Smart Agricultural Technology 12 (2025) 101304

Contents lists available at ScienceDirect

Smart Agricultural Technology

journal homepage: www.journals.elsevier.com/smart-agricultural-technology

Automated dairy cow identification and feeding behaviour analysis using a computer vision model based on YOLOv8

Claudia Giannone [a,*], Mohsen Sahraeibelverdy [a], Martina Lamanna [b], Damiano Cavallini [b], Andrea Formigoni [b], Patrizia Tassinari [a], Daniele Torreggiani [a], Marco Bovo [a]

[a] *Department of Agricultural and Food Sciences, University of Bologna, Bologna (BO), Italy*
[b] *Department of Veterinary Medical Sciences, University of Bologna, Ozzano dell'Emilia (BO), Italy*

ARTICLE INFO

Keywords:
Dairy cow
Behaviour analysis
Deep learning
AI
Feeding

ABSTRACT

Monitoring changes in the feeding behaviour of dairy cows is essential for assessing their feeding preferences, milk production, and health status. Sick cows often exhibit altered feeding patterns, such as reduced feeding time and frequency, making early detection crucial for effective farm management. Traditional methods for monitoring feeding behaviour are labour-intensive, time-consuming, and prone to errors. To address these challenges, precision livestock farming technologies have gained increasing attention. While wearable sensors, such as accelerometers and RFID tags, provide accurate data, they have limitations, including high costs and potential stress on animals. Alternatively, computer vision-based approaches offer a non-invasive and efficient solution for monitoring feeding behaviour. Deep learning techniques, particularly the YOLO (You Only Look Once) object detection model, have been widely applied in animal husbandry. Despite advancements in object detection, individual cow recognition in operational environment remains a challenge due to the lack of a standardized and viable approach.

The main aim of the paper is to evaluate the reliability and validate a deep learning-based computer vision model for automatically recognizing individual cows at the feeding lane in a relevant environment. By identifying individual cows, it is possible to determine their feeding time, feeding duration and daily frequency. The paper describes the work phases from data collection to analysis and validation of an improved YOLOv8n model that, after a fine-tuning on the collected video set, achieved a precision of 85 %, a recall of 62 % (F1 score 0.72) at IoU 0.5 and processes a 640 × 640 pixels frame in just 12 ms on an NVIDIA RTX 2080. The promising results presented here contribute to the advancement and validation of computer vision applications in herd monitoring, supporting the commercial adoption of these technologies for analysing cow behaviour so increasing animal welfare and the sustainability of the animal production.

1. Introduction

Monitoring the changes in the feeding behaviour of dairy cows is critical for evaluating their feeding preference and milk production [1–3]. Detecting feeding behaviour can also be used to predict health status [4,5] since sick and healthy cows express different feeding behaviours, including parameters such as number of accesses to the feeding lane and daily feeding time [6–8]. For instance, it has been reported that lame cows reduced daily feeding time and feeding frequency by 46 % and 44 %, respectively [9]. Similarly, daily feeding time decreases dramatically in cows with ketosis if compared to healthy cows [10]. Traditional diagnostic procedures and behaviour recognition methods are time-consuming, prone to error, and labour-intensive, resulting in major concerns for farmers [11–13]. To overcome these problems, precision livestock farming (PLF) technologies are becoming increasingly common in modern dairy farms. In fact, in recent years, there has been a growing interest in automated systems for monitoring the health and the welfare of dairy cattle [14]. Different wearable sensors, such as acoustic sensors, radio frequency identification (RFID), accelerometers, and pressure sensors can be used to detect sudden changes in the eating and health of animals [15,16]. Despite the wearable sensors providing accurate information, they have several restrictions such as high cost and damage-sensitive, which may lead to stress reaction in dairy cows [2]. On the other hand, computer vision approaches are non-invasive, offer a high-speed response, and can

---

* Corresponding author.
E-mail address: claudia.giannone2@unibo.it (C. Giannone).

https://doi.org/10.1016/j.atech.2025.101304
Received 2 July 2025; Received in revised form 9 August 2025; Accepted 9 August 2025

Nomenclature table

YOLO	(You Only Look Once)	SGD	(Stochastic Gradient Descent)
ROI	(Region Of Interest)	NMS	(Non-Maximum Suppression)
SPPF	(Spatial Pyramid Pooling – Fast)	IoU	(Intersection over Union)
PAN	(Pan Aggregation Network)	A	(Accuracy)
ICT	(Information Communication Technology)	P	(Precision)
RFID	(Radio Frequency IDentification)	R	(Recall)
PoE	(Power over Ethernet)	F_1	(F_1 score)
DFL	(Distribution Focal Loss)	mAP	(mean Average Precision)
CIoU	(Complete Intersection over Union)	TP	(True Positives)
CE	(Cross-Entropy)	TN	(True Negatives)
		FP	(False Positives)
		FN	(False Negatives)

prevent stress issues caused by wearable sensors [17].

In recent years, deep learning algorithms have been widely used in animal husbandry. YOLO (You Only Look Once) [18] is a popular object detection model, known for its real-time performance and accuracy, that has been used in several studies on dairy cows monitoring [19,20]. Different versions of YOLO have been developed to improve its efficiency [21]. Some research studies have been conducted using the YOLO model to biometric identification of dairy cows [22–24]. In an experiment on a group of housed Holstein heifers, it was reported that YOLOv3 predicted feeding frequency and feeding time with R^2 values of 0.55 and 0.99, respectively [2,25] compared the performances of YOLOv4 and DRN-YOLO in detection of feeding behaviour of dairy cows. Results showed that YOLOv4 and DRN-YOLO predicted feeding behaviour of cows with high precision (95.46 % and 97.16 %, respectively). However, they found that models mistakenly classified grass arching behaviour as a feeding behaviour.

The main aim of this study is to develop and test the reliability level of a computer vision system, based on deep learning techniques, for the automatic recognition of individual cows at the feeding lane within images representing their position. In particular, whilst developing a new software framework lies outside the scope of this paper, the paper focuses on methodological aspects related to a more efficient application of Information Communication Technology (ICT) in the dairy cows monitoring field. In fact, while object detection is already applied in commercial applications in various contexts, individual cow recognition still represents an open issue for both the research and commercial fields. Indeed, this topic has been gaining increasing interest in literature in the last few years [26–29]. Starting from the identification of the single cow is possible to calculate the daily time spent at the feeding lane and the feeding frequency during the day. On a practical level, an automatic recognition system could allow farmers to better manage their time, resources and intervention on any problems related to animal health such as ketosis, mastitis, and more, resulting in significant economic savings and increasing the sustainability of the animal production sector.

The study was intended to test whether it is possible to estimate with sufficient precision the time spent at the feeder in a small portion of the feeding lane used as preliminary case study. Our approach involved installing a camera in the barn, fine-tuning a model to identify cows within camera’s field of view and monitor the behaviour using a region-of-interest (ROI) filter.

The promising results reported here represents a relevant contribution to the progress and validation of computer vision for herd monitoring applications and could be an important support for the commercial applications of these technologies for the analysis of cow actions and behaviour. In fact, by scaling the current experimental setup and monitoring the entire length of the feeding lane, it will be possible to obtain the daily data related to the feeding behaviour of each animal.

2. Materials and methods

2.1. Experimental setup

The experimental campaign was conducted at the teaching and experimental dairy barn of the University of Bologna located in Ozzano dell’Emilia (Latitude 44.424; Longitude 11.479) in northern Italy (see Fig. 1). The facility, which was built in 2003 for institutional educational and research activities, houses in total about 170 animals, with about 90 lactating cows and approximately 80 dry and replacement heifers. Within the barn, cows were grouped based on the different phases of the production cycle.

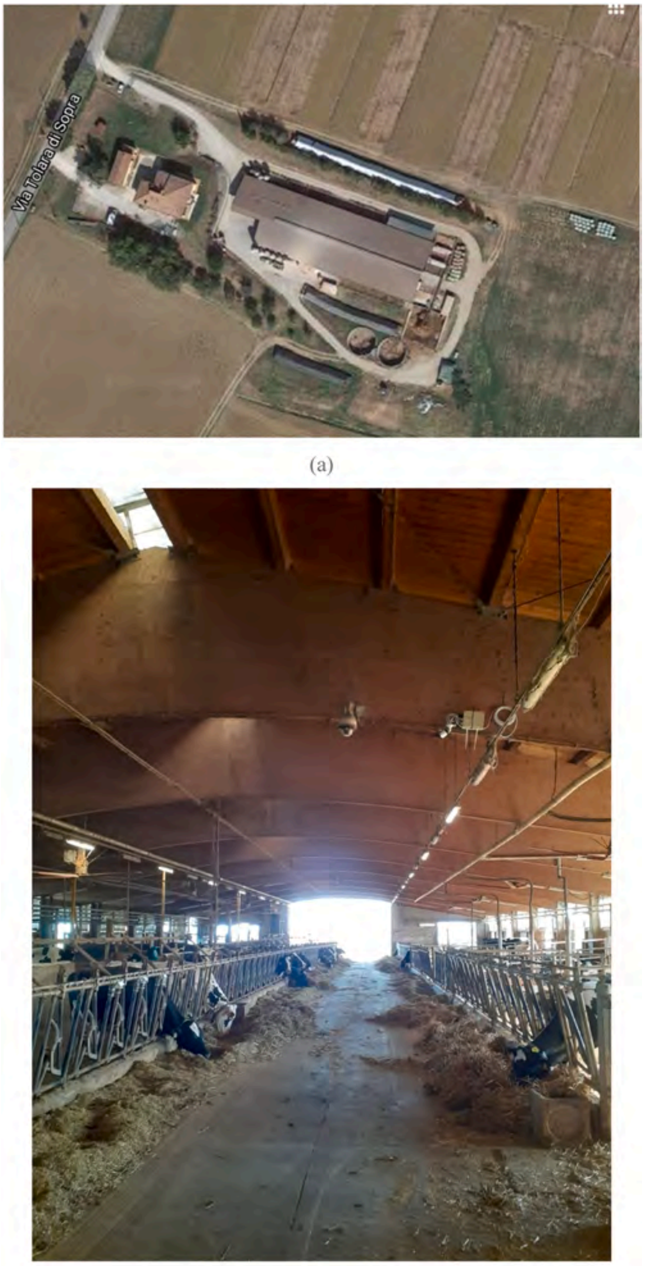
The facility was structured into four distinct areas: the feeding area, located along the two sides of the central alley, the exercise area, situated in the outdoor pens, the resting area, organized into cubicles, and the milking area. The latter was fully automated, utilizing two Merlin2 automatic milking systems (Fullwood Packo, Shropshire, UK) installed in 2021. The barn is provided with cooling ventilation system based on the Temperature-Humidity Index (THI) value.

A subset of nineteen lactating cows, which were the focus of the study, was housed in a restricted portion of the free-stall barn, with free access to cubicle stalls and the feeding area. These cows were fed twice per day with total mixed ration and auto-feeders for concentrate supplementation. They had free access to one of the two automatic milking systems installed in the barn, and for each cow, the behavioural, productive and health parameters are daily recorded automatically. Further information related to the herd consistency and facilities can be found in [30] and [31].

2.2. Recording system, data collection and pre-processing

A two-dimensional camera, positioned on a beam of the roof at about 4.5 meters of height at about 4 meters from the front feeding area, was used for video recording (see Fig. 1 b). The camera height and positioning (see Fig. 2) were optimized to maximize visibility of a portion of the feeding area, allowing detailed observation of feeding behaviour without disturb and influence routine and habits of the animals. Videos were recorded using a PoE (Power over Ethernet) powered HIK DS-2CD2T87G2HLI4 camera (HIKVISION, Hangzhou, China) for high quality imaging and connected to a workstation for continuous recording. The camera resolution was 3840 × 2160 pixels, i.e., 8 MP, and the videos were captured using the FFmpeg software (https://www.ffmpeg.org/, 2007). In addition to setting up the camera, to guarantee a stable connection for data transfer ethernet cables were also installed.

FFmpeg software was used to extract individual frames from the videos. The video data collection took place from 01/03/2024 to 11/03/2024 (dd/mm/year). Firstly, each video was cropped every 2 frames. The selected database consists of 7235 images in total and it has been split into three sub-datasets: training set (70 %), validation set (20 %) and test set (10 %) at a ratio of 16:9 via a Python script. During this



(a)

(b)

Fig. 1. Views of the experimental barn selected as case study. (a) Outdoor aerial view. (b) Indoor view from the central alley.

recording period a total of 19 different cows were observed and manually identified. We manually performed a data cleaning step to delete images where cows were not present or not recognizable. Despite the cleaning step, images that exhibits non ideal but realistic conditions have been intentionally retained in the dataset, such as daylight changes and partial occlusions, in order to have a trained model able to better generalize and perform in a real-world application scenario. Using the online software Roboflow (https://roboflow.com, 2023), the images were manually labelled. Roboflow permits users to produce the necessary YOLO text files for training and evaluation. For each frame, an operator manually drawn a bounding box around each cow with a unique ID basing our evaluation on the morphological aspects of the cow, as shown in Fig. 3. This means that each animal was differentiated from the others based on the different coat texture and characteristics. Table 1 summarizes the main parameters of the adopted dataset.

2.3. Model training

The training process was conducted using YOLOv8 (https://www.ultralytics.com/it/yolo), a state-of-the-art framework optimized for real-time applications able to perform object detection tasks [32]. The YOLOv8 architecture is distinguished by its five size variants, i.e. *n, s, m, l and x*. The model sizes, specifically the number of parameters and computational complexity, are indicated by these letters. YOLOv8n has a very small number of parameters (3.2 million) and 8.7 billion Floating Point Operations per second (FLOP_s) which is significantly lower compared to YOLOv8s (11.2 million parameters; 28.6 FLOP_s), YOLOv8m (25.9 million parameters; 78.9 FLOP_s), YOLOv8l (43.7 million parameters; 165.2 FLOP_s) and YOLOv8x (68.2 million parameters; 257.8 FLOP_s) [33]. This architecture contributes to its high detection speed. Although larger models such as YOLOv8m, YOLOv8l, and YOLOv8x achieve higher $\text{mAP}^{\text{val,}}$ their greater computational cost would both reduce inference speed and raise power consumption, undesirable in commercial barns where computing power are limited. Moreover, YOLOv8 performed better than YOLOv7 and YOLOv5 on the Roboflow 100 benchmarks. Specifically, YOLOv8 achieved a mean average precision (mAP) of 80.2 %, outperforming the mAP of YOLv5 that resulted equal to 73.5 % while using fewer learnable parameters [34]. Therefore, YOLOv8n was chosen as base model because, according to both prior studies [35] and our preliminary tests, it represents an optimal compromise between speed and detection performance for the application scenario investigated here. Fig. 4 shows a simplified architecture of YOLOv8n.

As we can see from the 3D diagram in Fig. 4, it describes the layers an input image goes trough before arriving at the final classification output. To differentiate the different network components the architecture is color-coded:

Yellow: the backbone is in charge of extracting characteristics from the input image. It includes a series of convolutional layers and Cross Stage Partial (CSP) which splits the feature map into two parts [37]. The first part undergoes convolutional processing, and the other is concatenated with the result of the previous part [37]. These layers reduce spatial resolution while improving the CNN's learning capacity [37];

Green: the Spatial Pyramid Pooling – Fast (SPPF) supports extracting the features at numerous scales. This block applies pooling operations and helps the model detect objects of varying sizes;

Blue: Pan Aggregation Network (PAN) and up-sampling layers represent the neck. The neck component of our architecture combines multi-scale feature maps from various network depths to integrate information across different layers [37];

Red: The head performs final bounding box regression, predicts the coordinates of the detected objects, object classification, assigns a class label, and confidence score calculations, gives the confidence that an object is present in the predicted box.

As mentioned above, the present model was fine tuned starting from YOLOv8n variant, pretrained on the COCO dataset [38]. The authors started from the pretrained weight array of YOLOv8n because it was already able to detect a plethora of animals, such as cows. The adoption of this weight array helped to speed up model convergence and improved initial performance, reducing the computational cost, the training time duration and the number of required training data. Additionally, libraries such as "Supervision" and "Roboflow" were installed to train an improved specific model in the work and assist data handling. The dataset was converted to the YOLO (.txt) annotation standard so that it could be compatible with the training framework. The YOLO annotation file for each frame defines the bounding box coordinates for the detected objects in every image. Each row corresponds to a single detected object and consists of five numerical values: *Class ID, X_center, Y_center, Width, and Height*, all of which are normalized between 0 and 1 in relation to the image dimensions. The Class ID represents the assigned category of the object, such as an individual cow ID in this case. The *X_center* and *Y_center* specify the normalized coordinates of the bounding

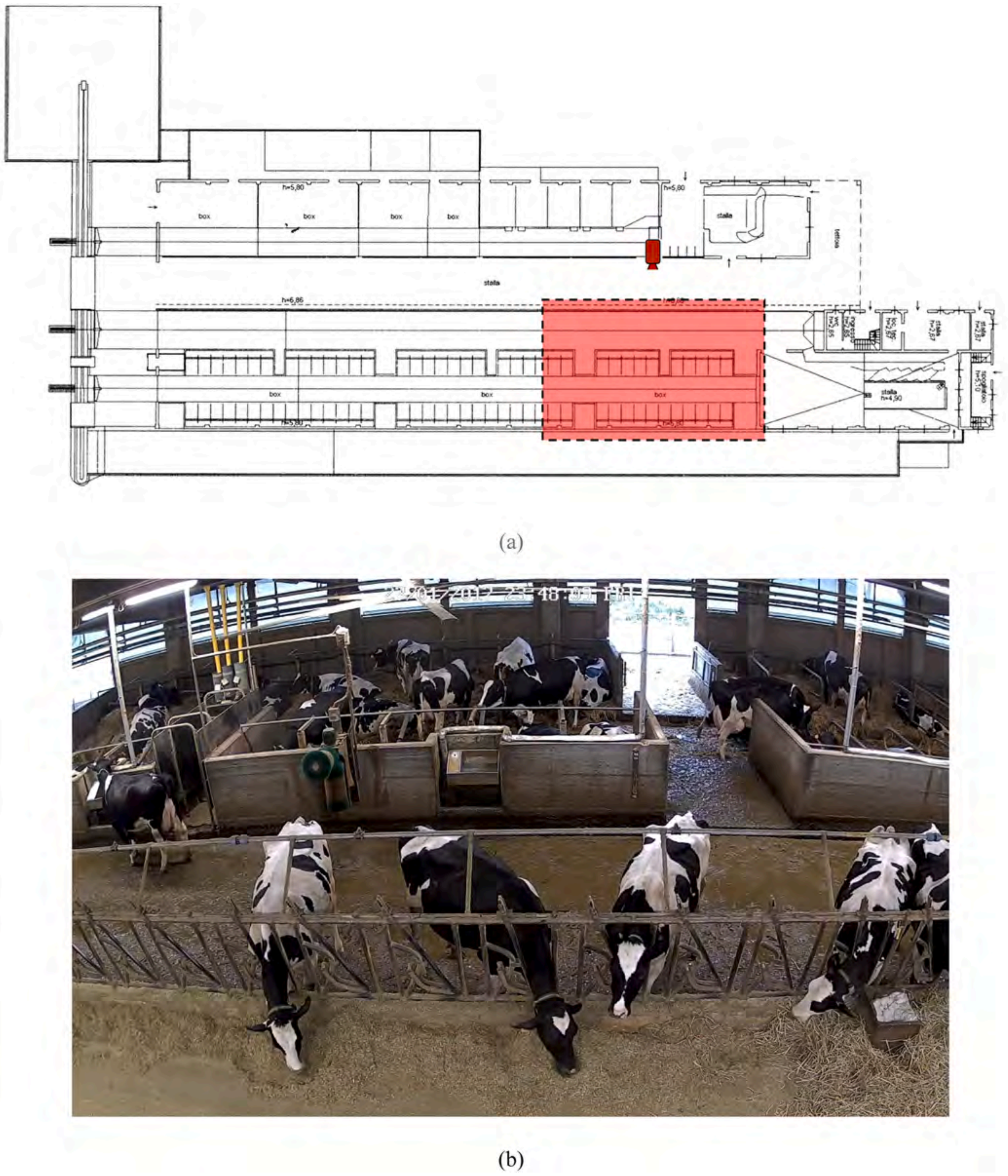
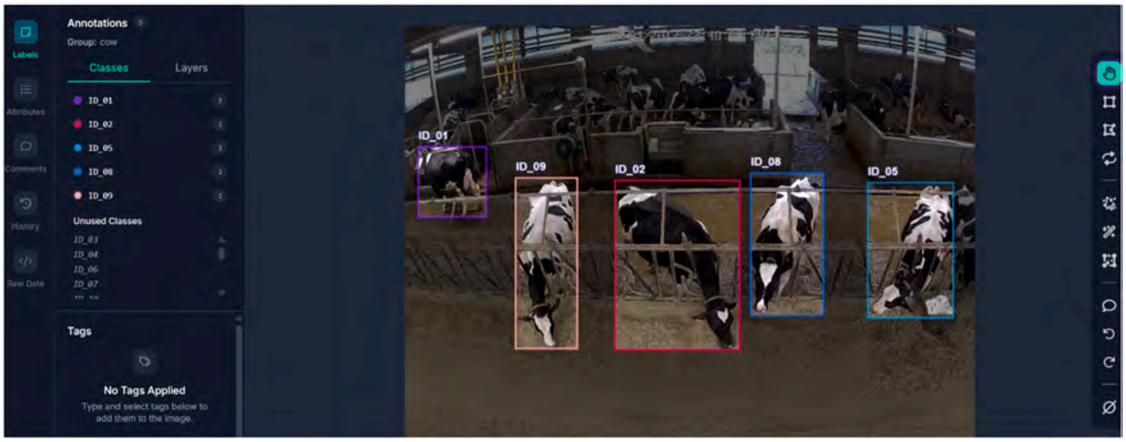


Fig. 2. Views of the experimental setup. (a) Layout of the experimental barn with the indication of the position of the camera and the area framed by the camera (see red coloured area). (b) Example view of the area framed by the camera.

box center, while the Width and Height define the size of the bounding box. Since these annotations give the training algorithm accurate information on object locations, they are essential for our model. The use of normalized values preserves model compatibility at different image resolutions. Then, a configuration file (data.yaml) was created to define essential parameters, including the paths to the training, validation, and test datasets, as well as the number of object classes to be identified. Regarding the core training parameters, the intended task was set at 'detection' (task: detect), and the execution mode was set to 'train' (mode: train). The training dataset images were resized to 640 × 640 pixels and the dataset was split into 70 % training, 20 % validation and 10 % testing to guarantee balanced evaluation. The model was trained over a range of 70 epochs. A patience threshold of 10 periods, automatically ended training if performance failed to improve over 10



No Tags Applied

Type and select tags below to add them to the image.

Fig. 3. View of the Roboflow software during the phase of cow labelling.

Table 1
Main parameters of the adopted dataset.

Item	Parameters
ID of cow categories	19 (ID_1; ID_2; ID_3;….ID_19)
Video Frame Rate	2 fps
Camera Resolution	3840 × 2160 pixels
Train/Val/Test Split	70 % / 20 % / 10 %
Type of Annotation	Bounding box + Classification
Aspect Ratio	16:9
Total number of frames	7235

consecutive periods. Computational efficiency, along with gradient stability, were balanced using a batch size of 16.

The model underwent training with the AdamW optimizer, which used a significantly low initial learning rate of 0.01, a high momentum of 0.937 and a small weight decay of 0.0005 to effectively prevent overfitting, following the default hyperparameter configuration suggested by the Ultralytics framework. In the pre-processing phase, firstly 7235 frames were imported, then each image was auto-oriented based on its metadata tag and subsequently was resized to 640 × 640 pixels using a bilinear interpolation. Then, model training was performed by fine-tuning a YOLOv8n network initialized from the pretrained weights. The other values were assumed equal to the official default settings for YOLOv8.

Different data augmentations techniques were also incorporated to increase dataset size and to increase generalizations. They considered: crop augmentation (0 % minimum zoom, 20 % maximum zoom), rotation augmentation (-15° to +15°), saturation variation (-25 % to +25 %), exposure variation (-10 % to +10 %). The post-processing confidence threshold was set at 0.25, while the Intersection over Union (IoU) threshold was set at 0.5. The training was conducted using GPU

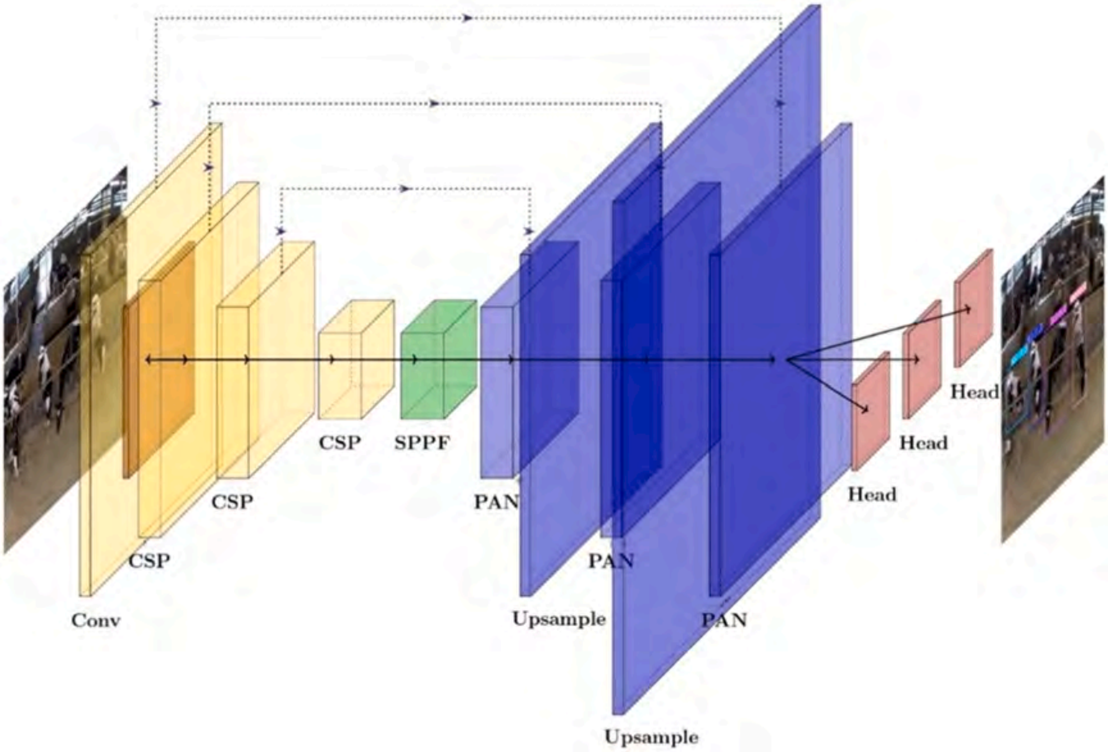


Fig. 4. YOLOv8n simplified network architecture: The model consists of a backbone for feature extraction, a PAN-based neck for feature fusion, and a multi-scale detection head. Image generated using PlotNeuralNet open-source tool [36].

Table 2
Training parameters used to fine-tune the YOLOv8 model.

Parameter	Value
Framework	YOLOv8
Pre-trained weights	YOLO8n (trained on COCO)
Fine-tuning dataset	Custom dairy cow dataset
Annotation format	YOLO (.txt)
Image Size	640 × 640
Batch Size	16
Epochs	70
Patience	10
Learning Rate	0.01
Optimizer	AdamW
Preprocessing	Auto-Orient
Confidence Threshold	0.25
IoU Threshold	0.5
Momentum	0.937
Weight Decay	0.0005
Augmentations	Crop: 0 % Min Zoom, 20 % Max Zoom Rotation: -15° to +15° Saturation: -25 % to + 25 % Exposure: -10 % to +10 %

acceleration (NVIDIA RTX 2080) and the model performance was evaluated using mean Average Precision (mAP@50 and mAP@50-95), Precision (P), Recall (R), and F1 score. The final training setup, including early stopping, was determined by monitoring performance on the validation set. The details of the training parameters used for fine-tuning the YOLOv8 model are collected in Table 2.

In the work, Ultralytics YOLOv8 framework (https://github.com/ultralytics/ultralytics) was used for model training and loss optimization. The used loss function consists of three fundamental components: Box Loss, Class Loss and Distribution Focal Loss (DFL), each of which targets a specific element of the model performance.

2.3.1. *Box loss*

The Complete Intersection over Union (CIoU) loss, which calculates the difference between predicted and ground truth bounding boxes was applied. CIoU integrates overlap area, center point distances, and aspect ratio so improving the accuracy of object localization. The complete definition of the adopted loss function is provided in Eq.s (1) [39]:

$$Loss_{box} = 1 - CIoU(b_p,\ b_t) \tag{1a}$$

$$CIoU(b_p, b_t) = 1 - IoU(b_p, b_t) + \frac{\rho^2(b_p, b_t)}{c^2} + \frac{v^2}{(1 - IoU(b_p, b_t)) + v} \tag{1b}$$

$$v = \frac{4}{\pi^2}\left(arctan\frac{w_{gt}}{h_{gt}} - arctan\frac{w_p}{h_p}\right)^2 \tag{1c}$$

Where:

$\rho(b_p,b_t)$ is the Euclidean distance between the centers of the predicted box;

c is the diagonal length of the smallest box covering both b_p and b_t;

v measures the consistency of the aspect ratios, with w and h width and height of a box.

2.3.2. *Class loss*

In order to test classification accuracy, we used the Cross-Entropy (CE) loss. This loss determines the difference between predicted class probabilities and the actual labels to permits multi-label classification tasks often encountered in object detection. The definition is in Eq. (1):

$$Loss_{cls} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{(i,\ c)}\log\left(p_{i,\ c}\right) \tag{1}$$

where y represents the ground truth label for class c of instance i, and p shows the model predicted probability that instance i belongs to class c.

2.3.3. *DFL loss*

During bounding box refining, DFL was used to improve the precision of bounding box regression by learning the distribution of box coordinates. DFL is effective in improving object localization quality, especially for small or overlapped objects [40]. The total weighted loss is a combination of the previous three, as shown in Eq. (2)

$$Total\ Loss = box \times Loss_{box} + cls \times Loss_{cls} + dfl \times Loss_{dfl} \tag{2}$$

These loss components were proportionally adjusted through hyperparameters defined in the “args.yaml” configuration file. Specifically, box loss, class loss, and DFL weights were set to the Ultralytics framework’s suggested values (box: 7.5; cls: 0.5; dfl: 1.5) to achieve a balanced trade-off among box regression, accuracy and classification.

2.4. *Experimental environment configuration*

All network models used in the experiment are implemented and run in Python to guarantee an equal comparison. The following hardware setup was utilized in this study: the CPU model is Intel Core i9-9700K @ 3.60 GHz x 8, the memory is 32 GB, the GPU model is NVIDIA GeForce RTX 2080 Ti (11 GB), the primary hard disk is 9.1 TB, the operating system is Ubuntu 22.04, the Driver is NVIDIA Driver 535.183.01, the CUDA version is 12.2 and the Python version is 3.11.5. Then we saved

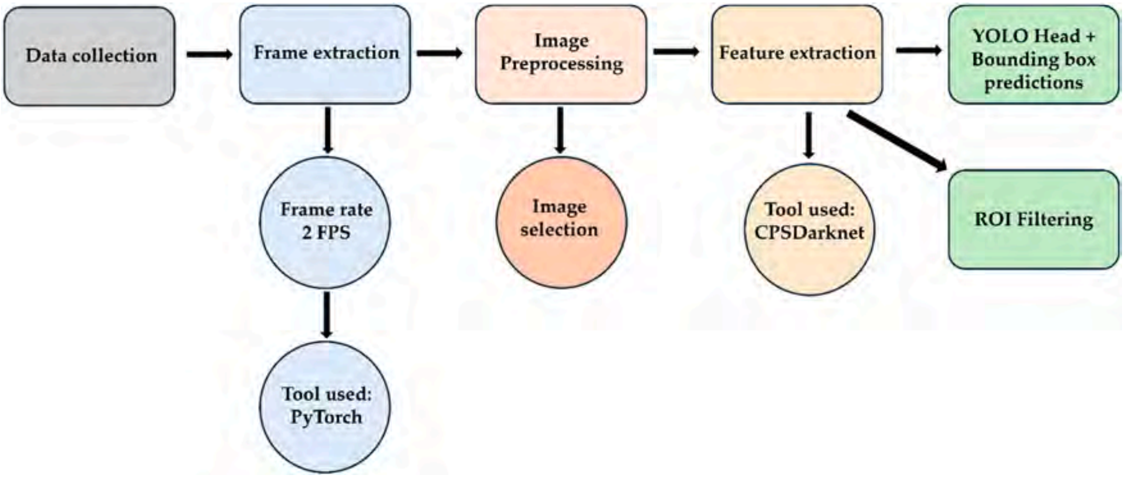


Fig. 5. Pre-processing pipeline for the deep learning model.

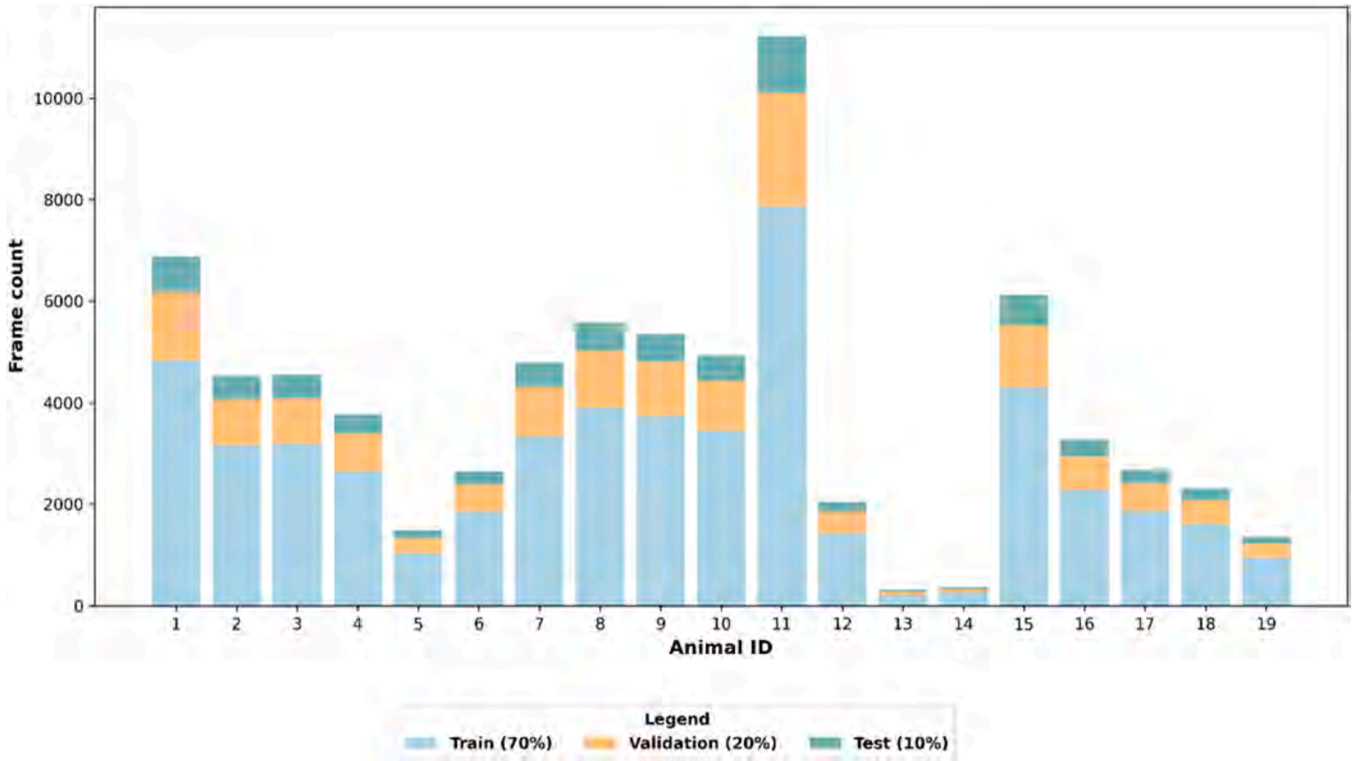


Fig. 6. Identification instance distribution. The bar chart represents the frame count for each animal ID, divided in 70 % for training, 20 % for validation, and 10 % for test.

the model parameters and evaluated the model.

2.5. Pre-processing pipeline

Fig. 5 illustrates the computer vision-based pipeline for detecting and tracking cows in a monitored environment. The pipeline begins with the collection and recording of video footage within a controlled environment of a cattle barn. Following this, frame extraction is carried out via PyTorch [41] sampling at 2 fps to have enough temporal resolution for tracking analysis. Afterward, pre-processed extracted frames undergo selection in such a way that images are discarded if found to be blurry or irrelevant. At this stage, process images are fed into CPSDarknet [42] to get relevant features for object detection, the backbone of the model. Our model based on YOLO is then applied to identify and classify every cow using unique tracking IDs. The system then calculates the ROI time to determine how much time every identified cow individually spent in a specific area, where in the present study it is the feeding lane. In our study, we considered a cow in the feeding area if more than 50 % of its bounding box intersects with the ROI polygon drawn around the feed barrier. This counter was used as indicator of time spend at the feeding lane. To verify the robustness of the analysis, we also measured how long each cow stay within the ROI. A structured pipeline allows the data to be processed effectively and thus proves to be a very useful tool for the automation of livestock monitoring or specific precision farming applications. All stages of the pipeline were implemented in Python.

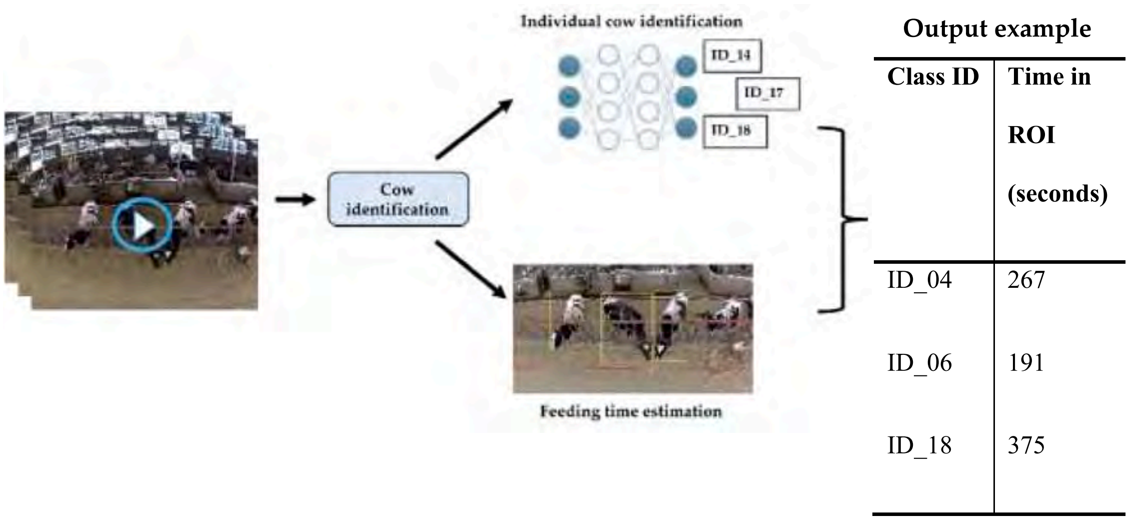
2.6. Identification process

To identify individual cows with a unique ID, a deep learning-based classification system using YOLOv8 in combination with Roboflow was developed. This tool was used for labelling the data, organizing the dataset, and exporting it for the training of YOLO. Manual annotations were performed by multiple operators and a standardized protocol was followed. A total of 19 cows were identified, with their classification instances distributed across the dataset. Using the dataset described in previous section, we use label instances, where the distribution is shown in Fig. 6, amongst the training, validation, and test sets.

The frame count per cow of the considered dataset differed in a considerable way and some animals had significantly more labelled instances than others. For example, ID_01 had 4815 frames for training, 1376 for validation, and 688 for testing, while ID_12 had only 322 frames for training, 92 for validation, and 46 for test, indicating potential class imbalance. Regardless of these differences, the model was trained and fine-tuned to generalize effectively across different animal appearances and perspectives. The balanced dataset split also allowed for more robust model evaluation, preventing overfitting and improving generalization for unseen instances. After training, the model was tested in the barn environment, processing real-time video streams for cow identification. Non-maximum suppression (NMS) post-processing steps was used to filter redundant detection [43]. In this way was possible to obtain a YOLO annotation file for each frame, which defines the bounding box coordinates for the detected objects in every image. As already mentioned in section 2.3, in the present study, Roboflow was used to produce these annotations during the dataset preparation process. The structured format of this file makes it possible for YOLOv8's training pipeline to decode it and process it efficiently, which helps the model learn and increase object recognition accuracy over time.

2.7. Feeding behaviour recognition

To monitor the feeding behaviour of cows, a ROI-based approach was applied to track the time each cow spent in a particular region of the feeding area. The ROI coordinates were manually annotated on an original image of 2560 × 1440 pixels and then rescaled to 640 × 640 pixels to match the input resolution of the YOLOv8 detection model. It was drawn manually based on the visible physical boundary of the feed barrier. Since the feeding area layout is fixed and consistent across all recordings, we validated the correctness of the ROI visually by



Class ID	Time in ROI (seconds)
ID_04	267
ID_06	191
ID_18	375

Fig. 7. Algorithm for individual cow identification and time spent in ROI.

overlaying it on multiple frames. Although the approach is straightforward, this method is practical and reproducible. Using bounding box data from model predictions, each detection was transformed from normalized YOLO format (X_center, Y_center, Width, Height) into absolute bounding box coordinates (x1, y1, x2, y2). A ground truth .csv file with cow detection was used for the analysis and the total time spent in the ROI for each individual cow was calculated by counting the number of frames in which its bounding box overlapped with the ROI. In our study, the 50 % threshold of overlaps between bounding box and ROI polygon was assumed as indicator of time at the feeding lane. Finally, the time calculation was saved as a .csv file for further analysis. Fig. 7 shows a graphic of the algorithm we used for individual cow identification and time spent in ROI.

3. Results and discussion

3.1. Evaluation of model performance

A 30-minute video taken five days after training was compared to a manually labelled ground truth dataset in order to assess the YOLOv8-base cow recognition model in a real-world test scenario. The intersection-over-union (IoU) coefficient, commonly known as the Jaccard similarity index [44] is the first metric that is to be introduced. This parameter describes the characteristics of the objects evaluated and then determines a standardized measure, as outlined in (3) representing the proportion of the intersection divided by the union of the two bounding boxes, i.e., the predicted and the ground truth bounding boxes [22].

$$IoU = \frac{Area\ of\ overlap}{Area\ of\ union} \tag{3}$$

Moreover, accuracy (A) measures the total percentage of correct classifications, precision (P) measures the percentage of true positive detections in the total number of retrieved instances in pattern recognition tasks. Meanwhile, recall (R) measures the percentage of true positive elements correctly detected out of all the instances. As a result, both precision and recall depend on identifying and quantifying applicability. The definitions of accuracy, precision and recall are in Eqs. (4), (5), (6).

$$A = \frac{TP + TN}{TP + TN + FN + FP} \tag{4}$$

$$P = \frac{TP}{TP + FP} \tag{5}$$

Table 3
Metrics of the overall model performance.

Precision	Recall	F_1-score	TP	FP	FN	IoU Threshold
0.85	0.62	0.72	968	167	569	0.85

$$R = \frac{TP}{TP + FN} \tag{6}$$

In this context, TP (true positive) signifies the number of times the detector correctly identifies a specific class withing an image, with an IoU greater than a predefined threshold. In contrast, FP (false positive) is the number of occasions in which the model incorrectly detects other objects as belonging to the target category or when the IoU falls below the specified threshold [22]. Meanwhile, FN (false negative) represents the number of times the model fails to identify the presence of the target class in an image.

The balanced *F1* score is a measure that represents the harmonic mean and combines *P* and *R* in the evaluation of classifications results [45]. When *P* and *R* have comparable values, this metric is mathematically defined as the square of the geometric mean of *P* and *R* and it's normalized by their arithmetic mean. The *F1*-score ranges from 0 to 1, where 0 means complete misclassification while 1 means perfect classification for each frame [22]. The definition of *F1* is provided in Eq. (7).

$$F1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{7}$$

As far as the above said metrics, in the present study the following values have been obtained: Precision equal to 0.85, Recall equal to 0.62, and F1 score equal to 0.72. These results indicate that the model predicts cows with high confidence but occasionally fails to predict individually which causes a false negative count of 569. Table 3 shows the overall model performance.

Unlike from previous YOLO-based cattle monitoring studies (see for instance [25] and [2]) the proposed fine-tuned YOLOv8n model deliver a 85 % precision, 62 % recall and a F1 score of 0.72 while running at 12 ms per 640 × 640 image input size. For example, [25] used YOLOv3 to detect group housed heifers with 96 % accuracy at 20FPS and [2] used YOLOv4 to monitor feeding behaviour at 31 ms per 416 × 416 image input size. Thanks to its anchor-free head and lightweight C2f backbone, YOLOv8n have comparable performance but runs faster and this is useful for continuous monitoring tasks.

Going into more detailed in respect to what was said before, in this context TP (equal to 968) correspond to the instances where model has

Table 4
Frame count, accuracy, and precision for each class ID in the test set.

Class ID	Frame count	Accuracy (%)	Precision (%)
1	6878	82	85
2	4515	74	86
4	3775	68	84
6	2651	64	83
8	2612	70	82
13	318	53	87
16	3267	67	86
17	2881	60	88
18	2307	65	85

detected a cow that was present in the ground truth, meanwhile FP equal to 167 counts, correspond to occasions where the model detected a cow that was not present in the ground truth. Conversely, FN (equal to 569) are those cases where a cow was present in the ground truth but not detected by the model, which means the model missed an identification. When analysing by class, some of the cows, e.g., ID_1, ID_2, and ID_8, had higher accuracy in the range 74 %-82 %, while others, e.g., ID_13, ID_17, had lower performance (either because of class imbalance, occlusions, or because it's affected by movement artifacts). Even though these variations, precision remained consistently high in all classes, in the range 82 %-88 %, meaning that the model is reliable in correctly identifying if the object is detected.

As reported in Table 4, in the test set there are fewer classes and for each we calculated the accuracy and precision. The frame count refers to the classes seen in the train set.

3.2. Learning rates and parameters of the model

A bootstrap resampling method considering 1000 iterations was used to assess performance uncertainty from calculating the 95 % confidence interval for *P* and *R*, in order to prove that model maintains consistency across multiple samples during training. Additionally, the training and validation loss curves were analysed over the 70 epochs, tracking bounding box regression (box loss), classification (cls loss), and DFL loss. Training loss gradually decreased while validation loss stabilized, suggesting that the model successfully learned significant features without

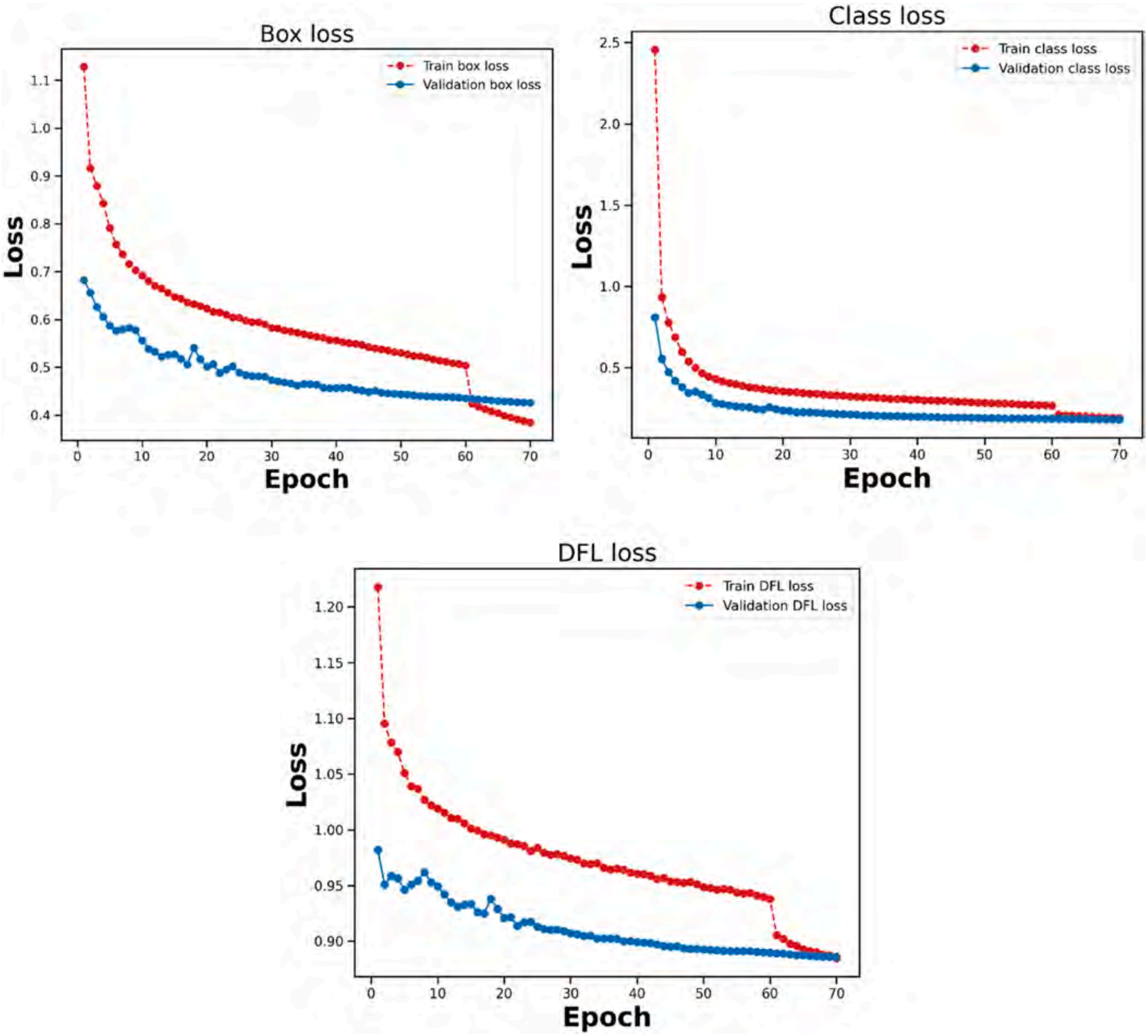


Fig. 8. Main learning rates for box loss, class loss and DFL loss for the YOLOv8 model.

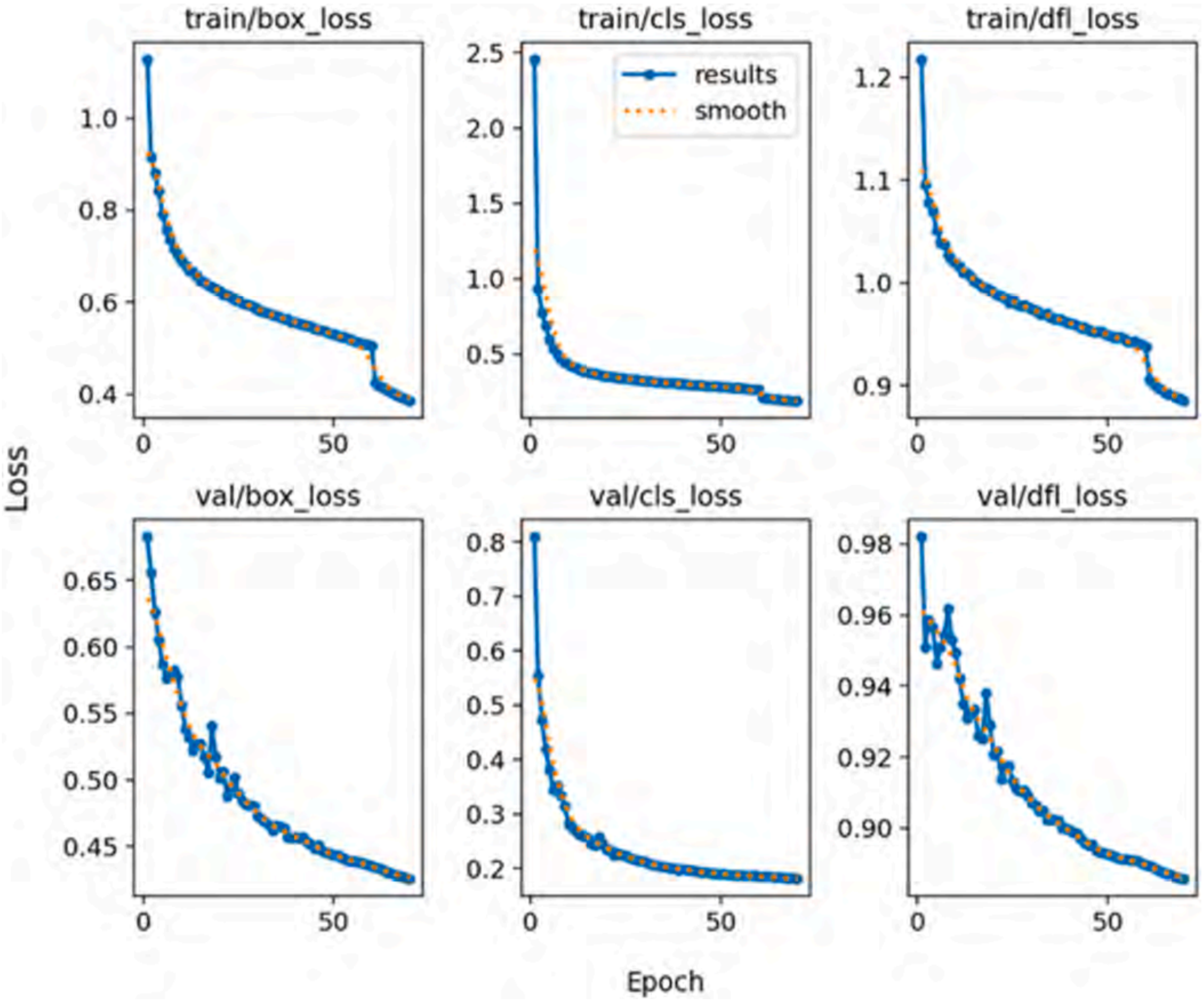


Fig. 9. Box loss, class loss and DFL loss curves resulting from training and validating the YOLOv8 network over 70 epochs.

severe overfitting. However, as the final epochs approach, minor differences between the training and validation loss curves point to the possibility of further fine-tuning using regularization strategies such as dropout or weight decay adjustments. It is worth noticing that values of validation loss are typically lower than values of training loss. Our hypothesis is that our dataset includes consecutive video images, which allows for validation frames to be easier to predict than training frames. Since consecutive images share similar visual features, the model may have learnt significant patterns from nearby training frames, reducing the complexity of predictions during validation.

Table 5
Average confidence per class on the test set.

Class ID	Frame count	Average confidence
1	661	0.74
2	19	0.81
4	2529	0.86
6	199	0.67
8	696	0.73
13	222	0.85
16	2862	0.82
17	197	0.77
18	1	0.54

Box loss graph in Fig. 8 assesses the discrepancy between the bounding boxes predicted by the model and those actually present for the objects in each image. Correct definition of bounding boxes is essential for object detection and to minimize this loss, the model must be able to predict boxes that nearly surround the target objects.

Another important information for YOLOv8 is the class loss parameter, which assess how well the model is able to classify the objects which are present in each frame. If this loss is low, it means the model has learned to distinguish classes of different objects properly.

Finally, DFL loss is a form of focal loss function that aims at solving the problem of class imbalance in object detection datasets. This imbalance can make the model overfit to the classes more frequently trained. Fig. 9 shows the train and validation loss curves of YOLOv8 parameters. The comparison between loss curves for training and validation over the 70 epochs demonstrate that the model for cow detection performed with success the classification task.

To evaluate the reliability of the YOLOv8-based cow identification model, the average confidence score per class was calculated on the test set, using predictions from a 30-minute test video that was recorded five days after training. Confidence scores indicate the model’s certainty in making predictions for each detected cow. Table 5 summarizes the average confidence per class and the corresponding frame count. This highlights variations in prediction certainty between different cow IDs.

Table 6
Metrics of the overall model performance applied to a few-unbalanced dataset.

Precision	Recall	mAP50	mAP50-95	F_1-score
0.98	0.92	0.93	0.86	0.94

Table 7
Frame count, accuracy, and precision for each class ID in the test set of the few-unbalanced dataset.

Class ID	Frame count	Precision (%)	Recall (%)
1	6878	82	91
2	4515	88	91
4	3775	95	97
6	2651	98	96
8	2612	98	97
16	3267	95	97
17	2881	98	97
18	2307	98	91

Table 8
Average confidence per class on the test set of the few-unbalanced dataset.

Class ID	Frame count	Average confidence
1	661	0.95
2	19	0.92
4	2529	0.94
6	199	0.94
8	696	0.94
16	2862	0.92
17	197	0.94
18	1	0.54

Class ID 4 (0.86), Class ID 13 (0.85), and Class ID 16 (0.82) had the highest confidence scores, suggesting that the model was highly confident in recognizing these cows, most likely as a result of their distinctive visual features and sufficient number of training data. In contrast, Class ID 18 had the lowest confidence (0.54), indicating possible issues such as motion blur, occlusions, or underrepresentation in the training set, or probably due to the fact that in the test set this class was seen only in one frame. The results in the table show that the average confidence level for most classes were relatively high, with only few exceptions. Overall, the confidence analysis suggests that the YOLOv8 model regularly assigns high confidence scores to correctly detected cows, showing its reliability for automated cattle identification. Low-confidence predictions in some classes, however, point to possible areas for development, such as better data augmentation, increased training samples for underrepresented classes, and improved post-processing criteria to filter low-confidence detections.

3.3. Evaluation of model performance applied to a few -unbalanced dataset

In order to test the response of the model to application with few-unbalanced dataset, the model has been trained also on a reduced version of the original dataset so avoiding problem of unbalanced identification of the majority classes. So, in this sub-section, the classes # 5, 12, 13, 14 and 19 have been not considered in the training, validation and testing of the model so considering a more balanced dataset. The results of this analysis are reported in Table 6, Table 7 and Table 8. As the comparison with the previous results shows, the adoption of a more balanced dataset improves the global performance of the model and the average confidence results higher than 90 % for all the classes except class 18 present in just one frame of the test dataset. It is worth noticing that the availability of a balanced dataset among the different classes is a condition that in general is not difficult to obtain also in commercial barn, if the data collection period is prolonged for a sufficient time span. So, if the data collection period is adequate and each cow on the herd is sufficiently framed, the application of this model in commercial barn could provide a reliable and effective tool for daily herd management.

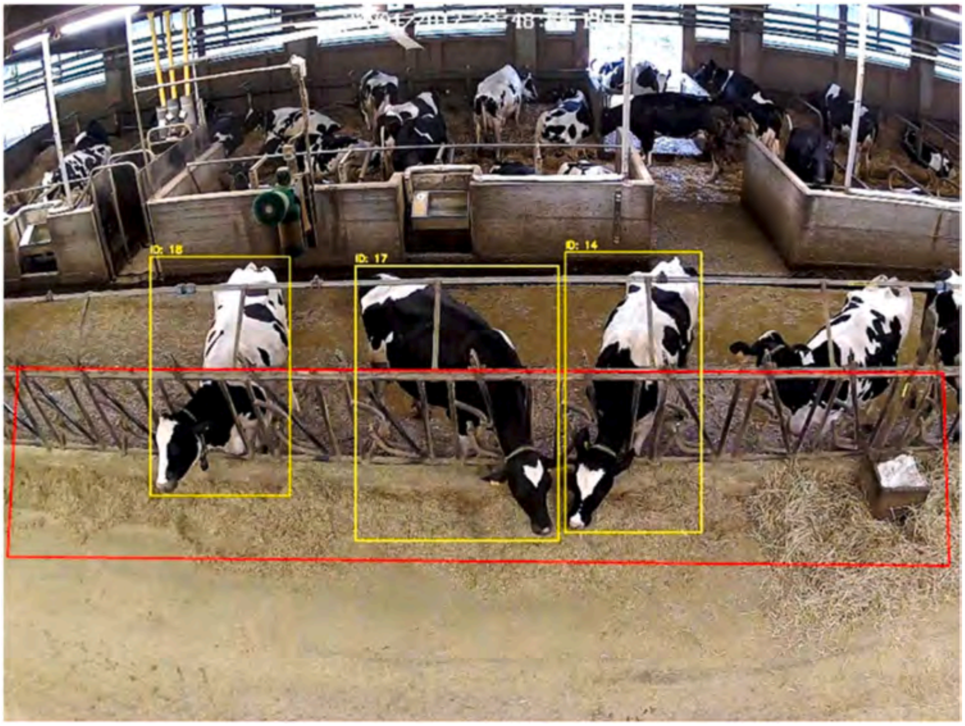


Fig. 10. Example of predicted bounding boxes (yellow rectangles) with ID class (yellow numbers) assessed for each bounding box and with overlapped the region of interest (red rectangle).

3.4. Behaviour analysis

The analysis of cow feeding behaviour was examined by calculating the time spent within the ROI that corresponded to the feeding area (see Fig. 10). In this work, the ROI was manually drawn based on visual alignment. While no geometric calibration was applied, the cows and the ROI share the same image space and perspective distortion, securing that their relative spatial relationships remain consistent. Using Python-based analysis, the bounding box coordinates of detected cows were compared to a predefined ROI polygon in order to identify whether an individual cow was present at the feeding area. The results were saved in an output file including a detailed summary of the time each cow spent in the ROI during the test video recorded five days after training. To validate the results, they were compared to the ground truth annotation file obtained by manually labelling of the same frames. The comparison allowed to evaluate the model's accuracy in identifying feeding behaviour and confirmed a high correlation between the automated detection and the ground truth data (average accuracy of 0.88). However, minor differences were observed, probably as a result of occlusions, missed detections, or changes in cow posture that affected the location of the bounding box within the ROI. It is essential to note that even if the head of a cow is inside the feeding trough, feeding behaviour may not occur and the further challenge will be identifying the feeding period, i.e., the time the cows spend eating, that represent a further important information for the herd management and for the monitoring of the single cow. The promising results obtained in this study contribute to the advancement and validation of computer vision applications in herd monitoring, supporting the commercial adoption of these technologies for analysing cow behaviour with the main objectives to increase animal welfare and the sustainability of the animal production chain.

4. Conclusions

This study investigated the use of computer vision in dairy farming, developing a deep learning model based on YOLOv8 to identify cows with different IDs and calculate the time spent at the feeding area. Using manual annotations and coat pattern recognition, the model was trained to achieve correct identification. The results show that the model reached accuracy between 53 % to 88 %. These outcomes indicated that, while the model is generally performing well, certain individual classes achieve lower accuracy, which may be caused by the pattern complexity, lighting conditions, or occlusions in the dataset. The highest accuracy (82 %) and precision of (85 %) were achieved for the most frequent class, demonstrating that performance improves with more training data. Despite these results, some challenges persist, such as the necessity for larger labelled datasets and the influence of environmental conditions (e.g., illumination variations or occlusions). While we adopted the standard YOLOv8 architecture, our innovation lies in adapting it to a specific and challenging context: individual cow recognition in farm environments. This was achieved through a custom dataset and a training pipeline adapted to our task. Rather than proposing architectural changes, our research focuses on evaluating YOLOv8's performance in settings underexplored in literature. We selected YOLOv8n for its balance of computational efficiency and ease of deployment, particularly ideal in scenarios commonly found in farm settings. Although we did not compare it with other models, our goals was to establish an initial benchmark using a lightweight model. In future work, we will expand the study to include such comparisons to better understand the trade-offs between speed and accuracy. While these findings highlight the potential of computer vision in precision livestock farming, future research might strengthen the model by including other data sources, such as RFID, to improve identification accuracy. Moreover, integrating this technology with real-time AI processing at the edge might allow for on-farm automation, reducing manual labour and improving overall efficiency. Dairy farms may move towards a data-driven and sustainable future by implementing computer vision to optimise resources and animal care. This could represent a step forward smarter farming, with advanced AI-based monitoring system contributing to increased productivity and better livestock health, and animal welfare.

Ethical statement

The authors declare that the research involved animals, but all procedures were carried out in accordance with relevant regulations and with respect for animal welfare

CRediT authorship contribution statement

Claudia Giannone: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Mohsen Sahraeibelverdy:** Writing – original draft, Software, Methodology, Investigation, Data curation. **Martina Lamanna:** Writing – original draft, Methodology, Data curation. **Damiano Cavallini:** Writing – review & editing, Supervision, Data curation. **Andrea Formigoni:** Writing – review & editing, Visualization, Supervision, Data curation. **Patrizia Tassinari:** Writing – review & editing, Supervision, Resources, Methodology, Formal analysis, Data curation. **Daniele Torreggiani:** Writing – review & editing, Visualization, Supervision, Resources, Formal analysis. **Marco Bovo:** Writing – review & editing, Supervision, Software, Resources, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] B.L. Nielsen, R.F. Veerkamp, A.B. Lawrence, Effects of Genotype, Feed Type and Lactational Stage on the Time Budget of Dairy Cows, Acta Agricult. Scandinavica, Section A - Animal Sci. 50 (4) (2000) 272–278, https://doi.org/10.1080/090647000750069467. Dec.

[2] Z. Yu, et al., Automatic Detection Method of Dairy Cow Feeding Behaviour Based on YOLO Improved Model and Edge Computing, Sensors 22 (9) (2022) 3271, https://doi.org/10.3390/s22093271. Apr.

[3] D. Cavallini, L.M.E. Mammi, M. Fustini, A. Palmonari, A.J. Heinrichs, A. Formigoni, Effects of ad libitum or restricted access to total mixed ration with supplemental long hay on production, intake, and rumination, J. Dairy Sci. 101 (12) (2018) 10922–10928, https://doi.org/10.3168/jds.2018-14770. Dec.

[4] A.J. Heinrichs, B.S. Heinrichs, D. Cavallini, M. Fustini, A. Formigoni, Limiting total mixed ration availability alters eating and rumination patterns of lactating dairy cows, JDS Commun. 2 (4) (2021) 186–190, https://doi.org/10.3168/jdsc.2020-0074. Jul.

[5] S. Aditya, E. Humer, P. Pourazad, R. Khiaosa-Ard, J. Huber, Q. Zebeli, Intramammary infusion of Escherichia coli lipopolysaccharide negatively affects feed intake, chewing, and clinical variables, but some effects are stronger in cows experiencing subacute rumen acidosis, J. Dairy Sci. 100 (2) (2017) 1363–1377, https://doi.org/10.3168/jds.2016-11796. Feb.

[6] G.M. Pereira, K.T. Sharpe, B.J. Heins, Evaluation of the RumiWatch system as a benchmark to monitor feeding and locomotion behaviors of grazing dairy cows, J. Dairy. Sci. 104 (3) (2021) 3736–3750, https://doi.org/10.3168/jds.2020-18952. Mar.

[7] D. Cavallini, et al., Effect of an immunomodulatory feed additive in mitigating the stress responses in lactating dairy cows to a high concentrate diet challenge, Animals 12 (16) (2022) 2129, https://doi.org/10.3390/ani12162129. Aug.

[8] M. Fadul, et al., Assessment of feeding, ruminating and locomotion behaviors in dairy cows around calving – a retrospective clinical study to early detect spontaneous disease appearance', PLoS. One 17 (3) (2022) e0264834 https://doi.org/10.1371/journal.pone.0264834. Mar.

[9] V.M. Thorup, et al., Lameness affects cow feeding but not rumination behavior as characterized from sensor data, Front Vet. Sci. 3 (2016), https://doi.org/10.3389/fvets.2016.00037. May.

[10] L.A. González, B.J. Tolkamp, M.P. Coffey, A. Ferret, I. Kyriazakis, Changes in feeding behavior as possible indicators for the automatic monitoring of health disorders in dairy cows, J. Dairy. Sci. 91 (3) (2008) 1017–1028, https://doi.org/10.3168/jds.2007-0530. Mar.

[11] G. Mattachini, E. Riva, F. Perazzolo, E. Naldi, G. Provolo, Monitoring feeding behaviour of dairy cows using accelerometers, J. Agric. Eng. 47 (2016) 54, https://doi.org/10.4081/jae.2016.498. Mar.

[12] S. Magro, A. Costa, D. Cavallini, E. Chiarin, M. De Marchi, Phenotypic variation of dairy cows' hematic metabolites and feasibility of non-invasive monitoring of the metabolic status in the transition period, Front. Vet. Sci. 11 (2024), https://doi.org/10.3389/fvets.2024.1437352. Nov.

[13] K. Džermeikaitė, D. Bačėninaitė, R. Antanaitis, Innovations in cattle farming: application of innovative technologies and sensors in the diagnosis of diseases, Animals 13 (5) (2023) 5, https://doi.org/10.3390/ani13050780. Art. noJan.

[14] B. Pichlbauer, J.M. Chapa Gonzalez, M. Bobal, C. Guse, M. Iwersen, M. Drillich, Evaluation of different sensor systems for classifying the behavior of dairy cows on pasture, Sensors 24 (23) (2024) 23, https://doi.org/10.3390/s24237739. Art. noJan.

[15] A.H. Stygar, et al., A systematic review on commercially available and validated sensor technologies for welfare assessment of dairy cattle, Front Vet. Sci. 8 (2021), https://doi.org/10.3389/fvets.2021.634338. Mar.

[16] M. Lamanna, M. Bovo, D. Cavallini, Wearable collar technologies for dairy cows: A systematized review of the current applications and future innovations in precision livestock farming, Animals 15 (3) (2025) 458, https://doi.org/10.3390/ani15030458. Feb.

[17] J.O. Chelotti, et al., Livestock feeding behaviour: A review on automated systems for ruminant monitoring, Biosyst. Eng. 246 (2024) 150–177, https://doi.org/10.1016/j.biosystemseng.2024.08.003. Oct.

[18] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection', arXiv: arXiv:1506.02640, 2016. May 09Accessed: Apr. 28, 2024. [Online]. Available: http://arxiv.org/abs/1506.02640.

[19] M.E. Hossain, M.A. Kabir, L. Zheng, D.L. Swain, S. McGrath, J. Medway, A systematic review of machine learning techniques for cattle identification: datasets, methods and future directions, Artif. Intell. Agric. 6 (2022) 138–155, https://doi.org/10.1016/j.aiia.2022.09.002.

[20] M. Jiang, Y. Rao, J. Zhang, Y. Shen, Automatic behavior recognition of group-housed goats using deep learning, Comput. Electron. Agric. 177 (2020), https://doi.org/10.1016/j.compag.2020.105706. Aug.

[21] J. Terven, D.-M. Córdova-Esparza, J.-A. Romero-González, A comprehensive review of YOLO architectures in computer Vision: from YOLOv1 to YOLOv8 and YOLO-NAS, MAKE 5 (4) (2023) 1680–1716, https://doi.org/10.3390/make5040083. Nov.

[22] P. Tassinari, et al., A computer vision approach based on deep learning for the detection of dairy cows in free stall barn, Comput. Electron. Agric. 182 (2021), https://doi.org/10.1016/j.compag.2021.106030. Mar.

[23] N. Bergman, Y. Yitzhaky, I. Halachmi, Biometric identification of dairy cows via real-time facial recognition, Animal 18 (3) (2024) 101079, https://doi.org/10.1016/j.animal.2024.101079. Mar.

[24] W. Andrew, C. Greatwood, and T. Burghardt, *Aerial animal biometrics: individual friesian cattle recovery and visual identification via an autonomous UAV with onboard deep inference*. 2019. doi: 10.48550/arXiv.1907.05310.

[25] T. Bresolin, R. Ferreira, F. Reyes, J. Van Os, J.R.R. Dórea, Assessing optimal frequency for image acquisition in computer vision systems developed to monitor feeding behavior of group-housed Holstein heifers, J. Dairy. Sci. 106 (1) (2023) 664–675, https://doi.org/10.3168/jds.2022-22138. Jan.

[26] Y. Qiao, C. Clark, S. Lomax, H. Kong, D. Su, S. Sukkarieh, Automated individual cattle identification using video data: A unified deep learning architecture approach, Front Anim. Sci. 2 (2021), https://doi.org/10.3389/fanim.2021.759147. Nov.

[27] R. Wang, et al., An ultra-lightweight method for individual identification of cow-back pattern images in an open image set, Expert. Syst. Appl. 249 (2024) 123529, https://doi.org/10.1016/j.eswa.2024.123529. Sep.

[28] L. Fu, et al., Lightweight individual cow identification based on Ghost combined with attention mechanism, PLoS. One 17 (10) (2022) e0275435, https://doi.org/10.1371/journal.pone.0275435. Oct.

[29] C. Clayton, A. Victory, K. Graves, A. Griffin, C. Hudson, M. McGee, 420 Automated identification and behavioral estrus detection in freestall-housed dairy cows using computer vision and machine learning: A two-layered algorithm approach, J. Anim. Sci. 102 (Supplement_3) (2024) 4–5, https://doi.org/10.1093/jas/skae234.005. Sep.

[30] E. Ferlizza, S. Fasoli, D. Cavallini, M. Bolcato, G. Andreani, and G. Isani, 'Preliminary study on urine chemistry and protein profile in cows and heifers', 2020, doi: 10.29261/pakvetj/2020.067.

[31] L.M.E. Mammi, et al., Calving difficulty influences rumination time and inflammatory profile in Holstein dairy cows, J. Dairy. Sci. 104 (1) (2021) 750–761, https://doi.org/10.3168/jds.2020-18867. Jan.

[32] T. Diwan, G. Anirudh, J.V. Tembhurne, Object detection using YOLO: challenges, architectural successors, datasets and applications, Multimed. Tools. Appl. 82 (6) (2023) 9243–9275, https://doi.org/10.1007/s11042-022-13644-y. Mar.

[33] Ultralytics, YOLOv8, Accessed: Aug. 08, https://docs.ultralytics.com/models/yolov8, 2025.

[34] I.J. Jacob, S. Piramuthu, P. Falkowski-Gilski (Eds.), *Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2023*. in Algorithms for Intelligent Systems, Singapore: Springer Nature Singapore, 2024, https://doi.org/10.1007/978-981-99-7962-2.

[35] K. Jeong, et al., A monitoring system for Cattle behavior detection using YOLO-v8 in IoT environments, in: 2024 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, IEEE, 2024, pp. 1–4, https://doi.org/10.1109/icce59016.2024.10444145. Jan.

[36] H. Iqbal, HarisIqbal88/PlotNeuralNet, TeX, 2025. Aug. 08Accessed: Aug. 08, 2025. [Online]. Available: https://github.com/HarisIqbal88/PlotNeuralNet.

[37] R.-Y. Ju, W. Cai, Fracture detection in pediatric wrist trauma X-ray images using YOLOv8 algorithm, Sci. Rep. 13 (1) (2023) 20077, https://doi.org/10.1038/s41598-023-47460-7. Nov.

[38] in *Computer Vision – ECCV 2014*, vol. 8693 T.-Y. Lin, et al., Microsoft COCO: Common objects in context, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), in Lecture Notes in Computer Science, in Lecture Notes in Computer Science, 8693, Springer International Publishing, Cham, 2014, pp. 740–755, https://doi.org/10.1007/978-3-319-10602-1_48.

[39] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression', arXiv: arXiv:1911.08287, 2019 https://doi.org/10.48550/arXiv.1911.08287. Nov. 19.

[40] X. Li, et al., Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection, arXiv: arXiv:2006.04388, 2020, https://doi.org/10.48550/arXiv.2006.04388. Jun. 08.

[41] A. Paszke, et al., PyTorch: An Imperative Style, High-Performance Deep Learning Library', arXiv: arXiv:1912.01703, 2019 https://doi.org/10.48550/arXiv.1912.01703. Dec. 03.

[42] C.-Y. Wang, H.-Y.Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, CSPNet: A new backbone that can enhance learning capability of CNN, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, IEEE, 2020, pp. 1571–1580, https://doi.org/10.1109/CVPRW50498.2020.00203. Jun.

[43] G. Wu, Y. Li, Non-maximum suppression for object detection based on the chaotic whale optimization algorithm, J. Vis. Commun. Image Represent. 74 (2021) 102985, https://doi.org/10.1016/j.jvcir.2020.102985. Jan.

[44] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, 'Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression', Apr. 15, 2019, arXiv: arXiv:1902.09630. doi: 10.48550/arXiv.1902.09630.

[45] X. Nie, M. Yang, and W. Liu, *Deep neural network-based robust ship detection under different weather conditions*. 2019. doi: 10.1109/ITSC.2019.8917475.