Journal of Field Robotics

WILEY

RESEARCH ARTICLE

Real-Time Behavior Recognition Using a Legged Robot for Animal–Robot Interaction

Edoardo Fazzari[1,2,3] | Donato Romano[1,2] | Fabrizio Falchi[1,4] | Cesare Stefanini[3]

[1]Sant'Anna School of Advanced Studies, The BioRobotics Institute, Viale Rinaldo Piaggio, Pontedera, Italy | [2]Department of Excellence in Robotics and AI, Sant'Anna School of Advanced Studies, Piazza Martiri della Libertà, Pisa, Italy | [3]Department of Robotics, Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE | [4]Institute of Information Science and Technologies, National Research Council of Italy, Pisa, Italy

Correspondence: Edoardo Fazzari (edoardo.fazzari@santannapisa.it)

Received: 15 May 2025 | **Revised:** 29 August 2025 | **Accepted:** 22 November 2025

Funding: Horizon Europe - Food, Bioeconomy Natural Resources, Agriculture and Environment 101181363.

Keywords: animal action recognition | animal-robot interaction | deep learning | ethology | knowledge distillation

ABSTRACT

Animal–robot interaction is an emerging interdisciplinary field that explores the dynamics between animals and robotic systems, as well as the design principles for effective engagement. While previous approaches have investigated animal responses to robotic stimuli, they have yet to integrate artificial intelligence (AI) for real-time behavioral analysis during the interaction. This paper addresses this gap by introducing an AI-driven framework that enables a robotic dog to autonomously monitor and analyze livestock behavior, specifically in cows and chickens. Our system processes real-time camera observations using deep-learning models to detect animal presence and recognize actions. It integrates three neural networks: YOLO-Chicken and YOLO-Cows, for accurate detection of chickens and cows, respectively, and DARTEMIS, a novel, distilled unimodal variant of a state-of-the-art Animal Action Recognition model. The networks communicate efficiently via Redis in a lightweight manner, with all processing conducted onboard the robot. We trained YOLO-Cow and YOLO-Chicken on a subset of the COCO data set for cows and a public data set for chickens, achieving mAP@50-95 scores of 0.67 and 0.56, respectively. DARTEMIS, trained on the Animal Kingdom data set like ARTEMIS, reached an mAP of 77.3. With these models, we tested our system in real-world conditions through field trials, evaluating its ability to accurately detect animals and classify their behaviors. This study presents the first successful integration of efficient deep- learning models into a robotic platform for real-time animal behavior analysis. The proposed framework paves the way for continuous automated livestock monitoring, with potential applications in improving animal welfare and farm management. The full implementation is publicly available and designed to be adaptable to various robotic platforms and related challenges.

1 | Introduction

The interaction between animals and robots is an emerging field that explores how animals engage with robotic systems and how robots can be designed to effectively interact with animals. This domain seeks to merge the natural and artificial worlds into a synergistic system (Romano et al. 2019). It is inherently interdisciplinary, drawing from robotics and engineering for robot design and control systems (Mo et al. 2020); ethology for understanding animal behavior and social patterns (Bonnet et al. 2019); biology for exploring sensory capabilities across species (Elmer et al. 2021); psychology for investigating cognition, learning, and conditioning (Macrì et al. 2020); and computer science, which is the focus of this study, for integrating processing systems and machine learning (Fazzari et al. 2024).

Prior research in animal–robot interaction has largely focused on how different species perceive and respond to robotic entities, particularly by optimizing robot appearance, movement, and behavior to foster meaningful interactions in controlled laboratory environments (Lykov et al. 2024). These studies have provided valuable insights into animal behavior within physical and social contexts, contributing to a deeper understanding of ecological complexities, an especially critical endeavor given the ongoing biodiversity crisis (Chen et al. 2023). Beyond laboratory settings, animal–robot interaction holds promise for various real-world applications, such as wildlife research and conservation (e.g., using robotic animals to study wild populations) (Melo et al. 2023) and agricultural automation (e.g., deploying robotic systems for livestock management) (Ren et al. 2020). Despite this potential, many existing approaches face limitations when applied in dynamic, real-world settings where conditions are less controlled, and unexpected challenges are commonplace (Hewawasam et al. 2022). We posit that machine learning, particularly reinforcement learning, has a pivotal role to play in addressing these challenges by enhancing the adaptability of robotic systems. Additionally, machine learning can improve animal–robot interaction by enabling autonomous monitoring and behavior analysis without requiring constant human supervision (Fazzari et al. 2025a).

In this context, we propose integrating deep object detection and animal action recognition models to analyze observations captured by a Unitree GO2 robotic dog monitoring cows and chickens. Specifically, we developed a lightweight system capable of running on resource-constrained device without the need for server access. For object detection, we employed models based on the YOLO architecture (Khanam and Hussain 2024) and a novel lightweight variant called LeYOLO (Hollard et al. 2024), which we tested for recognizing and localizing cows and chickens. For animal action recognition, we addressed the challenge of creating a lightweight model capable of processing video frames while maintaining high accuracy. Current state-of-the-art models for action recognition, such as our ARTEMIS Fazzari et al. (2025b), are computationally intensive, leveraging multiple visual and language modalities to achieve superior mean Average Precision (mAP) scores. To overcome this limitation, we applied Knowledge Distillation (KD) (Hinton 2015) to extract knowledge from ARTEMIS and create a distilled, unimodal variant called DARTEMIS, which operates exclusively on video frames. DARTEMIS, based on TimeSformer (Bertasius et al. 2021), reduces model size and improves inference speed without significant performance degradation (Banner et al. 2019).

The main contributions of this study are as follows:

- A novel robotic framework integrating deep learning for animal action recognition in cows and chickens, demonstrating the feasibility of incorporating machine learning into animal–robot interactions for behavior understanding.
- DARTEMIS, a distilled and quantized multilabel action recognition model that achieves comparable performance to the state-of-the-art ARTEMIS model while being only 1% of its size.
- We conducted field evaluations of our framework in a breeding facility and a privately owned farm, assessing its performance in real-world conditions. To support further research, we have released the code along with the collected data set, which includes 6549 frames spanning approximately 40 min of active interaction.

The remainder of this paper is structured as follows: Section 2 reviews related efforts in animal–robot interactions. Section 3 details the robotic dog platform, implemented deep- learning architectures and the experimental setup. In Section 4 we present and discuss the experimental results, including model training and field testing. Finally, Section 5 concludes the paper with key takeaways and future directions.

2 | Related Work

The use of robots to interact with animals gained attention in the 1990s, marking a growing interest in building artificial agents for animal–robot interactions (Abdai and Miklósi 2018). Early examples include a robot designed to reduce stress in caged chickens (Bohlen 1999), a robotic rat influencing food-seeking behavior in real rats (Takanishi et al. 1998), and a mechanical bee capable of communicating with forager bees inside a hive (Michelsen et al. 1992). These studies primarily focused on understanding animal responses to robotic stimuli. However, a critical aspect of animal–robot interaction, enabling robots to process and respond to their perceptions, either visually or through other sensory inputs, remains underexplored. This study aims to address this gap, as existing studies have predominantly concentrated on imparting behaviors to robots or examining how animals interact or coexist with them.

The field of animal–robot interaction can broadly be divided into two categories (De Schutter et al. 2001): *single interaction* and *multiple interactions*. Single interactions involve stimulus-response paradigms, where a decoy stimulates the animal to observe its reaction (Bulté et al. 2018). An effective decoy need not perfectly replicate the target animal but should act as an ideal stimulus to elicit specific responses. This is encapsulated in the concept of *supernormal stimulus* (De Schutter et al. 2001), where artificially constructed agents evoke reactions without resembling an animal. Examples include male sticklebacks reacting to red postal trucks (Tinbergen 2020) and dogs responding more strongly to robotic vocalizations than to appearance or movement (Morovitz et al. 2017).

Multiple interactions, on the other hand, involve creating a dynamic loop of interactions between animals and robots. This requires robots equipped with mechanical components capable of responding to animal behavior. Such interactions may involve either a linear sequence of exchanges between a single animal and a robot or large-scale interactions between groups of animals and robots. The former is particularly challenging, as the robot must detect subtle behavioral cues and adapt dynamically to the situation, a capability that is currently underdeveloped (De Schutter et al. 2001). The latter falls under the concept of Collective Intelligence (CI) (McMillen and Levin 2024), where robots mimic social behaviors to study phenomena such as schooling in fish (Porfiri 2018). CI research also explores social interactions between animals and robots within communities, such as rats displaying social behavior

toward pre-programmed robotic rats (del Angel Ortiz et al. 2016), or the use of robots as mediators to study inter-species interactions (Bonnet et al. 2019).

A critical limitation of previous studies is the lack of using of deep-learning techniques. While deep-learning research has advanced significantly, it has primarily focused on animals rather than the robotic aspect of interactions. This includes tasks like animal detection through classification or object detection (Eikelboom et al. 2019), identifying keypoints on animal joints for movement analysis (Pereira et al. 2022), and action recognition (Ng et al. 2022). In this study, we aim to bridge this gap by integrating deep-learning-based object detection and action recognition into anima–robot interactions. This represents a crucial step toward developing intelligent multi-interaction systems that enable more dynamic and meaningful interactions between animals and robots.

3 | Methods

3.1 | The Robotic Dog

The robotic dog used in this study is a Unitree Go2 Quadruped (educational version)[1], equipped with a 16GB NVIDIA Jetson Orin NX, capable of delivering between 40 and 100 Tera Operations Per Second (TOPS). The robot comes with a 4D (3D position + 1D grayscale) LiDAR-L1 pre-installed, enabling real-time 3D mapping of its environment and obstacle avoidance. The robot is also equipped with a camera that records video at 1080p resolution and 15 FPS, featuring an F2.2 aperture and a 120° field of view. However, this camera can only be accessed via the Unitree mobile application and is not programmable through code. To overcome this limitation, we integrated an Intel RealSense D435i camera, which can be accessed programmatically via Python. The Intel RealSense D435i provides depth image resolutions of 1280 × 720 at 30 FPS, enhancing the robot's perception capabilities. Furthermore, a remote controller is included, allowing users to manually maneuver the robot when necessary.

Regarding the impact of the robot's form factor, we believe it may have played a role. The quadrupedal appearance could make the robot more familiar to farm animals, potentially being perceived as similar to a real dog, and thus not entirely out of the ordinary in their environment. At the same time, the choice of the Unitree Go2 robotic dog was also motivated by practical considerations. We added the following clarification in Section 3.1:

3.2 | Object Detection

This section provides an overview of the object detection models considered in our study, along with the data set used for their training.

3.2.1 | Models

Object detection methods can broadly be divided into two categories: two-stage and single-stage approaches, each offering distinct methodologies for handling the detection pipeline. Two-stage detectors, such as Faster R-CNN (Ren et al. 2016) and CO-DETR (Zong et al. 2023), first generate region proposals for potential object locations and then perform classification and bounding box refinement sequentially on these proposed regions. In contrast, single-stage detectors, like YOLO (Jiang et al. 2022) and SSD (Liu et al. 2016), simplify object detection by treating it as a regression problem, directly predicting object classes and bounding box coordinates in a single forward pass of the network. By eliminating the need for a separate proposal generation step, single-stage detectors achieve real-time performance, which is crucial for applications like autonomous driving (Cao et al. 2023) and robotics (Ge et al. 2023). They also demonstrate competitive accuracy with modern architectures, making them particularly appealing for deployment in resource-constrained environments. Consequently, we adopted single-stage detectors for our object detection tasks.

The single-stage models we selected for this study are YOLOv11 (Khanam and Hussain 2024) and LeYOLO (Hollard et al. 2024). YOLOv11 represents the latest iteration in the YOLO series, offering enhanced performance over its predecessors through architectural innovations. These include the introduction of the C3k2 (Cross Stage Partial with kernel size 2) block, SPPF (Spatial Pyramid Pooling—Fast), and C2PSA (Convolutional block with Parallel Spatial Attention) components (Khanam and Hussain 2024). YOLOv11 is available in five configurations–nano, small, medium, large, and extra large–tailored to different computational capacities. In our evaluation, we tested all configurations using an input image size of 640.

LeYOLO is a recently developed lightweight and efficient implementation of YOLO, designed to reduce model parameters and FLOPs while maintaining performance comparable to YOLOv11n. Its efficiency stems from an optimized backbone inspired by the Inverted Bottlenecks principle (Sandler et al. 2018) and the introduction of two novel components: the Fast Pyramidal Architecture Network (FPAN) for fast multiscale feature sharing and Decoupled Network-in-Network (DNiN) detection for lightweight computations in classification and regression tasks. LeYOLO is available in four configurations: nano, small, medium, and large. To maintain consistency with the input size used in YOLOv11, we evaluated the small and medium configurations of LeYOLO, both trained on 640 × 640 input images.

Table 1 provides a detailed comparison of the YOLOv11 and LeYOLO models used in our study, focusing on their number of parameters and Floating Point Operations (FLOPs).

3.2.2 | Data Set

For training our YOLO-Chicken and YOLO-Cow models, we utilized two public data sets: a customized version of the COCO data set (Lin et al. 2014) containing only cow images and an open-source chicken data set from RoboFlow.[2] The latter was chosen due to the absence of a dedicated chicken category in COCO, which only includes a general “bird” class. Using these diverse public data sets helped enhance the robustness of our detection models, as they contain various breeds of cows and chickens and often feature a higher number of animals per image than those observed in our field evaluations.

TABLE 1 | Object detection models employed dimensions in terms of model parameters and FLOPs.

Model	Parameters (M)	FLOPs (M)
LeYOLOSmall	1.9	4.5
LeYOLOMedium	2.4	5.8
YOLO11n	2.6	6500
YOLO11s	9.4	21,500
YOLO11m	20.1	68,000
YOLO11l	25.3	86,900
YOLO11x	59.9	194,900

Note: Models are reported in increasing dimension. We reported the FLOPs indicated in the corresponding papers.

The curated COCO subset provided 1968 images for training and 87 images for validation, while the chicken data set included 3700 training images and 500 validation images. By isolating specific object categories, we optimized the model outputs to specialize in cow and chicken detection while benefiting from the strong feature representations learned during pre-training.

3.3 | Animal Action Recognition

This section examines the animal action recognition networks we evaluated, focusing on the distillation of a large neural network, ARTEMIS, to create a smaller, more efficient model. Additionally, we discuss the quantization strategy employed to further reduce the model size and enhance inference speed.

3.3.1 | Models

Since the release of the Animal Kingdom data set (Ng et al. 2022), research interest has surged in animal action recognition, particularly through multimodal architectures (Mondal et al. 2023). The data set is well-suited for this task, as it encompasses 140 distinct actions captured in 30,100 video clips across 850 different species, establishing itself as a definitive benchmark for multilabel animal action recognition. In multilabel AAR, the objective is to develop a model capable of identifying all actions present in a scene, irrespective of the number of animals or the specific actor performing each action. Formally, given an input sample x and a model function M, the task is to estimate the probability of each action $a = a_1, ..., a_n$ being observed as:

$$\mathrm{P\{a\}} = \mathrm{softmax}(M(x)) \tag{1}$$

For evaluation, these probabilities are compared to the ground truth after applying a threshold parameter τ, which can be automatically determined using the Multilabel Average Precision metric in torchmetrics[3].

The current state-of-the-art model for the Animal Kingdom data set is ARTEMIS, which integrates multimodal inputs by employing textual descriptions generated through captioning and summarization using BLIP2 (Li et al. 2023) and Llama 3 (Dubey et al. 2024), alongside image and video data. ARTEMIS is available in multiple configurations, differentiated by residual

TABLE 2 | Number of parameters in multimodal animal action recognition architectures proposed for the Animal Kingdom data set, alongside their respective video encoders, VideoMamba and TimeSformer.

Model	Parameters (M)	mAP
MSQNet	252	73.1
Mamba-MSQNet (Best)	123	74.6
ARTEMIS (Best 1)	10,953	77.3
ARTEMIS (Best 2)	10,953	77.2
ARTEMIS (Best 3)	10,955	77.3
ARTEMIS (Ensemble)	11,461	79.8
VideoMamba-Ti	7	62.9
VideoMamba-S	25	63.2
VideoMamba-M	74	70.6
TimeSformer	121	72.8

Note: For BLIP2 and Llama 3 (8B), used in ARTEMIS, we accounted precisely for 2.7 billion and 8 billion parameters, respectively. For ARTEMIS (Ensemble), a single usage of BLIP2 and Llama 3 was considered, as all textual descriptions were precomputed before model training for efficiency. We record also the results in term of mAP: for VideoMamba and TimeSformer results were obtained as part of this study for analysis reasons.

connections between multimodal branches and ensemble techniques. The ensemble comprises a weighted combination of the three most performant ARTEMIS configurations, designated as $BEST_1$, $BEST_2$, and $BEST_3$. However, the integration of video-language models and large language models like BLIP2 and Llama 3 introduces significant computational overhead, rendering ARTEMIS challenging to deploy on resource-constrained devices such as our robotic platform. Table 2 provides a comprehensive comparison of parameter counts across different ARTEMIS configurations (single and ensemble), along with two previously proposed multimodal models, MSQNet and Mamba-MSQNet. The table additionally details the dimensions of their respective video encoders, TimeSformer (Bertasius et al. 2021) and VideoMamba (Li et al. 2024).

While previous models demonstrate lower mean Average Precision compared to ARTEMIS, they do not rely on BLIP2 and Llama 3 for input generation, resulting in substantially reduced computational complexity. Since we would like to achieve performance comparable to ARTEMIS (Ensemble) while mitigating computational demands, without opting for a lower mAP, we conducted the distillation of ARTEMIS (Ensemble) into a more computationally efficient unimodal network.

Conceptually, the transformation from the multimodal ARTEMIS ensemble to the unimodal DARTEMIS configuration can be described as an input-to-logit mappings. Let v, i and t denote the video, image and text inputs, respectively (with t produced by the BLIP2-LLaMA3 captioning/summarization pipeline). Each ARTEMIS variant $M^{(k)}$, $k \in \{1, 2, 3\}$ (corresponding to the configurations referred to as $BEST_1$, $BEST_2$ and $BEST_3$), consumes the three modalities and produces a vector of logits

$$\mathbf{z}^{(k)} = M^{(k)}(v, i, t) \in \mathbb{R}^n, \tag{2}$$

where n is the number of action labels. The ensemble aggregates these logits with nonnegative weights α_k obtained via Genetic

Algorithm as described in ARTEMIS original paper (Fazzari et al. 2025b), yielding ensemble logits and ensemble probabilities

$$\mathbf{z}_{\text{ens}} = \sum_{k=1}^{3} \alpha_k \, \mathbf{z}^{(k)}, \alpha_k \geq 0, \sum_{k=1}^{3} \alpha_k = 1, \tag{3}$$

$$\mathbf{p}_{\text{ens}} = \sigma(\mathbf{z}_{\text{ens}}), \tag{4}$$

where $\sigma(\cdot)$ denotes the element-wise sigmoid used to convert logits to per-action probabilities in the multilabel setting.

The DARTEMIS configuration is a unimodal, video-only model g_ψ parameterized by ψ. It produces logits and probabilities using only the video input:

$$\mathbf{z}_{\text{dart}} = g_\psi(v) \in \mathbb{R}^n, \mathbf{p}_{\text{dart}} = \sigma(\mathbf{z}_{\text{dart}}). \tag{5}$$

Informally, the aim of the transformation is for the video-only outputs $\mathbf{z}_{\text{dart}}$ (or $\mathbf{p}_{\text{dart}}$) to mirror the ensemble outputs $\mathbf{z}_{\text{ens}}$ (or $\mathbf{p}_{\text{ens}}$) as closely as possible while removing the dependency on image- and text-generation pipelines. The concrete distillation procedure that implements this transformation, together with its optimization details and evaluation protocol, is described in the following section.

3.3.2 | Knowledge Distillation

Knowledge Distillation (KD) (Hinton 2015) is a technique for transferring knowledge from a complex model or ensemble (teacher) to a smaller, more efficient model (student) while maintaining comparable performance. In this process, the student is trained not only on the ground-truth labels but also to match the class probabilities, or *soft targets*, produced by the teacher. This alignment is achieved by minimizing a loss that combines the primary task loss with a similarity loss, computed using Mean Squared Error (MSE). To address the multilabel nature of our problem, we employed the sigmoid function instead of the softmax function usually used to compute the soft targets:

$$\hat{y}_{\text{soft}} = \frac{\text{logits}}{T}, \tag{6}$$

where T represents the temperature parameter that smooths the teacher's output probabilities. The total loss is defined as:

$$\mathcal{L} = \alpha \ \text{MSE}\left(\hat{y}_{\text{soft}}^{\text{teacher}}, \hat{y}_{\text{soft}}^{\text{student}}\right) + (1-\alpha) \ \text{BCE}(\text{logits}_{\text{student}}, \text{targets}), \tag{7}$$

where α is a weighting factor, and BCE denotes the binary cross-entropy loss:

$$\text{BCE}(\mathbf{z}, \mathbf{y}) = -\frac{1}{n}\sum_{j=1}^{n} [y_j \log \sigma(z_j) + (1-y_j) \log(1-\sigma(z_j))], \tag{8}$$

with $\sigma(\cdot)$ denoting the sigmoid activation, $\mathbf{z} \in \mathbb{R}^n$ the predicted logits, and $\mathbf{y} \in \{0,1\}^n$ the ground-truth binary labels.

To further enhance the distillation process, we investigated three advanced strategies: temperature annealing (Malinin et al. 2019), dynamic temperature adjustment (Wen et al. 2021), and noise-augmented teacher predictions (Sau and Balasubramanian 2016). Temperature annealing starts with a high temperature ($T = 5$) to allow the student to learn from softened teacher predictions and linearly decreases it to a lower value ($T = 1$) by the end of training, enabling the student to focus on sharper predictions. The linear annealing schedule can be expressed as:

$$T(e) = T_{\text{start}} - \frac{e}{E_{\max}}(T_{\text{start}} - T_{\text{end}}), \tag{9}$$

where e is the current epoch, $E_{\max}$ is the total number of training epochs, $T_{\text{start}} = 5$, and $T_{\text{end}} = 1$.

Dynamic temperature adjustment, on the other hand, adapts T based on the Mean Absolute Error (MAE) between the teacher's and student's predictions, ensuring that the temperature responds to their evolving alignment during training. The temperature is computed as:

$$T = \max\left(\frac{T_{\text{base}}}{\text{MAE}(\text{logits}_{\text{teacher}}, \text{logits}_{\text{student}})}, T_{\min}\right), \tag{10}$$

where $T_{\text{base}} = 5$ and $T_{\min} = 1$.

Lastly, noise-augmented predictions involve adding Gaussian noise to the teacher's outputs, which mitigates overconfidence and improves generalization. This can be expressed as:

$$\tilde{\mathbf{z}}_{\text{teacher}} = \mathbf{z}_{\text{teacher}} + \varepsilon, \varepsilon \sim \mathcal{N}(0, \sigma^2 I), \tag{11}$$

where $\mathbf{z}_{\text{teacher}}$ are the teacher's logits, σ^2 controls the noise variance, and I is the identity matrix.

Additionally, we aimed to distill multimodal knowledge from ARTEMIS into unimodal architectures, such as VideoMamba or TimeSformer, to simplify the model and reduce computational demands. Multimodal-to-unimodal distillation, though less commonly studied, has shown promise in various domains, including egocentric action recognition (Radevski et al. 2023) and medical imaging (Ahmad et al. 2024; Xiong et al. 2023). This approach enables the student model to learn implicit intermodal correlations from the teacher, improving generalization and yielding superior performance compared to models trained solely on unimodal data.

3.4 | Implementation Details

The workflow designed for the robotic system to observe and analyze animal actions is illustrated in Figure 1. The diagram represents the components running within the robot and their communication via Redis streams. The system comprises three main components: the camera, which operates as a background process, and two Docker containers–one for object detection and the other for action recognition. The camera process runs independently outside of Docker to simplify setup, allowing

direct Python access to the USB port. This approach avoids the additional configuration required to enable USB access within a Docker container. Conversely, the deep-learning models and the Redis server are containerized using Docker, as this facilitates the installation of dependencies for GPU utilization and ensures compatibility by leveraging the latest Redis and nvidia/l4t-pytorch images available on Docker Hub.

The camera captures images at regular intervals of 0.375 s. This interval was carefully selected to align with the requirements of state-of-the-art action recognition models, which typically process sequences of 16 frames, while also considering the average clip length of 6 s in the Animal Kingdom data set ($6/16 = 0.375$). Once an image is captured, it is saved to disk, and its file path is published to the Frames Redis stream. The object detection container reads from this stream to determine whether animals are present in the frame. Based on the detections, it generates a results file and publishes the image path along with a boolean flag indicating the presence or absence of animals to the Detection stream. The action recognition module uses this information to decide whether a given frame should be processed. When a frame containing an animal is detected, the module stores it in a buffer, maintaining the most recent 16 frames. Once exactly 16 frames are accumulated, the model performs action recognition, saving the results to a log file. This allows for both real-time notifications and later human evaluation. If more than 16 consecutive meaningful frames (i.e., those containing animals) are encountered, the oldest frame is discarded to ensure that only the most recent 16 frames are retained. Conversely, if a frame without any detected animals is encountered, the recognition module clears its buffer, waiting for a new meaningful sequence to begin. This strategy prevents unnecessary processing of irrelevant frames, optimizing computational efficiency.

3.5 | Experimental Setup

We evaluated our system at CiRAA[4] (Centro di Ricerche Agro-Ambientali 'E. Avanzi'), one of Europe's largest research centers dedicated to the study of sustainable agricultural systems. The center is situated within the natural park "Migliarino—San Rossore—Massaciuccoli," near Pisa, and includes facilities specifically designed for animal breeding. These facilities house cows, providing an ideal environment to test our model. For the chicken evaluation, we tested the robot in a privately owned farm.

4 | Results

In this section, we present the results of our experiments across all neural network components. This includes object detection performance on COCO subsets focusing on cows and chickens, the outcomes of distillation and quantization processes aimed at deriving a smaller, high-performing, and efficient model from ARTEMIS, and the evaluation of the integrated system deployed on our robotic dog for field applications.

4.1 | Object Detection

The selected LeYOLO and YOLO models were trained separately to recognize cows and chickens using the COCO data set and the mentioned chicken data set from RoboFlow. Tables 3 and 4 summarize the detection performance for cows and chickens, respectively, in terms of mean Average Precision (mAP) at different thresholds, including mAP@50, mAP@75, and the overall mAP@50-95. Among the evaluated models, YOLOv11x demonstrated the highest performance for both tasks. The results indicate a clear trend: larger and more complex architectures yield better performance.

For the cow detection task, the difference in overall mAP@50-95 between YOLOv11x and smaller models was more pronounced compared to the chicken detection task, where the medium, large, and extra-large versions of YOLOv11 achieved the same performance levels. The LeYOLO configurations were the least effective, with slightly lower performance than YOLOv11n.

Upon inspecting predictions made by the LeYOLO models, we observed that their accuracy improved significantly when the animals were clearly visible and unobstructed. Additionally, since the COCO data set includes various cow species labeled under a single

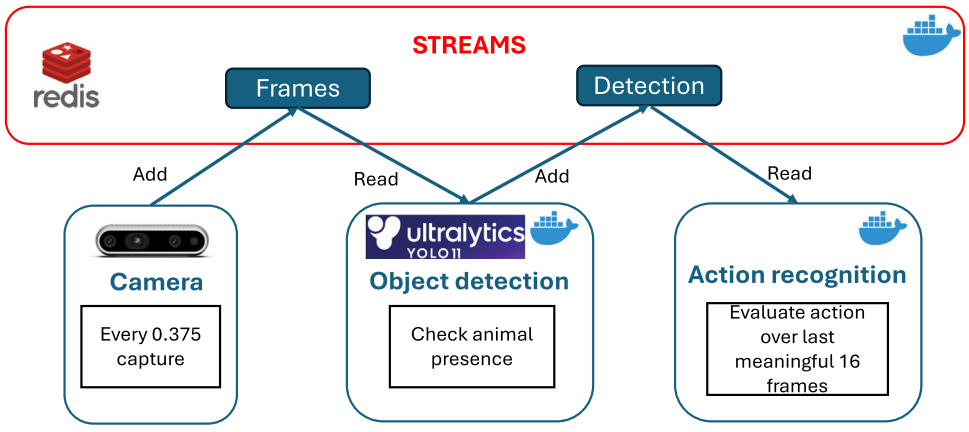

FIGURE 1 | Workflow illustrating the operations of our complete system. The systems comprises four components: Redis for handling two streams for the communication, a process for capturing frames using the RealSense camera, and two Docker containers running the object detection and the action recognition model. Everything is implemented on the robot. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 3 | Object detection results for recognizing cows in terms of mean Average Precision at different thresholds.

Model	mAP 50	mAP 75	mAP 50-95
LeYOLOSmall	0.76	0.55	0.51
LeYOLOMedium	0.77	0.56	0.51
YOLOv11n	0.78	0.60	0.55
YOLOv11s	0.80	0.67	0.60
YOLOv11m	0.82	0.68	0.61
YOLOv11l	0.83	0.69	0.61
YOLOv11x	**0.86**	**0.74**	**0.67**

Note: Best results are in bold.

TABLE 4 | Object detection results for recognizing chickens in terms of mean Average Precision at different thresholds.

Model	mAP 50	mAP 75	mAP 50-95
LeYOLOSmall	0.84	0.55	0.51
LeYOLOMedium	0.84	0.54	0.52
YOLOv11n	0.84	0.57	0.53
YOLOv11s	0.85	0.60	0.55
YOLOv11m	**0.86**	**0.61**	**0.56**
YOLOv11l	**0.86**	**0.61**	**0.56**
YOLOv11x	**0.86**	**0.61**	**0.56**

Note: Best results are in bold.

class, all models, both YOLO and LeYOLO, performed better at detecting animals with colors and appearances corresponding to species more commonly found in farming environments, such as the classic black-and-white or brown cow.

Based on these results and the model dimensions presented in Table 1, we selected YOLOv11m for both tasks. For chicken detection, this choice was justified by the absence of performance gains when using larger versions, leading us to designate this model as YOLO-Chicken. For cow detection, while the extra-large version offered a slight improvement in performance, the difference was marginal compared to the significant reduction in model size, approximately one-third of the extra-large version. Therefore, we opted for YOLOv11m, referred to as YOLO-Cow.

4.2 | Distilled ARTEMIS: DARTEMIS

The student models we used had not previously been evaluated on the Animal Kingdom data set and were only tested as video encoders. To provide a fair comparison, we trained these models using the same data augmentation strategies, optimizer, and scheduler employed in ARTEMIS. Each model was trained for 300 epochs. The results, summarized in Table 2, reveal several insights.

VideoMamba performed poorly in its tiny and small configurations, achieving a mAP of approximately 63. However, the medium configuration demonstrated significant improvement, reaching 70.6 mAP. Among the single-modality models

TABLE 5 | Distillation results using ARTEMIS (ensemble) as teacher model.

Model	Teacher noise	Temperature	mAP
VideoMamba-Ti	No	1	63.72
VideoMamba-Ti	No	2	63.48
VideoMamba-Ti	No	5	62.71
VideoMamba-Ti	No	10	62.82
VideoMamba-Ti	No	Dynamic	72.07
VideoMamba-Ti	Yes	Dynamic	71.88
VideoMamba-Ti	No	Annealing	71.95
VideoMamba-Ti	Yes	Annealing	71.91
VideoMamba-S	No	Dynamic	73.16
VideoMamba-S	Yes	Dynamic	73.66
VideoMamba-S	No	Annealing	73.02
VideoMamba-S	Yes	Annealing	73.23
VideoMamba-M	No	Dynamic	74.23
VideoMamba-M	Yes	Dynamic	73.45
VideoMamba-M	No	Annealing	73.82
VideoMamba-M	Yes	Annealing	73.73
Timesformer	No	Dynamic	76.87
Timesformer	Yes	Dynamic	77.04
Timesformer	No	Annealing	76.66
Timesformer	Yes	Annealing	**77.30**

Note: Best result is indicated in bold.

(processing only video clips), TimeSformer achieved the highest score, with an mAP of 72.8. While MSQNet slightly outperformed TimeSformer with an mAP of 73.1, its architecture is considerably more complex. MSQNet integrates TimeSformer with the CLIP image encoder for extracting frame-specific features, and the CLIP text encoder for incorporating textual information from class names. This marginal performance gain suggests that MSQNet's added complexity was not entirely justified. In contrast, Mamba-MSQNet, which has complexity comparable to TimeSformer achieved performance improvement over its video encoder, VideoMamba-Ti, increasing mAP from 62.9 to 74.6. Here, we aimed to achieve similar improvements without increasing model complexity by leveraging knowledge distillation from the ARTEMIS (ensemble), which achieved the highest mAP on the Animal Kingdom data set for action recognition.

The distillation process was first conducted using the baseline strategy described in the Methods section, where we varied the temperature parameter to explore both hard and soft targets ($T \in \{1, 2, 5, 10\}$), while fixing the weighting factor at $\alpha = 0.5$ (kept constant across all experiments). This simple strategy was evaluated exclusively on VideoMamba-Ti, as reported in Table 5. Although performance improved marginally, reaching 63.7 mAP, the gains remained limited. To address this, we employed more advanced strategies, namely, Dynamic Temperature Distillation (Wen et al. 2021) and temperature annealing (Malinin et al. 2019), both of which substantially

enhanced performance. Under these methods, VideoMamba-Ti achieved 72.1 and 72.0 mAP, respectively, results comparable to TimeSformer, while requiring 94% fewer parameters. Further experimentation introduced Gaussian noise injection into the teacher logits; however, this led to a slight performance decrease, with both methods yielding 71.9 mAP.

Building on these findings, we performed an ablation study on larger configurations of VideoMamba and TimeSformer, systematically testing DTD and temperature annealing, with and without noise-augmented teacher predictions. The results confirm that distillation consistently and substantially improves model performance relative to their non-distilled counterparts. Across architectures, DTD typically outperformed temperature annealing, though the benefit of teacher noise proved model-dependent. Specifically, Gaussian noise improved performance for TimeSformer and VideoMamba-S, while reducing mAP for VideoMamba-Ti and VideoMamba-M. Importantly, when improvements were observed, they applied consistently across both DTD and annealing, and conversely, performance degradation also manifested in both methods.

A comparison between Tables 5 and 2 further highlights the efficiency of our approach. VideoMamba-S, distilled with DTD and teacher noise, achieved 73.66 mAP, surpassing MSQNet while being only 10% of its size and relying solely on the video modality. Scaling up, VideoMamba-M achieved a modest improvement, reaching 74.23 mAP under DTD. Notably, TimeSformer distilled with temperature annealing and teacher noise attained 77.3 mAP, representing the highest performance among all distilled models. This result not only matches ARTEMIS (single) but does so with slightly fewer parameters than Mamba-MSQNet, only 1% of the parameter count of ARTEMIS (ensemble), and again using only a single modality. Given this superior performance, coupled with a relatively modest parameter increase over the next-best architecture (VideoMamba-M) yet yielding nearly a 3-point mAP gain, we selected the distilled TimeSformer as our final action recognition model. We designate this distilled variant of ARTEMIS as **DARTEMIS**, denoting "Distilled ARTEMIS."

4.3 | System Real-Time Performance

For real-time operation, the system must efficiently process information with minimal delay. Therefore, we evaluated the performance of each component to assess potential latencies in our predictions. The camera process is scheduled to capture an image every 375 ms. Once an image is acquired, the object detection module processes it in 50.1 ms, with 2.5 ms for preprocessing, 44 ms for inference, and 3.6 ms for postprocessing. Subsequently, assuming that 16 meaningful frames have been accumulated, DARTEMIS processes the batch (consisting of a single sample) in 970 ms.

Using this approach, by the time a sample is fully processed, two new frames will have been collected but remain unprocessed. If this pattern continues, a cumulative delay of approximately 2.5 frames per processed sample will be introduced, compromising real-time performance. To mitigate this issue, we experimented with an alternative batching strategy to ensure that the processing delay remains bounded. Instead of processing a single sample at a time, we modified the approach to create a batch containing all frames collected while the action recognition module was running. This batch is then fed to the network, allowing multiple samples to be processed simultaneously within the same time frame.

With this strategy, the system processes two or three samples together, resulting in an upper bound delay of 1 s for obtaining recognition results. This delay remains within an acceptable range for real-time analysis, ensuring the system's responsiveness while maintaining accurate action recognition.

4.4 | Field Evaluation

We evaluated our system by deploying the robotic platform in real-world environments to perform behavior recognition on chickens and cows. Figure 2 illustrates the robot in both scenarios, alongside examples of visual observations processed by the system. These include bounding boxes generated by YOLO-Chicken and YOLO-Cow, and the top three predicted behaviors from DARTEMIS, obtained by applying a softmax function to the model's logits. This visualization effectively represents the robot's perception and interpretation of its surroundings in real time.

During those field testing a data set was collected from the 6549 frames recorded by our robot, corresponding to approximately 40 min of data. To assess the system's accuracy, we annotated the collected video frames using the Animal Kingdom taxonomy, which aligns with the label structure of our model. The videos were segmented into nonoverlapping clips of 16 consecutive frames, the input format required by DARTEMIS, and each clip was manually labeled. Quantitative results are reported in Table 6, with separate evaluations provided for chicken and cow observations.

For the chicken experiments, the robot was placed inside a hen coop, allowing close-range observations without risk of being trampled, as chickens are approximately the same size as the robot. The robot's presence initially triggered cautious behavior: the chickens maintained a safe distance and fled when the robot moved rapidly or approached directly. To minimize disruption, the robot was kept mostly stationary or moved at slow speeds. Under these conditions, the chickens resumed natural behaviors such as "moving," "standing still," "eating," and "sensing," which were effectively recognized by the model. DARTEMIS achieved a mAP of 82.9, demonstrating robust performance. Errors in prediction were primarily caused by adverse lighting conditions. When the robot was oriented toward the sun, camera artifacts such as lens flares, color blooming, and localized overexposure degraded the visual quality of the input frames, as reported in Figure 3a, negatively impacting model performance.

In the cow experiments, the robot interacted with two breeds: Holstein Friesian and Pisana, a rare and endangered breed native to the Pisa region. Trials were conducted both in a barn and in an open pasture. Unlike chickens, cows did not exhibit fear. Instead, they demonstrated curiosity, often approaching the robot, sniffing it, and observing it at close range. However, this interest waned over time if the robot remained still. Once the robot resumed motion, the cows' attention was often re-engaged. For safety reasons, all observations were conducted with the cows behind a protective fence to prevent accidental damage to the robot.

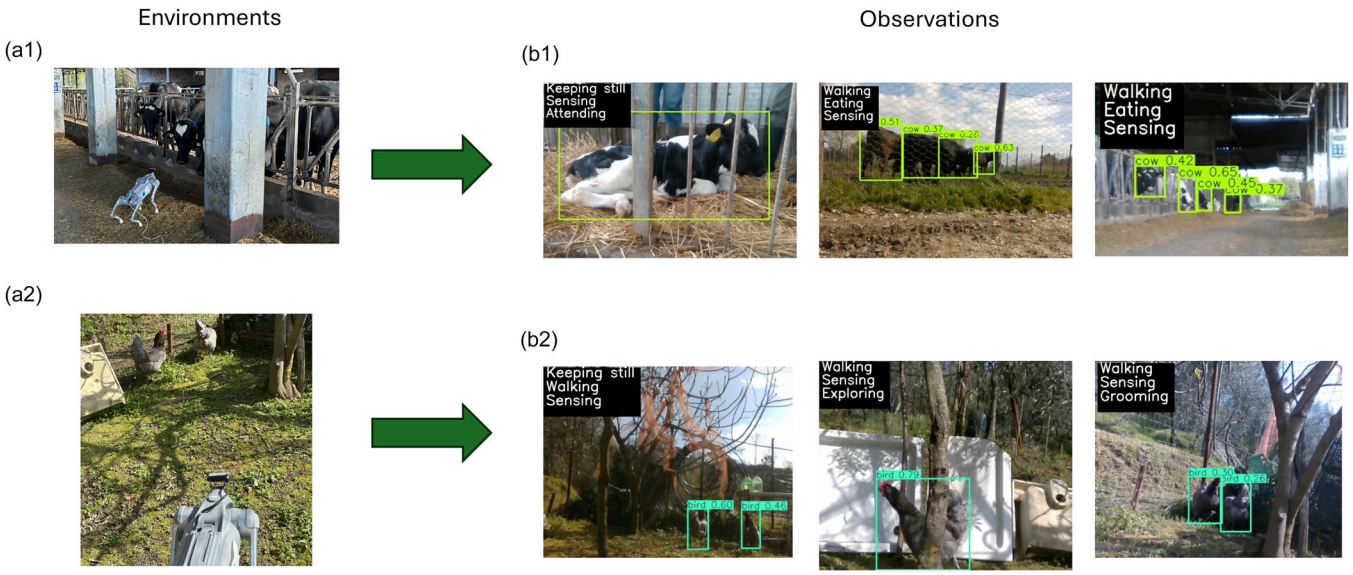

FIGURE 2 | Evaluation of our system in two environments, interacting and analyzing actions of cows (a) and chickens (b). In the observations, the action labels correspond to the three most probability actions detected by DARTEMIS. [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 6 | Field performance of DARTEMIS using the observations collected by the robot.

	mAP
All	54.0 (65.3)
Only chickens	82.9
Only cows	46.9 (61.8)

Note: In parenthesis the results after removing clips with significant occlusions or excessive motion.

Although the mAP achieved for cow action recognition was lower at 46.9, DARTEMIS demonstrated impressive qualitative performance. Accurate predictions were made when the cows were fully or mostly visible in the frame. However, when occlusions occurred, due to foreground haystacks or structural elements like metal bars, prediction accuracy decreased, as shown in Figure 3b. This can be attributed to the training distribution of DARTEMIS, which was based on documentary-style video clips that typically feature unobstructed and well-framed views of animals. Such conditions are not always met in real-world farm environments. Another challenge arose from the robot's movement, which occasionally introduced motion blur and camera shake. In these cases, the model sometimes misclassified actions. For instance, stationary cows engaging in behaviors such as "eating" or "sensing" were occasionally labeled as "walking," likely due to background motion being interpreted as animal movement. After excluding frames with significant occlusions or excessive motion from the evaluation data set, the mAP improved to 61.8, highlighting the model's potential under better observational conditions. Future work could address these issues by incorporating head stabilization control (HSC) (Manfredi et al. 2013) to mitigate camera shake and developing automated systems to optimize the robot's positioning: ensuring that the camera consistently captures animals at an appropriate distance and with minimal occlusion could further enhance detection accuracy and action recognition performance.

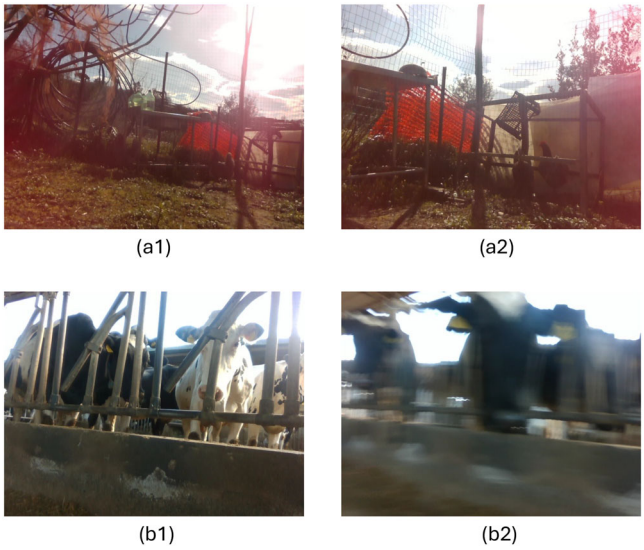

FIGURE 3 | Cases with adverse lighting, occlusions, or motion blur. Illustrating common challenges encountered in real-world deployments. [Color figure can be viewed at wileyonlinelibrary.com]

A noteworthy ambiguity emerged between the classes "sensing" and "attending." According to the Animal Kingdom definitions, "attending" refers to an animal fixating on a specific stimulus while remaining still, whereas "sensing" implies nonspecific scanning behavior, often involving head movement. In practice, cows frequently transitioned from one behavior to the other within the same sequence, initially focusing intently, for example, on the robot, and later displaying more general exploratory head movements. This dynamic made annotation and classification challenging and suggests that these categories might be better unified under a broader behavioral class in future work. We also observed occasional confusion between the actions "moving" and "walking," although these cases were rare. The "moving" label, being more general, may be applied in cases where movement was inferred but not explicitly detected

in one of the specific action labels, introducing ambiguity. Despite these challenges, DARTEMIS demonstrated strong performance when animals were well-framed and clearly visible. Notably, the model successfully identified complex behaviors, such as a cow "manipulating" a metal bar for extending its head through to investigate the robot. This indicates that the model not only captures common locomotor behaviors but can also identify fine-grained, context-dependent actions, even in uncontrolled outdoor settings.

5 | Conclusion

This paper presents a real-time behavior recognition system for animal–robot interaction, leveraging a Unitree Go2 robotic dog to analyze the actions of two animal species: cows and chickens. Our system seamlessly integrates three key components, the RealSense camera, an object detection module, and an action recognition module, to enable robust and efficient behavioral analysis in real-world settings.

The robot is equipped with a RealSense camera that captures frames at 0.375 s intervals. These frames are then processed by YOLOv11x, which has been specifically trained for cow and chicken detection using the COCO data set and a public data set from RoboFlow, respectively. Our object detection model achieves a high mAP@50 of 0.86 for both species, demonstrating its reliability in identifying animals in diverse environments. To facilitate real-time processing, the camera module communicates with a Redis container, publishing frame path information to a Redis Stream upon capturing a new frame. Once object detection is performed, results are published to a separate Stream, which the action recognition module subsequently reads. This module employs DARTEMIS, a distilled version of ARTEMIS that we developed to simplify its multimodal processing into unimodal. Unlike its predecessor, DARTEMIS operates solely on video input, significantly reducing computational complexity and enabling real-time inference on edge devices like the robotic dog. Despite its smaller footprint–99% smaller than ARTEMIS, DARTEMIS achieves an impressive mAP of 77.3, nearly matching the performance of the original model.

In conclusion, our system represents a significant step forward in the field of animal–robot interaction, equipping a robotic agent with the capability to recognize and analyze animal behavior autonomously. While our approach achieves strong detection and recognition performance, some limitations remain. The object detection module, though highly accurate, may occasionally fail to detect all animals in a given frame or, in rare cases, mistakenly classify nonanimal objects. Similarly, the action recognition model, trained on the Animal Kingdom data set, provides a broad understanding of animal behaviors but lacks species-specific specialization. While we see value in a generalized model capable of recognizing actions across multiple species, certain applications may benefit from species-specific models for enhanced accuracy. As another possible direction for future work, extending the current multilabel AAR paradigm to assign actions to entire video clips rather than treating them independently for each animal could provide deeper insights into which animals are performing specific actions and enable more accurate assessments of their welfare.

Overall, our work lays the foundation for intelligent, real-time animal behavior monitoring using robotic systems. By further refining detection accuracy and expanding species-specific adaptations, this technology has the potential to revolutionize precision livestock farming, wildlife conservation, and ethological research.

Acknowledgments

European Commission, HORIZON 2020 EXCELLENT SCIENCE - Future and Emerging Technologies (FET), Grant agreement ID: 899520.

Data Availability Statement

The data that supports the findings of this study are available in the following GitHub repository https://github.com/edofazza/behavior-recognition-animal-robot-interaction.

Endnotes

[1] https://www.unitree.com/go2

[2] https://universe.roboflow.com/thesis-3c51t/chicken-counting

[3] https://lightning.ai/docs/torchmetrics/stable/classification/average_precision.html#multilabelaverageprecision

[4] https://avanzi.unipi.it

References

Abdai, J., and Á. Miklósi. 2018. "Poking the Future: When Should We Expect That Animal-Robot Interaction Becomes a Routine Method in the Study of Behavior." *Animal Behavavior and Cognition* 5, no. 4: 321–325.

Ahmad, S., Z. Ullah, and J. Gwak. 2024. "Multi-Teacher Cross-Modal Distillation With Cooperative Deep Supervision Fusion Learning for Unimodal Segmentation." *Knowledge-Based Systems* 297: 111854.

Banner, R., Y. Nahshan, and D. Soudry. 2019. *Post Training 4-Bit Quantization of Convolutional Networks for Rapid-Deployment*. Curran Associates Inc.

Bertasius, G., H. Wang, and L. Torresani. 2021. " Is Space-Time Attention All You Need for Video Understanding?" In ICML, volume 2, 4.

Bohlen, M. 1999. "A Robot in a Cage-Exploring Interactions Between Animals and Robots." In *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No. 99EX375)*, 214–219. IEEE.

Bonnet, F., R. Mills, M. Szopek, et al. 2019. "Robots Mediating Interactions Between Animals for Interspecies Collective Behaviors." *Science Robotics* 4, no. 28: eaau7897.

Bulté, G., R. J. Chlebak, J. W. Dawson, and G. Blouin-Demers. 2018. "Studying Mate Choice in the Wild Using 3d Printed Decoys and Action Cameras: A Case of Study of Male Choice in the Northern Map Turtle." *Animal Behaviour* 138: 141–143.

Cao, Y., C. Li, Y. Peng, and H. Ru. 2023. "Mcs-Yolo: A Multiscale Object Detection Method for Autonomous Driving Road Environment Recognition." *IEEE Access* 11: 22342–22354.

Chen, J., M. Hu, and D. J. Coker, et al. 2023. "Mammalnet: A Large-Scale Video Benchmark for Mammal Recognition and Behavior Understanding." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13052–13061. IEEE.

del Angel Ortiz, R., C. M. Contreras, A. G. Gutiérrez-Garcia, and M. F. M. González. 2016. "Social Interaction Test Between a Rat and a Robot: A Pilot Study." *International Journal of Advanced Robotic Systems* 13, no. 1: 4.

Dubey, A., A. Jauhri, A. Pandey, et al. 2024. The llama 3 Herd of Models. *arXiv preprint arXiv:2407.21783*.

Eikelboom, J. A., J. Wind, E. van de Ven, et al. 2019. "Improving the Precision and Accuracy of Animal Population Estimates With Aerial Image Object Detection." *Methods in Ecology and Evolution* 10, no. 11: 1875–1887.

Elmer, L. K., C. L. Madliger, D. T. Blumstein, et al. 2021. "Exploiting Common Senses: Sensory Ecology Meets Wildlife Conservation and Management." *Conservation Physiology* 9, no. 1: coab002.

Fazzari, E., F. Carrara, F. Falchi, C. Stefanini, and D. Romano. 2024. "Using Ai to Decode the Behavioral Responses of an Insect to Chemical Stimuli: Towards Machine-Animal Computational Technologies." *International Journal of Machine Learning and Cybernetics* 15, no. 5: 1985–1994.

Fazzari, E., D. Romano, F. Falchi, and C. Stefanini. 2025a. "Animal Behavior Analysis Methods Using Deep Learning: A Survey." *Expert Systems With Applications* no. 289: 128330.

Fazzari, E., D. Romano, F. Falchi, and C. Stefanini. 2025b. "Artemis: Animal Recognition Through Enhanced Multimodal Integration System." *International Journal of Machine Learning and Cybernetics* no. 16: 5877–5892.

Ge, W., S. Chen, H. Hu, et al. 2023. "Detection and Localization Strategy Based on Yolo for Robot Sorting Under Complex Lighting Conditions." *International Journal of Intelligent Robotics and Applications* 7, no. 3: 589–601.

Hewawasam, H., M. Y. Ibrahim, and G. K. Appuhamillage. 2022. "Past, Present and Future of Path-Planning Algorithms for Mobile Robot Navigation in Dynamic Environments." *Journal of the Industrial Electronics Society* 3: 353–365.

Hinton, G. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Hollard, L., L. Mohimont, N. Gaveau, and L. -A. Steffenel. 2024. Leyolo, New Scalable and Efficient CNN Architecture for Object Detection. *arXiv preprint arXiv:2406.14239*.

Jiang, P., D. Ergu, F. Liu, Y. Cai, and B. Ma. 2022. "A Review of Yolo Algorithm Developments." *Procedia Computer Science* 199: 1066–1073.

Khanam, R., and M. Hussain. 2024. Yolov11: An Overview of the Key Architectural Enhancements. *arXiv preprint arXiv:2410.17725*.

Li, J., D. Li, S. Savarese, and S. Hoi. 2023. "Blip-2: Bootstrapping Language-Image Pre-Training With Frozen Image Encoders and Large Language Models." In *International Conference on Machine Learning*, 19730–19742. PMLR.

Li, K., X. Li, Y. Wang, et al. 2024. Videomamba: State Space Model for Efficient Video Understanding. *arXiv preprint arXiv:2403.06977*.

Lin, T. -Y., M. Maire, and S. Belongie, et al. 2014. "Microsoft Coco: Common Objects in Context." In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755. Springer.

Liu, W., D. Anguelov, and D. Erhan, et al. 2016. "Ssd: Single Shot Multibox Detector." In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, 21–37. Springer.

Lykov, A., M. Litvinov, and M. Konenkov, et al. 2024. "Cognitivedog: Large Multimodal Model Based System to Translate Vision and Language Into Action of Quadruped Robot." In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, 712–716. IEEE.

Macrì, S., M. Karakaya, C. Spinello, and M. Porfiri. 2020. "Zebrafish Exhibit Associative Learning for an Aversive Robotic Stimulus." *Lab animal* 49, no. 9: 259–264.

Malinin, A., B. Mlodozeniec, and M. Gales. 2019. Ensemble Distribution Distillation. *arXiv preprint arXiv:1905.00076*.

Manfredi, L., T. Assaf, S. Mintchev, et al. 2013. "A Bioinspired Autonomous Swimming Robot as a Tool for Studying Goal-Directed Locomotion." *Biological Cybernetics* 107: 513–527.

McMillen, P., and M. Levin. 2024. "Collective Intelligence: A Unifying Concept for Integrating Biology Across Scales and Substrates." *Communications Biology* 7, no. 1: 378.

Melo, K., T. Horvat, and A. J. Ijspeert. 2023. "Animal Robots in the African Wilderness: Lessons Learned and Outlook for Field Robotics." *Science Robotics* 8, no. 85: eadd8662.

Michelsen, A., B. B. Andersen, J. Storm, W. H. Kirchner, and M. Lindauer. 1992. "How Honeybees Perceive Communication Dances, Studied by Means of a Mechanical Model." *Behavioral Ecology and Sociobiology* 30: 143–150.

Mo, X., W. Ge, M. Miraglia, et al. 2020. "Jumping Locomotion Strategies: From Animals to Bioinspired Robots." *Applied Sciences* 10, no. 23: 8607.

Mondal, A., S. Nag, J. M. Prada, X. Zhu, and A. Dutta. 2023. "Actor-Agnostic Multi-Label Action Recognition With Multi-Modal Query." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 784–794. IEEE.

Morovitz, M., M. Mueller, and M. Scheutz. 2017. " Animal–Robot Interaction: The Role of Human Likeness on the Success of Dog–Robot Interactions." In *1st International Workshop on Vocal Interactivity In-and-Between Humans, Animals and Robots (VIHAR) Conference*, 22–26.

Ng, X. L., K. E. Ong, Q. Zheng, Y. Ni, S. Y. Yeo, and J. Liu. 2022. "Animal Kingdom: A Large and Diverse Dataset for Animal Behavior Understanding." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19023–19034. IEEE.

Pereira, T. D., N. Tabris, A. Matsliah, et al. 2022. "Sleap: A Deep Learning System for Multi-Animal Pose Tracking." *Nature Methods* 19, no. 4: 486–495.

Porfiri, M. 2018. "Inferring Causal Relationships in Zebrafish-Robot Interactions Through Transfer Entropy: A Small Lure to Catch a Big Fish." *Animal Behavior and Cognition* 5, no. 4: 341–367.

Radevski, G., D. Grujicic, M. Blaschko, M. -F. Moens, and T. Tuytelaars. 2023. "Multimodal Distillation for Egocentric Action Recognition." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5213–5224. IEEE.

Ren, G., T. Lin, Y. Ying, G. Chowdhary, and K. Ting. 2020. "Agricultural Robotics Research Applicable to Poultry Production: A Review." *Computers and Electronics in Agriculture* 169: 105216.

Ren, S., K. He, R. Girshick, and J. Sun. 2016. "Faster R-Cnn: Towards Real-Time Object Detection With Region Proposal Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, no. 6: 1137–1149.

Romano, D., E. Donati, G. Benelli, and C. Stefanini. 2019. "A Review on Animal–Robot Interaction: From Bio-Hybrid Organisms to Mixed Societies." *Biological Cybernetics* 113: 201–225.

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L. -C. Chen. 2018. "Mobilenetv2: Inverted Residuals and Linear Bottlenecks." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* : 4510–4520.

Sau, B. B., and V. N. Balasubramanian. 2016. Deep Model Compression: Distilling Knowledge From Noisy Teachers. *arXiv preprint arXiv:1610.09650*.

De Schutter, G., G. Theraulaz, and J. -L. Deneubourg. 2001. "Animal–Robots Collective Intelligence." *Annals of Mathematics and Artificial Intelligence* 31: 223–238.

Takanishi, A., T. Aoki, M. Ito, Y. Ohkawa, and J. Yamaguchi. 1998. "Interaction Between Creature and Robot: Development of an Experiment System for Rat and Rat Robot Interaction." In *Proceedings. 1998*

IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No. 98CH36190), 1975–1980. IEEE.

Tinbergen, N. 2020. *The Study of Instinct*. Pygmalion Press, an imprint of Plunkett Lake Press.

Wen, T., S. Lai, and X. Qian. 2021. "Preparing Lessons: Improve Knowledge Distillation With Better Supervision." *Neurocomputing* 454: 25–33.

Xiong, F., C. Shen, and X. Wang. 2023. "Generalized Knowledge Distillation for Unimodal Glioma Segmentation From Multimodal Models." *Electronics* 12, no. 7: 1516.

Zong, Z., G. Song, and Y. Liu. 2023. "Detrs With Collaborative Hybrid Assignments Training." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6748–6758. IEEE.