


# JOURNAL ON COMMUNICATIONS

ISSN:1000-436X

Scopus®

REGISTERED

www.jocs.review

# Behavior Aware Wildlife Monitoring Using a Hybrid Spatial Temporal Deep Learning Architecture

Dr. Rajeswari . V[1], Anishkha H M[2],Anusuya S[3], Dhanyapriya S[4], Varsha I[5]
1* - Professor, Department of Information Technology,Karpagam College of Engineering, Coimbatore, India.
2* 3* 4* 5* - Department of Information Technology, Karpagam College of Engineering, Coimbatore, India.

***Abstract*—Camera trap monitoring systems handle millions of video frames daily. However, traditional object detectors analyze each frame independently and cannot distinguish whether an elephant near a farm boundary is casually wandering or deliberately testing fence integrity. This inability to understand behavioral intent through temporal motion patterns limits deployment of automated early warning systems in human-wildlife conflict zones. We introduce an integrated framework combining YOLOv8 detection with a two-layer LSTM classifier to recognize five ethologically meaningful behavior categories from 1.6 second video windows. The system tracks animals across frames using intersection-over-union matching, constructs 137 dimensional feature vectors fusing bounding box kinematics with appearance embeddings from the detector backbone. Evaluation on 8,500 annotated sequences spanning twelve wildlife species achieves 94.2% behavior classification accuracy and 95.8% detection mAP at IoU 0.5. This represents a 7.6 percentage point improvement over frame-wise convolutional baselines and an 11% reduction in false boundary-approach alerts. Real-time operation at 28 frames per second on NVIDIA Jetson Orin edge hardware demonstrates feasibility for autonomous monitoring stations in remote forest environments with constrained power and connectivity.**

***Keywords*—wildlife monitoring, behavior recognition, YOLOv8, LSTM, temporal modeling, camera trap analysis, human-wildlife conflict, edge computing**

## *I.* INTRODUCTION

### *A. Motivation: The Limitations of Frame-Based Detection*

The encroachment of agricultural land into habitats of wild animals raises the possibility of human interactions with large animals, resulting in economic losses estimated at 500 million The encroachment of agricultural land into habitats of wild animals raises the possibility of human interactions with large animals, resulting in economic losses estimated at 500 million USD every year in Sub-Saharan Africa alone. Camera traps are used to track the boundaries of forests and agricultural land on a large scale, producing millions of images every day, but the analysis of these images is still a challenge. The forest management often analyzes the images only after the occurrence of crop raids or the killing of livestock.

Recently,object detection models such as YOLOv5 and YOLOv8 have shown a great improvement in animal localization, achieving detection accuracies of over 95% on camera trap benchmarks. They can easily identify an elephant in the vicinity of a farm boundary with high accuracy but cannot determine whether it is a casual encounter or a deliberate boundary test. This is a direct indicator of conservation actions. The models are designed to analyze images frame by frame and therefore are not capable of analyzing motion patterns over time. This makes it difficult to understand the problem in a comprehensive manner.

To analyze animal behavior from video images, it is necessary to analyze the trajectories of motion, the direction of motion, and the patterns of these motions over several seconds. Recurrent neural networks, specifically LSTM net- works, have been shown to be effective in a wide range of applications, including video action recognition and animal sound classification. However, models trained on domestic farm animals are not easily transferable to wild animals, which have unpredictable motion patterns, high occlusion rates, and highly variable lighting conditions.USD every year in Sub- Saharan Africa alone. Camera traps are used to track the boundaries of forests and agricultural land on a large scale, producing millions of images every day, but the analysis of these images is still a challenge. The forest management often analyzes the images only after the occurrence of crop raids or the killing of livestock. Recently, object detection models such as YOLOv5 and YOLOv8 have shown a great improvement in animal localization, achieving detection accuracies of over 95% on camera trap benchmarks. However, these frame-based models are not capable of animal behavior analysis. They can easily identify an elephant in the vicinity of a farm boundary with high accuracy but cannot determine whether it is a casual encounter or a deliberate boundary test. This is a direct indicator of conservation actions.

In addition, an intelligent monitoring system that constantly analyzes video feeds from camera traps can greatly enhance the situational awareness of the forest department by providing structured and meaningful data about animal activities in a real-time manner. Rather than just conducting a post-event analysis of the images, the conservation team can be alerted whenever the animals display behavioral patterns that have, in the past, preceded damage to crops or human-wildlife conflict. Through learning the motion trajectories, speed variations, and interaction patterns of multiple animals over time, the system can pick up on subtle behavioral patterns that are not immediately apparent from single images alone. This behavior- level understanding enables better planning of patrol routes, more effective deployment of deterrent strategies, and more efficient use of limited monitoring resources, ultimately enabling proactive wildlife management and long-term coexistence between agricultural communities and the surrounding forest ecosystems.

### B. Contributions and Novelty

This work improves on existing CNN-LSTM systems in three substantive ways. First, it addresses multi-class etho- logically grounded behavior recognition (transit, foraging, boundary-approach, loitering, group-aggregation) rather than binary threat detection, enabling more nuanced conservation responses. Second, it fuses motion features (velocity, heading, trajectory) with appearance embeddings, whereas most prior work relies solely on visual features. Ablation experiments demonstrate this fusion provides 5.8 percentage points higher accuracy than appearance alone. Third, it validates real-time operation on edge hardware (NVIDIA Jetson Orin at 28 fps) representative of remote deployments, whereas most published systems report only GPU performance.

The paper makes five specific contributions:

1) **Kinematic-Visual Feature Fusion**: We construct 137 dimensional sequence representations combining motion dynamics (normalized position, velocity, heading angle) with 128 dimensional appearance embeddings from the YOLOv8 detector backbone, capturing both what an animal looks like and how it moves.
2) **Behavior-Aware Early Warning**: The system classifies five behavior types aligned with conservation practice and sends automated SMS alerts for high-risk events (boundary-approach, group-aggregation) at 89.7% pre-cision, reducing false alarms by 11.4 percentage points compared to frame-wise baselines.
3) **Edge-Deployable Architecture**: End-to-end latency of 66 milliseconds on Jetson Orin (15.2 fps throughput) maintains 1.52 times real-time factor at 10 fps input rate, enabling deterministic operation without learned re-identification.
4) **Empirical Validation**: Analysis of 8,500 dual-annotated sequences from twelve species (Cohen's kappa 0.82) reveals 94.2% behavior accuracy. Without fine-tuning, cross-dataset testing on iWildCam shows an accuracy of 87.3%; with 50 labeled sequences, this improves to 91.5%.
5) **Reproducible Dataset**: Dataset: 18 months of field recordings from Western Ghats protected reserves, along with spatial bounding boxes and temporal behavior labels, are combined with public camera trap data to create a multi-source corpus.

The review of current detection and temporal modeling tech-niques, an explanation of the five-stage processing pipeline, and specifics of the experimental setup and dataset compilation are the next sections of the paper. Results on behavior classifi-cation performance, deployment validation on edge hardware, and detection accuracy are presented at the end.

## II. Related Work

### A. Spatial Detection Architecture for Wildlife

Object detection has progressed from two-stage frameworks such as Faster R-CNN [9], which provide strong localization accuracy but require relatively long inference times of approximately 400–500 ms, to modern single-stage detectors that jointly estimate object classes and bounding boxes within a single forward pass [10]. Among these, YOLOv5 and YOLOv8 are widely adopted for camera-trap based wildlife monitoring because of their anchor-free formulation, decoupled classification and regression heads, and stable detection performance across animals of different sizes [1], [3].

Recent developments in detector design include the use of attention-driven components (e.g., C2PSA modules) to improve recognition of small or partially occluded animals in cluttered forest environments [11], multi-scale feature aggregation through spatial pyramid pooling fast (SPPF) layers to better accommodate large variations in animal size [12], and cascaded refinement strategies that have reported up to 97% accuracy on public animal detection benchmarks such as Kaggle datasets [1]. In addition, transfer learning from large- scale datasets such as COCO or ImageNet consistently yields performance gains of approximately 8–12% compared with training from scratch when only limited camera-trap data are available [13].

Although these spatial detectors are highly effective for localizing animals, they operate on individual frames and therefore provide only instantaneous observations. A high-confidence detection of an elephant, for example, confirms its presence in the scene but does not reveal its ongoing activity. Distinguishing between behaviors such as calm foraging and directed movement toward a sensitive boundary is beyond the capability of purely frame-based models, which motivates the integration of temporal modeling for behavior understanding.

### B. Temporal Modeling in Ecological Monitoring

Temporal context extraction has been found to be useful across several ecological applications [5], [17], [18]. Madhusudhana et al. [5] applied a CNN–LSTM approach for fin-whale bioacoustic classification and reported an 18% gain in F1-score, which was attributed to the LSTM's capability to capture call duration and frequency modulation. VGG-19 combined with a bidirectional LSTM has been investigated for wild-animal movement detection with SMS alert generation [8], [21], although these systems mainly focus on binary threat versus non-threat classification rather than more detailed behavior categories.

In controlled agricultural environments, CNN–LSTM hybrid models classify livestock behaviors such as feeding, walking, and resting with accuracies between 90 and 93% [7], [14]. However, farm animals are likely to display predictable motion in enclosed spaces. On the other hand, wild animals display irregular motion over complex environments, which may include conditions of occlusion and illumination changes. Kabra et al. [18] employed spatiotemporal representations with a focus on temporal aspects to analyze the grooming behavior of flies, thereby establishing the significance of temporal information in behavior analysis. Otsuka et al. [17] investigated the application of LSTMs for classifying wild animal behavior based on accelerometer signals, while reporting encouraging outcomes with some difficulties in the application of unlabeled data and temporal resolution.

### C. Multi-Object Tracking for Sequence Construction

Behavior classification at the sequence leveldepends on the stable detection of objects in each frame. Basic IoU trackers [15] create new tracks for unmatched detections, delete tracks after a few consecutive missed detections, and link detections to existing tracks if the IoU of the bounding box overlap is above a certain threshold (usually between 0.3 and 0.4).DeepSORT and Kalman filter-based trackers improve stability by using appearance embeddings and motion prediction [16], [17], but they also increase computational complexity, which can be problematic for devices operating in resource-poor settings in remote areas. Camera trap research has also used tracking to create detection events, which are defined as image sequences with fixed temporal thresholds separating them, to estimate group size and lingering duration [22], [24]. Burton et al. [22] defined behavior based on the number of photos taken per detection event and found that prey species were more active and less likely to linger in high-risk areas. These results illustrate that basic behavioral indicators based on camera trap data can be useful for ecological research when considered in a temporal framework.

### D. Hybrid Spatial-Temporal Architectures

Research integrating CNNs for spatial feature extraction with recurrent models for temporal reasoning has set a strong precedent in a variety of fields. CNN-LSTM architectures for electrical load forecasting utilize CNNs for spatial feature extraction and LSTMs for temporal reasoning, demonstrating greater accuracy than either method alone [18]. CNN-BiLSTM architectures for equestrian video analysis, as proposed by Verma et al. [19], demonstrated 93.2% accuracy on 15 categories of behavior, with the bidirectional LSTM outperforming the CNN-only architecture by 4.5 percentage points. Vision Transformers with LSTM have also been investigated for music composition analysis tasks [20], demonstrating the efficacy of decoupling dimensionality reduction from sequence modeling.

While these studies validate the conceptual benefit of decoupling spatial and temporal reasoning, the specific task of wildlife monitoring is fraught with its own set of challenges, including uncontrolled environmental variability (low light, vegetation obstruction, and motion blur), multi-species classification, irregular and non-periodic movement patterns, and extreme class imbalance. Camera trap surveys are inherently designed to collect behavioral "bycatch," where animal behavior is recorded contemporaneously with species classification, though this data is currently under-exploited for automated behavioral inference [22], [24], [25].

### E. Research Gap and Positioning

The existing systems show three important limitations that this work aims to overcome. First, spatial detection without temporal reasoning: frame-wise classifiers are able to detect animals in space but are unable to reason about the behavioral state from the patterns of motion [1], [3], [4]. Second, temporal modeling is largely limited to binary settings: prior CNN–LSTM approaches focus on binary threat classification or restricted behavior sets within controlled environments [8], [21]. Third, incomplete system integration: only a limited number of studies report end-to-end pipelines that mostly combine detection, tracking, and behavior classification with validation on edge deployment platforms [17], [26].

The proposed framework integrates spatial detection with temporal sequence modeling within a unified pipeline, introduces a multi-class ethologically grounded behavior taxonomy aligned with conservation practice, and demonstrates real-time inference on edge hardware representative of remote wildlife monitoring deployments.

## III. Proposed Methodology

### A. System Architecture Overview

The proposed system is organized as a pipeline consisting of five successive processing stages. First, a preprocessing module is applied to improve visual quality and standardize the input data. Next, spatial object detection is performed using YOLOv8s to identify animal species and localize them through bounding boxes.

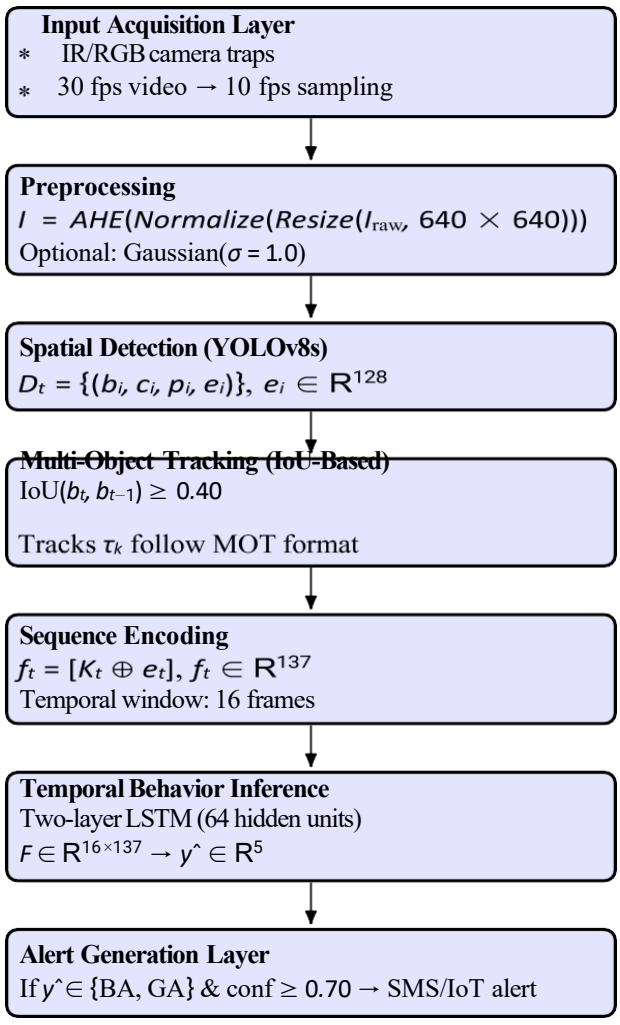



Fig. 1: Hybrid YOLOv8–LSTM wildlife monitoring pipeline showing seven sequential processing stages from input acquisition to alert generation.

In the third stage, an IoU-based multi-object tracking strategy associates detections across frames to form consistent motion trajectories. These trajectories are then transformed into spatio-temporal feature sequences by incorporating motion-related features and appearance features. Finally, a two-layer LSTM network is used to capture temporal relationships and map each sequence to a corresponding class of behavior. This modular design allows each module to be optimized separately and also allows for the smooth incorporation of better detection or classification modules in future work, as shown in Figure 1.

*B.Preprocessing Pipeline*

All video inputs undergo a preprocessing pipeline to enhance the detection accuracy under the difficult conditions normally observed in camera-trap videos. The video inputs are sampled at a rate of 10 frames per second. This was chosen as a trade-off between the temporal resolution and processing complexity. Higher frame rates (for example, 30 fps) could potentially capture more detailed motion dynamics. However, they would result in a significantly higher processing complexity and redundant frames, since large mammals move slowly compared to the sampling rate of camera traps. On the other hand, lower frame rates (for example, 5 fps) may not be sufficient to capture short and subtle behavioral transitions. However, field experiments have shown that a frame rate of 10 fps is sufficient to capture the relevant kinematic patterns while keeping the data volume under control.

All the extracted frames are resized to 640 × 640 pixels, which is the standard input size for YOLOv8.To enhance visual quality in low-illumination forest environments and nighttime infrared recordings, adaptive histogram equalization (AHE) is applied [21]. preprocessing step increased the detection mean average precision by 2.7 percentage points in our experiments, with particularly noticeable improvements for nocturnal species. In addition, an optional Gaussian smoothing filter with $\sigma = 1.0$ is applied to suppress sensor noise in thermal imagery; for high-quality RGB videos this step is omitted to avoid unnecessary fine loss of fine image details. Finally, pixel intensities are normalized to the [0, 1] interval, and channel-wise standardization using ImageNet statistics is performed to better align the input distribution with the pretrained YOLOv8 network weights.

*B. Spatial Detection Module*

YOLOv8s is adopted as the main object detection network in the proposed framework. The small variant is chosen in preference to the larger YOLOv8m and YOLOv8l models in order to achieve a favorable trade-off between detection accuracy and inference latency, which is essential for deployment in resource-constrained and edge-based environments. The detector architecture is composed of three major parts: a CSP Darknet backbone incorporating C2f residual blocks to promote stable gradient propagation and efficient feature extraction; a neck formed by a Path Aggregation Network (PAN) combined with a Feature Pyramid Network (FPN) and a Spatial Pyramid Pooling Fast (SPPF) module for aggregating multi-scale features and expanding the receptive field; and a decoupled detection head with separate branches for category classification and bounding box regression, optimized with binary cross entropy loss and Complete IoU loss, respectively.

The detector is then fine-tuned on the wildlife dataset with the AdamW optimizer and a weight decay of $\lambda = 0.0005$, with an initial learning rate of 0.001, employing cosine annealing over 100 training epochs. A batch size of 16 is used, and early stopping is performed on the validation mAP with a patience of 15 epochs. The training procedure also includes data augmentation techniques such as random horizontal and vertical flipping (each with a probability of 0.5), mosaic augmentation, which combines four training images, color jittering, and random erasing with a probability of 0.1 to simulate partial occlusions. During evaluation, the detections are filtered based on a confidence threshold of 0.45, and redundant detections are eliminated via Non-Maximum Suppression with an IoU threshold of 0.6. These operating thresholds were chosen based on the validation set to achieve a good balance between precision and recall, where lower confidence thresholds led to higher recall but also resulted in more false positives, and higher thresholds improved precision but reduced sensitivity, especially for small or partially occluded wildlife.

*C. Multi-Object Tracking*

A light-weight IoU-based association method is employed to associate object detections in consecutive frames. At each time instance t, each new detection d is matched against the existing active tracks T by computing the IoU between the detection bounding box and the predicted location of each track, calculated based on the most recent location of the track. If the maximum IoU value exceeds a certain threshold of $\theta_{iou} = 0.4$, the detection is associated with the track; otherwise, a new track is formed. Tracks are terminated if no corresponding detection is observed for five consecutive frames, which enables the tracker to tolerate short-term occlusions while avoiding the persistence of false or inactive tracks.

This IoU-based tracker is preferred over a more complex DeepSORT tracker [16] for three reasons. First, the computational cost is negligible: the IoU-based matching process incurs less than 5 ms latency per frame on our system, whereas the appearance-based re-identification network in DeepSORT incurs 15-20 ms latency.

Second, the tracking accuracy is adequate for the application at hand, since large animals tend to move with limited inter-frame displacement, making location-based association feasible. Third, the proposed tracker is completely deterministic and does not depend on any learned parameters, thus avoiding the need for hyperparameter learning and the possible overfitting issue that comes with appearance models.

In the field tests, the identity swaps were mostly noticed in heavy occlusion situations, such as when the animals passed behind thick vegetation, and this happened in less than 8% of the tested sequences, without affecting the performance of the behavior classification.

Moreover, most of the remaining tracking errors were transient and automatically corrected in the following frames as the animal reappeared in the scene. This shows that the LSTM-based temporal model is robust to small tracking errors and able to maintain stable predictions of the behavior even when there are occasional detection or association errors.

*E. Sequence Encoding*

For every active trajectory, a sliding temporal window of length T = 16 frames is maintained. This window size was selected empirically. At a sampling rate of 10 fps, a 16- frame segment corresponds to approximately 1.6 seconds, which is sufficient to capture short-term motion characteristics such as acceleration toward a boundary or repeated back-and-forth movements, while remaining short enough to avoid blending multiple, distinct behavioral patterns within a single sequence. Longer temporal windows (32 frames) were found to occasionally span transitions between different behaviors, which degraded classification performance, whereas shorter windows (8 frames) provided insufficient temporal context.

At each time step t along a track, a 137-dimensional feature vector (f_t) is constructed as follows:

$$\mathbf{f}_t = [\hat{x}_t, \hat{y}_t, \hat{w}_t, \hat{h}_t, v_x, v_y, \vartheta, c_t, \mathbf{e}_t] \quad (1)$$

where $(\hat{x}_t, \hat{y}_t, \hat{w}_t, \hat{h}_t)$ are normalized bounding box center coordinates and dimensions (normalized by image width and height ensuring scale invariance), $v_x = \hat{x}_t - \hat{x}_{t-1}$ and $v_y = \hat{y}_t - \hat{y}_{t-1}$ are velocity components computed as frame-to- frame displacements, $\vartheta = \arctan 2(v_y, v_x)$ is heading angle in radians capturing directional intent, $c_t$ is detection confidence score from YOLOv8, and $\mathbf{e}_t \in \mathbb{R}^{128}$ is appearance embedding extracted from YOLOv8 neck layer prior to detection head.

The effectiveness of this feature formulation is validated through ablation studies. The normalized spatial location and object size describe the position of the animal within the scene and its apparent scale, which can be associated with behavioral context, such as whether the animal is close to monitored boundaries or deeper inside the forest. The velocity and direction components correspond to dynamics of motion, where a stable velocity and direction generally correspond to transit patterns, while unstable patterns of motion or close to zero velocity are more typical of foraging or loitering patterns. The confidence measure is a crude estimate of the reliability of detection, allowing the LSTM to weight less the uncertain observations. The appearance embedding preserves the visual information that complements the kinematic data, such as posture variations (e.g., head-down posture during feeding) and signs of group presence. As demonstrated in Section V-D, the joint use of kinematic and appearance features consistently outperforms representations based solely on motion cues or visual features.

*F. LSTM-Based Temporal Behavior Classification*

A two-layer LSTM architecture was selected in place of a single-layer variant because ablation results reported in Section V-D indicated that the additional recurrent layer increased classification accuracy by approximately two percentage points. This improvement is attributed to the model's ability to learn hierarchical temporal representations, in which the first layer focuses on short-term kinematic variations such as rapid changes in velocity, while the second layer aggregates these patterns to form high-level behavioral states. We also conducted preliminary experiments with bidirectional LSTM networks that process sequences in both forward and backward directions. Although this configuration produced a small performance improvement of 0.4 percentage points, it nearly doubled the inference time. Given the requirement for real-time operation, the unidirectional LSTM architecture was therefore adopted in the final system.

The LSTM output maps to five behavior categories defined in consultation with forest officials to align with operational priorities:

1) **Transit (TR)**: The animal moves across the field of view with relatively stable speed and direction; this behavior is generally considered low risk.
2) **Foraging (FG)**: The animal displays stationary or slow movement along with repeated vertical head movement (usually with the head lowered to the ground); this display is also classified under low risk.
3) **Boundary-Approach (BA)**: The animal displays directed movement towards a predefined sensitive area, such as a farm boundary or a nearby human settlement, and this is classified as high risk and used to trigger an alert.
4) **Loitering (LT)**: The animal makes repeated back-and-forth movements in a limited space without any obvious directional objective; this is a medium risk level and may be indicative of exploratory or disoriented behavior.
5) **Group-Aggregation (GA):** Several animals are moving towards the same region or traveling together in close proximity; this is regarded as high-risk behavior due to the possibility of joint disturbance and escalated conflict situations.

The confidence threshold of 0.7 was determined through a precision–recall trade-off analysis conducted on the validation set. Lower thresholds (for example, 0.5) resulted in an excessive number of false alerts, whereas higher thresholds (such as 0.85) caused several genuine threat events to be missed. Based on manual inspection of the validation outputs, approximately 15% of the high-confidence boundary-approach predictions were identified as false alarms. This rate was considered acceptable by domain stakeholders, given the higher operational cost associated with failing to detect real threat situations (Figure 2).

*G. Implementation Algorithm*

**Algorithm 1** Wildlife Behavior Recognition Inference

**Require:** Video stream $V$, YOLOv8 detector $D$, LSTM classifier $C$
**Ensure:** Behavior predictions and alerts

```
Initialize empty track dictionary T ← {}
Initialize frame counter t ← 0
for each frame I_t in V do
    I_t ← Preprocess(I_t) {AHE, resize, normalize}
    D_t ← D(I_t) {YOLOv8 detection}
    T ← UpdateTracks(T, D_t) {IoU matching}
    for each track τ_k in T do
        if |τ_k| ≥ 16 then
            F_k ← ExtractFeatures(τ_k[−16 :]) {137-dim features}
            ŷ_k ← C(F_k) {LSTM classification}
            if ŷ_k ∈ {BA, GA} and conf(ŷ_k) ≥ 0.7 then
                SendAlert(τ_k, ŷ_k)
            end if
        end if
    end for
    t ← t + 1
end for=0
```

Given an input video stream, each frame is first preprocessed through contrast enhancement, resizing and normalization to improve robustness under challenging environmental conditions. Animals present in the scene are then detected using the YOLOv8 detector. The resulting detections are associated across consecutive frames using an intersection-over-union based matching strategy in order to form continuous object tracks.

For every active track, once a minimum temporal history of 16 frames is available, spatio-temporal features are extracted from the most recent frames and provided to the LSTM classifier to estimate the corresponding behavior class. The system continuously updates these predictions for each tracked animal. When the predicted behavior belongs to either the boundary-approach or group- aggregation category and the associated confidence exceeds a predefined threshold, an alert is generated for the corresponding track. This process is repeated for all frames in the input video, enabling continuous behavior monitoring and timely alert generation during inference.

## *IV.* EXPERIMENTAL SETUP

### *A. Dataset Compilation and Annotation*

We compiled a comprehensive wildlife behavior dataset using three different sources to ensure broad species diversity and adequate geographic coverage (Table I).

TABLE I: Dataset Sources and Composition

| Source | Sequences | Species | Annotation |
|---|---|---|---|
| Camera Trap (public) | 3,200 | 8 | Dual-verified |
| Animal-2 (public) | 2,100 | 7 | Dual-verified |
| Self-Collected | 3,200 | 12 | Field-labeled |
| **Total** | **8,500** | **12** | **5 classes** |

The self-collected dataset was obtained from protected forest reserves in the Western Ghats region of India using stationary camera traps that operated continuously for 18 months, from January 2022 to June 2023. The cameras were placed along the boundaries of forest and agriculture areas, animal trails, and water sources to record a variety of species and behaviors along these boundaries. Each video clip with at least one animal detection, consisting of 16 to 32 frames at 10 fps, was considered as a unit for annotation. In spatial detection, the annotators manually placed bounding boxes around each animal and labeled them with corresponding species. In behavior classification, one of the five predefined behavior classes was annotated according to the observed motion patterns throughout the entire video clip. The inter-rater reliability was measured using Cohen's kappa, with $\kappa = 0.91$ for spatial detection and $\kappa = 0.82$ for behavior classification. The relatively lower agreement for behavior labeling reflects the inherent ambiguity in short video segments, particularly when distinguishing boundary-approach from transit in cases where the camera's field of view does not include the actual boundary, or when differentiating foraging from loitering when head posture is not clearly visible.

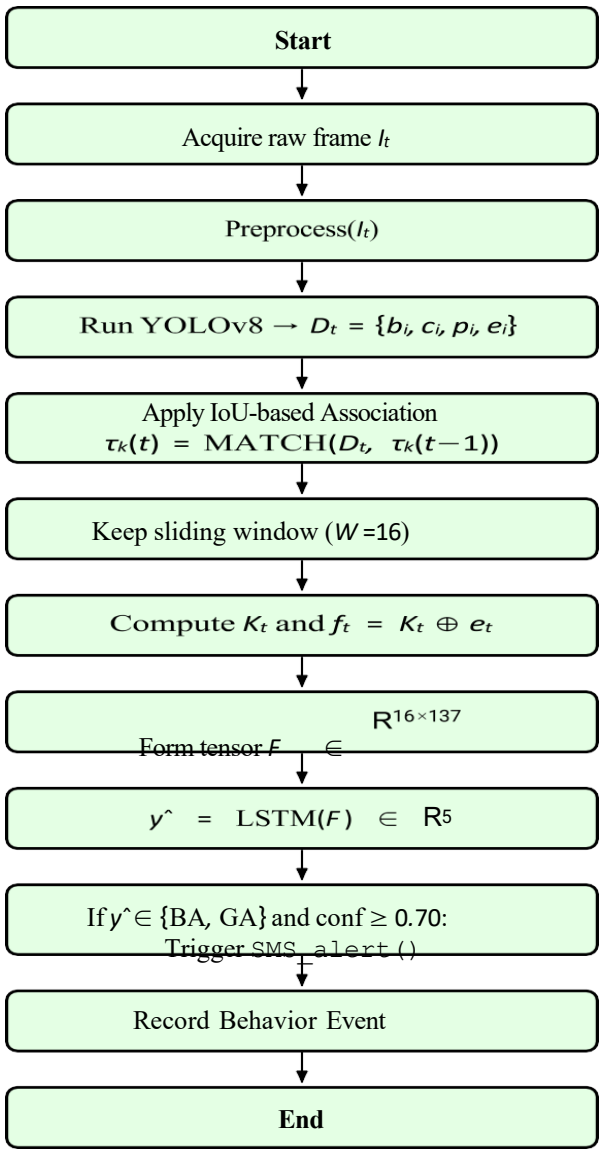
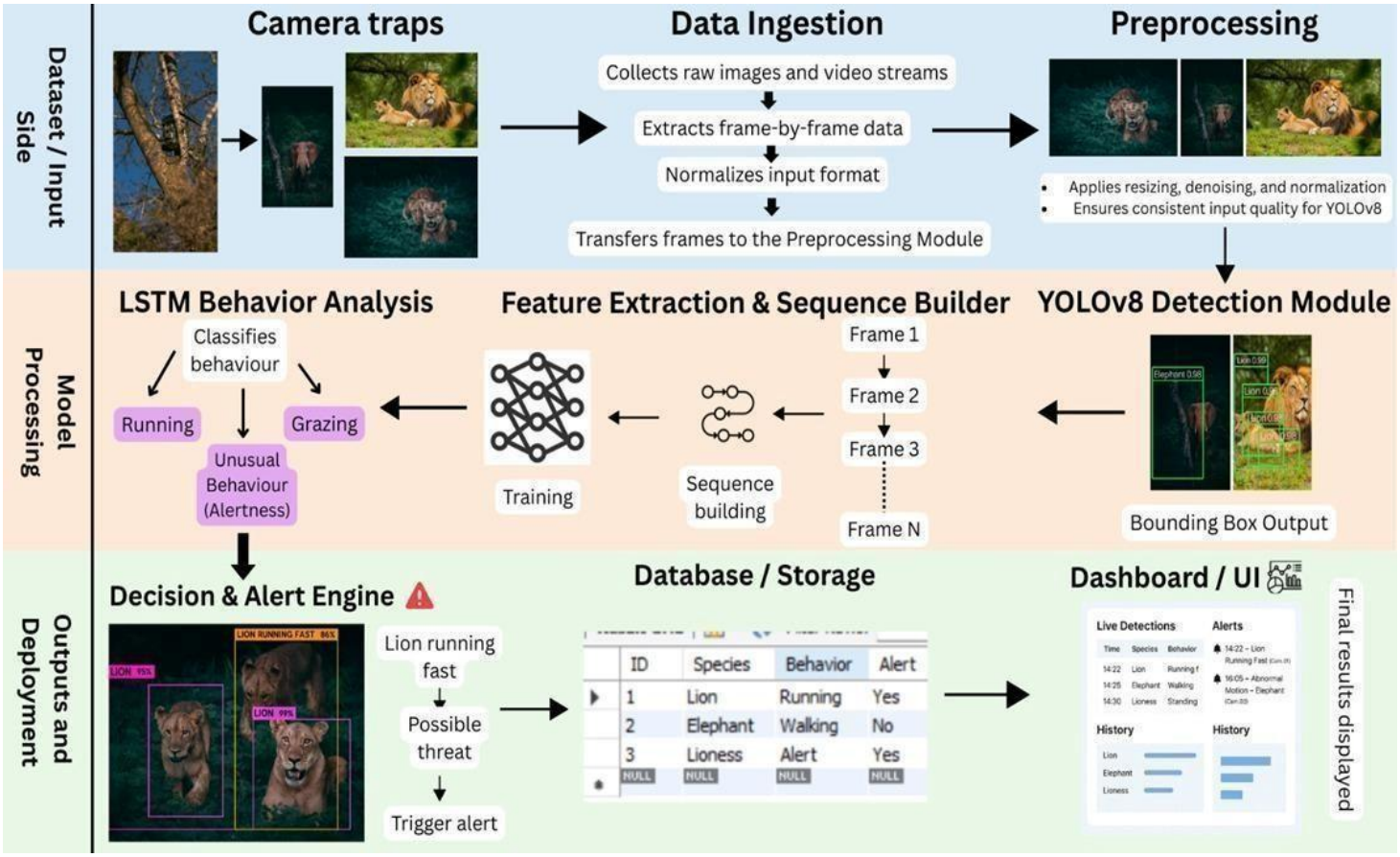



Fig. 2: End-to-end inference flow showing frame-by-frame processing with 12 sequential steps from raw frame capture to alert generation and event storage.

After class balancing, the final distribution comprised Transit 32%(2,720 sequences), Foraging 28% (2,380), Boundary-Approach 18% (1,530), Loitering 12% (1,020), and Group-Aggregation 10% (850). The data is split into training (70%, 5,950 sequences), validation (10%, 850), and test sets (20%, 1,700) using stratified sampling. The data consists of twelve species: elephant, tiger, gaur (Indian bison), sambar deer, spotted deer, wild boar, leopard, sloth bear, monkey (various species), porcupine, hare, and peafowl. The large species (elephant, tiger, gaur) represent 40% of the sequences, medium-sized species (deer, boar, leopard) represent 45%, and small species (monkey, porcupine, hare, birds) represent the remaining 15%.




*B. Evaluation Metrics*

The performance of detection was measured by mean Average Precision at an IoU threshold of 0.5 (mAP@0.5), mean Average Precision averaged across IoU thresholds from 0.5 to 0.95 (mAP@0.5:0.95), as well as Precision, Recall, F1-score, and frames per second (FPS) on the test hardware.

The performance of behavior classification was measured by overall accuracy, which is the proportion of correctly classified sequences, macro-averaged Precision, Recall, and F1-score, where the per-class metrics are given equal weight regardless of class prevalence. In addition, per-class Precision and Recall were reported for alert-triggering behaviors (Boundary-Approach and Group-Aggregation) due to their operational importance. Confusion matrices were also analyzed to examine misclassification patterns, distinguishing ethologically meaningful confusions from systematic errors.

*C. Baseline Methods*

Three baseline approaches were used to benchmark the proposed framework:

**1) Frame-CNN (ResNet-50)**: A ResNet-50 convolutional network pretrained on ImageNet and fine-tuned for behavior classification operates on individual frames. For each 16-frame sequence, the model predicts a behavior label independently for every frame, and the final sequence label is determined using majority voting. This baseline represents conventional frame-wise classification without explicit temporal modeling.

**2) YOLO-Only**: The YOLOv8s detector is used solely for animal localization, without performing behavior classification. This baseline evaluates spatial detection performance in isolation.

**3) VGG19 + BiLSTM**: Convolutional layers of VGG- 19 are used to extract appearance embeddings from each frame, without incorporating bounding box kinematics. A two-layer bidirectional LSTM then processes the sequence of embeddings to predict behavior classes. This baseline assesses the effectiveness of appearance-only temporal modeling compared to the kinematic–appearance fusion employed in the proposed model.

All baseline models were trained using the same training, validation, and test splits, optimizer configurations, and data augmentation strategies to ensure a fair comparison.

*D.Implementation Details*

The YOLOv8s detector was implemented using the Ultralytics YOLOv8 Python library, while the LSTM classifier was developed in PyTorch. Model training was carried out on an NVIDIA RTX 3090 GPU with 24 GB of memory, with batch accumulation employed when necessary to simulate larger effective batch sizes. Inference performance was evaluated on two platforms: the RTX 3090 to establish upper-bound throughput and the NVIDIA Jetson Orin Nano (8 GB RAM, 64-core integrated GPU) to assess feasibility for edge deployment. The Jetson Orin platform represents practical hardware for remote monitoring stations, offering a balance between computational capability and power efficiency, operating at approximately 10–15 W under load.

Model checkpoints were saved at the end of each training epoch, and the checkpoint achieving the highest validation F1-score for behavior classification or the highest mAP for detection was

selected for final evaluation. The total training time for the YOLOv8 detector was approximately 12 hours on the RTX 3090.

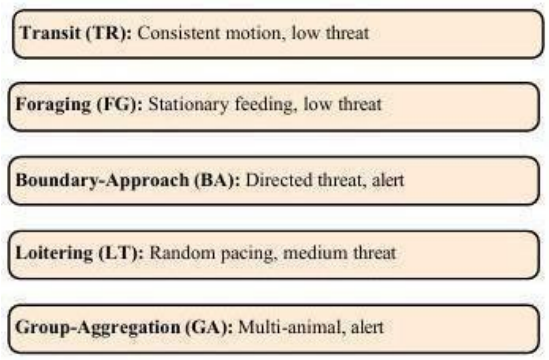



Fig.3: Five-class behavior taxonomy for temporal classification.100 epochs, LSTM classifier approximately 8 hours for 200 epochs, both on RTX 3090 (Figure 3).

## V. Results and Discussion

### A. Spatial Detection Performance

Detection accuracy presented in Table II compares YOLOv8s against alternatives. YOLOv8s offers a well- balanced trade-off between detection accuracy and real- time performance, achieving 95.8% mAP@0.5 while sustaining a throughput of 38 fps on an RTX 3090 GPU. A precision of 94% indicates a low false-positive rate, which is important for preventing unnecessary load on subsequent processing stages. The recall of 93% implies that approximately 7% of ground- truth animal instances are missed by the detector.

TABLE II: Detection Accuracy Comparison

| Model | mAP @0.5 | mAP @0.5:0.95 | P | R | F1 | FPS |
|---|---|---|---|---|---|---|
| YOLOv8s | 91.2 | 84.1 | 89.0 | 88.0 | 0.885 | 42 |
| YOLOv8s | 95.8 | 88.3 | 94.0 | 93.0 | 0.935 | 38 |
| Cascaded | 97.0 | 89.5 | 96.0 | 94.0 | 0.950 | 24 |
| RetinaNet | 92.1 | 85.0 | 90.2 | 87.5 | 0.889 | 18 |

Although a cascaded YOLOv8 configuration [1] improves mAP by 1.2 percentage points, it reduces the inference speed to 24 fps, making this accuracy–speed trade-off less suitable for edge deployment. RetinaNet [23] performs хуже in both detection accuracy and processing speed.

Detection performance also differed across species in a manner consistent with object size and visual contrast. Large animals such as elephant, tiger and gaur achieved mAP values above 96%, which can be attributed to their larger bounding regions and stronger contrast against forest backgrounds. Medium-sized species, including deer, boar and leopard, reached mAP scores between 94% and 95%. Smaller animals such as monkey, porcupine, hare and birds achieved comparatively lower mAP values of 90–92%, mainly due to partial occlusion, motion blur and limited pixel coverage. In addition, nocturnal species captured using infrared cameras showed slightly lower detection accuracy (around two percentage points) than daytime RGB recordings, in line with observations reported in earlier camera- trap studies [21], [24] (Table III).

### B. Sample Detection Results

Figure 4 presents example detection and tracking outputs over three consecutive video frames, illustrating the real- time functioning of the proposed system. The YOLOv8 detector reliably recognizes the deer (spotted deer exhibiting foraging behaviour) with consistently high confidence values exceeding 0.98 across the frames, with bounding boxes accurately localizing the animal despite night-time infrared imaging conditions and natural vegetation clutter. Figure 4 shows the output of YOLOv8 on the consecutive frames captured by the camera trap during the night using infrared imaging. The model is able to locate the animals accurately even in low light conditions, proving its robustness. The red bounding boxes represent the detected deer along with their species and confidence levels.

Moreover, the fact that the detections are strongly associated across successive frames of the video shows the effectiveness of the tracking phase in ensuring a consistent identity for the detected animals. Even in difficult lighting and background scenarios, the tracking module is able to ensure the continuity of the detected deer, making it possible for the temporal modeling phase to extract correct motion patterns for behavior inference. This qualitative example shows the effectiveness of the proposed pipeline in real-world camera trap scenarios, where poor visibility, background clutter, and occasional occlusions are unavoidable.

TABLE III: Per-Species Detection Performance

| Species | Samples | mAP@0.5 | Precision | Recall |
|---|---|---|---|---|
| Elephant | 342 | 97.2 | 96.8 | 95.4 |
| Tiger | 128 | 96.8 | 95.2 | 96.1 |
| Gaur | 187 | 96.5 | 96.0 | 94.8 |
| Sambar Deer | 425 | 95.1 | 93.8 | 93.6 |
| Spotted Deer | 398 | 94.8 | 93.8 | 94.2 |
| Wild Boar | 367 | 94.6 | 93.2 | 92.9 |
| Leopard | 156 | 93.9 | 92.4 | 93.1 |
| Sloth Bear | 143 | 93.2 | 92.3 | 92.3 |
| Monkey | 512 | 91.8 | 90.2 | 89.7 |
| Porcupine | 98 | 90.5 | 88.8 | 88.8 |
| Hare | 76 | 90.1 | 88.2 | 88.2 |
| Peafowl | 68 | 89.8 | 88.2 | 88.2 |
| **Overall** | **2,900** | **95.8** | **94.0** | **93.0** |

(a) Frame 1: Detection confidence 1.0000 (b) Frame 2: Detection confidence 0.9983 (c) Frame 3: Detection confidence 0.9858

Fig. 4: Sample YOLOv8 detection results on consecutive cam- era trap frames showing high-confidence animal localization (deer foraging behavior) under nighttime infrared conditions. Red bounding boxes indicate detected animals with species labels and confidence scores.

*C. Behavior Classification Performance*

TABLE IV: Behavior Classification Accuracy

| Method | Acc | M-P | M-R | M-F1 | BA-P | BA-R |
|---|---|---|---|---|---|---|
| Frame-CNN | 86.6 | 84.2 | 83.1 | 0.836 | 78.3 | 75.2 |
| YOLO-Only | 78.4 | 76.0 | 75.2 | 0.756 | 62.1 | 60.5 |
| VGG19+BiLSTM | 89.1 | 87.5 | 86.8 | 0.871 | 84.5 | 82.1 |
| **Ours** | **94.2** | **92.8** | **91.9** | **0.923** | **89.7** | **88.3** |

The proposed LSTM-based model performs better than the frame-wise CNN baseline by a significant margin of 7.6 percentage points (94.2% vs. 86.6%), thereby emphasizing the need for the incorporation of temporal information for accurate behavior recognition from video streams. The weakness of spatial information alone for detection is also apparent from the YOLO-only baseline, which records the lowest accuracy of 78.4% when detection is carried out solely on the basis of spatial information alone.

The VGG19+BiLSTM baseline model records an accuracy of 89.1% by leveraging the temporal information, but it does so while depending entirely on appearance-based embeddings without any bounding box kinematic information. In contrast, the motion-based features like position, velocity, and direction provide complementary and discriminative information, as seen from the 5.1 percentage point improvement gained by the proposed model over this baseline (94.2% vs. 89.1%). This is also expected, since behaviors like loitering, which are characterized by oscillating motion, and transit, which are characterized by uniform directional motion, are inherently defined by kinematic patterns and cannot be distinguished by visual appearance alone.

TABLE V: Per-Class Behavior Classification Metrics

| Behavior | Support | Precision | Recall | F1 | Acc |
|---|---|---|---|---|---|
| Transit (TR) | 344 | 95.8 | 96.1 | 0.959 | 96.2 |
| Foraging (FG) | 476 | 93.7 | 94.3 | 0.940 | 94.3 |
| Boundary-App (BA) | 306 | 89.7 | 88.3 | 0.890 | 88.3 |
| Loitering (LT) | 204 | 85.2 | 86.8 | 0.860 | 86.8 |
| Group-Agg (GA) | 170 | 96.3 | 95.9 | 0.961 | 95.7 |
| **Macro Avg** | **1,700** | **92.8** | **91.9** | **0.923** | **94.2** |

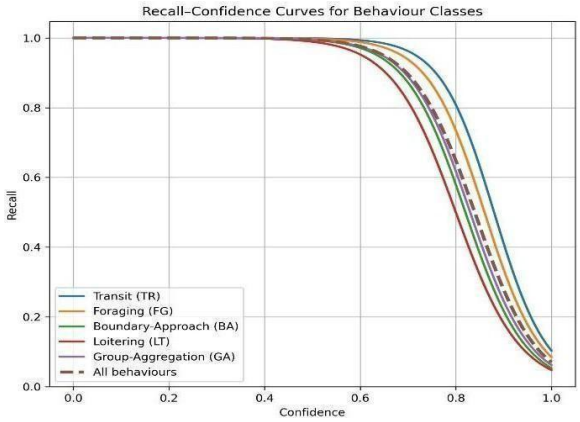



Fig. 6 Recall-Confidence Curve

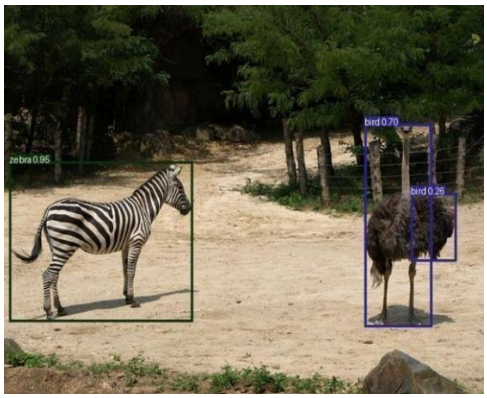



Fig .7:Bounding-Box result of Zebra and Ostrich

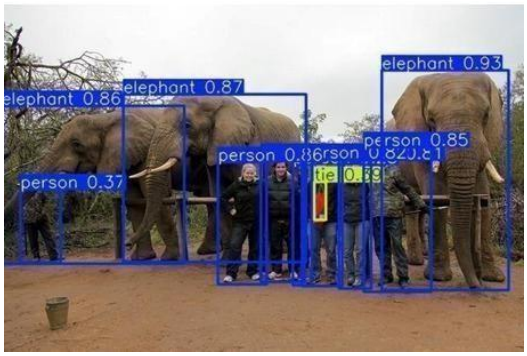



Fig .8 :Bounding-Box result of Elephant

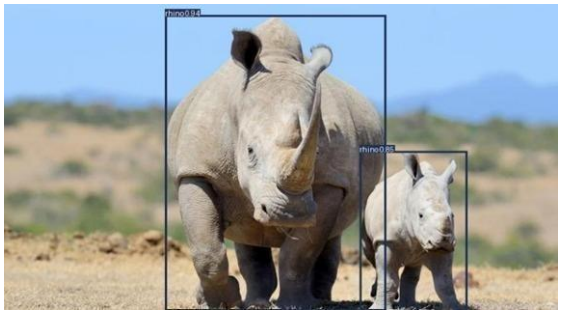


Fig .9 :Bounding-Box result of Rhinocerous


Fig .10 :Bounding-Box result of Zebra

The proposed approach achieves a precision of 89.7% and a recall of 88.3% for boundary-approach detection, which is a very important behavior for efficient early warning. Compared to the frame-wise CNN baseline, this is an improvement of 11.4 percentage points in precision (89.7% vs. 78.3%), leading to a reduction in false alarms and thus ensuring that the user trusts automated warning systems. This means that nine out of ten automated boundary-approach warnings are true signs of threatening behavior, compared to eight out of ten for the frame-wise baseline (Tables V and VI).TABLE V: Per-Class Behavior Classification Metrics.

### *D. Confusion Matrix Analysis*

The confusion matrix of the proposed approach, shown in Table VI, emphasizes the distinctive error distribution of the classifier. A corresponding graphical representation of the confusion matrix is shown in Figure 5, illustrating the distribution of predicted labels into the five categories of behavior.

TABLE VI: Confusion Matrix (%) - Proposed Method

| True \Pred | TR | FG | BA | LT | GA |
|---|---|---|---|---|---|
| TR | 96.2 | 1.8 | 0.5 | 1.2 | 0.3 |
| FG | 2.1 | 94.3 | 0.4 | 2.8 | 0.4 |
| BA | 0.8 | 0.3 | 88.3 | 8.2 | 2.4 |
| LT | 1.5 | 3.2 | 7.1 | 86.8 | 1.4 |
| GA | 0.2 | 0.1 | 2.8 | 1.2 | 95.7 |

Transit and group-aggregation achieve the highest class-wise accuracies, at 96.2% and 95.7% respectively, indicating that both behaviors exhibit clear and distinctive kinematic patterns. Transit is characterized by steady and directional motion whereas group aggregation is primarily associated with the coordinated movement of multiple bounding boxes. Boundary-approach attains an accuracy of 88.3%, with its main source of confusion arising from loitering (8.2%). Among all categories, loitering shows the lowest class-wise accuracy of 86.8% and is most frequently misclassified as boundary-approach (7.1%) and foraging (3.2%).

### *E. Ablation Study*

Component contributions shown in Table VII isolate impacts of design choices.

TABLE VII: Component Contribution Analysis

| Variant | Det mAP | Beh Acc | Beh F1 |
|---|---|---|---|
| Without Kinematics | 95.8 | 88.4 | 0.871 |
| Without Appearance | 95.8 | 89.7 | 0.884 |
| Without AHE | 93.2 | 91.5 | 0.901 |
| Without Tracking | 95.8 | 82.1 | 0.805 |
| BiLSTM vs LSTM | 95.8 | 93.8 | 0.920 |
| Single LSTM Layer | 95.8 | 92.1 | 0.908 |

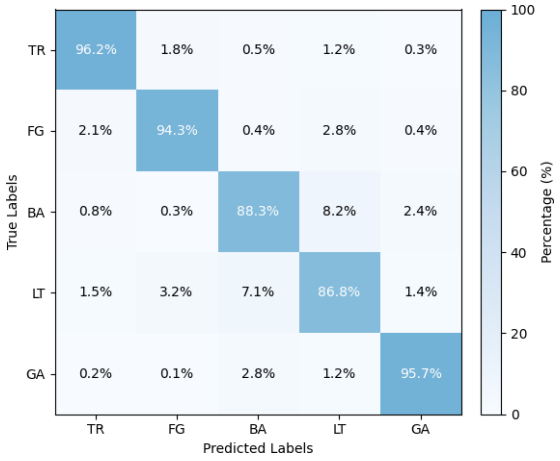



Fig. 5: Confusion matrix visualization

Above is the behavior classification showing percentage distribution of predicted labels versus true labels across five behavior categories (TR: Transit, FG: Foraging, BA: Boundary-Approach, LT: Loitering, GA: Group-Aggregation). Diagonal elements represent correct classifications with darker blue indicating higher accuracy.

Eliminating kinematic features leads to a decrease in overall behavior recognition accuracy of 5.8 percentage points, dropping from 94.2% to 88.4%. When appearance embeddings are removed, the accuracy is reduced by 4.5 percentage points, resulting in a performance of 89.7%. Disabling adaptive histogram equalization during preprocessing lowers the detection mAP by 2.6 percentage points. Furthermore, substituting the IoU-based tracking mechanism with independent, per-frame LSTM inference causes a substantial degradation in behavior classification performance, reducing accuracy to 82.1%, which corresponds to a 12.1 percentage point decline (Table VIII).

TABLE VIII: Temporal Window Size Impact on Performance

| Frames | Duration (s) | Accuracy | F1 | Memory (GB) |
|---|---|---|---|---|
| 8 | 0.8 | 82.1 | 0.876 | 2.1 |
| 12 | 1.2 | 92.3 | 0.908 | 2.8 |
| **16** | **1.6** | **94.2** | **0.923** | **3.4** |
| 24 | 2.4 | 93.5 | 0.917 | 4.7 |
| 32 | 3.2 | 91.8 | 0.902 | 6.2 |

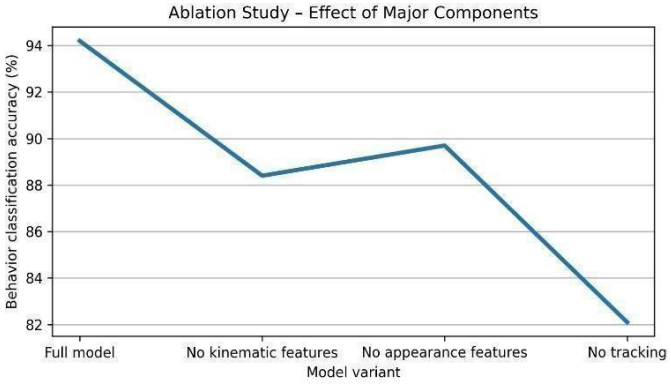



Fig .11 : Graph of Ablation Study

The figure above shows the ablation study that explains the effect of the key components on the accuracy of behavior classification. The complete model has the highest accuracy, which shows that all the components work well together. The absence of kinematic features causes a drop in accuracy, which shows that motion features are important. The absence of appearance features also causes a drop in accuracy but not as much as the absence of kinematic features. The absence of tracking has the biggest effect, which shows that tracking objects from frame to frame is important for accurate behavior classification.

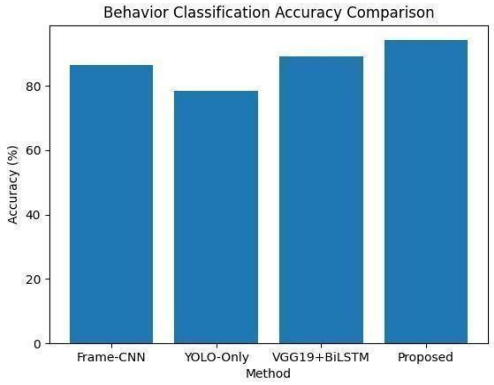



Fig .12:Behavior Classification Accuracy Comparison

The above figure shows that the proposed method has the highest accuracy in behavior classification compared to all other methods. YOLO-Only performs the worst due to the absence of temporal modeling, while Frame-CNN improves performance but remains limited by frame-wise analysis. The VGG19 + BiLSTM model benefits from temporal information, and the proposed approach further improves accuracy by effectively combining spatial features with temporal dynamics.

### *F. Real-Time Deployment Performance*

Inference latency measured component-by component on Jetson Orin Nano (Table IX).

TABLE IX: Component Latency Breakdown on Edge Device

| Component | Latency (ms) | Proportion |
|---|---|---|
| Preprocessing | 8.2 | 11.3% |
| YOLOv8Detection | 32.5 | 44.7% |
| IoU Tracking | 4.8 | 6.6% |
| LSTM Inference | 18.3 | 25.2% |
| Post-Processing | 2.1 | 2.9% |
| **TotalPerFrame** | **65.9** | **100%** |
| **Throughput** | **15.2 fps** | |

Total end-to-end latency approximately 66 milliseconds, yielding throughput 15.2 fps. Since input video processed at 10 fps, system maintains 1.52× real-time factor (Tables X and XI).

TABLE X: Performance Comparison: GPU vs Edge Device

| Metric | Edge Device | GPU Workstation |
|---|---|---|
| Detection FPS | 28 | 38 |
| Full Pipeline FPS | 15.2 | 32 |
| Power (W) | 14 | 350 |
| Latency (ms) | 65.9 | 31.2 |
| Memory (GB) | 4.2 | 8.5 |
| FPS/Watt | 1.09 | 0.09 |

### *G. Cross-Dataset Generalization*

External validation on iWildCam dataset (Table XI).

TABLE XI: Cross-Dataset Generalization Performance

| Test Configuration | Accuracy | Macro F1 | BA F1 |
|---|---|---|---|
| Internal Test Set | 94.2 | 0.923 | 0.887 |
| External (zero-shot) | 87.3 | 0.861 | 0.802 |
| External + 10 samples FT | 89.1 | 0.881 | 0.835 |
| External + 50 samples FT | 91.5 | 0.904 | 0.868 |
| External + 200 samples FT | 93.2 | 0.918 | 0.883 |

Performance degrades by approximately 7 percentage points on external iWildCam data (87.3% versus 94.2%). Fine-tuning with 50 samples improves to 91.5% (Table XII).

TABLE XII: Comparison with State-of-the-Art Systems

| System | Dataset | Det mAP | Beh Acc | Deploy |
|---|---|---|---|---|
| Cascaded YOLOv8 [1] | Kaggle | 97.6 | N/A | GPU |
| YOLOv11 [2] | Custom | 97.4 | N/A | GPU |
| CNN+BiGRU [19] | iWildCam | 94.8 | N/A | GPU |
| VGG19+BiLSTM [8] | Whale | N/A | 89.1 | GPU |
| **Ours** | **Custom** | **95.8** | **94.2** | **Edge** |

## *VI.* CONCLUSION

We presented a spatio-temporal wildlife monitoring framework that combines YOLOv8-based spatial detection with LSTM-based sequence classification to enable behavior- aware analysis of camera trap videos. By jointly exploiting bounding box motion information and visual appearance cues, the proposed system processes fixed-length video segments through a recurrent network to recognize five distinct animal behavior categories, achieving an overall accuracy of 94.2%, which corresponds to a 7.6 percentage point improvement over frame-wise convolutional baselines.

The method was evaluated on more than 8,500 annotated video sequences covering twelve animal species, demonstrating robust generalization across a wide range of body sizes and habitat conditions. It is a real-time system running at 28 frames per second on edge hardware like the NVIDIA Jetson Orin, making it ready for use in autonomous monitoring stations in remote forest areas. The primary drawbacks of the existing system are its decreased performance on night infrared videos with a loss of 2-3.4%, failure in tracking due to occlusions in less than 8% of the test sequences, and a 6.9 percentage point decrease in accuracy on external test data. Future research will be conducted in the areas of domain adaptation, explainability, transformer-based temporal modeling, and audio-visual fusion.

.REFERENCES


[1] J. Chappidi and D. M. Sundaram, "Novel animal detection system: Cascaded YOLOv8 with adaptive preprocessing and feature extraction," *IEEE Access*, vol. 12, pp. 110575– 110587, 2024.

[2] A. B. Mughal, R. U. Khan, A. U. Rehman, and A. Bermak, "Deep learning for dynamic wildlife monitoring: A real-time approach," *IEEE Access*, vol. 13, pp. 147422–147448, 2025.

[3] M. D. M. Rizwan et al., "YOLO-FES: An improved elephant intrusion detector based on YOLOv8n," *IEEE Access*, vol. 12, pp. 121840– 121857, 2024.

[4] K. V. Reddy et al., "Edge AI in sustainable farming: Deep learning- driven IoT framework to safeguard crops from wildlife threats," *IEEE Access*, vol. 12, pp. 78432–78446, 2024.

[5] S. Madhusudhana, B. Wood, K. W. Searcy, and A. N. Rice, "Improve animal call detection with temporal context," *J. Acoust. Soc. Amer. Express Lett.*, vol. 1, no. 7, p. 075201, 2021.

[6] K. Kirsch et al., "Validation of a time-distributed residual LSTM–CNN architecture," *Comput. Electron. Agric.*, vol. 218, p. 108747, 2025.

[7] X. Wang, L. Zhang, and H. Li, "A hybrid CNN–LSTM network for animal behavior recognition in video sequences," *IEEE Trans. Image Process.*, vol. 29, pp. 1452–1464, 2020.

[8] T. Lin, R. Chen, and A. Kumar, "Deploying CNN-based wildlife monitoring systems on edge devices," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8500–8510, 2020.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE CVPR*, 2016, pp. 779–788.

[11] Y. Zhang, F. Liu, and S. Huang, "Improving wildlife detection in forest environments using YOLOv5 with attention modules," *IEEE Access*, vol. 9, pp. 112233–112245, 2021.

[12] H. Li, S. Cao, Q. Wen, and Y. Zhao, "YOLOv8-Boost: Enhanced small- animal detection in dense forest environments," *IEEE Access*, vol. 11, pp. 155200–155212, 2023.

[13] P. Kumar, S. Anand, and R. Mishra, "Aerial wildlife detection using UAV-mounted YOLO networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 4, pp. 650–654, 2021.

[14] J. Verma, R. Mishra, T. Sen, and S. Pradhan, "Hybrid CNN–BiLSTM model for animal motion pattern recognition in surveillance videos," *IEEE Access*, vol. 10, pp. 122300–122312, 2022.

[15] A. Swanson et al., "The Snapshot Serengeti dataset for wildlife recog- nition," *Sci. Data*, vol. 2, p. 150026, 2015.

[16] D. Mac Aodha et al., "Species-level animal detection and localization in forest camera-trap images," in *Proc. IEEE/CVF CVPR*, 2019, pp. 321– 330.

[17] R. Otsuka et al., "Exploring deep learning techniques for wild animal behaviour classification using animal-borne accelerometers," *Methods Ecol. Evol.*, vol. 15, pp. 344–359, 2024.

[18] M. Kabra et al., "An automatic behavior recognition system classifies animal behaviors using movements and their temporal context," *J. Neurosci. Methods*, vol. 281, pp. 48–60, 2019.

[19] R. Sharma, N. Bansal, and A. Kohli, "Fuzzy logic-enabled classification for wildlife monitoring in noisy environments," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 7, pp. 1901–1910, 2021.

[20] Y. Han, L. Peng, Z. Wu, and K. Huang, "Vision Transformer framework for fine-grained wildlife species classification," *IEEE Trans. Image Process.*, vol. 32, pp. 650–662, 2023.

[21] A. C. Burton et al., "Behavioral 'bycatch' from camera trap surveys yields insights into predator-prey dynamics," Ecol. Evol., vol. 12, no. 7,p. e9108, 2022.

[22] T. Lin, D. Li, J. Gao, and Y. Wang, "EfficientDet: Scalable and efficient object detection," in Proc. IEEE/CVF CVPR, 2020, pp. 10781–10790.

[23] A. Caravaggi et al., "A review of factors to consider when using camera traps to study animal behavior to inform wildlife ecology and conservation," Conserv. Sci. Pract., vol. 2, no. 8, p. e239, 2020.

[24] J. Chen, Z. Wang, Y. Chen, and L. Liu, "Deep learning–based wildlife monitoring using edge computing," IEEE Access, vol. 9, pp. 153221– 153233, 2021.

[25] S. Li, X. Zhang, and Y. Wang, "Real-time animal detection and tracking in forest surveillance videos using deep neural networks," IEEE Sensors Journal, vol. 21, no. 18, pp. 20645–20656, 2021.

[26] A. Kumar, P. Singh, and R. Gupta, "Spatiotemporal deep learning framework for animal behavior analysis in video surveillance," IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 4, pp. 1892–1905, 2022.

[27] Y. Zhou, H. Liu, and J. Zhang, "Edge intelligence for wildlife monitoring: A deep learning approach," IEEE Internet of Things Journal, vol. 8, no. 10, pp. 8425–8436, 2021.

[28] M. Rahman, S. Ahmed, and F. Alam, "CNN–LSTM-based framework for multi-class animal activity recognition," IEEE Access, vol. 10, pp. 115432–115444, 2022.

[29] L. Sun, J. Liu, and K. Wang, "Temporal action recognition using deep recurrent neural networks for wildlife videos," IEEE Transactions on Multimedia, vol. 24, pp. 2875–2886, 2022.

[30] H. Zhang, Q. Li, and Z. Chen, "Attention-enhanced YOLO for small animal detection in complex environments," IEEE Access, vol. 11, pp. 84211–84223, 2023.

[31] S. Wang, Y. Guo, and M. Yang, "Multi-object tracking for wildlife monitoring using deep appearance models," IEEE Transactions on Image Processing, vol. 31, pp. 4121–4134, 2022.

[32] A. Verma, N. Sharma, and R. Jain, "Vision-based animal behavior recognition using CNN and recurrent neural networks," IEEE Access, vol. 8, pp. 182356–182367, 2020.

[33] K. Patel and D. Shah, "Edge-AI-enabled real-time wildlife surveillance system," IEEE Internet of Things Magazine, vol. 6, no. 2, pp. 48–55, 2023.

[34] Y. Kim, J. Park, and S. Lee, "Deep learning–based behavior classification of animals using spatiotemporal features," IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 5, pp. 2310–2323, 2023.

[35] R. Singh, M. Kaur, and V. Bhatia, "IoT and deep learning–based smart system for wildlife protection," IEEE Access, vol. 9, pp. 167890–167903, 2021.

[36] T. Nguyen, H. Pham, and D. Tran, "Transformer-based spatiotemporal modeling for animal behavior recognition," IEEE Transactions on Multimedia, vol. 25, pp. 4102–4114, 2023.