frontiers in Signal Processing

TYPE Original Research
PUBLISHED 14 April 2026
DOI 10.3389/frsip.2026.1778118

OPEN ACCESS

EDITED BY
Farook Sattar,
University of Victoria, Canada

REVIEWED BY
André Moreira Souza,
University of São Paulo, Brazil
Usha Divakarla,
NMAM Institute of Technology, India

*CORRESPONDENCE
Francesco Beritelli,
francesco.beritelli@unict.it

RECEIVED 30 December 2025
REVISED 25 February 2026
ACCEPTED 20 March 2026
PUBLISHED 14 April 2026

CITATION
Avanzato R, Beritelli L, Bognanno S, Beritelli F and Avondo M (2026) Automatic monitoring herbage prehensions in grazing cows using audio signals and deep learning techniques.
Front. Signal Process. 6:1778118.
doi: 10.3389/frsip.2026.1778118

Automatic monitoring herbage prehensions in grazing cows using audio signals and deep learning techniques

Roberta Avanzato[1], Ludovica Beritelli[2], Salvatore Bognanno[3], Francesco Beritelli[1]* and Marcella Avondo[3]

[1]Department of Electrical, Electronic and Computer Engineering, University of Catania, Catania, Italy, [2]Department of Mathematics and Computer Science, University of Catania, Catania, Italy, [3]Department of Agricoltura, Alimentazione e Ambiente, University of Catania, Catania, Italy

Background: Accurate monitoring of feeding behavior in grazing ruminants, particularly the detection of prehension events, is a central challenge for Precision Livestock Farming (PLF). Traditional methods, such as accelerometers, show limitations in the reliable identification of individual events. Acoustic analysis based on deep learning is emerging as a non-invasive and promising alternative.
Methods: This study presents two main contributions: (i) a web-based software platform (built on React.js and TensorFlow.js) for the annotation, visualization, and in-browser inference of audio signals; (ii) a comparative analysis of several 2D-CNN architectures (DenseNet-121, ResNet-101, EfficientNet-B7, and YOLO11s-cls) for the classification of prehension events. Models were trained and tested on a dataset of logarithmic spectrograms (500 ms) derived from audio recordings acquired via collars on cattle.
Results: Analysis revealed high performance across all architectures. Although DenseNet-121 achieved the highest weighted metrics (Accuracy 83.7%, AUC 0.90), the YOLO11s-cls model demonstrated remarkable competitiveness, achieving nearly identical accuracy (83.1%) but with significantly superior computational efficiency (4.5 ms inference time). Crucially for field applications, YOLO exhibited excellent rejection of non-relevant sounds, with a 91% Specificity on the "no-prehension" class.
Conclusions: The study validates the efficacy of spectrogram-based 2D-CNNs for ingestion monitoring and identifies YOLO as a promising candidate for efficiency-oriented deployment scenarios, offering a favorable trade-off between predictive reliability and low-latency requirements. The developed platform further supports this transition from research to in-field application.

KEYWORDS
acoustic monitoring, bite detection, deep learning, grazing cows, precision livestock farming, spectrograms

1 Introduction

Recent advances in smart livestock farming have demonstrated the potential of data-driven approaches for monitoring animal behavior and improving management efficiency. In this context, different sensing modalities have been explored, including computer vision techniques for indoor monitoring, such as posture and behavior recognition of dairy cows in barns (Avanzato et al., 2022). While these approaches are well suited for controlled

environments, other sensing technologies are required for extensive and semi-extensive systems, where animals graze freely and feeding behavior is more difficult to observe.

Proper management of grazing animals can play an important role in improving productions and their quality, as well as animal health and welfare, and in optimizing pasture availability in terms of biomass and chemical-nutritional characteristics. Therefore, the availability of effective and easily applicable tools for measuring their behavior is important for increasing the efficiency of extensive and semi-extensive livestock farming systems, where grazing is the main, if not the only source of food for animals. Grazing is the least controllable feeding system for ruminants, due to the animals' extensive grazing areas. In such situations of wide freedom of movement, monitoring parameters such as time spent eating, ruminating, or resting with precision farming tools has yielded good results (Penning, 1983; Rutter, 2000; Werner et al., 2019; Pereira et al., 2020).

Accelerometers, positioned under the lower jaw of the animal, provide good responses as they are able to perceive and recognize head movements associated with actual grazing (Giovanetti et al., 2017) or, potentially, the less intense and more regular chewing activity typical of rumination. Counting individual grasping actions on grass tufts represents a further step in understanding grazing activity since, being associated with actual herbage input, it represents a starting point for the subsequent calculation of daily dry matter intake by estimating the weight of individual grasps.

Recent advances in signal processing and machine learning have enabled automatic classification of complex acoustic patterns in biological systems, ranging from animal vocalizations to ingestive behaviors (Schneider et al., 2024; Jobarteh et al., 2025; Clark and Dunn, 2022). Although much of this work has focused on vocalization or multimodal approaches, there remains a gap in the specific application of audio-based deep learning for the quantification of grazing and prehension events in free-ranging ruminants, which this study aims to address. More generally, audio signal processing combined with deep learning has been widely adopted for a variety of classification and detection tasks in complex acoustic scenarios. Time–frequency representations and spectrogram-based features, coupled with convolutional neural networks, have proven effective for robust sound event classification under noisy and real-world conditions, including environmental sound recognition and biomedical audio analysis (Damiano et al., 2024; Silva et al., 2022). These studies demonstrate the versatility and robustness of modern audio-based learning pipelines, highlighting their ability to generalize across different domains and application contexts. Building upon these advances, the present work applies similar signal processing and deep learning principles to the automatic detection of grazing-related prehension events, addressing a domain-specific challenge that has received comparatively limited attention.

Acoustic technology offers a non-invasive and promising alternative for real-time monitoring (Shorten, 2023). The sounds produced during biting and chewing exhibit distinctive patterns in terms of duration, energy, and spectral content, allowing for the characterization of ingestive activities (prehension, chewing, rumination) (Chelotti et al., 2016; Galli et al., 2018; Laca and WallisDeVries, 2000; Chelotti et al., 2023; Vanrell et al., 2020). Several studies have shown that audio-based systems can achieve accuracy levels comparable or superior to accelerometers, offering the advantages of a scalable and easily integrable technology.

Specific algorithms, such as the Real-Time Chews-Boluses Recognition Algorithm (CBRTA) (Chelotti et al., 2016) and the Jaw Movement Food Activity Recognizer (JMFAR) (Chelotti et al., 2023), automatically analyze jaw movement sounds. Other studies have used signal processing (Milone et al., 2012; Vanrell et al., 2020) or deep learning to classify ingestion events based on forage characteristics (Li et al., 2021), or even to differentiate ingestive sounds between different pasture types like grass and shrubs (Avondo et al., 2025). In this context, the approach combining the transformation of audio signals into spectrograms with the application of latest-generation convolutional neural networks (CNNs) has shown particularly promising results.

In our previous work (Avanzato et al., 2023), we proposed a method based on 1D CNNs trained on raw audio segments. Although initial results showed high accuracy, the limited dataset and reliance on raw audio raised concerns regarding overfitting and generalization. In this study, we propose an updated approach by conducting a comparative analysis of several state-of-the-art CNN architectures, including YOLO11 and Densenet, applied to spectrogram images generated from labeled audio recordings. We believe this approach, which evaluates the trade-offs between different models, improves robustness and provides a clear path for generalization on unseen data.

In light of these premises, the present study has the following specific objectives:

- The statistical characterization of spectral features to assess the discriminative power of spectrograms in distinguishing prehension from non-prehension events.
- The implementation and comparison of deep learning pipelines for the automatic classification of prehension events.
- The estimation of the average daily feed intake of grazing dairy cows by linking prehension event detection to grasp-level analysis, with the perspective of integrating this parameter into the proposed monitoring tool.
- The development and validation of a software platform for the management, annotation, and analysis of audio data recorded from grazing cattle.

The proposed approach is not limited to mere automatic classification but aims to provide a comprehensive and integrated analysis platform.

This paper is structured as follows. Section 2 is made up of subsection 2.1 that details the experimental dataset and the audio analysis methodology, including data acquisition, annotation procedures, signal preprocessing, and spectrogram generation; Section 2.2 provides a statistical characterization of prehension and no-prehension events, assessing the discriminative power of spectral features prior to deep learning; Section 2.3 describes the deep learning models adopted for automatic prehension classification, the training strategies, and the evaluation metrics; Section 2.4 presents the architecture and functionalities of the developed web-based software platform, describing its modular design, user interface, and operational workflow. Section 3 reports the experimental results and the comparative analysis of the tested architectures. Section 4 discusses the implications of the

findings, with particular attention to real-world deployment and computational efficiency. Finally, Section 5 summarizes the main conclusions and outlines future research directions.

2 Materials and methods

2.1 Dataset and audio analysis methodology

The platform's validation and the development of classification models were based on an experimental audio dataset, for which the collection and preparation followed a rigorous methodology.

2.1.1 Dataset description and acquisition

The audio dataset was acquired under in-field experimental conditions, monitoring some cows during free grazing on different types of pasture. The animals were equipped with microphone-enabled collars, allowing acoustic signals to be captured in close proximity to the source, thereby reducing the impact of background environmental noise. Audio files were recorded in stereo WAV format at a sampling rate of 44.1 kHz. During pre-processing, the signals were converted to mono and downsampled to 16 kHz. This reduction in sampling rate was chosen as an optimal compromise between preserving the informative spectral components of the signal and the computational efficiency required for processing. The recordings, averaging approximately 1 hour in duration, captured a wide variety of behavioral events. Despite the presence of environmental noise (e.g., wind, movement), preliminary analyses confirmed that prehension events maintain distinctive acoustic patterns, making them suitable for automatic identification.

Overall, the dataset was collected from a limited number of animals in a controlled experimental setting. In particular, the *training* and *validation* sets include recordings from three different bovines, while the *testing* set contains data from a fourth bovine that was never observed during model training. For the training and validation sets, the first bovine contributed eight distinct recording sessions, the second five sessions, and the third eight sessions. The testing dataset consists of eight recording sessions acquired from a single, distinct animal. This design allows the evaluation of the models both under seen-animal conditions and under a realistic unseen-animal scenario.

The creation of a ground truth dataset required a meticulous manual annotation phase, supported by both visual inspection and dedicated software tools. When necessary, audio recordings were synchronized with videos acquired using GoPro cameras to assist expert operators in accurately identifying feeding-related behaviors. Each audio file was carefully analyzed and segmented, precisely marking the start and end of individual prehension events.

The annotation process was further supported by the web-based platform described in Section 2.4, which provided an interactive interface for audio visualization, temporal navigation, and event labeling. Through this tool, operators were able to manually annotate prehension and no-prehension segments directly on the waveform, ensuring temporal precision and consistency across recordings. The platform also enabled the import and refinement of pre-existing annotations, facilitating iterative validation and correction of labels. All annotations, including timestamps and class labels (e.g., *prehension*, *no-prehension*), were exported and formalized in CSV format to guarantee interoperability and reproducibility.

TABLE 1 Distribution of sample sizes across classes and data splits.

Class	Training	Validation	Testing
No-prehension	17172	4294	10549
Prehension	14146	3537	6,545

2.1.2 Data preparation and preprocessing

To make the data compatible with the inputs of the machine learning models, the continuous audio signals were segmented into short-duration subsequences (chunks). Each event (both *prehension* and *no-prehension*) was split into fixed 500 ms time windows. For segments with a shorter duration, a zero-padding technique was applied to meet the required length. This approach generates a dimensionally homogeneous dataset that is easily processed by the models.

As is typical in behavioral datasets, the class distribution in the raw data was significantly imbalanced, with a clear predominance of *no-prehension* segments. To prevent model bias (overfitting) towards the majority class, a balancing strategy using under-sampling of the *no-prehension* class was applied to construct the training sets. The balanced data was then split into *training (80%)* and *validation (20%)* sets. It is important to note that the *testing dataset* was created using distinct recordings, never seen by the models during the training phase, to ensure an impartial and robust evaluation of generalization capability. The final dataset distribution (500 ms approach) is reported in Table 1.

Considering the entire dataset, the cumulative recording time amounts to approximately 7 h and 49 min. In particular, the *no-prehension* class accounts for about 4 h and 27 min of audio, while the *prehension* class covers approximately 3 h and 22 min. This overview highlights a moderate class imbalance that reflects realistic grazing conditions.

2.1.3 Feature representation

As input for the 2D-CNN and YOLO-type architectures, each audio segment was converted into a two-dimensional time-frequency representation. Specifically, logarithmic spectrograms were generated. In this representation, the x-axis corresponds to time, the y-axis to frequency, and the color intensity (mapped on a logarithmic scale from −80 dB to 0 dB) is proportional to the signal's energy in that specific time-frequency band. This transformation allows for emphasizing the most relevant acoustic components for class discrimination (Figure 1).

2.2 Statistical characterization of prehension events

To assess the discriminative power of the generated spectrograms prior to deep learning model training, an exploratory statistical analysis was conducted. This step aimed to verify whether the "Prehension" and "No-prehension" classes

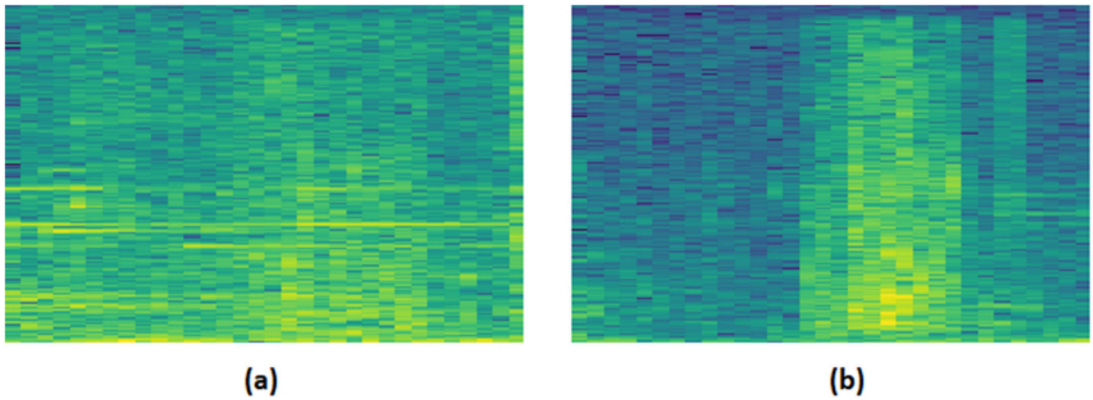

FIGURE 1
Example of a spectrogram related to an audio segment from **(a)** the "no-prehension" class and **(b)** the "prehension" class.

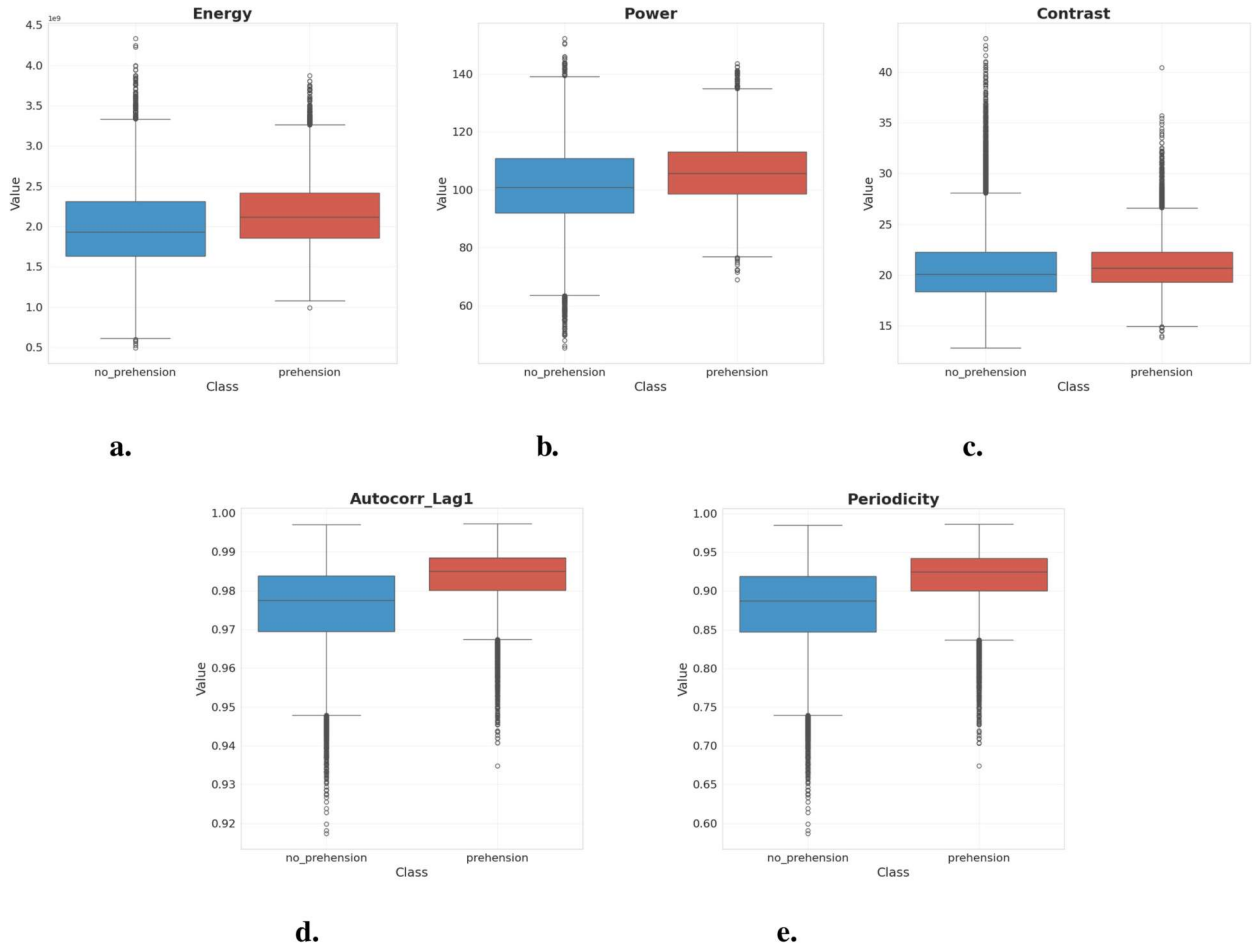

FIGURE 2
Statistical distribution of spectral features extracted from the training dataset ($N = 31,318$). **(a)** Spectral Energy. **(b)** Mean Power. **(c)** Spectral Contrast. **(d)** Lag-1 Autocorrelation. **(e)** Rhythmic Periodicity. The red boxes represent the *Prehension* class, showing distinctively higher intensity and temporal structure compared to the *No-prehension* class (blue).

exhibited distinct patterns in terms of energy and temporal structure, justifying the use of image-based classification.

2.2.1 Feature extraction methodology

Since the spectrograms were generated as pure signal representations (without axes or annotations), we extracted global statistical features directly from the pixel intensity values $I(x, y)$. The images were converted to grayscale, where pixel values range from 0 (silence) to 255 (peak intensity). Five descriptors were computed for each sample in the training set ($N = 31,318$):

1. Spectral Energy Proxy: Calculated as the sum of squared pixel intensities ($\sum I(x, y)^2$), representing the total acoustic energy of the 500 ms segment.
2. Mean Power: The arithmetic mean of pixel intensities, indicating the average signal strength independent of image dimensions.
3. Spectral Contrast: The standard deviation of pixel intensities, representing the dynamic range between background noise and signal peaks.
4. Temporal Consistency (Autocorr Lag-1): The spectrogram was reduced to a 1D temporal profile by averaging frequencies. The lag-1 autocorrelation of this profile was calculated to measure signal smoothness and continuity.
5. Periodicity: Defined as the magnitude of the secondary peak in the autocorrelation function, capturing rhythmic patterns typical of chewing or biting cycles.

2.2.2 Analysis of acoustic patterns

The distribution of these features for both classes is presented in Figure 2. The analysis reveals significant acoustic differences between prehension events and the background environment.

As shown in the *Energy* and *Power* boxplots (Figures 2a,b), the *Prehension* class (red) exhibits consistently higher median values compared to the *No-prehension* class (blue). This confirms that the act of tearing forage generates a distinct high-energy acoustic signature that clearly emerges from the background noise.

Regarding the temporal domain, the *Autocorr_Lag1* and *Periodicity* metrics indicate that prehension events are not only louder but also more structured. The *Prehension* distributions are more compact and shifted towards higher values, suggesting that the biting sound possesses a specific temporal envelope and rhythmicity that differentiates it from the stochastic nature of environmental sounds (e.g., wind, footsteps). Conversely, *Contrast* showed a significant overlap between classes, indicating that pixel variance alone is a weaker discriminator.

These statistical findings confirm that the spectrograms contain promising informative patterns, combining intensity and temporal structure, providing a solid basis for the subsequent training of Convolutional Neural Networks.

2.3 Deep learning models

For the automatic classification of prehension events, several deep learning architectures operating on 2D time-frequency representations were investigated and compared. In particular, the YOLO11 architecture was selected as the primary model due to its computational efficiency, while other state-of-the-art CNNs were considered for comparative evaluation.

2.3.1 YOLO11-based classification model

The primary approach proposed is based on the YOLO11 architecture (Khanam and Hussain, 2024), which, although renowned for object detection, was adapted here for a pure classification task. The small variant of the architecture (YOLO11s-cls) was adopted, as it provides an effective trade-off between classification accuracy and inference time. Unlike the standard detection models that output an $S \times S$ spatial grid, the -cls variant terminates with a global classification head. This head applies Global Average Pooling (GAP) to the final feature maps, followed by a linear layer to output class probabilities, completely bypassing the anchor-box decoding overhead. Furthermore, YOLO11 integrates advanced structural optimizations, such as the C3k2 blocks (an evolution of Cross Stage Partial networks), which enhance feature extraction efficiency while maintaining a highly parallelizable gradient flow. The model was trained on the spectrogram dataset (described in Section 2.1), and the final evaluation was performed on a disjoint test set: 31,318 images for training, 7,831 for validation, and 17,094 for testing. The inputs to the model are 500-millisecond duration logarithmic spectrograms. Data augmentation techniques, such as contrast variation and noise injection into the spectrograms, were adopted to increase the model's robustness. The final output of the model consists of the probability that a given spectrogram represents a prehension event.

The YOLO11s-cls model architecture used consists of 47 layers, totaling approximately 6.72 million parameters, with a computational complexity of 12.14 GFLOPs. The entire training and validation process was executed on an NVIDIA A100-PCIE-40GB GPU. Training and validation were managed using the Ultralytics framework (v8.3.174) and the PyTorch library (v2.4.1). A transfer learning strategy was employed, using a model pre-trained on the ImageNet dataset and adapting the final classifier to our binary classification task (2 classes). The training pipeline was executed for 100 epochs with a batch size of 16. Key hyperparameters included an image resolution of 640 × 640 pixels, an initial learning rate of 0.01, and the Stochastic Gradient Descent (SGD) optimizer (with 0.937 momentum). The training was completed in 4.459 h. At the conclusion, the model with the best performance on the validation set was saved. The validated model demonstrated high computational efficiency, with an average inference time of 4.53 ms per image and 0.2 ms for preprocessing.

2.3.2 Comparative 2D-CNN architectures

The second category of models operates on two-dimensional data (tensors [batch, channels, height, width]), treating the audio spectrograms (described in Section 2.1) as images. For this analysis, several 2D-CNN architectures, known for their effectiveness in hierarchical feature extraction, were compared:

- ResNet-101: An architecture based on residual connections (skip connections) that allows for training very deep networks

while mitigating the vanishing gradient problem (He et al., 2016).
- Densenet-121: Characterized by dense connections, where each layer receives the feature maps from all preceding layers as input, promoting feature reuse and more efficient gradient propagation (Huang et al., 2017).
- Efficientnet-b7: A model that uses a compound scaling strategy to optimally balance network depth, width, and resolution, achieving high computational efficiency (Tan and Le, 2019).

The entire pipeline was implemented using the PyTorch framework. The data was split into training and validation sets. A transformation pipeline was defined for spectrogram preprocessing: images were resized to 256 × 256 pixels. All images were converted to PyTorch tensors. Standard normalization (based on the ImageNet mean and standard deviation) was not applied. A transfer learning strategy was adopted. Each model was initialized with weights pre-trained on the ImageNet dataset. The final classification layer of each network was removed and replaced with a new layer to adapt the output to our binary classification task (two classes: no-prehension vs. prehension). Training was performed on an NVIDIA A100 GPU. The models were trained for 100 epochs using a batch size of 32. CrossEntropyLoss was used as the loss function, while optimization was handled by Stochastic Gradient Descent (SGD) with a learning rate of 1e-5 (0.00001) and a momentum of 0.9. During training, performance (Loss and Accuracy) was monitored on both the training and validation sets at each epoch using TensorBoard's SummaryWriter.

2.3.3 Evaluation metrics

To quantitatively evaluate the performance of the proposed models, standard metrics for binary classification tasks were adopted. The main metrics include:

- Accuracy (ACC): Measures the total proportion of correctly classified samples. (TP = True Positives, TN = True Negatives, FP = False Positives, FN = False Negatives). Equation 1 gives the formula for calculating accuracy:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

- Precision (P): Measures the proportion of positive predictions that were correct (True Positives) among all positive predictions. Equation 2 gives the formula for calculating precision:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

- Recall (R): Also known as Sensitivity, measures the proportion of True Positives correctly identified among all actual positive samples. Equation 3 gives the formula for calculating recall:

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

- F1-score: The harmonic mean of Precision and Recall, used to balance the trade-off between the two metrics. Equation 4 gives the formula for calculating F1_score:

$$\text{F1} - \text{score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

- Specificity: Measures the proportion of True Negatives correctly identified. Equation 5 gives the formula for calculating specificity:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{5}$$

- Confusion Matrix: A summary table visualizing True Positives, True Negatives, False Positives, and False Negatives.

All metrics were calculated on the test set, using a standard classification threshold of 0.5.

2.4 Software platform

The software platform, which is the subject of this study, was designed to unify the fundamental phases of the analysis process into a single operational environment: from raw audio data acquisition, to manual annotation (ground-truth labeling), up to the training and validation of deep learning models for the automatic recognition of prehension events.

2.4.1 Platform architecture

The system's architecture is based on a modular approach, designed to ensure scalability, maintainability, and the future integration of additional components, such as video analysis modules. The platform was implemented as an interactive web application, accessible via a browser, eliminating the need for dedicated user-side hardware infrastructure. This architecture was conceived to support not only researchers but also technical operators or farmers for in-field use.

The system is structured into three main functional macro-layers:

- Annotation and Data Management Interface: Allows for the uploading of audio files (WAV format), interactive signal visualization, and manual labeling of events (prehension, rumination, neutral segments). Annotations are managed via CSV files to ensure interoperability.
- Inference and Classification Engine: Integrates pre-trained machine learning models on spectrograms. It performs real-time (in-browser) predictions thanks to the integration of TensorFlow.js, without the need for dedicated computing servers (GPUs) or server-side processing.
- Results Visualization and Analysis Module: Allows for a synoptic (visual) comparison between manually annotated segments (ground truth), segments imported from external files, and those predicted by the model. It includes interactive charts for classification probabilities and supports data exportation for offline analysis.

FIGURE 3
Visualization of the audio signal in the time domain within the analysis platform.

The choice of the technology stack was guided by stringent objectives of accessibility, scalability, and computational lightness:

- Frontend: Developed in React. js (Meta Platforms, Inc., 2024), a modern framework that ensures modularity and ease of updating.
- Audio Visualization: Interaction with the waveform is managed by the Wavesurfer. js library (Wavesurfer.js team, 2024), which allows for navigation and precise selection of audio segments.
- Machine Learning (Client-Side): The execution of pre-trained models directly in the browser is made possible by the integration of TensorFlow.js (Smilkov et al., 2019).
- UI/UX: The graphical user interface utilizes the Material UI library (MUI, 2024), while reporting and visualization of classification charts are based on ApexCharts (ApexCharts, 2024).

Such an architecture makes the platform multi-user and accessible from any station equipped with a modern browser, without requiring complex software installations. Furthermore, the use of TensorFlow.js opens up future scenarios for re-training or fine-tuning models directly in-browser, leveraging new data labeled by users in the field. Due to ongoing national project constraints, the platform source code and trained model weights are not publicly released at this stage; however, academic access can be granted upon reasonable request to the corresponding author.

2.4.2 User interface and operational workflow

The user interface was designed to guide the operator through a sequential and standardized workflow, articulated in the following phases:

1. Audio Data Uploading: The user uploads the WAV audio files acquired via monitoring devices (e.g., collar-mounted microphones) (Figure 3).
2. Labeling (Ground Truth Creation): The platform supports both direct manual annotation on the signal (ground truth creation) and the import of pre-existing labels via CSV files (Figure 4).
3. Pre-processing and Model Selection: The user can configure analysis parameters, such as the offset window. Upon selecting the model, the platform manages the automatic transformation of audio segments into logarithmic spectrograms (−80 dB–0 dB), normalized or at original duration, before sending them to the inference engine (Figure 5).
4. Inference and Threshold Adjustment: The user initiates the automatic classification. Predictions are calculated and displayed as an overlay on the waveform (Figure 6a). The platform also allows for the dynamic setting and modification of the confidence threshold (Figure 6b), enabling the user to balance the model's Precision and Recall based on specific analysis needs.
5. Result Visualization and Comparison: The system allows for a comparative visual analysis between manual labels, imported CSV labels, and those automatically generated by the model, immediately highlighting any discrepancies (Figure 7).
6. Export and Analysis: The final results, including manual, imported, and predicted annotations, can be exported in CSV format for subsequent offline statistical analysis or to feed secondary modules (e.g., ingested biomass estimation) (Figure 8).
7. Settings Section: A dedicated section (Figure 9) summarizes the audio file's metadata (e.g., sample rate, duration, size) and the status of the loaded inference model (e.g., model ID, chunk duration).

3 Experimental results

The models were trained and evaluated on the labeled spectrogram dataset, using the data split defined in Section 2.1. The final model was selected based on performance on the validation set and tested on unseen data (the test set) to ensure impartiality.

Table 2 summarizes the main evaluation metrics for the YOLO11s-cls model, calculated on the test set.

FIGURE 4
Examples of manual labeling and file-based labeling on the audio signal loaded into the analysis platform.

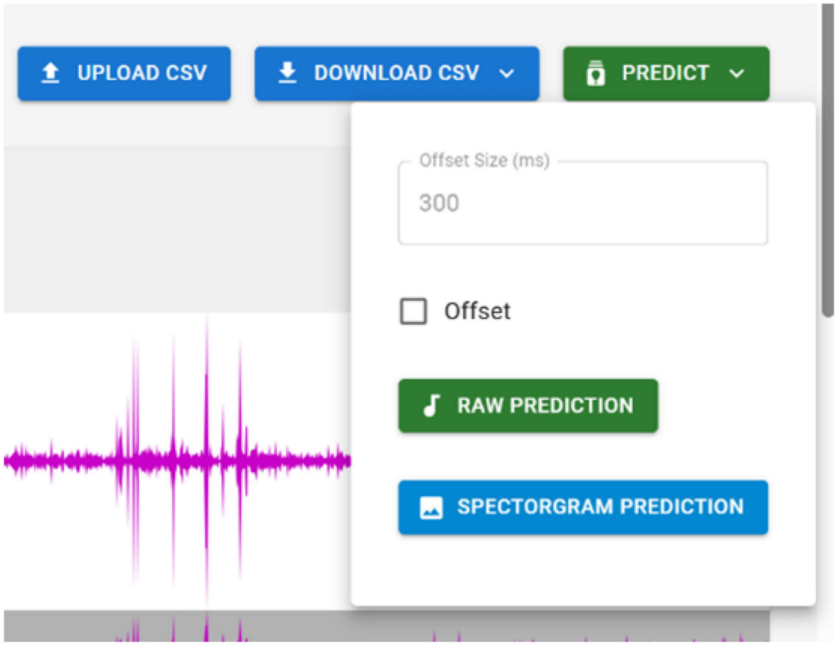

FIGURE 5
Dropdown menu for selecting the offset and the network model to be used for the automatic prehension classification phase.

A more detailed analysis of the YOLO11s-cls model's performance, based on its confusion matrix, highlights a robust classification behavior, albeit with a clear trade-off between the two classes. The model excels at identifying the negative class (*No-prehension*), achieving a True Negative Rate (Specificity) of 91%. This is a particularly positive result, as it implies a False Positive Rate (FPR) of only 9%. In practical terms, this means the model is highly reliable at avoiding the misclassification of "no-prehension" segments (such as background noise or footsteps) as prehension events. For the positive class (*Prehension*), the model achieves a True Positive Rate (Recall or Sensitivity) of 70%. Although the model correctly identifies the majority of events, the analysis reveals that its primary source of error lies in False Negatives (FNR): 30% of actual prehension events are missed and incorrectly classified by the model as "no_prehension". Despite this imbalance (high Specificity and

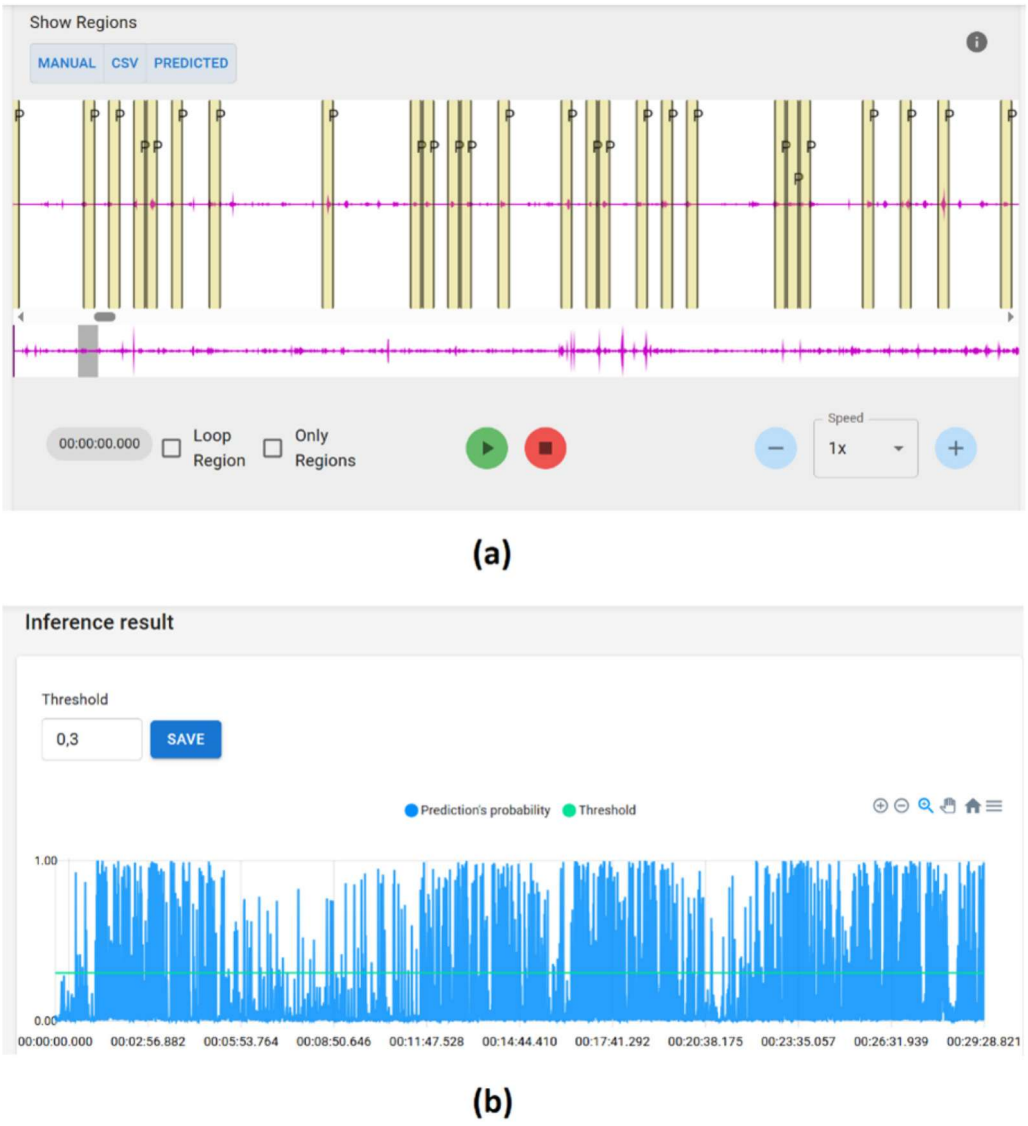

FIGURE 6
(a) Visualization of the model-predicted prehension labels and **(b)** the confidence threshold curve for each 500 ms chunk, with threshold adjustment capability.

moderate Recall), the overall results confirm the effectiveness of the approach, especially in minimizing false positive errors.

To evaluate the impact of architectural choice on classification performance, several deep learning models were trained and compared. Table 3 summarizes the performance metrics obtained by each architecture on the test set.

The comparative analysis reveals a highly competitive landscape where the *YOLO11s-cls* architecture demonstrates remarkable capability, rivaling traditional heavyweight classifiers despite being optimized for speed and efficiency.

Although *DenseNet-121* recorded the highest absolute values across the metrics (Accuracy 83.7%), its advantage over *YOLO11s-cls* (Accuracy 83.1%) is marginal. The performance gap is less than 1%, a small difference that may not justify the higher computational cost of dense architectures, depending on the target deployment constraints. While DenseNet's feature reuse mechanism proves effective for spectrogram analysis, YOLO demonstrates that a streamlined, efficiency-first architecture can capture the same complex acoustic patterns with virtually equivalent precision.

The *YOLO11s-cls* model stands out as the most balanced solution. It outperformed both *ResNet-101* (82.6%) and *Efficientnet-b7* (81.2%), establishing itself as a strong efficiency-oriented alternative for practical deployment, with comparable predictive performance and lower latency. While DenseNet shows a slightly higher weighted Specificity (80.4% vs. 78.1%), closer inspection of the confusion matrices (Figure 10) reveals that YOLO maintains excellent reliability in rejecting False Positives (91% True Negative Rate on the critical 'no_prehension' class), mitigating the impact of the minor numerical discrepancy in weighted specificity.

FIGURE 7
Comparison between manual labels, labels loaded from a CSV file, and those predicted by the CNN model.

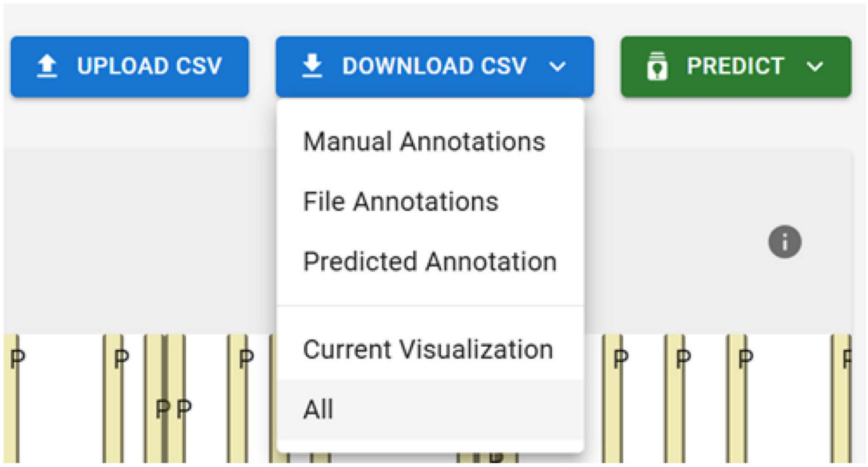

FIGURE 8
Dropdown menu for selecting which CSV file to download.

4 Discussion

Beyond the purely methodological comparison of deep learning architectures, this study is primarily motivated by an applicative objective of central importance for precision livestock farming: the estimation of the temporal distribution and average daily amount of feeding activity in grazing dairy cows. In particular, the automatic detection of prehension events enables the estimation of the number of feeding acts over time, potentially aggregated into hourly or diurnal time slots, which represents a fundamental prerequisite for indirectly assessing daily feed intake. This information is of paramount relevance for farmers, as it supports nutritional management, pasture planning, and early detection of anomalies in feeding behavior.

Within this framework, the final selection of the YOLO11s-cls architecture prioritizes the optimal balance between predictive reliability and operational feasibility, which is essential for large-scale and continuous monitoring in real farming conditions. The rationale for this choice is threefold:

- Performance–Efficiency Ratio: Although DenseNet-121 achieves slightly higher peak metrics, this improvement comes at the cost of significantly increased computational complexity. YOLO11s-cls delivers nearly equivalent Accuracy and F1-scores while maintaining a lightweight footprint (5.4M parameters). As detailed in Table 4, a purely theoretical look at floating-point operations reveals a paradox: DenseNet-121 requires fewer theoretical GFLOPs compared to YOLO11s-cls. However, theoretical FLOPs are an incomplete metric for real-world inference speed, as they do not account for Memory Access Cost (MAC) and hardware-level parallelism. The architecture of DenseNet-121 relies on continuous concatenation of feature maps from all preceding layers, creating severe memory fragmentation and high read/write traffic. Conversely, YOLO11s-cls

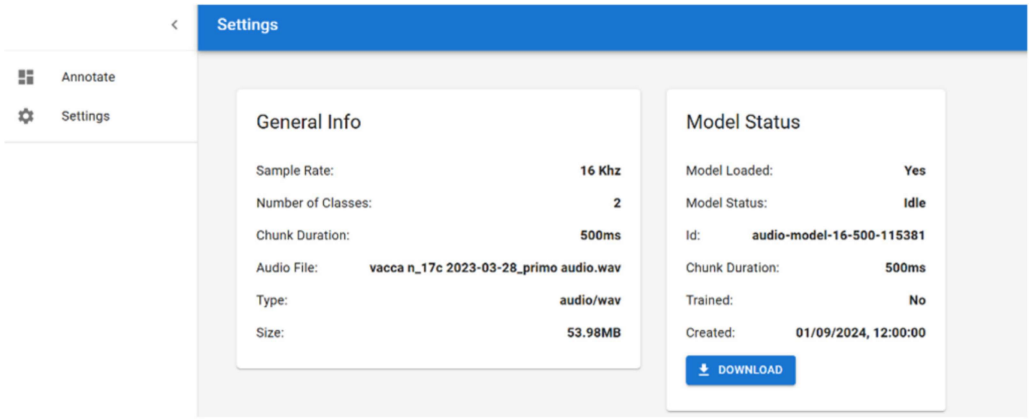

FIGURE 9
Screenshot of the settings information.

TABLE 2 Per-class performance metrics for the YOLO11s-cls model on the test set.

Class	Accuracy	Precision	Recall	F1-score	Specificity
No-prehension	-	0.831	0.912	0.870	0.700
Prehension	-	0.832	0.700	0.760	0.912
Weighted	0.831	0.831	0.831	0.828	0.781

TABLE 3 Comparison of performance metrics (weighted average) of the models on the test set.

Model	Accuracy	Precision	F1-score	Specificity
YOLO11s-cls	0.831	0.831	0.831	0.781
Densenet-121	0.837	0.836	0.837	0.804
ResNet-101	0.826	0.825	0.826	0.780
Efficientnet-b7	0.812	0.811	0.812	0.784

leverages a streamlined, highly parallelizable structure (e.g., C3k2 blocks) that minimizes MAC. Furthermore, the `-cls` variant bypasses the standard spatial detection grid entirely, utilizing a Global Average Pooling (GAP) classification head. Thus, despite a higher FLOP count, YOLO provides a significantly faster and more hardware-friendly inference process. From an applicative perspective, the marginal performance gain of heavier models may not justify their reduced practicality in deployment scenarios where latency and sustained throughput are critical.

To rigorously evaluate this 0.6% performance gap, McNemar's test was applied to the paired sample-wise predictions of the two models on the test set. The test confirmed that the performance difference between DenseNet-121 and YOLO11s-cls is not statistically significant ($\chi^2 = 2.618$, $p = 0.106$). Therefore, the accuracy trade-off is genuinely marginal, further justifying the selection of the YOLO architecture for its significantly superior computational efficiency and suitability for real-time edge deployment.

- Inference Speed for Real-Time and Long-Term Monitoring: As detailed in Section 2.3, YOLO11s-cls achieves an inference time of approximately 4.5 ms on an NVIDIA A100 GPU. This level of efficiency enables high-frequency inference over extended recording periods, which is a key requirement for reconstructing feeding patterns across different times of the day without introducing latency or computational bottlenecks. Such capability is crucial for translating event-level detection into meaningful daily intake indicators.
- Scalability and Robustness: Despite being originally conceived for object detection, the YOLO architecture shows promise when adapted to spectrogram-based classification. Its ability to match the performance of more complex CNNs highlights its versatility and makes it particularly suitable for future extensions of the monitoring system, such as the inclusion of additional ingestive behaviors (e.g., rumination) or deployment on resource-constrained edge devices. However, it must be noted that executing a full-precision (FP32) model on a desktop GPU does not constitute full edge readiness; to achieve true edge deployment, future work will require model quantization (e.g., FP32 to INT8) and direct empirical benchmarking on ARM-based microprocessors (e.g., Raspberry Pi or NVIDIA Jetson).

In this context, YOLO11s-cls emerges as a practical solution for bridging the gap between high-performance research models and the constraints of on-farm deployment. Its combination of accuracy, speed, and computational efficiency enables the continuous monitoring of prehension activity, which constitutes the foundation for estimating daily feed intake in grazing cattle.

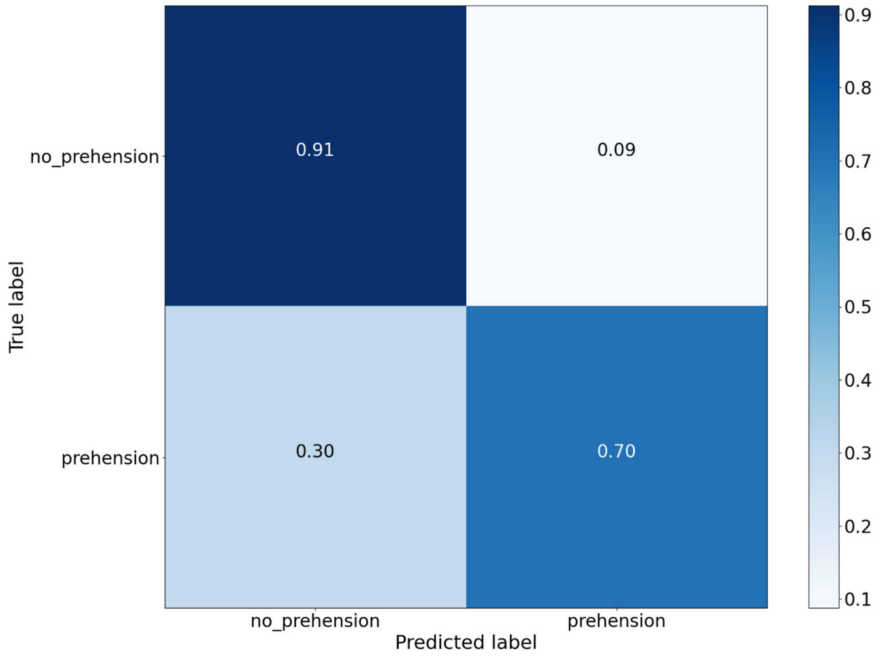

FIGURE 10
Normalized confusion matrix for the YOLO11s-cls model, calculated on the test set. The axes indicate the predicted label and the true label. The values on the diagonal represent the percentage of correct classifications for each class.

TABLE 4 **Empirical comparison of computational complexity, memory footprint, and inference efficiency for the evaluated architectures. Metrics were computed with a batch size of 1 on an NVIDIA GPU (FP32 precision), matching the respective input resolutions (640 × 640 for YOLO11s-cls and 256 × 256 for 2D-CNNs).**

Model	Parameters (M)	Memory (MB)	GFLOPs	Inference time (ms)
YOLO11s-cls	6.72	26.90	12.14	4.53 ± 0.03
DenseNet-121	6.96	27.82	7.57	11.62 ± 0.06
ResNet-101	42.50	170.02	20.54	8.67 ± 0.05
EfficientNet-b7	63.79	255.17	13.95	21.47 ± 0.21

A qualitative analysis of the remaining 9% False Positive rate indicates that misclassifications are primarily triggered by the animal walking through the pasture. The acoustic friction and the physical snapping of the vegetation against the animal's hooves or body create transient, high-energy broadband noises. These specific environmental sounds can temporarily mimic the spectral energy signature and temporal envelope of an actual grass tearing (prehension) event, occasionally bypassing the model's rejection filters.

4.1 Analysis of low recall and real-time constraints

While the YOLO11s-cls model achieved an excellent Specificity (91%), effectively rejecting background noise, its Recall for the Prehension class was 0.700. Although a 30% False Negative Rate might initially seem to heavily impact the calculation of daily dry matter intake, its practical criticality can be mitigated. Grazing behavior typically occurs in dense temporal bursts (meal bouts) rather than isolated acoustic events. Consequently, the missed detection of scattered individual prehensions can be mathematically compensated through standard post-processing techniques, such as temporal smoothing or heuristic sequence-matching algorithms, effectively bridging the 30% gap to reconstruct the overall feeding session. To better visualize the absolute distribution of these classifications, the non-normalized confusion matrix is provided in Figure 11.

A primary technical cause for this reduced sensitivity lies in the strictly causal windowing strategy employed. In this study, continuous audio signals were segmented into rigid, non-overlapping 500 ms chunks. While implementing an overlapping sliding window (e.g., 50%) is a standard technique that typically improves recall in offline bioacoustics by preserving boundary events, we deliberately opted against it. An overlapping strategy inevitably requires continuous audio buffering, which introduces processing latency and increases memory overhead. These factors are strictly incompatible with the primary objective of deploying a zero-latency, real-time inference system on low-power edge devices.

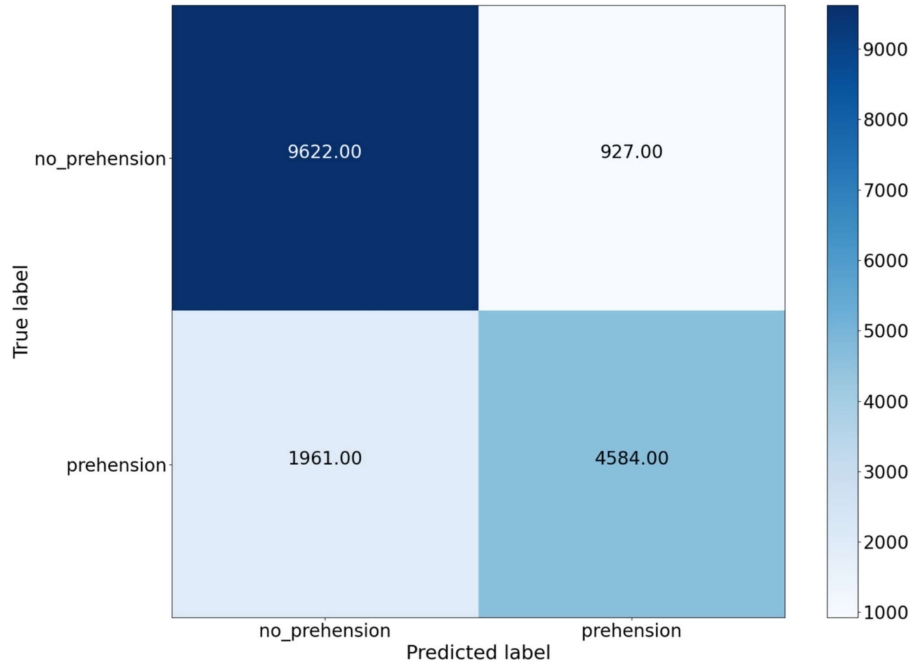

FIGURE 11
Non-normalized confusion matrix for the YOLO11s-cls model on the test set, displaying the absolute number of detected and missed events.

Therefore, the non-overlapping segmentation is considered a design choice aligned with strict real-time constraints in low-power settings. The inevitable boundary losses are instead compensated through the aforementioned downstream post-processing techniques, ensuring accurate meal bout reconstruction without compromising the strict real-time hardware constraints.

Furthermore, regarding the loss function, the model was trained using standard Cross-Entropy. This choice was dictated by the fact that the class imbalance within the training set was relatively moderate (17,172 *no-prehension* versus 14,146 *prehension* samples), preventing the network from collapsing toward the majority class. However, as the natural imbalance in the test set is more pronounced, future iterations will explore the implementation of Focal Loss. By dynamically down-weighting well-classified background noise and imposing heavier penalties on missed minority class events, Focal Loss represents an optimal strategy to further enhance the model's sensitivity without compromising its real-time constraints.

4.2 Sample size, generalization, and biological variability

Despite the promising computational efficiency and classification metrics achieved by the YOLO11s-cls architecture, it is essential to contextualize the biological limitations of the current study. The dataset was derived from a limited sample size of four animals (three for model training, one for unseen testing). Consequently, the model's reported accuracy and 'robustness' strictly apply to these specific experimental conditions and the single test animal utilized. We acknowledge that cattle masticatory and prehension sounds vary significantly across the species due to differences in cranial morphology, animal age, and the specific structural properties of the grazed forage. However, to partially mitigate this limitation and account for dietary variability, it should be noted that the audio data from these four animals were acquired under highly contrasting grazing conditions. Specifically, the recordings were conducted on two different dates across distinct pasture areas: a poor pasture, predominantly composed of grasses in an advanced growth stage, and a rich pasture with abundant biomass, primarily consisting of sulla in the vegetative stage. Nevertheless, given the overall constrained sample size of monitored individuals, there remains an inherent risk that the deep learning models may overfit to the specific acoustic phenotypes rather than learning generalized prehension features. Therefore, while this study validates the technological and computational pipeline–from the web platform to edge-capable inference–future research must prioritize biological validation. Deploying and evaluating the model on a significantly larger, unseen herd across diverse pasture environments will be a mandatory next step to definitively confirm the cross-subject generalization and true applicability of the proposed acoustic monitoring system.

4.3 Quantitative and qualitative acoustic monitoring

The proposed framework opens the door to more advanced acoustic-based analyses of feeding behavior. As already demonstrated in previous work (Avondo et al., 2025), the spectral characteristics of prehension sounds can also provide qualitative information about the type of forage being ingested, enabling discrimination between different pasture components such as grasses and legumes. When combined with the quantitative estimation of prehension frequency and temporal distribution

presented in this study, these approaches pave the way toward a comprehensive, non-invasive assessment of both the amount and the quality of feed intake in grazing systems.

5 Conclusion and future work

This study presented the development and validation of an innovative software platform for monitoring the feeding behavior of grazing dairy cows, with a specific focus on the automatic detection of prehension events using acoustic signals and deep learning techniques. The proposed platform is modular and user-friendly, allowing users to upload, visualize, annotate, and automatically classify audio recordings through an integrated web-based interface. The graphical user interface, implemented with modern technologies such as React. js and Wavesurfer. js, was designed to ensure accessibility and usability even for non-expert operators.

From a methodological perspective, the platform incorporates an efficient audio analysis pipeline based on spectrogram representations and deep learning models. A comparative evaluation of several state-of-the-art architectures (including ResNet-101, DenseNet-121, EfficientNet-b7, and YOLO11s-cls) demonstrated that YOLO11s-cls provides the most effective trade-off between predictive performance and computational efficiency. In particular, its low inference latency (4.5 ms per sample) and high reliability in rejecting non-prehension events (91% Specificity on the *no-prehension* class) make it well suited for continuous monitoring and real-world deployment.

The applicative implications for precision livestock farming are substantial. The ability to automatically detect and temporally aggregate prehension events represents a fundamental step toward estimating feeding activity patterns over the course of the day and, indirectly, daily feed intake. Such information is of primary importance for farmers, as it supports nutritional management, pasture optimization, and early detection of deviations in feeding behavior that may indicate health or welfare issues.

Despite these promising results, the present study also exhibits some limitations that will guide future research efforts. First, the experimental dataset was collected from a limited number of bovines and recording sessions. Although the evaluation included an unseen-animal testing scenario, increasing both the number of monitored animals and the duration of recordings will be essential to further diversify the dataset and improve model robustness and generalization across different breeds, environments, and seasonal conditions.

Second, while the proposed models demonstrated strong performance under real grazing conditions, environmental noise remains a critical challenge in outdoor acoustic monitoring. Future work will therefore focus on the integration of noise-robust signal processing techniques and data augmentation strategies specifically designed to improve resilience to wind, movement-related artifacts, and background sounds commonly encountered in pasture environments.

Third, although this study establishes a reliable framework for prehension event detection, the estimation of ingested biomass is currently indirect. A key future development will involve the quantitative estimation of feed intake at the individual cow level by linking the automatically detected number of prehension events to grasp-level intake models. This will enable the calculation of average daily feed intake and its temporal distribution, followed by validation against reference measurements, thereby strengthening the practical relevance of the system for farm-level decision support.

In parallel, future research will extend the range of behaviors analyzed by the platform to include rumination, idle phases, and vocalizations, enabling a more comprehensive characterization of animal behavior. The platform will also be optimized for deployment on edge devices, such as Raspberry Pi or NVIDIA Jetson platforms, leveraging the computational efficiency demonstrated by YOLO11s-cls.

Finally, multimodal extensions represent a promising direction for further enhancing system accuracy and robustness. The integration of audio-based monitoring with computer vision techniques, such as posture recognition and pasture biomass estimation, will enable the development of a fully integrated, non-invasive, and scalable monitoring framework for precision livestock farming.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Ethics statement

Ethical approval was not required for the study involving animals in accordance with the local legislation and institutional requirements because the audio recordings were made simply by wearing a collar equipped with a lightweight digital audio recorder.

Author contributions

RA: Supervision, Software, Writing – review and editing, Writing – original draft, Visualization, Conceptualization, Validation, Methodology. LB: Software, Writing – review and editing, Writing – original draft, Validation, Data curation, Visualization, Formal Analysis. SB: Visualization, Data curation, Writing – review and editing. FB: Methodology, Writing – review and editing, Supervision, Conceptualization. MA: Data curation, Methodology, Funding acquisition, Writing – review and editing, Conceptualization.

Funding

The author(s) declared that financial support was received for this work and/or its publication. This work was supported in part by the European Union Next-GenerationEU (PIANO NAZIONALE DI RIPRESA E RESILIENZA (PNRR) – MISSIONE 4 COMPONENTE 2, INVESTIMENTO 1.4 – D. D. 1032 17/06/2022, CN00000022).

Acknowledgements

This study was carried out within the Agritech National Research Center.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author(s) FB declared that they were an editorial board member of Frontiers, at the time of submission. This had no impact on the peer review process and the final decision.

Generative AI statement

The author(s) declared that generative AI was used in the creation of this manuscript. ChatGPT was used exclusively for linguistic rephrasing of the text and did not contribute to the generation of original content or the development of the ideas presented.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

Author disclaimer

This manuscript reflects only the authors' views and opinions, neither the European Union nor the European Commission can be considered responsible for them.

References

ApexCharts (2024). *Apexcharts.js - open source javascript charts for your website*. Available online at: https://apexcharts.com/ (Accessed April 20, 2024).

Avanzato, R., Beritelli, F., and Puglisi, V. F. (2022). "Dairy cow behavior recognition using computer vision techniques and cnn networks," in 2022 IEEE international conference on internet of things and intelligence systems (IoTaIS) *(IEEE)*, 122–128.

Avanzato, R., Avondo, M., Beritelli, F., Di Franco, F., and Tumino, S. (2023). "Detecting the number of bite prehension of ggrazing cows in an extensive system using an audio recording method," in *Icyrime*, 27–31.

Avondo, M., Bognanno, M., Beritelli, F., Avanzato, R., Biondi, L., Gimmillaro, F., et al. (2025). Sound and video detection as a tool to estimate free grazing behavior in sheep on different swards. *Animals* 15, 2671. doi:10.3390/ani15182671

Chelotti, J. O., Vanrell, S. R., Milone, D. H., Utsumi, S. A., Galli, J. R., Rufiner, H. L., et al. (2016). A real-time algorithm for acoustic monitoring of ingestive behavior of grazing cattle. *Comput. Electron. Agric.* 127, 64–75. doi:10.1016/j.compag.2016.05.015

Chelotti, J. O., Vanrell, S. R., Martinez-Rau, L. S., Galli, J. R., Utsumi, S. A., Planisich, A. M., et al. (2023). Using segment-based features of jaw movements to recognise foraging activities in grazing cattle. *Biosyst. Eng.* 229, 69–84. doi:10.1016/j.biosystemseng.2023.03.014

Clark, F. E., and Dunn, J. C. (2022). From soundwave to soundscape: a guide to acoustic research in captive animal environments. *Front. Vet. Sci.* 9, 889117. doi:10.3389/fvets.2022.889117

Damiano, S., Cramer, B., Guntoro, A., and van Waterschoot, T. (2024). Synthetic data generation techniques for training deep acoustic siren identification networks. *Front. Signal Process.* 4, 1358532. doi:10.3389/frsip.2024.1358532

Galli, J. R., Cangiano, C. A., Pece, M., Larripa, M., Milone, D. H., Utsumi, S., et al. (2018). Monitoring and assessment of ingestive chewing sounds for prediction of herbage intake rate in grazing cattle. *Animal* 12, 973–982. doi:10.1017/S1751731117002415

Giovanetti, V., Decandia, M., Molle, G., Acciaro, M., Mameli, M., Cabiddu, A., et al. (2017). Automatic classification system for grazing, ruminating and resting behaviour of dairy sheep using a tri-axial accelerometer. *Livest. Sci.* 196, 42–48. doi:10.1016/j.livsci.2016.12.011

He, K., Zhang, X., Ren, S., and Sun, J. (2016). "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 770–778. doi:10.1109/CVPR.2016.90

Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 4700–4708. doi:10.1109/CVPR.2017.243

Jobarteh, B., Mincu-Iorga, M., Gavojdian, D., and Neethirajan, S. (2025). Integrating multi-modal data fusion approaches for analysis of dairy cattle vocalizations. *Front. Veterinary Sci.* 12, 1704031. doi:10.3389/fvets.2025.1704031

Khanam, R., and Hussain, M. (2024). *Yolov11: an overview of the key architectural enhancements*. arXiv preprint arXiv:2410.17725.

Laca and WallisDeVries (2000). Acoustic measurement of intake and grazing behaviour of cattle. *Grass Forage Sci.* 55, 97–104. doi:10.1046/j.1365-2494.2000.00203.x

Li, G., Xiong, Y., Du, Q., Shi, Z., and Gates, R. S. (2021). Classifying ingestive behavior of dairy cows *via* automatic sound recognition. *Sensors* 21, 5231. doi:10.3390/s21155231

Meta Platforms, Inc. (2024). React: the library for web and native user interfaces. Available online at: https://react.dev (Accessed May 30, 2024).

Milone, D. H., Galli, J. R., Cangiano, C. A., Rufiner, H. L., and Laca, E. A. (2012). Automatic recognition of ingestive sounds of cattle based on hidden markov models. *Comput. Electronics Agriculture* 87, 51–55. doi:10.1016/j.compag.2012.05.004

MUI (2024). Material ui: react components that implement material design. Available online at: https://mui.com/material-ui/ (Accessed May 30, 2024).

Penning, P. (1983). A technique to record automatically some aspects of grazing and ruminating behaviour in sheep. *Grass Forage Sci.* 38, 89–96. doi:10.1111/j.1365-2494.1983.tb01626.x

Pereira, G., Heins, B., O'Brien, B., McDonagh, A., Lidauer, L., and Kickinger, F. (2020). Validation of an ear tag–based accelerometer system for detecting grazing behavior of dairy cows. *J. Dairy Science* 103, 3529–3544. doi:10.3168/jds.2019-17269

Rutter, S. M. (2000). Graze: a program to analyze recordings of the jaw movements of ruminants. *Behav. Res. Methods, Instrum. and Comput.* 32, 86–92. doi:10.3758/bf03200791

Schneider, S., von Fersen, L., and Dierkes, P. W. (2024). Acoustic estimation of the manatee population and classification of call categories using artificial intelligence. *Front. Conservation Sci.* 5, 1405243. doi:10.3389/fcosc.2024.1405243

Shorten, P. (2023). Acoustic sensors for detecting cow behaviour. *Smart Agric. Technol.* 3, 100071. doi:10.1016/j.atech.2022.100071

Silva, L., Valadão, C., Lampier, L., Delisle-Rodríguez, D., Caldeira, E., Bastos-Filho, T., et al. (2022). Covid-19 respiratory sound analysis and classification using audio textures. *Front. Signal Process.* 2, 986293. doi:10.3389/frsip.2022.986293

Smilkov, D., Thorat, N., Assogba, Y., Yuan, A., Kreeger, N., Yu, P., et al. (2019). "Tensorflow.js: machine learning for the web and beyond," in *Proceedings of machine learning and systems (MLSys)*, 1.

Tan, M., and Le, Q. V. (2019). "Efficientnet: rethinking model scaling for convolutional neural networks," in Proceedings of the 36th international conference on machine learning (ICML) *(PMLR), vol. 97 of* proceedings of machine learning research, 6105–6114.

Vanrell, S. R., Chelotti, J. O., Bugnon, L. A., Rufiner, H. L., Milone, D. H., Laca, E. A., et al. (2020). Audio recordings dataset of grazing jaw movements in dairy cattle. *Data Brief* 30, 105623. doi:10.1016/j.dib.2020.105623

Wavesurfer.js team (2024). Wavesurfer.js audio waveform player javascript library. Available online at: https://wavesurfer.xyz (Accessed May 30, 2024).

Werner, J., Umstatter, C., Leso, L., Kennedy, E., Geoghegan, A., Shalloo, L., et al. (2019). Evaluation and application potential of an accelerometer-based collar device for measuring grazing behavior of dairy cows. *Animal* 13, 2070–2079. doi:10.1017/S1751731118003658