

A Computer Vision Pipeline for Individual-Level Behavior Analysis: Benchmarking on the Edinburgh Pig Dataset

Haiyu Yang

hy625@cornell.edu

Cornell University

Enhong Liu

Cornell University

Jennifer Sun

Cornell University

Sumit Sharma

Cornell University

Meike van Leerdam

Cornell University

Sebastien Franceschini

University of Liège

Puchun Niu

Cornell University

Miel Hostens

Cornell University

Article

Keywords: animal behavior analysis, precision livestock farming, deep learning, automated monitoring, video tracking

Posted Date: June 15th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9971180/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on August 24th, 2026.
See the published version at <https://doi.org/10.1038/s41598-026-67803-4>.

A Computer Vision Pipeline for Individual-Level Behavior Analysis: Benchmarking on the Edinburgh Pig Dataset

Haiyu Yang^{*1}, Enhong Liu¹, Jennifer Sun¹, Sumit Sharma¹, Meike van Leerdam¹, Sebastien Franceschini², Puchun Niu¹, and Miel Hostens¹

¹Department of Animal Science, College of Agriculture and Life Sciences, Cornell University, Ithaca, NY 14853, USA

²University of Liège, Gembloux Agro-Bio Tech (ULiège-GxABT), 5030 Gembloux, Belgium

Abstract

Animal behavior analysis is central to understanding welfare, health, and productivity in livestock, yet manual observation is time-consuming, subjective, and difficult to scale. We present a modular pipeline that integrates open-source, state-of-the-art computer vision models to automate individual-level behavior analysis in group-housing environments. The pipeline combines zero-shot object detection, motion-aware segmentation and tracking, and vision-transformer feature extraction for robust behavior recognition, addressing challenges such as occlusion and crowding in indoor pig monitoring. We validated the system on the Edinburgh Pig Behavior Video Dataset across detection, tracking, and behavior-classification tasks. A temporal model achieved 94.2% overall accuracy on nine behaviors, a 21.2 percentage-point improvement over the previous benchmark, while tracking reached 93.3% identity preservation (IDF1) and detection reached 89.3% average precision. The modular design lets individual components be replaced as methods advance and suggests potential for adaptation to other species and settings, although cross-species use would require further validation. By releasing an open-source, end-to-end implementation, this work provides a reproducible and scalable tool for automated, objective, and continuous behavior monitoring in precision livestock farming and welfare assessment.

Keywords: animal behavior analysis; precision livestock farming; deep learning; automated monitoring; video tracking

1 Introduction

Welfare is an increasing concern in livestock production^{1,2}, and animal behavior is a vital, early indicator of physiological and psychological state³⁻⁵. Automated detection of behavioral change can flag health and welfare problems before clinical signs appear, both in pigs⁶ and in cattle^{7,8}. Traditional monitoring, however, relies on human observation that is labor-intensive, subjective, and limited in temporal coverage and scale^{9,10}; reliable welfare assessment can require hundreds of

^{*}Corresponding author: hy625@cornell.edu

animals, a level of coverage manual observation cannot achieve¹¹. This has created a pressing need for automated, objective, and continuous monitoring.

Sensor-based systems help but cannot capture the complex social behaviors central to welfare and face adoption barriers such as low farmer confidence^{12–15}. Computer vision is an attractive alternative—non-contact, lower-cost, and continuous^{16–18}—but farm environments are unusually challenging¹⁹: infrastructure and overlapping animals cause severe occlusion²⁰; lighting variation can degrade performance by 20–30%^{21,22}; mounting constraints force partial viewing angles²³; and group housing produces frequent identity switches^{24,25}.

Recent deep-learning models address individual stages of this problem. Open-vocabulary detectors such as OWLv2²⁶ enable zero-shot animal detection without species-specific training; tracking has progressed from SAM²⁷ to SAM2²⁸ and SAMURAI²⁹, which adds motion-aware memory and Kalman filtering; and self-supervised encoders such as DINOv2³⁰, together with the vision–language model CLIP³¹, provide strong features without task-specific labels. These components are typically evaluated in isolation or in controlled settings, and integrated, end-to-end pipelines for specific agricultural applications remain scarce^{32–34}.

To address this gap, we integrate these open-source models into a single modular pipeline with six stages: (1) optimized video decoding; (2) zero-shot object detection (OWLv2, or YOLOv12 where appropriate); (3) motion-aware segmentation and tracking with SAMURAI; (4) automated per-animal cropping; (5) feature extraction with DINOv2 or CLIP; and (6) flexible behavior classification. We validate the complete pipeline on individual-level pig behavior recognition in group housing, using top-mounted camera footage to limit occlusion despite its partial view^{20,23}.

2 Results

2.1 Validation on the Edinburgh Pig Behavior Video Dataset

2.1.1 Dataset Preparation

We decoded the 12 annotated video sequences using the stride specified on the dataset’s official website, following the procedure described in the Methods (Video Decoding and Frame Organization). This produced 600 frames per sequence, accumulating 7,200 labeled frames with 8 labeled pigs each, for a total of 57,600 individual pig annotations. The annotations were verified by overlaying the bounding boxes on the first frame, and two issues were discovered. First, in one sequence the initial annotations did not align with the objects and that sequence was therefore excluded from later benchmarking (Figure 1, left). Second, in two other sequences the starting frame contained pigs mounting one another in a corner, with some pigs not visible from the camera, and these sequences were also excluded (Figure 1, right). The remaining 9 sequences were used for thorough benchmarking, yielding 43,200 labeled individuals.

2.1.2 Object Detection Benchmarking

Visual comparison between YOLO and OWLv2 revealed that YOLO could not generate predictions for all the objects present in the frame without fine-tuning on the pig dataset, contrary to our earlier findings on the cow dataset. We therefore selected OWLv2, which performed reliably out of the box. The detailed comparison is provided in the Supplementary Material.

In our experiments, all pigs were detected at a confidence threshold of 0.5, which we adopted for validation on the decoded frames.

At the standard IoU threshold of 0.50, the model achieved an Average Precision (AP) of 89.28%, demonstrating robust detection capabilities. As shown in Table 1, the system maintained a favorable

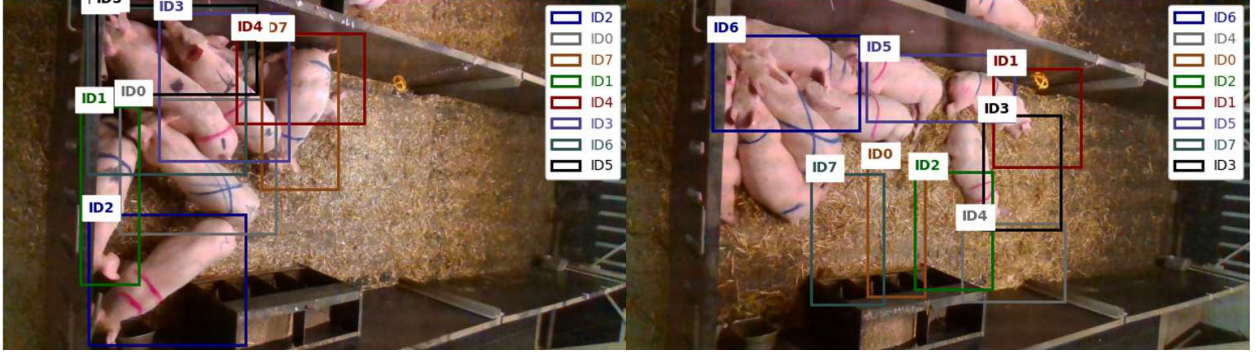


Figure 1. Examples of sequences excluded from benchmarking: (left) dataset showing misalignment between ground truth bounding boxes and actual pig locations, and (right) challenging tracking scenario with invisible objects on the first frame due to severe overlapping and mounting behaviors in the corner.

balance between precision (80.19%) and recall (88.05%), resulting in an F1 score of 83.94%. The true positive rate of 88.05% indicates that the model successfully detected the vast majority of pigs, while maintaining a relatively low false positive rate of 19.81%. The mean IoU of correctly matched detections was 0.747, suggesting accurate localization beyond the minimum threshold requirement.

The model’s counting accuracy yielded a Mean Absolute Error (MAE) of 1.53 pigs per frame, indicating reasonable performance in estimating the number of animals present in each image. Counting accuracy is a critical metric for livestock monitoring applications where accurate population assessment is essential.

Table 1. Object detection performance metrics for OWLv2 on the Edinburgh Pig Behavior dataset at IoU threshold 0.50.

Metric	Value
Average Precision (AP)	89.28%
Precision	80.19%
Recall	88.05%
F1 Score	83.94%
True Positive Rate	88.05%
False Positive Rate	19.81%
Missed Detection Rate	11.95%
Average IoU	0.747

2.1.3 Object Segmentation and Tracking Benchmarking

The proposed tracking system achieved an average Multiple Object Tracking Accuracy (MOTA) of 86.68%. The IDF1 score, which measures the system’s ability to correctly identify and maintain object identities throughout sequences, reached 93.33%. The system exhibited remarkable consistency in identity management, with an average of only 0.44 identity switches across all sequences. Track fragmentations averaged 83.89, with the number of tracklets matching the actual number of pigs in each sequence, while maintaining a perfect average tracklet length of 600 frames. This demonstrates the system’s ability to maintain continuous tracks throughout entire video sequences without interruption.

Individual sequence analysis revealed consistently high performance across diverse scenarios. The system achieved IDF1 scores ranging from 86.40% to 99.90% as shown in Table 2, with five sequences exceeding 90% identity preservation. The best performance was observed in sequence 2019_12_10_000060, which achieved 99.90% IDF1 and 99.80% MOTA with only 5 missed detections and no identity switches, as shown in Figure 2.



Figure 2. Visualization example of SAMURAI tracking performance demonstrating robust identity preservation during severe occlusion events. The green mask represents the object being tracked. Across different timestamps (time evolving from left to right), the mask identity of the pig stays the same even when the tracked object overlaps with other entities from the view of the camera.

Table 2. Multi-object tracking performance metrics across nine validation sequences from the Edinburgh Pig Behavior dataset.

Validation Sequence	IDF1	Recall	Precision	ML	Switches	MOTA	Avg. Length	Tracklets
2019_11_05_000002	91.00%	91.00%	91.00%	0	0	82.00%	600	8
2019_11_11_000028	87.40%	87.40%	87.40%	0	2	74.80%	600	8
2019_11_11_000036	91.60%	91.60%	91.60%	0	0	83.20%	600	8
2019_11_22_000010	86.40%	86.40%	86.40%	0	0	72.80%	600	8
2019_11_28_000113	98.30%	98.30%	98.30%	0	0	96.60%	600	8
2019_12_02_000005	94.50%	94.50%	94.50%	0	0	89.10%	600	8
2019_12_02_000208	98.10%	98.10%	98.10%	0	0	96.20%	600	8
2019_12_10_000060	99.90%	99.90%	99.90%	0	0	99.80%	600	8
2019_12_10_000078	92.80%	92.80%	92.80%	0	2	85.60%	600	8
Average	93.33%	93.33%	93.33%	0	0.44	86.68%	600	8

ML: Mostly Lost. Switches: Number of identity switches. Avg. Length: Average tracklet length (frames).

The tracking system’s precision and recall both averaged 93.33%, indicating balanced performance in detecting true positives while minimizing false detections. The consistency of these metrics across sequences suggests reliable performance suitable for automated behavioral monitoring applications. The complete elimination of tracklet fragmentation in the majority of sequences, with all animals tracked as “mostly tracked” or “partially tracked” and none classified as “mostly lost,” further validates the system’s effectiveness for continuous monitoring applications in precision pig farming.

2.1.4 Feature Extraction and MLP Classification Model Benchmarking

We first cropped pig regions from the background using the bounding box annotations, following the procedure described in the Methods (Automated Object Cropping), and then extracted high-dimensional visual embeddings from these cropped frames for use in behavior classification.

The feature extraction process used parallel processing across 12 workers to efficiently handle the computational demands of the DINOv2-large model. Because the dataset annotations propagate forward in time, we cropped each object using its current descriptor until a new descriptor superseded it, yielding a total of 43,200 unique behavioral instances across 16 distinct behavior categories. Each cropped pig image, standardized to 224×224 pixels, was processed through the pre-trained transformer to generate 1024-dimensional feature vectors. These embeddings captured rich visual representations suitable for distinguishing between behavioral patterns.

The distribution of behaviors in our dataset revealed significant class imbalance, with ‘investigating’ (10,281 instances) and ‘walk’ (2,766 instances) being the most frequent, while rare behaviors such as ‘chase’ (1 instance) and ‘jumpontopof’ (6 instances) had insufficient samples for reliable classification. To ensure robust model training and evaluation, we selected nine behaviors with adequate representation: standing (3,168), lying (3,187), eating (5,475), drinking (638), sitting (327), sleeping (15,256), running (90), playing with toy (126), and nose-to-nose interactions (431).

To visualize the learned feature space, we employed t-SNE dimensionality reduction on a subset of embeddings representing nine key behaviors: drinking, eating, lying, nose-to-nose, playing with toy, running, sitting, sleeping, and standing. The resulting visualization revealed distinct clustering patterns, with certain behaviors forming well-separated groups while others showed expected overlap due to visual similarities. For instance, stationary behaviors such as eating and drinking formed cohesive clusters, while dynamic behaviors like running and playing with toy exhibited more dispersed distributions in the feature space.

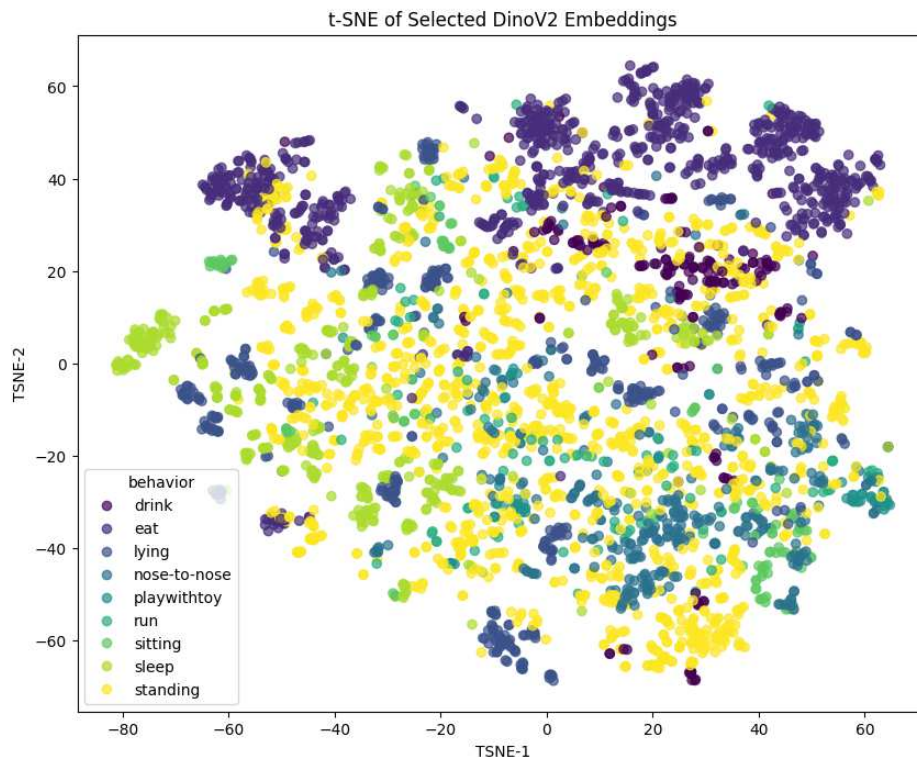


Figure 3. t-SNE visualization of DINOv2 embeddings revealing natural clustering of pig behaviors with distinct separation between stationary (eating, sleeping) and dynamic (running, playing) activities.

(1) MLP Classification Results We first evaluated a Multi-Layer Perceptron (MLP) classifier on the extracted DINOv2 features. After filtering to nine well-represented behaviors (standing, lying, eating, drinking, sitting, sleeping, running, playing with toy, and nose-to-nose interactions), we obtained 28,698 examples with a 70/15/15 train/validation/test split.

The MLP classifier achieved a test accuracy of 92.9%, as shown in Table 3. The model demonstrated high performance across most behavior categories, with eating achieving the highest F1-score (0.980), followed by sleeping (0.964) and lying (0.883). The system showed particularly strong recall for eating (99.6%), lying (96.2%), and nose-to-nose interactions (93.8%), indicating reliable detection when these behaviors occurred.

The confusion matrix in Table 3 also revealed that while standing behavior achieved high precision (89.2%), it showed moderate recall (76.2%), with misclassifications distributed across lying, eating, and nose-to-nose interactions. Dynamic behaviors like running presented challenges, achieving 64.3% recall but lower precision (47.4%), with confusion primarily occurring with standing and nose-to-nose interactions.

(2) LSTM Classification Results To better capture temporal dependencies in behavioral sequences, we implemented an LSTM-based classifier that processes sequences of DINOv2 embeddings. Using a sliding window approach with majority-based filtering, we generated 14,255 temporal windows from the original dataset.

The LSTM classifier achieved a test accuracy of 94.2%, representing a 1.3 percentage point improvement over the MLP baseline as shown in Table 4. This temporal model demonstrated superior performance in several key areas. Notably, the LSTM achieved more balanced precision-recall trade-offs across behaviors, with weighted average metrics of 94.9% precision, 94.3% recall, and 94.4% F1-score.

The LSTM showed improvements in challenging behavior categories. Standing behavior maintained high precision (90.7%) while achieving 78.4% recall, an improvement over the MLP’s 76.2%. The model demonstrated exceptional performance on eating (97.2% F1-score) and sleeping (97.8% F1-score). Social behaviors such as nose-to-nose interactions achieved 90.3% recall with the LSTM, though precision remained moderate at 51.7%; this suggests that temporal context helped identify these interactions, but distinguishing them from other close-proximity behaviors remained challenging.

Sitting behavior showed lower recall with the LSTM (76.0%) compared to the MLP (87.8%), and also lower precision, resulting in a lower overall F1-score (66.7% vs. 75.4%). This suggests the temporal model was more conservative in identifying sitting postures.

3 Discussion

3.1 Comparison with Former Research

3.1.1 Benchmarking Results Comparison

Our zero-shot OWLv2 detector reached an AP of 89.28%, 5.93 percentage points below the 95.21% AP reported by Bergamini et al.³⁵ with a fine-tuned YOLOv3 model; the gap is consistent with the difference between zero-shot and fine-tuned detection, and a direct comparison is limited because the original validation split was not disclosed. For tracking, our pipeline improved substantially over the same baseline: MOTA rose modestly from 84.88% to 86.68%, while IDF1 increased from 71.78% to 93.33% (a 21.55 percentage-point gain), identity switches fell from 16 to 0.44, and track fragmentations dropped from 115 to 83.89. To keep the comparison fair given our 600-frame

Table 3. Per-class classification performance of MLP model with DINOv2 features.

Behavior	Precision	Recall	F1-Score	Support
Standing	0.892	0.762	0.821	475
Lying	0.816	0.962	0.883	478
Eating	0.965	0.996	0.980	821
Drinking	0.860	0.896	0.878	96
Sitting	0.662	0.878	0.754	49
Sleeping	0.992	0.937	0.964	2289
Running	0.473	0.643	0.546	14
Playing with toy	0.900	0.947	0.923	19
Nose-to-nose	0.492	0.938	0.645	64
Weighted Average	0.940	0.929	0.932	4305

Table 4. Per-class classification performance of LSTM model with DINOv2 features.

Behavior	Precision	Recall	F1-Score	Support
Standing	0.907	0.784	0.841	236
Lying	0.901	0.958	0.929	238
Eating	0.985	0.958	0.972	409
Drinking	0.793	0.958	0.868	48
Sitting	0.594	0.760	0.667	25
Sleeping	0.983	0.973	0.978	1136
Running	0.400	0.571	0.471	7
Playing with toy	0.818	1.000	0.900	9
Nose-to-nose	0.517	0.903	0.700	31
Weighted Average	0.949	0.943	0.944	2139

sequences, we benchmarked against validation sequences A and D from Bergamini et al.³⁵, which have comparable tracklet lengths.

In behavior classification, the MLP reached 92.9% accuracy and the LSTM 94.2% on nine behaviors, against the 73% average reported by Bergamini et al.³⁵ on five behaviors—improvements of 19.9 and 21.2 percentage points, respectively, despite a more diverse behavior set. We attribute these gains to richer DINOv2 features relative to traditional CNN descriptors, high-quality individual crops from the preprocessing stage, and, for the LSTM, the temporal context that benefits dynamic and transitional behaviors (weighted F1 improved from 93.2% to 94.4%). That even the MLP exceeded the baseline indicates that pre-trained vision transformers encode much behavioral information without temporal modeling, while the LSTM’s consistent gains confirm the value of temporal context. These improvements were obtained specifically for pigs and would require validation before extending to other species or settings.

3.2 Practical Benefits of the Modular Design

The modular design provides three concrete benefits. First, processing-intensive operations can be optimized independently: the decoding and tracking modules include specialized memory management to handle long video sequences, while feature extraction employs parallel processing to maximize throughput. Resource-constrained environments can also implement only a subset of the pipeline—for example, deploying only the feature extraction and classification modules on pre-recorded video when real-time processing is not required. Second, errors in one module do not necessarily cascade through the system; if tracking temporarily fails due to occlusion, downstream stages can recover in subsequent frames. Third, individual modules can be replaced as conditions or technologies change, as demonstrated when we switched from YOLOv12 to OWLv2 for pig detection without redesigning the rest of the pipeline. This same property suggests potential for adaptation to other species and environments, though, as our YOLOv12 experience shows, each new context requires empirical validation rather than guaranteed transfer.

3.3 Limitations

Several limitations apply. First, the pipeline requires that animals in the initial frame are not severely overlapping, mounting, or out of view, since the first-frame detection seeds the entire downstream process, and it depends on strict adherence to the standardized data-flow conventions. Second, camera placement involves a trade-off: side views capture limb and postural detail useful for fine behavior classification but suffer heavy occlusion in groups, whereas the top view we used reduces occlusion and simplifies segmentation at the cost of limb information that can be important for some behaviors³⁶; multi-camera setups address both but add hardware, calibration, and computational cost. Third, higher resolution improves detection but raises computation quadratically, and real-time operation trades accuracy for speed. Finally, our experience of YOLOv12 failing on pigs while OWLv2 succeeded shows that even “zero-shot” model selection remains context-specific, so each new species or setting requires its own validation rather than guaranteed transfer.

3.4 Future Work

Future work spans both method and deployment. On the method side, multi-modal integration could augment visual features with audio, environmental sensors, or genomic data through fusion networks^{37–39}, though additional modalities can add redundancy and each requires evaluation^{34,40}; targeted fine-tuning of pretrained components such as SAM-2-Video is also possible, but is best done with a small, diverse annotation set while freezing most weights. On the deployment side, knowledge

distillation, quantization, and pruning could enable edge use, and streaming or distributed processing could support real-time analysis on standard farm cameras. Practical adoption also depends on integration with farm-management systems; usefully, the embeddings are storage-efficient—a typical 2 GB video reduces to roughly 0.05 GB while preserving behavioral information, easing sharing and meta-analysis²¹. The pipeline is released as open source (see Code availability) to support these directions and community contributions across new species and environments.

3.5 Conclusions

We presented a modular, open-source pipeline for individual-level livestock behavior analysis and validated it on pig monitoring in group housing. Integrating OWLv2 for detection, SAMURAI for tracking, and DINOv2 for feature extraction, the pipeline reached 94.2% accuracy on nine-class behavior recognition with a temporal model, alongside a 21.55 percentage-point improvement in identity preservation (IDF1) over the existing benchmark on the Edinburgh Pig Behavior Video Dataset.

The main contribution is a validated, reproducible integration of existing models rather than a new architecture. Its limitations—reliance on high-quality initial frames and context-specific model selection—underscore that validation on further species and environments is needed before broader generalization claims. By releasing the implementation, we provide a tested framework for pig behavior monitoring and a starting point for adaptation to other precision-livestock-farming contexts.

4 Methods

4.1 Datasets Description

We deployed our pipeline on two open-sourced datasets: (1) the CBVD-5 dataset from Li et al.⁴¹ and (2) the Edinburgh Pig Behavior Video Dataset from Bergamini et al.³⁵. As we only validated the feature extraction ability of the model on the CBVD-5 dataset, the details of that experiment are presented in the Supplementary Material.

4.1.1 Edinburgh Pig Behavior Video Dataset

The Edinburgh Pig Behavior Video Dataset from Bergamini et al.³⁵ represents a comprehensive video dataset specifically designed for automated pig behavior recognition and welfare monitoring research. The dataset captures focused video recordings from a nearly overhead perspective of a single pen housing with eight growing pigs. Videos were recorded in RGB color format at six frames per second with a resolution of 1280×720 pixels, providing sufficient temporal and spatial resolution for behavior analysis while maintaining manageable data volumes. The pen environment featured standard commercial pig farming infrastructure designed to reflect typical industry conditions. This included a three-space feeder and two nipple water drinkers, with flooring consisting of partially slatted surfaces supplemented with straw and shredded paper bedding. Such a realistic setup ensures the dataset’s relevance for developing automated livestock monitoring systems that can be deployed in an actual commercial farming setting. The ground truth annotations were meticulously prepared to provide comprehensive information for each visible pig in every labeled frame. These annotations included three key elements: axis-aligned bounding boxes that precisely delineate each animal’s spatial location; persistent tracking identifiers that maintain individual pig identity across consecutive frames; and behavior labels categorized into 17 distinct predefined classes. The dataset

comprises 7,200 annotated frames, with eight pigs tracked in each frame, providing approximately 57,600 individual pig annotations. This substantial volume of detailed annotations offered a robust foundation for training and evaluating computer vision algorithms across multiple tasks, including pig detection and localization, individual tracking in group housing environments, and behavior recognition at the individual level. The combination of detailed annotations and consistent labeling protocols made this dataset particularly suitable for thoroughly evaluating the comprehensive capabilities of our proposed pipeline.

For evaluation, nine video sequences (600 frames each) were utilized for tracking performance assessment and 800 frames were randomly sampled from the complete dataset for object detection benchmarking. This comprehensive annotation scheme enabled us to evaluate all components of our pipeline: detection, tracking, and behavior classification.

4.2 Computing Environment

Our experiments were conducted on a computing cluster featuring an NVIDIA V100 GPU (16 GB memory), 6 core CPUs, 112 GB RAM, and a local NVMe SSD array provided in Databricks. We implemented the pipeline using PyTorch 2.0, with additional libraries including OpenCV for video processing and scikit-learn for evaluation metrics. The selection of a GPU cluster is mainly due to the hardware requirements of the SAMURAI tracking model. For other sections in the pipeline, a CPU cluster would be efficient for all data manipulation.

4.3 System Architecture Overview

Our framework is designed as a modular pipeline that processes video input through six distinct stages: video decoding, object detection, segmentation and tracking, object cropping, feature extraction, and behavior classification, as shown in Figure 4. All those stages will be explained in the following sections.

The pipeline begins with raw video input from farm cameras, demonstrated here with overhead-mounted cameras in pig housing, and then goes through our framework:

1. The videos are first decoded into sequential frames with optimized sampling to balance temporal resolution with computational efficiency.
2. The decoded frames then undergo object detection using OWLv2 or YOLOv12 to identify and localize individual animals in the first frame.
3. Once initial detections are established, the SAMURAI model performs continuous tracking and segmentation throughout the video sequence, producing precise masks and bounding boxes for each animal across all frames.
4. The tracked objects are then cropped from their original frames, creating individual image sequences for each animal.
5. These cropped sequences undergo feature extraction using either DINOv2 or CLIP, generating rich embeddings that capture both visual appearance and behavioral patterns.
6. Finally, these embeddings serve as input to various classification architectures, from simple MLPs to sophisticated temporal models, depending on the specific behavior analysis requirements.

This modular design ensures flexibility and allows for easy replacement or upgrading of individual components as new technologies emerge.

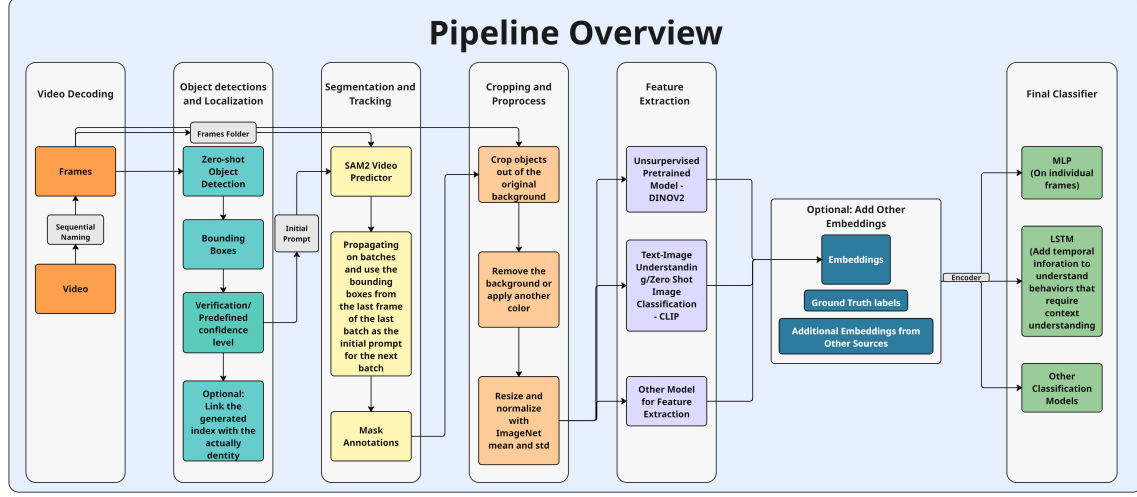


Figure 4. System architecture overview showing the complete workflow from video decoding to behavior classification: raw video is decoded into sequential frames, and objects are localized to seed segmentation and temporal tracking with the SAM2 video predictor. Each detected instance is then cropped (background removed or recolored), resized, and normalized before feature extraction using unsupervised pretrained models (e.g., DINOv2) or zero-shot classifiers (e.g., CLIP). Finally, per-frame embeddings are fed to an MLP for instantaneous behavior labels, while sequences of embeddings are passed through an LSTM (or other temporal model) to capture behavioral dynamics.

4.4 Video Decoding and Frame Organization

The video decoding process is critical for managing computational resources while maintaining sufficient temporal resolution for behavior analysis. Our approach implements several optimizations to handle the large data volumes typical in continuous farm monitoring.

Videos are processed using OpenCV’s efficient video capture capabilities, with configurable stride parameters to control frame sampling density. The stride value determines how many frames are skipped between saved frames, allowing us to adjust the temporal resolution based on the specific behaviors being analyzed. For instance, fast movements like jumping require higher temporal resolution (lower stride values), while behaviors like walking can be adequately captured with lower temporal resolution (higher stride values).

To prevent Graphics Processing Unit (GPU) memory overflow during the tracking phase, we organize decoded frames into subfolders with a maximum of 3,000 frames each. This limit was determined empirically through extensive testing on various GPU configurations. When tracking four objects simultaneously, we found that processing more than 5,000 frames would consistently cause memory issues on standard GPUs like the NVIDIA V100 with 16 GB memory.

A global sequential numbering scheme (e.g., 0000001.jpg, 0000002.jpg, ...) was employed for naming image frames, ensuring temporal continuity across subfolders. This is crucial for preserving the temporal relationships needed for behavior analysis, especially when behaviors span multiple subfolders.

4.5 Zero-Shot Object Detection

4.5.1 OWLv2

OWLv2²⁶ is an open-vocabulary object detector that supports zero-shot detection via text-based queries, making it well-suited to agricultural environments with variable object categories. Building on OWL-ViT (Open-World Localization Vision Transformer)⁴², OWLv2 introduces a self-training approach (OWL-ST) that uses pseudo-box annotations generated from over a billion web images and their associated text, enabling the model to learn at web scale without manual annotations. OWLv2 also introduces several architectural optimizations that further enhance efficiency (removing low-variance image patches) and improve instance selection using an object head, reducing computation by approximately 50% compared to the original OWL-ViT. After object detection, a two-stage filtering process was applied: (1) automatic filtering based on target object classes (e.g., “pig” in our implementation) and confidence thresholds defined according to the specific application, where objects not meeting these criteria were excluded; and (2) manual refinement through bounding box overlays (Supplementary Fig. S2), where the model’s output was examined visually. The evaluation employed an Intersection over Union (IoU) threshold of 0.5 to assess detection quality.

We also experimented with the YOLOv12 model⁴³. However, in our experiments with the Edinburgh Pig Dataset, OWLv2 proved more effective than YOLOv12 for detecting pigs in overhead views, demonstrating the importance of selecting appropriate models for specific applications.

4.6 Motion-Aware Segmentation and Tracking with SAMURAI

Segmentation and tracking is an essential step that generates sequential images of the same animal chronologically. SAM2, the successor of SAM, enables promptable video segmentation. SAMURAI builds upon the foundation of SAM2, introducing critical enhancements for video tracking in agricultural settings. Its core innovation is a motion-aware approach that integrates Kalman filter-based motion modeling with selective memory mechanisms.

Compared to the vanilla SAM2 video detector, SAMURAI demonstrates substantial performance improvements across multiple challenging benchmarks. On the LaSOT dataset⁴⁴, which encompasses 70 object categories including livestock, amphibians, reptiles, arthropods, and other mammals across more than 3.87 million frames, SAMURAI achieves up to 5.69% AUC gain and 6.53% P_{norm} gain, making it particularly well-suited for long-term single-object tracking applications. The model further exhibits a 7.1% AUC improvement on LaSOText⁴⁵, showcasing its robustness in handling diverse tracking scenarios. Additionally, SAMURAI achieves a 3.5% AO gain on the GOT-10k benchmark⁴⁶, which features 563 object classes and 87 distinct motion patterns, demonstrating its effectiveness in generic, class-agnostic tracking tasks. These consistent improvements across varied benchmarks establish SAMURAI as the current state-of-the-art video tracking model, offering superior performance for applications requiring robust object tracking in complex, real-world scenarios.

To handle long video sequences, a batch processing strategy with memory management was implemented. After each subfolder of frames was processed, the final bounding boxes were extracted to serve as initial prompts for the next batch. GPU memory was cleared between batches using garbage collection and PyTorch’s memory management functions to prevent the accumulation of memory fragmentation.

To evaluate tracking performance, the MOTA Challenge metric was used to comprehensively assess tracking accuracy, identity preservation, and trajectory continuity.

4.7 Automated Object Cropping

The cropping process isolates individual animals from their background, creating standardized inputs for feature extraction. This step is crucial for ensuring consistent feature quality regardless of the animal’s position or size in the original frame.

Our implementation uses parallel processing with `ProcessPoolExecutor`, a high-level Python API for running callables in a pool of separate processes rather than threads. Each cropping task processes a single frame–annotation pair through the following steps: (1) load the original frame using OpenCV; (2) extract the bounding box coordinates $[x, y, w, h]$ from the annotation; (3) create a binary mask using the contour information; (4) apply the mask to isolate the object from the background; (5) fill external regions with a specified background color (typically black or white); and (6) resize the cropped region to a standard dimension (e.g., 224×224 pixels).

The output filename convention preserves traceability: `[global_frame_index]_[object_name].jpg`. This naming scheme maintains the temporal sequence while identifying individual animals, essential for subsequent temporal analysis.

Parallel processing significantly improves throughput, with the number of concurrent workers configurable based on available CPU cores. However, we implement safeguards to prevent system overload by leaving 2–3 cores idle to avoid the disk I/O bottleneck, and we monitor system load to adjust the worker count dynamically when necessary.

4.8 Feature Extraction

Feature extraction was performed on cropped images to generate high-dimensional embeddings using two complementary approaches: DINOv2 for visual features and Contrastive Language-Image Pre-training (CLIP) for multimodal representations. Both approaches produce embeddings well-suited to downstream behavior analysis. The performance of CLIP versus DINOv2 was compared using the CBVD-5 dataset; details are provided in the Supplementary Material.

4.8.1 DINOv2 Architecture and Implementation

DINOv2³⁰ is a self-supervised vision model that learns powerful visual features without requiring labeled data. Starting with 1.2B uncured images, DINOv2 uses deduplication and self-supervised retrieval to create a curated dataset of 142M diverse images (LVD-142M). The model is trained efficiently using techniques like FlashAttention and sequence packing, with resolution adaptation in the final training phase.

For smaller models, DINOv2 employs knowledge distillation from the largest ViT-g model. The result is a family of models (ViT-S/B/L/g) that achieve state-of-the-art performance on various vision tasks without fine-tuning, making them excellent general-purpose visual encoders for both image-level and pixel-level applications.

Our implementation uses DINOv2-large, which offers an optimal balance between accuracy and computational efficiency. The model processes each cropped image through the following pipeline: (1) image preprocessing with standard normalization; (2) forward pass through the Vision Transformer; (3) extraction of the [CLS] token representation; (4) optional mean pooling over spatial dimensions; and (5) saving the resulting embedding as a `.pt` file.

The resulting embedding has shape (1, 1024), capturing rich, learned representations of the cropped images that can be fed into a behavior-classification model.

4.9 Behavior Classification

In order to translate the generated embeddings into behavior classifications of the nine different pig behaviors, two different deep learning models were applied. The model performance was compared using the evaluation metrics: precision, recall, F1-score, and support.

4.9.1 Basic MLP Classifier

For simple behavior classification tasks, we implement a straightforward Multi-Layer Perceptron (MLP) architecture to evaluate the pipeline’s performance even with basic classification models. The architecture consists of a three-layer feedforward neural network that processes the extracted feature embeddings. The input layer accepts the embedding vectors from our feature extraction stage, which are 1024-dimensional when using DINOv2-large. The first hidden layer reduces this dimensionality to 512 neurons using a fully connected transformation, followed by ReLU activation to introduce non-linearity and a dropout rate of 0.5 for regularization. The second hidden layer further compresses the representation to 256 neurons, maintaining the same ReLU activation and 0.5 dropout rate to prevent overfitting during training. Finally, the output layer maps these features to the number of target behavior classes through a linear transformation followed by softmax activation, producing normalized probability distributions over the possible behaviors.

4.9.2 Temporal Models

For behaviors requiring temporal context, we implement a Bidirectional LSTM (BiLSTM) architecture that captures both forward and backward temporal dependencies. The model consists of a single-layer bidirectional LSTM with 128 hidden units per direction, resulting in a 256-dimensional output when concatenating both forward and backward states. This BiLSTM layer processes sequences of 1024-dimensional DINOv2 embeddings extracted from consecutive frames, enabling the model to learn temporal patterns that are crucial for behavior recognition.

The architecture employs a classification head consisting of two fully connected layers. The first linear layer reduces the 256-dimensional BiLSTM output to 128 neurons, followed by a ReLU activation function and dropout regularization with a rate of 0.3 to prevent overfitting. The second linear layer maps these features to the final number of behavior classes. We extract the hidden state from the last timestep of the sequence as the comprehensive temporal representation, which encapsulates the accumulated behavioral patterns throughout the observed time window.

4.9.3 Model Training and Evaluation

The MLP classifier was trained using the backpropagation algorithm⁴⁷ with the Adam optimizer⁴⁸, employing a learning rate of 1×10^{-3} and weight decay of 1×10^{-5} for regularization. To address class imbalance in the dataset, we computed class weights using inverse frequency weighting, where each class weight was calculated as the inverse of its frequency normalized by the sum of all inverse frequencies. These weights were incorporated into the cross-entropy loss function to ensure balanced learning across all behavior categories.

The dataset was split using stratified sampling to maintain class distributions, with 70% for training, 15% for validation, and 15% for testing. Training proceeded for a maximum of 50 epochs with early stopping based on validation loss, using a patience of 10 epochs to prevent overfitting. The model achieving the lowest validation loss was saved and used for final evaluation. All experiments were conducted on NVIDIA GPUs with a batch size of 64 and 4 parallel data loading workers to optimize computational efficiency.

4.10 Modular Architecture Design

Our pipeline is designed with a modular architecture to address the inherent complexity and diversity of animal behavior analysis in agricultural settings. The modular approach follows principles established in software engineering⁴⁹ and computer vision systems⁵⁰ that promote component isolation, independent development, and flexible reconfiguration.

The six core modules (decoding, detection, tracking, cropping, feature extraction, and classification) are designed with well-defined interfaces that facilitate: (1) independent optimization, where each module can be separately refined or replaced without disrupting the entire pipeline; (2) flexible deployment, where different subsets of modules can be deployed based on specific research or application requirements; and (3) incremental improvement, where new techniques can be incorporated into specific modules as they become available.

Data availability

The datasets analysed during the current study are publicly available. The Edinburgh Pig Behavior Video Dataset is available from the source described by Bergamini et al.³⁵, and the CBVD-5 dataset from the source described by Li et al.⁴¹. No new datasets were generated in this study.

Code availability

The open-source implementation of the pipeline is openly available on Zenodo at <https://doi.org/10.5281/zenodo.20595459> (archived from the GitHub repository <https://github.com/Bovi-analytics/Individual-Behavior-Analysis-with-CV>, release V1) under the MIT License.

Ethics declaration

This study used previously published, publicly available video datasets and did not involve any new experiments on, or handling of, live animals by the authors. No animal ethics approval was therefore required.

Author contributions

H.Y. designed and implemented the study and drafted the first version of the manuscript. All other authors reviewed and edited the manuscript. M.H. supervised the project. All authors read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

- [1] A. Cornish, D. Raubenheimer, and P. McGreevy. What we know about the public’s level of concern for farm animal welfare in food production in developed countries. *Animals*, 6(11):74, 2016.
- [2] B. Clark, L. A. Panzone, G. B. Stewart, I. Kyriazakis, J. K. Niemi, T. Latvala, and L. J. Frewer. Consumer attitudes towards production diseases in intensive production systems. *PLoS One*, 14(1):e0210432, 2019.
- [3] D. Berckmans. Precision livestock farming technologies for welfare management in intensive livestock systems. *Revue Scientifique et Technique*, 33(1):189–196, 2014.
- [4] I. Halachmi, M. Guarino, J. Bewley, and M. Pastell. Smart animal agriculture: application of real-time sensors to improve animal well-being and production. *Annual Review of Animal Biosciences*, 7(1):403–425, 2019.
- [5] A. Bugeiro, R. Fouz, and F. J. Diéguez. Associations between on-farm welfare, milk production, and reproductive performance in dairy herds in northwestern Spain. *Journal of Applied Animal Welfare Science*, 24(1):29–38, 2021.
- [6] S. G. Matthews, A. L. Miller, J. Clapp, T. Plötz, and I. Kyriazakis. Early detection of health and welfare compromises through automated detection of behavioural changes in pigs. *The Veterinary Journal*, 217:43–51, 2016.
- [7] R. Antanaitis, K. Džermeikaitė, A. Bepalovaitė, I. Ribelytė, A. Rutkauskas, S. Japertas, and W. Baumgartner. Assessment of ruminating, eating, and locomotion behavior during heat stress in dairy cattle by using advanced technological monitoring. *Animals*, 13(18):2825, 2023.
- [8] K. Džermeikaitė, D. Bačėninaitė, and R. Antanaitis. Innovations in cattle farming: application of innovative technologies and sensors in the diagnosis of diseases. *Animals*, 13(5):780, 2023.
- [9] S. Neethirajan. The use of artificial intelligence in assessing affective states in livestock. *Frontiers in Veterinary Science*, 8:715261, 2021.
- [10] S. G. Matthews, A. L. Miller, T. Plötz, and I. Kyriazakis. Automated tracking to measure behavioural changes in pigs for health and welfare monitoring. *Scientific Reports*, 7(1):17582, 2017.
- [11] J. O. Hampton, D. I. MacKenzie, and D. M. Forsyth. How many to sample? statistical guidelines for monitoring animal welfare outcomes. *PLoS One*, 14(1):e0211417, 2019.
- [12] R. Firk, E. Stamer, W. Junge, and J. Krieter. Automation of oestrus detection in dairy cows: a review. *Livestock Production Science*, 75(3):219–232, 2002.
- [13] C. Rial, M. L. Stangaferro, M. J. Thomas, and J. O. Giordano. Effect of automated health monitoring based on rumination, activity, and milk yield alerts versus visual observation on herd health monitoring and performance outcomes. *Journal of Dairy Science*, 107(12):11576–11596, 2024.
- [14] E. A. Eckelkamp and J. M. Bewley. On-farm use of disease alerts generated by precision dairy technology. *Journal of Dairy Science*, 103(2):1566–1582, 2020.

- [15] A. H. Stygar, Y. Gómez, G. V. Berteselli, E. Dalla Costa, E. Canali, J. K. Niemi, and M. Pastell. A systematic review on commercially available and validated sensor technologies for welfare assessment of dairy cattle. *Frontiers in Veterinary Science*, 8:634338, 2021.
- [16] D. A. B. Oliveira, L. G. R. Pereira, T. Bresolin, R. E. P. Ferreira, and J. R. R. Dorea. A review of deep learning algorithms for computer vision systems in livestock. *Livestock Science*, 253: 104700, 2021.
- [17] Hongkun Tian, Tianhai Wang, Yadong Liu, Xi Qiao, and Yanzhou Li. Computer vision technology in agricultural automation—a review. *Information Processing in Agriculture*, 7(1): 1–19, 2020.
- [18] J. McDonagh, G. Tzimiropoulos, K. R. Slinger, Z. J. Huggett, P. M. Down, and M. J. Bell. Detecting dairy cow behavior using vision technology. *Agriculture*, 11(7):675, 2021.
- [19] G. L. Menezes, G. Mazon, R. E. Ferreira, V. E. Cabrera, and J. R. Dorea. Artificial intelligence for livestock: a narrative review of the applications of computer vision systems and large language models for animal farming. *Animal Frontiers*, 14(6):42–53, 2024.
- [20] G. Li, Y. Huang, Z. Chen, G. D. Chesser, J. L. Purswell, J. Linhoss, and Y. Zhao. Practices and applications of convolutional neural network-based computer vision systems in animal farming: A review. *Sensors*, 21(4):1492, 2021.
- [21] K. Wurtz, I. Camerlink, R. B. D’Eath, A. P. Fernández, T. Norton, J. Steibel, and J. Siegford. Recording behaviour of indoor-housed farm animals automatically using machine vision technology: A systematic review. *PLoS One*, 14(12):e0226669, 2019.
- [22] A. Fuentes, S. Han, M. F. Nasir, J. Park, S. Yoon, and D. S. Park. Multiview monitoring of individual cattle behavior based on action recognition in closed barns using deep learning. *Animals*, 13(12):2020, 2023.
- [23] E. T. Psota, T. Schmidt, B. Mote, and L. C. Pérez. Long-term tracking of group-housed livestock using keypoint detection and MAP estimation for individual animal identification. *Sensors*, 20(13):3670, 2020.
- [24] Q. Guo, Y. Sun, C. Orsini, J. E. Bolhuis, J. de Vlieg, P. Bijma, and P. H. de With. Enhanced camera-based individual pig detection and tracking for smart pig farms. *Computers and Electronics in Agriculture*, 211:108009, 2023.
- [25] M. Vidal, N. Wolf, B. Rosenberg, B. P. Harris, and A. Mathis. Perspectives on individual animal identification from biology and computer vision. *Integrative and Comparative Biology*, 61(3):900–916, 2021.
- [26] M. Minderer, A. Gritsenko, and N. Houlsby. Scaling open-vocabulary object detection. *Advances in Neural Information Processing Systems*, 36:72983–73007, 2023.
- [27] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.
- [28] N. Ravi, V. Gabeur, Y. T. Hu, R. Hu, C. Ryali, T. Ma, and C. Feichtenhofer. SAM 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

- [29] C. Y. Yang, H. W. Huang, W. Chai, Z. Jiang, and J. N. Hwang. SAMURAI: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. *arXiv preprint arXiv:2411.11922*, 2024.
- [30] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, and P. Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.
- [31] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.
- [32] A. Rohan, M. S. Rafaq, M. J. Hasan, F. Asghar, A. K. Bashir, and T. Dottorini. Application of deep learning for livestock behaviour recognition: A systematic literature review. *Computers and Electronics in Agriculture*, 224:109115, 2024.
- [33] M. Bonneau, J. A. Vayssade, W. Troupe, and R. Arquet. Outdoor animal tracking combining neural network and time-lapse cameras. *Computers and Electronics in Agriculture*, 168:105150, 2020.
- [34] Y. Liu, W. Li, X. Liu, Z. Li, and J. Yue. Deep learning in multiple animal tracking: A survey. *Computers and Electronics in Agriculture*, 224:109161, 2024.
- [35] L. Bergamini, S. Pini, A. Simoni, R. Vezzani, S. Calderara, R. B. Eath, and R. B. Fisher. Extracting accurate long-term behavior changes from a large pig dataset. In *16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, pages 524–533. SciTePress, 2021.
- [36] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge. DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. *Nature Neuroscience*, 21(9):1281–1289, 2018.
- [37] A. B. Alvarenga, H. R. Oliveira, S. Y. Chen, S. P. Miller, J. N. Marchant-Forde, L. Grigoletto, and L. F. Brito. A systematic review of genomic regions and candidate genes underlying behavioral traits in farmed mammals and their link with human disorders. *Animals*, 11(3):715, 2021.
- [38] X. Wang, Y. Wang, J. Yang, X. Jia, L. Li, W. Ding, and F. Y. Wang. The survey on multi-source data fusion in cyber-physical-social systems: Foundational infrastructure for industrial metaverses and industries 5.0. *Information Fusion*, page 102321, 2024.
- [39] R. Bokade, A. Navato, R. Ouyang, X. Jin, C. A. Chou, S. Ostadabbas, and A. V. Mueller. A cross-disciplinary comparison of multimodal data fusion approaches and applications: Accelerating learning through trans-disciplinary information sharing. *Expert Systems with Applications*, 165:113885, 2021.
- [40] R. Arablouei, Z. Wang, G. J. Bishop-Hurley, and J. Liu. Multimodal sensor data fusion for in-situ classification of animal behavior using accelerometry and GNSS data. *Smart Agricultural Technology*, 4:100163, 2023.
- [41] K. Li, D. Fan, H. Wu, and A. Zhao. A new dataset for video-based cow behavior recognition. *Scientific Reports*, 14(1):18702, 2024.

- [42] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, and N. Houlsby. Simple open-vocabulary object detection. In *European Conference on Computer Vision (ECCV)*, pages 728–755. Springer Nature Switzerland, 2022.
- [43] Y. Tian, Q. Ye, and D. Doermann. YOLOv12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*, 2025.
- [44] H. Fan, L. Lin, F. Yang, P. Chu, G. Deng, S. Yu, and H. Ling. LaSOT: A high-quality benchmark for large-scale single object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5374–5383, 2019.
- [45] H. Fan, H. Bai, L. Lin, F. Yang, P. Chu, G. Deng, and H. Ling. LaSOT: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129: 439–461, 2021.
- [46] L. Huang, X. Zhao, and K. Huang. GOT-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(5):1562–1577, 2019.
- [47] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986.
- [48] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.
- [49] L. Bass, P. C. Clements, and R. Kazman. *Software architecture in practice*. Addison-Wesley Professional, 4 edition, 2021.
- [50] C. Orhei, M. Mocofan, S. Vert, and R. Vasiu. End-to-end computer vision framework. In *2020 International Symposium on Electronics and Telecommunications (ISETC)*, pages 1–4. IEEE, 2020.