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

Authors: H. Yang¹, E. Liu¹, J. Sun¹, S. Sharma¹, M. van Leerdam¹, S. Franceschini², P. Niu¹, M. Hostens¹

Institution:

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

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

Abstract

Animal behavior analysis plays a crucial role in understanding animal welfare, health status, and productivity in agricultural settings. However, traditional manual observation methods are time-consuming, subjective, and limited in scalability. We present a modular pipeline that leverages open-sourced state-of-the-art computer vision techniques to automate animal behavior analysis in a group housing environment. Our approach combines state-of-the-art models for zero-shot object detection, motion-aware tracking and segmentation, and advanced feature extraction using vision transformers for robust behavior recognition. The pipeline addresses challenges including animal occlusions and group housing scenarios as demonstrated in indoor pig monitoring. We validated our system on the Edinburgh Pig Behavior Video Dataset for multiple behavioral tasks. Our temporal model achieved 94.2% overall accuracy, representing a 21.2 percentage point improvement over existing methods. The pipeline demonstrated robust tracking capabilities with 93.3% identity preservation score and 89.3% object detection precision. The modular design suggests potential for adaptation to other contexts, though further validation across species would be required. The open-source implementation provides a scalable solution for behavior monitoring, contributing to precision pig farming and welfare assessment through automated, objective, and continuous analysis.

Highlight:

- Modular pipeline achieves 94.2% accuracy in recognition of nine different behaviors in pigs
- Real-time tracking maintains 93.3% identity preservation in group housing
- Open-source pipeline processes video to behavioral classification end-to-end

Keywords:

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

1. Introduction

Welfare is an increasing concern in the livestock industry. Cornish et al. (2016) highlighted growing public awareness and Clark et al. (2019) documented increased regulations regarding farm animal conditions. In modern agricultural practice, measuring animal behavior has become increasingly important for ensuring optimal welfare, productivity, and health management (Berckmans et al., 2014; Halachmi et al., 2019). Animal behavior serves as a vital indicator of various physiological and psychological states (Bugueiro et al., 2021). Matthews et al. (2016) demonstrated that automated detection of behavioral changes in pigs could identify health and welfare compromises before clinical manifestation. Similarly, Antanaitis et al. (2023) showed how monitoring ruminating, eating, and locomotion behaviors using sensors could assess cattle responses to heat stress. Behavioral monitoring often provides early warning signs of disease, stress, or environmental discomfort before clinical symptoms become apparent. Džermeikaitė et al. (2023) discussed how continuous behavioral monitoring through artificial intelligence and machine learning enables early disease detection in cattle farming, while Matthews et al. (2016) found that changes in activity patterns and feeding behavior could predict health issues in pigs days before traditional clinical diagnosis.

However, as Neethirajan et al. (2021) noted, traditional monitoring methods, which rely heavily on human observation, are labor-intensive, prone to subjective interpretation, and limited in both temporal coverage and scalability. Similarly, Matthews et al. (2017) emphasized that continuous observation of livestock by farm staff is impractical in commercial settings to the degree required for detecting behavioral changes relevant for early intervention. Hampton et al. (2019) further indicated that sample sizes required for reasonable levels of precision in animal welfare assessments often exceed 300 animals, which is typically unfeasible for traditional observation-based methods due to the time, labor, and logistical constraints involved. These limitations have created a pressing need for automated, objective, and continuous monitoring solutions.

While sensor-based monitoring has shown benefits for health detection and improved outcomes (Firk et al., 2002; Rial et al., 2024), these systems face limitations including low farmer confidence and inability to assess complex social behaviors crucial for welfare assessment (Eckelkamp, 2020; Stygar et al., 2021). Computer vision and artificial intelligence have emerged as promising alternatives, offering advantages such as eliminating physical attachments, reducing labor demands, and providing continuous behavioral monitoring at lower costs (Oliveira et al., 2021; Tian et al., 2020; McDonagh et al., 2021). However, deploying computer vision in real farm environments presents unique challenges beyond typical applications (Menezes et al., 2024).

Traditional livestock farming presents several complex challenges for computer vision systems. First, farm infrastructure creates severe occlusions through fences, feeding equipment, and overlapping animals with similar appearances (Li et al., 2021). Second, dramatic lighting variations from natural and artificial sources can decrease model performance by 20-30% without adaptation (Wurtz et al., 2019; Fuentes et al., 2023). Third, camera mounting constraints often result in suboptimal viewing angles limited to partial top-down or side views (Psota et al., 2020). Fourth, group housing causes frequent

occlusions and identity switches, as demonstrated by Guo et al. (2023) who documented multiple bounding box switching events across 759 videos. Finally, individual-level identification must handle challenges like odd poses and appearance changes (Vidal et al., 2021).

Rohan et al. (2024) highlighted how recent advances in deep learning, particularly in object detection, tracking, and feature extraction, have opened new possibilities for addressing these challenges. Behavior detection in animals using video analysis involves a multi-step computational process. A video is treated as a sequence of consecutive images, where the first step is to locate the target animal in the initial frame. Once detected, the same animal must be tracked across subsequent frames to ensure consistency throughout the analysis. The animal is then cropped from each frame, removing redundant information such as the background and other animals. This series of cropped images forms the input for feature extraction, where visual information, such as shape, movement, or pixel intensity, is mathematically represented. These features serve as inputs to a classification model. Finally, this classification model is trained using an annotated dataset, also known as ground truth, enabling it to recognize key behaviors, such as distinguishing a lying cow from a standing one.

For object detection, open-vocabulary models like OWLv2 (Minderer et al., 2023) enable zero-shot detection of animals without requiring species-specific training, offering advantages over traditional architectures in agricultural settings. Video tracking has been transformed by the progression from SAM (Kirillov et al., 2023) to SAM 2 (Ravi et al., 2024) and most recently SAMURAI (Yang et al., 2024), which incorporates motion-aware memory and Kalman filtering for robust tracking in dynamic farm environments. Feature extraction has evolved from manual descriptors to self-supervised learning approaches, with DINOv2 (Oquab et al., 2023) learning high-quality visual features without annotations and CLIP (Radford et al., 2021) enabling zero-shot classification through vision-language alignment—particularly valuable when labeled agricultural data is scarce. These advances enable behavior classification models to achieve high accuracy, as demonstrated by Domun et al. (2019) who achieved 95% accuracy for pig behavior recognition, though challenges remain in handling group dynamics and individual tracking in complex farm settings.

Despite these advances, Rohan et al. (2024) observed that more recent deep learning approaches, while showing promise, typically focus on specific tasks or controlled environments, lacking the flexibility and robustness required for comprehensive behavior analysis in real-world agricultural settings. Bonneau et al. (2020) found that even advanced hybrid systems combining deep learning and time-lapse cameras for outdoor animal monitoring achieve sensitivity rates ranging from only 70.7% to 94.8%, with performance varying dramatically based on environmental factors. Liu et al. (2024) concluded that conventional animal tracking methods consistently fail to meet the precision and real-time speed requirements necessary for practical application due to persistent challenges including occlusion, complex backgrounds, and identification switches. Most critically, while individual components have shown promise in isolation, there remains a lack of integrated pipelines that combine these various techniques into a cohesive, end-to-end solution for specific agricultural applications. This motivates the

development of modular approaches that can generate usable feature representations on a frame level from videos, though such pipelines require validation for each specific use case.

To address these limitations, we propose a modular pipeline that integrates multiple open-sourced stateof-the-art computer vision techniques into a cohesive pipeline. Our pipeline consists of six main components: (1) optimized video decoding for efficient frame processing to overcome the temporal coverage limitations of traditional methods that Hampton et al. (2019) identified as requiring large sample sizes; (2) zero-shot object detection using OWLv2 (or YOLOv12 as needed) for initial animal localization that addresses the poor accuracy of conventional systems in farm environments highlighted by Fernandes et al. (2020); (3) motion-aware segmentation and tracking with SAMURAI model from Yang et al. (2024) for continuous individual monitoring to overcome the specific challenges of occlusion and animal overlapping documented by Guo et al. (2023); (4) automated object cropping for isolation of individual animals, which mitigates the identification difficulties in group housing described by Vidal et al. (2021); (5) feature extraction using DINOv2 or CLIP for robust representation learning that can handle the variable lighting conditions and performance decreases of 20-30% noted by Fuentes et al. (2023); and (6) flexible classification architectures for behavior recognition. Aside from the pipeline architecture itself, we tried camera footage from top-mounted cameras as a potential solution to mitigate the farm's intricate layouts mentioned by Li et al. (2021), even though they only include partial information (Psota et al., 2020).

In short, we demonstrate how open-sourced state-of-the-art algorithms can be integrated into a modular pipeline for individual-level behavior analysis, validated specifically on pig behavior recognition in group housing environments.

2. Materials and Methods

2.1 Datasets Description

We deployed our pipeline on two open-sourced datasets: 1) CBVD-5 dataset from Li et al. (2024) and 2) The Edinburgh Pig Behavior Video Dataset from Bergamini et al. (2021). As we only validated the feature extraction ability of the model on the CBVD-5 dataset, the details of that experiment will be shared in the supplementary.

2.1.1 Edinburgh Pig Behavior Video Dataset

The Edinburgh Pig Behavior Video Dataset from Bergamini et al. (2021) represented 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.

2.2 Computing Environment

Our experiments were conducted on a computing cluster featuring NVIDIA V100 GPU (16GB 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.

2.3 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 Fig. 1. 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.

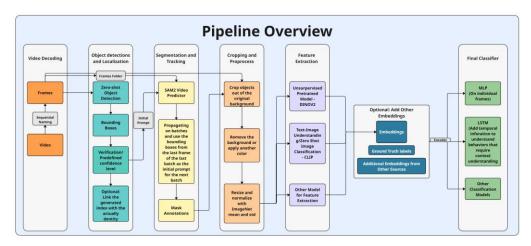


Fig.1. System architecture overview showing the complete workflow from video decoding to behavior classification: raw video is decoded into sequential frames, and objects are localized to seed segmentation and temporal tracking with the SAM2 video predictor. Each detected instance is then cropped (background 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.

2.4 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 16GB 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.

2.5 Zero-Shot Object Detection

2.5.1 OWLv2

OWLv2 (Minderer et al., 2023) 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) (Minderer et al., 2022), OWLv2 introduces a self-training approach (OWL-ST), using pseudo-box annotations generated from over a billion web images using their associated text, enabling the model to learn at web scale without manual annotations. Also, OWLv2 introduces several architectural optimizations that further enhance efficiency (removing low-variance image patches) and instance selection using an object head, reducing computation by ~50% compared to the original OWL-ViT. After the object detection with one of the two models, 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 automatically excluded; and (2) manual refinement through bounding box overlays (Fig. 6), where the output from the object detection model was examined visually. The evaluation employed an Intersection over Union (IoU) threshold of 0.5 to assess detection quality at varying levels of localization precision.

We also experimented with YOLOv12 model (Tian et al., 2025). But 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.

2.6 Motion-Aware Segmentation and Tracking with SAMURAI

Segmentation and tracking is an essential step that generates sequential images of the same animal chronologically. SAM 2, the successor of SAM makes 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 dataset (Fan et al., 2019), 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 LaSOT_{ext} (Fan et al., 2021), showcasing its robustness in handling diverse tracking scenarios. Additionally, SAMURAI achieves a 3.5% AO gain on the GOT-10k benchmark (Huang et al., 2019), 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 fragments.

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

2.7 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); 6) Resize the cropped region to a standard dimension (e.g., 224x224 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, monitoring network I/O and adjusting worker count dynamically if necessary.

2.8 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 Pretraining (CLIP) for multimodal representations, making them suitable for behavior analysis. The performance of CLIP versus DINOv2 was compared using the CBVD-5 dataset, the result of which can be found in the supplementary.

2.8.1 DINOv2 Architecture and Implementation

DINOv2 (Oquab et al., 2023) is a self-supervised vision model that learns powerful visual features without requiring labeled data. Starting with 1.2B uncurated 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; 5) Saving the resulting embedding as a .pt file.

The resulting embedding has shape (1, 2024), capturing rich, learned representations of the cropped images that can be fed into a behavior-classification model.

2.9 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.

2.9.1 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.

2.9.2 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.

2.9.3 Model Training and Evaluation

The MLP classifier was trained using the backpropagation algorithm (Rumelhart et al., 1986) with the Adam optimizer (Kingma & Ba, 2014), 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.

2.10 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 engineering (Bass et al., 2021) and computer vision systems (Orhei et al., 2020) 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: Each module can be separately refined or replaced without disrupting the entire pipeline; 2) Flexible deployment: Different subsets of modules can be deployed based on specific research or application requirements; 3) Incremental improvement: New techniques can be incorporated into specific modules as they become available.

3. Results

3.3 Validation on The Edinburgh Pig Behavior Video

3.3.1 Dataset Preparation

We decoded the 12 sequences of videos that were annotated using the stride specified on the official website of the dataset, following the procedure of 2.2.2. 600 frames are generated for each sequence and thereby accumulated 7200 labeled frames, each with 8 labeled pigs, meaning we have 57600 labeled individuals from the annotations. The annotations were verified by overlaying the bounding boxes on the first frame and it was discovered that: for one sequence, the initial annotations do not align with the objects and therefore are not used in the later benchmarking as shown in Figure 3. And for two other sequences, the starting frame has the targeted objects mounted on each other in one corner, and some objects are not visible from the camera and are therefore also not used. And the rest of the 9 sequences were used for the throughout benchmarking, resulting in a total of 43,200 labeled individuals from the annotations.

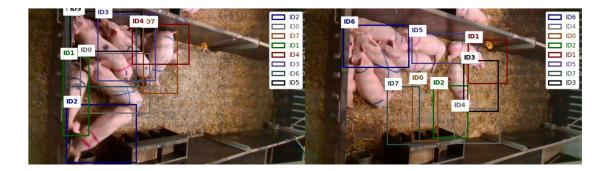


Fig. 2. Examples of sequences excluded from benchmarking: (left) dataset showing misalignment between ground truth bounding boxes and actual pig locations, and (right) challenging tracking scenario with invisible objects on the first frame due to severe overlapping and mounting behaviors in the corner.

3.3.2 Object Detection Benchmarking

Comparing YOLO and OWLv2 visually revealed that YOLO could generate predictions for all the objects present in the frame without fine-tuning on the pig dataset, contradictory to our earlier findings on the cow dataset. The OWLV2 model was working and therefore selected. The comparisons are revealed in the supplementary.

Based on our experiment, at the confidence level of 0.5, all the pigs were detected. This threshold was used to validate on the decoded frame.

At the standard IoU threshold of 0.50, the model achieved an Average Precision (AP) of 89.28%, demonstrating robust detection capabilities. As we can see from Table 1, the system maintained a favorable 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 in the dataset, 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. This counting error represents a critical metric for livestock monitoring applications where accurate population assessment is essential.

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

Metrics	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	

3.3.3 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 3.



Fig. 3. 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.

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.

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

Deliavioi dataset								
Validation Sequences	Idf1	Recall	Precision	Mostly Lost	Num Switches	Mota	Avg. Tracklet Length	Num 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	0.87%	600	8

3.3.4 Feature Extraction and MLP Classification Model Benchmarking

We first processed cropped pig regions from the background as we designed in 2.2.5 using the bounding boxes annotation and then extracted high-dimensional visual embeddings on these cropped frames, which were subsequently used for behavior classification.

The feature extraction process utilized parallel processing across 12 workers to efficiently handle the computational demands of the DinoV2-large model. Due to the forward propagation nature of our annotations, we crop the objects forward with the descriptors until a new descriptor overrides the last

one, resulting in 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 different 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.

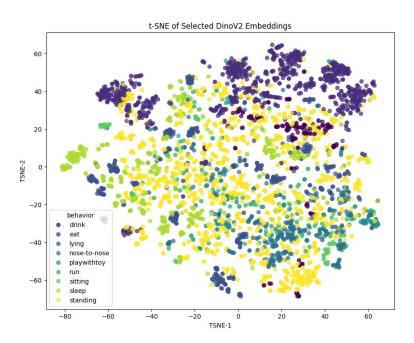


Fig. 4. t-SNE visualization of DINOv2 embeddings revealing natural clustering of pig behaviors with distinct separation between stationary (eating, sleeping) and dynamic (running, playing) activities

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 we can see from Table 3. The model demonstrated high performance across most behavior categories, with eating behaviors achieving the highest F1-score (0.980), followed by sleep (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 we can see from 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 from the MLP's 76.2%. The model demonstrated exceptional performance on eating (97.2% F1-score) and sleeping behaviors (97.8% F1-score). Social behaviors like nose-to-nose interactions achieved 90.3% recall with the LSTM, though precision remained moderate at 57.1%, suggesting the 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.

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	

4. Discussion:

4.1 Comparison with Former Research:

4.1.1 Benchmarking Results Comparison:

Our object detection results using OWLv2 achieved an AP of 89.28%, which is 5.93 percentage points lower than the 95.21% AP reported by Bergamini et al. (2021) using a fine-tuned YOLOv3 model. However, direct comparison is challenging as the details about which sequences were chosen as the validation set were not disclosed in the original study. The difference in performance may be attributed to the zero-shot nature of our approach versus their fine-tuned model, as well as potential differences in validation set composition.

For tracking performance, our pipeline demonstrated substantial improvements over the baseline model reported by Bergamini et al. (2021). The MOTA increased from 84.88% to 86.68%, representing a modest but meaningful improvement. More significantly, the IDF1 score improved from 71.78% to 93.33%, a 21.55 percentage point increase that indicates superior identity preservation capabilities. The reduction in identity switches from 16 to 0.44 (97.2% reduction) and decrease in track fragmentations from 115 to 83.89 demonstrate the effectiveness of our approach. To ensure a fair comparison given the 600-frame constraint of our annotated sequences, we specifically benchmarked our results against validation sequences A and D from Bergamini et al. (2021), which exhibit comparable average tracklet lengths.

In behavior classification, our framework demonstrated substantial improvements over previous approaches. The MLP classifier achieved 92.9% accuracy on nine behavior categories, while the LSTM classifier further improved performance to 94.2%. Both models significantly outperformed the 73% average accuracy reported by Bergamini et al. (2021) on five behaviors (standing, lying, moving, eating, and drinking).

This represents a 19.9 percentage point improvement for MLP and 21.2 percentage point improvement for LSTM over the baseline, despite evaluating on a more diverse and challenging set of nine behaviors. The superior performance can be attributed to several factors: (1) the use of modern vision transformer features (DINOv2) that capture richer visual representations compared to traditional CNN features, (2) the effectiveness of our preprocessing pipeline that ensures high-quality individual animal crops, and (3) for the LSTM model, the incorporation of temporal context that captures behavioral dynamics.

The LSTM's advantage over MLP was particularly evident in behaviors with temporal characteristics. While both models achieved similar performance on static behaviors like lying (MLP: 88.3% F1, LSTM: 92.9% F1), the LSTM showed marked improvements for dynamic and transitional behaviors. The temporal modeling also improved the precision-recall balance, as evidenced by the weighted average F1-score improvement from 93.2% (MLP) to 94.4% (LSTM).

Notably, even our simpler MLP architecture substantially exceeded the original baseline, suggesting that modern pre-trained vision transformers like DINOv2 can effectively encode behavioral information without requiring complex temporal modeling for many applications. However, LSTM's consistent improvements across most behavior categories validate the importance of temporal context for comprehensive behavior analysis.

These results demonstrate that our modular pipeline, combining state-of-the-art vision models with appropriate architectural choices, can achieve high accuracy in automated behavior classification while handling increased behavioral complexity. These improvements were achieved specifically on pig behavior analysis, and similar gains would need to be validated for other species or agricultural contexts.

4.1.2 Pipelines Benefits:

The modular design of our pipeline provides several concrete benefits for both researchers and practitioners:

- 1) Computational efficiency: Processing-intensive operations can be optimized independently. For example, the decoding and tracking modules include specialized memory management to handle long video sequences, while feature extraction employs parallel processing to maximize throughput.
- 2) Error isolation: Problems in one module do not necessarily compromise the entire pipeline. If tracking temporarily fails due to occlusion, the system can recover in subsequent frames without cascading errors through the entire system.
- 3) Adaptation to environmental conditions: Different farm environments may require different configurations. Low-light conditions might benefit from enhanced detection models, while crowded scenes may require specialized tracking approaches. Our modular design allows these adaptations as demonstrated when we switched from YOLOv12 to OWLv2 for pig detection without redesigning the entire system.

- **4) Potential for adaptation:** The modular design theoretically facilitates adaptation to other contexts. By replacing individual modules while maintaining the core pipeline architecture, the system could potentially be adapted for other applications, though this would require validation for each new use case. Our experience switching from YOLOv12 to OWLv2 demonstrates that even state-of-the-art models may require species-specific selection.
- **5) Incremental deployment:** Resource-constrained environments can implement a subset of the pipeline. For example, if real-time processing is not required, users can deploy only the feature extraction and classification modules on pre-recorded video.

4.2 Limitations

4.2.1 High-quality Start Point and Delicate Dataflow

The pipeline requires that animals in the first frame are not severely overlapping, mounting, or missing, as these conditions prevent successful initial detection and compromise the entire downstream process. Additionally, the modular design necessitates strict adherence to standardized data flow conventions, which users must follow precisely for proper functionality.

4.2.2 Camera Positioning Trade-offs

Camera positioning represents a critical design decision with inherent trade-offs that significantly impact system performance. Side-view camera configurations provide superior visibility of limb movements and postural details, enabling more nuanced behavioral classification; however, they suffer from frequent occlusions when monitoring multiple animals, potentially compromising tracking continuity. Conversely, top-view camera installations substantially reduce inter-animal occlusions and simplify instance segmentation but significantly limit access to limb information that Mathis et al. (2018) demonstrated to be essential for accurate behavior detection and classification. Multi-camera systems that integrate both perspectives offer comprehensive coverage and redundancy, theoretically overcoming the limitations of single-view approaches; however, this solution introduces substantially higher system complexity in terms of hardware requirements, calibration procedures, and computational demands for data fusion, alongside proportionally increased implementation costs that may limit practical deployment in commercial agricultural settings.

4.2.3 Computational Trade-offs

Higher image resolutions improve detection accuracy but incur quadratic increases in computational demands. Real-time processing, while valuable for immediate interventions, requires architectural compromises that may reduce accuracy compared to offline analysis. These trade-offs necessitate careful optimization based on specific deployment requirements and available resources.

4.2.4 Species-Specific Validation Requirements

Our experience with YOLOv12 failing on pig detection while OWLv2 succeeded highlights that model selection remains species-specific. Despite using state-of-the-art "zero-shot" models, each new

application context requires careful validation and potentially different model choices, limiting immediate generalizability.

4.3 Future

4.3.1 Multi-Modal Integration

Our current pipeline relies solely on visual information extracted through DINOv2 embeddings. Future development could explore incorporating complementary data modalities to enhance behavioral analysis accuracy, including audio analysis for vocalizations that lack distinct visual signatures, environmental sensors for context-aware behavioral adaptation analysis, and genomic data for individual-specific modeling (Alvarenga et al., 2021). These diverse data streams could be integrated through hierarchical fusion networks with specialized branches (Wang et al., 2024) and weight-aware modules that dynamically adjust each modality's importance (Bokade et al., 2021). While probability-based fusion may yield superior results compared to simple feature concatenation (Arablouei et al., 2023), multimodal integration must be approached judiciously, as additional modalities may introduce redundant information that diminishes model effectiveness (Liu et al., 2024), necessitating empirical evaluation of each new data source's specific contribution.

4.3.2 Potential Applications to Other Species

While our pipeline's modular architecture theoretically facilitates adaptation to other species and contexts, such extensions would require careful validation. The zero-shot capabilities of models like OWLv2 and the general-purpose design of SAMURAI suggest potential applicability beyond pig monitoring. However, our experience with YOLOv12's failure on pig detection demonstrates that even state-of-theart models may require context-specific selection and adaptation. Future work should systematically evaluate the pipeline's performance across different species, housing conditions, and camera configurations.

4.3.3 Optimization for Practical Deployment

Current computational requirements limit deployment in resource-constrained farm environments. Future optimization strategies could include knowledge distillation to create compact models, post-training quantization for edge devices, and structured pruning to reduce memory footprint. Additionally, implementing streaming pipelines and distributed processing architectures would enable real-time analysis on standard farm surveillance systems. The goal is to make sophisticated behavioral analysis accessible without requiring extensive computational resources or technical expertise.

4.3.4 Integration with existing systems

The modular design facilitates integration with current farm management infrastructure. Classification outputs can be formatted for compatibility with common farm management platforms, enabling automated data flow without manual intervention. For research applications, the system supports export formats compatible with animal behavior databases, facilitating data sharing and meta-analyses across studies (Wurtz et al., 2019). The feature embeddings generated by our pipeline offer particularly efficient

storage—a typical 2GB video can be compressed to approximately 0.05GB of numerical representations while preserving behavioral information.

4.3.5 Fine-tuning Pretrained Tracking Model

Each component of our pipeline can, in principle, be fine-tuned to meet a specific research objective; however, indiscriminate fine-tuning can degrade the model's overall performance. Consider SAM-2-Video, which was pre-trained on approximately 35.5 million mask instances—far beyond what most labs can realistically produce. Successful adaptation therefore hinges on two coordinated strategies: first, curating a compact yet high-quality and diverse set of mask annotations, and second, freezing the majority of model weights to safeguard the knowledge acquired during large-scale pre-training. Observing these constraints allows the model to assimilate new domain-specific cues without erasing the strengths it has already learned.

4.3.6 Open-source nature and Implementation Guidelines

Our pipeline will be released as an open-source project on GitHub to encourage community contributions and adaptations. Configuration management through standardized files specifies essential parameters including memory limits, batch sizes, and model paths, enabling users to adapt the system to their hardware constraints without code modifications. Consistent naming conventions and directory structures ensure seamless data transfer between modules. Comprehensive documentation and example configurations for common scenarios reduce barriers to adoption while maintaining reproducibility across deployments. We encourage the research community to contribute improvements, report performance on new species or environments, and share adaptations that could benefit others working in animal behavior analysis.

5. Conclusions and Perspectives

We have presented a modular pipeline for automated behavior analysis validated on pig monitoring in group housing environments. By integrating state-of-the-art deep learning techniques including OWLv2 for detection, SAMURAI for tracking, and DINOv2 for feature extraction, our pipeline achieved 94.2% accuracy on nine-class pig behavior recognition using temporal models.

Our experiments on the Edinburgh Pig Behavior Video Dataset demonstrated substantial improvements over existing methods, including a 21.2 percentage point increase in classification accuracy and a 21.55 percentage point improvement in identity preservation (IDF1) during tracking. The modular architecture enabled us to adapt to specific challenges—such as switching from YOLOv12 to OWLv2 when the former failed on pig detection—highlighting both the flexibility of the design and the need for empirical validation in each application context.

While our results on pig behavior analysis are promising, several limitations must be acknowledged. The pipeline requires high-quality initial frames for successful detection, and even state-of-the-art "zero-shot"

models required species-specific selection. These findings underscore that validation on additional species and environments would be necessary before broader claims about generalizability can be made.

This work demonstrates how existing computer vision models can be effectively integrated for livestock behavior analysis when properly validated and configured. The open-source implementation provides researchers with a tested framework for pig behavior monitoring and a potential starting point for adaptation to other contexts. As precision livestock farming continues to evolve, such automated monitoring tools—when properly validated for each specific application—can contribute to improved animal welfare assessment and management decisions.

6. Supplementary

6.1 Considerations and Trials for Deciding Model

In order to identify the best model, we conducted some evaluations of several open - source models. Two distinct video datasets were used: a proprietary recording of our own dairy cows and the publicly available Edinburgh Pig Behavior Video (Bergamini et al., 2021).

6.1.1 Object Detection and Localization

Aside from the OWLv2 model, we also experimented with the YOLOv12 model. YOLOv12 represents a significant advancement in the YOLO family by integrating an innovative area attention (A2) mechanism that overcomes the computational limitations of traditional attention approaches, which typically scales quadratically with input size and is impractical for high-resolution agricultural imagery (Tian et al., 2025). By dividing feature maps into distinct areas and computing attention locally, A2 reduces complexity to linear time, enabling efficient processing of large images while preserving global context. This attention-centric, real-time object detection framework outperforms previous YOLO versions and other detectors across all model scales (N, S, M, L, X), with YOLOv12-L achieving a state-of-the-art 53.7% mAP which is 0.4% higher than YOLOv11-L, demonstrated through superior object localization and background suppression as shown in Fig. 2.

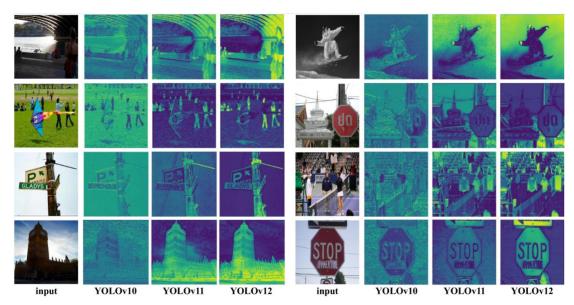


Fig. 5. Comparative visualization of attention heat maps across YOLO versions demonstrating YOLOv12's superior object localization and background suppression (Tian et al., 2025). YOLOv12 exhibits better ability at extracting the contours of the same images.

In our implementation, YOLOv12 performed zero-shot detection on initial video frames within approximately 15 seconds, generating bounding boxes for all detected objects; however, filtering is often necessary to exclude non-target items (like water troughs or feeding equipment commonly present in environments such as dairy barns).



Fig. 6. Bounding boxes overlay with predictions from YOLOv12(Right) showed the successful detections of cows within the frame and OWLv2 (Minderer et al., 2023) detection output demonstrated false positive generation at default confidence settings.

We have experimented with other zero-shot object detection and localization models like OWLv2 (Minderer et al., 2023), but the performance is not as good as YOLO shown in Fig. 6. There are more false predictions and require more adjustments of the confidence level to find the useful predictions.

However, the performance of the two models varies when applied to the Edinburgh Pig Behavior Video from Bergamini et al. (2021).

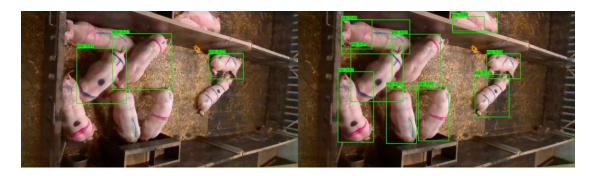


Fig. 7. Detection results on overhead pig housing footage showing YOLOv12's underdetection and OWLv2's complete coverage at 0.5 confidence threshold. YOLOv12 detected three of the eight pigs visible in the frame, demonstrating substantial underdetection. In contrast, OWLv2 achieved complete coverage by identifying all eight target pigs and even two additional, highly occluded pigs from an adjacent pen.

6.1.2 Segmentation and Tracking

We experimented with other video segmentation and tracking models like Xmem (Cheng et al., 2022) but the performances are not as ideal and may struggle to deploy in complicated barns.



Fig. 8. Comparative segmentation performance on occluded cattle showing SAMURAI's precise instance separation (in the middle) versus XMem's mask merging (on the right).

With SAM2 video predictor (in Fig. 8 on the left), we could see the clear separation of the cows, while the Xmem (Cheng et al., 2022) video predictor generated one mask over three overlapping cows, failing to make precise mask predictions in challenging tasks.

6.1.3 Choice of Individual Modules:

For the object detection and localization component, initial experiments with YOLOv12 demonstrated promising efficiency and accuracy characteristics (see Figure 7); however, the model failed to generate reliable predictions when applied to pig detection tasks. Consequently, we adopted OWLv2 as our primary object detection and localization model. OWLv2 exhibited robust performance in our pig detection experiments, demonstrating its suitability for this specific agricultural application.

The selection of the segmentation and tracking module was based on comparative evaluation between SAMURAI and XMem models. SAMURAI demonstrated superior segmentation capabilities, producing more precise object boundaries and maintaining clearer separation between individual animals, particularly in challenging scenarios involving overlapping subjects (Figure 8). The model's ability to

preserve distinct contours even during close animal interactions proved critical for accurate individual tracking. These performance advantages led to the selection of SAMURAI as the segmentation and tracking component within our pipeline.

For the feature extraction backbone, we evaluated both DINOv2 and CLIP models to determine their suitability for visual behavior analysis. As the current application does not require cross-modal textimage understanding, and given that DINOv2 demonstrated marginally superior performance in visual feature discrimination (Figure 9), we selected DINOv2 as our feature extraction model. The t-SNE visualizations revealed more distinct behavioral clusters with DINOv2 embeddings, suggesting enhanced discriminative power for behavior classification tasks.

6.2 Validation on CBVD-5

6.2.1 CBVD-5 dataset description

The CBVD-5 dataset from Li et al. (2024) served as a collection of dairy cow behavior keyframes with corresponding annotations, enabling us to evaluate the cross-species generalizability of our framework. The dataset originates from continuous video surveillance of dairy cows' daily activities on a commercial farm, with cameras operating 24 hours per day over a five-day period. This extensive monitoring period ensured comprehensive coverage of various behaviors across different times of day and environmental conditions.

From continuous video footage, keyframes were systematically selected for behavior annotation, resulting in 27,501 valid labeled data points. Each annotation contained precise spatial information about individual cow locations paired with one of five behavioral categories: standing, lying down, feeding, drinking, and rumination, capturing the primary activities essential for monitoring dairy cow welfare and productivity.

Given that the dataset provided sparse annotations across multiple cows without persistent individual identifiers, it presents different evaluation opportunities compared to the pig dataset. Consequently, we do not utilize this dataset for benchmarking our object detection, localization, tracking, and segmentation modules, which require consistent individual identification across frames. Instead, the primary evaluation focus centers on assessing the feature extraction capabilities of our framework and the classifier's ability to differentiate between behavioral patterns based on these extracted features.

For our evaluation, we concentrated on the standing and lying down categories, which gives us a basic touch on the feature extraction capability of our pipeline. This decision was motivated by two factors: the subtle nature of rumination behavior requires exceptionally high-quality video for accurate assessment, and the feeding and drinking classes exhibited significant sample imbalance that could bias the evaluation results. After filtering these two primary behaviors, we obtained 30,391 individual cropped samples suitable for analysis, providing a substantial dataset for evaluating behavioral classification performance in dairy cattle.

6.2.2 Dataset Preparation

We applied our automated cropping module (Section 3.5) to extract individual animal images from the CBVD-5 dataset annotations using the bounding boxes provided along with the pictures. The existing ground truth bounding boxes were utilized directly.

6.2.3 CLIP Implementation

The second model selected for the feature extraction was CLIP (Radford et al., 2021). CLIP is a neural network model that learns to understand both images and text by training on a large dataset of image-text pairs. The model consists of two main components: an image encoder that processes images and a text encoder that processes text descriptions. During training, CLIP learns to map images and their corresponding text descriptions to similar points in a shared embedding space, while pushing unrelated image-text pairs apart.

The training process uses a contrastive learning approach with a symmetric cross-entropy loss function. For a batch of image-text pairs, CLIP computes the cosine similarity between all possible combinations of images and texts: $S_{ij} = \frac{I_i T_j}{\|I_i\| \cdot \|T_j\|}$

These similarities are scaled with a temperature parameter,

$$L_{ij} = \tau \cdot S_{ij}$$

And then CLIP optimized the model to maximize the similarity scores for matching pairs while minimizing scores for non-matching pairs.

Image-to-text loss:

$$\mathcal{L}_{i} = -\log \frac{\exp(L_{ij})}{\sum_{j=1}^{N} \exp(L_{ij})}$$

Text-to-image loss:

$$\mathcal{L}_t = -\log \frac{\exp(L_{ij})}{\sum_{j=1}^{N} \exp(L_{ij})}$$

Total loss:

$$\mathcal{L} = \frac{1}{2}(L_i + \mathcal{L}_t)$$

This bidirectional loss ensures that both the image and text encoders learn complementary representations. CLIP can perform zero-shot classification by comparing an input image to text descriptions of different classes. The model computes the similarity between the image embedding and

text embeddings for phrases like "a photo of a [class]" and predicts the class with the highest similarity score.

Computing the image embedding: $I_{test} = I(test_image)$

Computing text embeddings for each class: $T_c = T("a\ photo\ of\ a\ \{class\}")$

Predicting the class with maximum similarity: $\hat{y} = \operatorname{argmax}_c \frac{I_{test} \cdot T_c}{\|I_{test}\| \cdot \|T_c\|}$

This ability to generalize to new tasks without additional training makes CLIP particularly versatile for various computer vision applications. To get more versatile features, CLIP-ViT-Large-Patch14 was selected for its strong zero-shot accuracy.

The implementation followed a similar pipeline to DINOv2 but includes additional capabilities for text-based queries, enabling zero-shot behavior classification without fine-tuning. However, in our experiments, we used these models primarily for feature extraction rather than zero-shot classification, as we trained supervised classifiers on the extracted embeddings.

6.2.4 Feature Extraction Comparison on Cows

We compared DINOv2 and CLIP embeddings using t-SNE visualization to evaluate their discriminative power for behavior classification. Since this is an image classification job, we will only use Dinov2 to extract image embeddings.

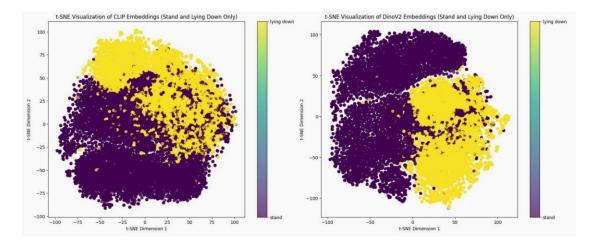


Fig. 9. t-SNE visualization of DINOv2 embeddings for cattle behaviors showing distinct cluster separation(left). t-SNE visualization of CLIP embeddings for cattle behaviors revealing overlapping behavioral clusters(right).

The visualizations reveal that DINOv2 produces more distinct clusters for different behaviors, while CLIP embeddings show moderate separation with overlapping regions.

6.2.5 Classification of Cow Behaviors

We evaluated MLP classifiers (described in Section 2.2.7) on the extracted features from both DINOv2 and CLIP models to assess their discriminative power for behavior classification. The training typically converging within 3-5 epochs.

Table 5: Binary classification performance of MLP models using DINOv2 and CLIP features for standing versus lying behavior detection of dairy cows.

Model	Feature Type	Accuracy	Precision	Recall	F1-Score
MLP	DINOv2	98.3%	0.982	0.982	0.982
MLP	CLIP	98.2%	0.981	0.983	0.982

Both feature extraction methods achieved high performance, with nearly identical results. The MLP classifier using DINOv2 features achieved 98.3% accuracy with a test loss of 0.0645, while the CLIP-based classifier achieved 98.2% accuracy with a slightly lower test loss of 0.0565.

Detailed performance metrics by class reveal consistent performance across both behaviors (Table 6).

Table 6: Class-specific performance metrics for standing and lying behavior classification using DINOv2 and CLIP features

Feature Extractor	Behavior	Precision	Recall	F1-score	Support
DINOv2	Standing	0.987	0.984	0.985	3022
	Lying	0.977	0.980	0.978	2042
CLIP	Standing	0.990	0.980	0.985	3022
	Lying	0.971	0.985	0.978	2042

The high performance of both feature extraction methods indicates that our framework successfully captures discriminative features for behavior classification. The marginal difference between DINOv2 and CLIP suggests that both approaches effectively encode behavioral information, with DINOv2 showing a slight edge in overall accuracy. The balanced performance across classes demonstrates the robustness of our approach for identifying both standing and lying behaviors with minimal bias toward either category.

7. Glossary

A2: Area Attention

AI: Artificial Intelligence

AO: Average Overlap

API: Application Programming Interface

AUC: Area Under the Curve

CBVD-5: Cow Behavior Video Dataset (5 categories)

CLS: Class token (in transformers)

CLIP: Contrastive Language-Image Pre-training

CNN: Convolutional Neural Network

CPU: Central Processing Unit

DINOv2: Self-Distillation with NO labels version 2

GB: Gigabyte

GOT-10k: Generic Object Tracking benchmark (10,000 videos)

GPU: Graphics Processing Unit

I/O: Input/Output

IoU: Intersection over Union

LaSOT: Large-scale Single Object Tracking

LaSOText: LaSOT extension dataset

LSTM: Long Short-Term Memory

LVD-142M: Large Vision Dataset with 142 Million images

mAP: mean Average Precision

MLP: Multi-Layer Perceptron

NVMe: Non-Volatile Memory Express

OWLv2: Open-World Localization Vision model version 2

Pnorm: Normalized Precision

RAM: Random Access Memory

ReLU: Rectified Linear Unit

RGB: Red, Green, Blue (color model)

SAM: Segment Anything Model

SAM2: Segment Anything Model version 2

SAMURAI: Segment Anything Model Upgraded for Real-time Adaptation and Inference

SSD: Solid State Drive

t-SNE: t-distributed Stochastic Neighbor Embedding

V100: NVIDIA Volta 100 GPU

ViT: Vision Transformer

Xmem: Extended Memory (video segmentation model)

YOLO: You Only Look Once

YOLOv3: You Only Look Once version 3

YOLOv11-L: You Only Look Once version 11 large

YOLOv12: You Only Look Once version 12

8. Reference

Alvarenga, A. B., Oliveira, H. R., Chen, S. Y., Miller, S. P., Marchant-Forde, J. N., Grigoletto, L., & Brito, L. F. (2021). 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.

Antanaitis, R., Džermeikaitė, K., Bespalovaitė, A., Ribelytė, I., Rutkauskas, A., Japertas, S., & Baumgartner, W. (2023). Assessment of ruminating, eating, and locomotion behavior during heat stress in dairy cattle by using advanced technological monitoring. Animals, 13(18), 2825.

Arablouei, R., Wang, Z., Bishop-Hurley, G. J., & Liu, J. (2023). Multimodal sensor data fusion for insitu classification of animal behavior using accelerometry and GNSS data. Smart Agricultural Technology, 4, 100163. Bass, L., Clements, P. C., & Kazman, R. (2021). Software architecture in practice (4th ed.). AddisonWesley Professional.

Berckmans, D. "Precision livestock farming technologies for welfare management in intensive livestock systems." Revue Scientifique et Technique 33, no. 1 (2014): 189-196.

Bergamini, L., Pini, S., Simoni, A., Vezzani, R., Calderara, S., Eath, R. B., & Fisher, R. B. (2021). 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 2021 (pp. 524-533). SciTePress.

Bokade, R., Navato, A., Ouyang, R., Jin, X., Chou, C. A., Ostadabbas, S., & Mueller, A. V. (2021). A cross-disciplinary comparison of multimodal data fusion approaches and applications: Accelerating learning through trans-disciplinary information sharing. Expert Systems with Applications, 165, 113885.

Bonneau, M., Vayssade, J. A., Troupe, W., & Arquet, R. (2020). Outdoor animal tracking combining neural network and time-lapse cameras. Computers and Electronics in Agriculture, 168, 105150.

Bugueiro, A., Fouz, R., & Diéguez, F. J. (2021). 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.

Cheng, H. K., & Schwing, A. G. (2022). Xmem: Long-term video object segmentation with an atkinson-shiffrin memory model. In European Conference on Computer Vision (pp. 640-658). Cham: Springer Nature Switzerland.

Clark, B., Panzone, L. A., Stewart, G. B., Kyriazakis, I., Niemi, J. K., Latvala, T., ... & Frewer, L. J. (2019). Consumer attitudes towards production diseases in intensive production systems. PloS one, 14(1), e0210432.,

Cornish, A., Raubenheimer, D., & McGreevy, P. (2016). What we know about the public's level of concern for farm animal welfare in food production in developed countries. Animals, 6(11), 74.

Domun, Y., Pedersen, L.J., White, D., Adeyemi, O., and Norton, T. "Learning patterns from time-series data to discriminate predictions of tail-biting, fouling and diarrhoea in pigs." Computers and Electronics in Agriculture 163 (2019): 104878.

Džermeikaitė, K., Bačėninaitė, D., & Antanaitis, R. (2023). Innovations in cattle farming: application of innovative technologies and sensors in the diagnosis of diseases. Animals, 13(5), 780.

Eckelkamp, E. A., & Bewley, J. M. (2020). On-farm use of disease alerts generated by precision dairy technology. Journal of dairy science, 103(2), 1566-1582.

Fan, H., Bai, H., Lin, L., Yang, F., Chu, P., Deng, G., ... & Ling, H. (2021). Lasot: A high-quality large-scale single object tracking benchmark. International Journal of Computer Vision, 129, 439-461.

Fan, H., Lin, L., Yang, F., Chu, P., Deng, G., Yu, S., ... & Ling, H. (2019). Lasot: A high-quality benchmark for large-scale single object tracking. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5374-5383).

Fernandes, A. F. A., Dórea, J. R. R., & Rosa, G. J. D. M. (2020). Image analysis and computer vision applications in animal sciences: an overview. Frontiers in Veterinary Science, 7, 551269.

Firk, R., Stamer, E., Junge, W., & Krieter, J. (2002). Automation of oestrus detection in dairy cows: a review. Livestock Production Science, 75(3), 219-232.

Fuentes, A., Han, S., Nasir, M. F., Park, J., Yoon, S., & Park, D. S. (2023). Multiview monitoring of individual cattle behavior based on action recognition in closed barns using deep learning. Animals, 13(12), 2020.

Guo, Q., Sun, Y., Orsini, C., Bolhuis, J. E., de Vlieg, J., Bijma, P., & de With, P. H. (2023). Enhanced camera-based individual pig detection and tracking for smart pig farms. Computers and Electronics in Agriculture, 211, 108009.

Halachmi, I., Guarino, M., Bewley, J., & Pastell, M. (2019). Smart animal agriculture: application of real-time sensors to improve animal well-being and production. Annual review of animal biosciences, 7(1), 403-425.

Hampton, J. O., MacKenzie, D. I., & Forsyth, D. M. (2019). How many to sample? Statistical guidelines for monitoring animal welfare outcomes. PLoS One, 14(1), e0211417.

Huang, L., Zhao, X., & Huang, K. (2019). 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.

Kinga, D., & Adam, J. B. (2015, May). A method for stochastic optimization. In International conference on learning representations (ICLR) (Vol. 5, No. 6).

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., Dollár, P., and Girshick, R. "Segment Anything." arXiv preprint arXiv:2304.02643 (2023).

Li, G., Huang, Y., Chen, Z., Chesser, G.D., Purswell, J.L., Linhoss, J., and Zhao, Y. "Practices and applications of convolutional neural network-based computer vision systems in animal farming: A review." Sensors 21, no. 4 (2021): 1492.

Li, K., Fan, D., Wu, H., & Zhao, A. (2024). A new dataset for video-based cow behavior recognition. Scientific Reports, 14(1), 18702.

Liu, Y., Li, W., Liu, X., Li, Z., & Yue, J. (2024). Deep learning in multiple animal tracking: A survey. Computers and Electronics in Agriculture, 224, 109161.

Mathis, A., Mamidanna, P., Cury, K. M., Abe, T., Murthy, V. N., Mathis, M. W., & Bethge, M. (2018). DeepLabCut: markerless pose estimation of user-defined body parts with deep learning. Nature neuroscience, 21(9), 1281-1289.

Matthews, S. G., Miller, A. L., Clapp, J., Plötz, T., & Kyriazakis, I. (2016). Early detection of health and welfare compromises through automated detection of behavioural changes in pigs. The Veterinary Journal, 217, 43-51.

Matthews, S. G., Miller, A. L., PlÖtz, T., & Kyriazakis, I. (2017). Automated tracking to measure behavioural changes in pigs for health and welfare monitoring. Scientific reports, 7(1), 17582.

McDonagh, J., Tzimiropoulos, G., Slinger, K. R., Huggett, Z. J., Down, P. M., & Bell, M. J. (2021). Detecting dairy cow behavior using vision technology. Agriculture, 11(7), 675.

Menezes, G. L., Mazon, G., Ferreira, R. E., Cabrera, V. E., & Dorea, J. R. (2024). 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.

Minderer, M., Gritsenko, A., & Houlsby, N. (2023). Scaling open-vocabulary object detection. Advances in Neural Information Processing Systems, 36, 72983-73007.

Minderer, M., Gritsenko, A., Stone, A., Neumann, M., Weissenborn, D., Dosovitskiy, A., ... & Houlsby, N. (2022, October). Simple open-vocabulary object detection. In European conference on computer vision (pp. 728-755). Cham: Springer Nature Switzerland.

Neethirajan, S. (2021). The use of artificial intelligence in assessing affective states in livestock. Frontiers in Veterinary Science, 8, 715261.

Oliveira, D. A. B., Pereira, L. G. R., Bresolin, T., Ferreira, R. E. P., & Dorea, J. R. R. (2021). A review of deep learning algorithms for computer vision systems in livestock. Livestock Science, 253, 104700.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., ... & Bojanowski, P. (2023). Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:2304.07193.,

Orhei, C., Mocofan, M., Vert, S., & Vasiu, R. (2020). End-to-end computer vision framework. In 2020 International Symposium on Electronics and Telecommunications (ISETC) (pp. 1-4). IEEE.

Psota, E.T., Mittek, M., Pérez, L.C., Schmidt, T., and Mote, B. "Multi-pig part detection and association with a fully-convolutional network." Sensors 19, no. 4 (2019): 852.

Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. "Learning transferable visual models from natural language supervision." In International Conference on Machine Learning, 8748-8763. PMLR, 2021.

Ravi, N., Gabeur, V., Hu, Y. T., Hu, R., Ryali, C., Ma, T., ... & Feichtenhofer, C. (2024). Sam 2: Segment anything in images and videos. arXiv preprint arXiv:2408.00714.

Rial, C., Stangaferro, M. L., Thomas, M. J., & Giordano, J. O. (2024). 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.

Rohan, A., Rafaq, M. S., Hasan, M. J., Asghar, F., Bashir, A. K., & Dottorini, T. (2024). Application of deep learning for livestock behaviour recognition: A systematic literature review. Computers and Electronics in Agriculture, 224, 109115.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. nature, 323(6088), 533-536.

Stygar, A. H., Gómez, Y., Berteselli, G. V., Dalla Costa, E., Canali, E., Niemi, J. K., ... & Pastell, M. (2021). A systematic review on commercially available and validated sensor technologies for welfare assessment of dairy cattle. Frontiers in veterinary science, 8, 634338.

T. Psota, E., Schmidt, T., Mote, B., & C. Pérez, L. (2020). Long-term tracking of group-housed livestock using keypoint detection and map estimation for individual animal identification. Sensors, 20(13), 3670.

Tian, Hongkun, Tianhai Wang, Yadong Liu, Xi Qiao, and Yanzhou Li. "Computer vision technology in agricultural automation—A review." Information processing in agriculture 7, no. 1 (2020): 1-19.

Tian, Y., Ye, Q., & Doermann, D. (2025). Yolov12: Attention-centric real-time object detectors. arXiv preprint arXiv:2502.12524.

Vidal, M., Wolf, N., Rosenberg, B., Harris, B. P., & Mathis, A. (2021). Perspectives on individual animal identification from biology and computer vision. Integrative and comparative biology, 61(3), 900-916.

Wang, X., Wang, Y., Yang, J., Jia, X., Li, L., Ding, W., & Wang, F. Y. (2024). The survey on multi-source data fusion in cyber-physical-social systems: Foundational infrastructure for industrial metaverses and industries 5.0. Information Fusion, 102321.

Wurtz, K., Camerlink, I., D'Eath, R. B., Fernández, A. P., Norton, T., Steibel, J., & Siegford, J. (2019). Recording behaviour of indoor-housed farm animals automatically using machine vision technology: A systematic review. PloS one, 14(12), e0226669.

Yang, C. Y., Huang, H. W., Chai, W., Jiang, Z., & Hwang, J. N. (2024). Samurai: Adapting segment anything model for zero-shot visual tracking with motion-aware memory. arXiv preprint arXiv:2411.11922.