beheshti, and M. Asadnia, "An intelligence cattle reidentification system over transport by siamese neural networks and YOLO," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2351–2363, Jan. 2024, doi: 10.1109/JIOT.2023.3294944.

[28] J. Uddin, "GitHub - JiaUddinPhD/Cow-Muzzle-Dataset," GitHub. [Online]. Available: https://github.com/JiaUddinPhD/Cow-Muzzle-Dataset

[29] I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli, and L. Vanneschi, "Structural similarity index (SSIM) revisited: a data-driven approach," *Expert Systems with Applications*, vol. 189, p. 116087, Mar. 2022, doi: 10.1016/j.eswa.2021.116087.

[30] D. Brunet, E. R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1488–1495, Apr. 2012, doi: 10.1109/TIP.2011.2173206.

[31] OpenAI *et al.*, "GPT-4o system card," arXiv preprint arXiv:2410.21276, 2024, [Online]. Available: http://arxiv.org/abs/2410.21276

[32] M. Buckland and F. Gey, "The relationship between recall and precision," *Journal of the American Society for Information Science*, vol. 45, no. 1, pp. 12–19, Jan. 1994, doi: 10.1002/(SICI)1097-4571(199401)45:1¡12::AID-ASI2¿3.0.CO;2-L.

[33] M. S. M. Sajjadi, O. Bousquet, O. Bachem, M. Lucic, and S. Gelly, "Assessing generative models via precision and recall," *Advances in Neural Information Processing Systems*, vol. 2018-December, pp. 5228–5237, 2018.

[34] X. Li, W. Wang, L. Sun, B. Hu, L. Zhu, and J. Zhang, "Deep learning-based defects detection of certain aero-engine blades and vanes with DDSC-YOLOv5s," *Scientific Reports*, vol. 12, no. 1, p. 13067, Jul. 2022, doi: 10.1038/s41598-022-17340-7.

[35] D. Reis, J. Kupec, J. Hong, and A. Daoudi, "Real-time flying object detection with YOLOv8," *arXiv preprint arXiv:2305.09972, 2024*, [Online]. Available: http://arxiv.org/abs/2305.09972

BIOGRAPHIES OF AUTHORS

Rifat Sarker Aoyon received the B.S. degree in computer science from BRAC University, Dhaka, Bangladesh, in 2023. He is currently pursuing a master's degree in computer engineering at Chosun University, Gwangju, South Korea. He has authored several publications in conference proceedings and journals. He is currently working as a graduate research assistant at Chosun University, South Korea. His research interests include deep learning, federated learning, mobile computing, and computer vision. He can be contacted at email: rifat.sarker.aoyon@g.bracu.ac.bd.

Ismail Hossain is an educator, scientist, and software engineer. He is a software engineer in MissionSquare Retirements, Washington DC, USA. He is also a lecturer (currently on leave) in the CSE department at Brac University, Bangladesh. He received his M.S. in Computer Science Degree from George Mason University, Virginia, USA, and completed his B.Sc. in Computer Science from Brac University, Bangladesh. His research interest spans large language models, machine translations, social data mining, and lifelong learning. He can be contacted at email: ihossai4@gmu.edu.

Jia Uddin is an associate professor in the AI and Big Data Department, Endicott College, Woosong University, Korea. He was an associate professor in the CSE department at BRAC University, Bangladesh. He received a Ph.D. degree (Computer Engineering) from the University of Ulsan, South Korea in January 2015 and an M.Sc. in Telecommunications from Blekinge Institute of Technology, Sweden, in 2010. He was a member of the Self-Assessment Team (SAC) of CSE, BRACU in the HEQEP project funded by the World Bank and the University of Grant Commission Bangladesh in 2016–2017. His research area includes fault diagnosis using AI, audio and image processing. He can be contacted at email: jia.uddin@wsu.ac.kr.