Characterizing and Modeling AI-Driven Animal Ecology Studies at the Edge

Jenna Kline, Austin O'Quinn, Tanya Berger-Wolf, and Christopher Stewart
Department of Computer Science and Engineering
The Ohio State University
{kline.377, oquinn.18, berger-wolf.1}@osu.edu, cstewart@cse.ohio-state.edu

2024 IEEE/ACM Symposium on Edge Computing (SEC) | 979-8-3503-7828-3/24/$31.00 ©2024 IEEE | DOI: 10.1109/SEC62691.2024.00025

***Abstract*— Platforms that run artificial intelligence (AI) pipelines on edge computing resources are transforming the fields of animal ecology and biodiversity, enabling novel wildlife studies in animals' natural habitats. With emerging remote sensing hardware, e.g., camera traps and drones, and sophisticated AI models in situ, edge computing will be more significant in future AI-driven animal ecology (ADAE) studies. However, the study's objectives, the species of interest, its behaviors, range, and habitat, and camera placement affect the demand for edge resources at runtime. If edge resources are under-provisioned, studies can miss opportunities to adapt the settings of camera traps and drones to improve the quality and relevance of captured data. This paper presents salient features of ADAE studies that can be used to model latency, throughput objectives, and provision edge resources. Drawing from studies that span over fifty animal species, four geographic locations, and multiple remote sensing methods, we characterized common patterns in ADAE studies, revealing increasingly complex workflows involving various computer vision tasks with strict service level objectives (SLO). ADAE workflow demands will soon exceed individual edge devices' compute and memory resources, requiring multiple networked edge devices to meet performance demands. We developed a framework to scale traces from prior studies and replay them offline on representative edge platforms, allowing us to capture throughput and latency data across edge configurations. We used the data to calibrate queuing and machine learning models that predict performance on unseen edge configurations, achieving errors as low as 19%.**

***Index Terms*—autonomous systems, Edge AI, imageomics, drone, camera trap, animal ecology, distributed inference**

I. Introduction

Camera traps and drones can automatically capture visual data on animals, their morphology and behaviors, and biodiversity within an ecosystem, transforming the fields of animal ecology and biodiversity (Figure 1). It is now common for field-based animal ecological studies to use more than 70 camera traps [14]. Between 2015 and 2020, at least 19 academic studies were driven by aerial drone imagery [16]. Drones are especially promising for animal behavior studies that require tracking wildlife over vast, remote landscapes [19], [43], [46], [57], [61]. Computer vision and machine learning approaches have sped up post hoc processing for visual data collected in the field [13], [43], [56], [69], [74], [79]. However, as camera traps and drones flood ecologists with data, it is challenging to curate, process, and manage the data to discover ecological insights in a timely fashion [21], [73], [80]. Further, AI pipelines that infer complex ecological traits require images with prescribed pixel resolution, angles, and timing: factors related to data quality that is determined at runtime. Images with low resolution or occlusions require expert analysis to decipher insights or must be discarded altogether.

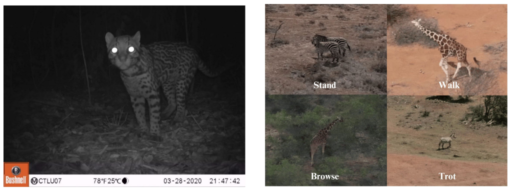

Fig. 1. Data captured from animal ecology studies in the field: 1) A camera trap captures large cats' species range and nightly activities, adapted from [71]. 2) Drones capture animal behavior and poses, adapted from [43].

Edge AI, the application of AI pipelines on edge computing systems [66], can enable AI-driven animal ecology (ADAE) studies. ADAE studies control remote sensing systems at runtime, filtering images, adjusting angles, and changing camera or drone positions to improve data quality through adaptive sampling [15], [17], [21], [27], [73], [80]. Ecologists are beginning to use edge AI platforms to conduct ADAE studies using networks of smart camera traps [1], [3], [70]. Drones are innately adaptive if they are piloted well. Edge AI can reduce the burden on pilots, allowing ADAE studies to employ multiple drones, capture data from vast areas, and improve data quality [9], [10], [45], [50].

Animal ecology studies collect data from predefined locations by placing camera traps or flying predefined missions. Our study provides a critical insight: *adaptive data collection enabled by edge computing can improve study efficacy.* We present the first characterization of ADAE studies. ADAE study workflows employ image analytics at the edge, composing inter-dependent, inference pipelines from complex AI computer vision models [37], [78]. These workflows must be executed under strict SLOs to support runtime adaptations. In this paper, our contribution is a characterization of ADAE studies, their definitive features, workload demands, and the factors affecting their performance.

The remainder of the paper is organized as follows. We

2837-4827/24/$31.00 ©2024 IEEE
DOI 10.1109/SEC62691.2024.00025

characterize ADAE workloads in Section II and describe our methodology to model and scale ADAE workloads in Section III. Section IV describes our framework for experimenting on representative hardware and analyzes factors affecting SLO attainment for prior studies. Section V examines performance modeling for ADAE studies. Section VI reviews related works. Section VII summarizes our findings and future work.

II. Characterization of ADAE Studies

We analyze datasets from prior AI-driven animal ecology (ADAE) studies to characterize their workloads [43], [71]. However, instead of searching for ecological insights, we examine when the data was collected and what software components were triggered using timestamps provided by camera traps and drones. To our knowledge, this work is a first attempt to apply ADAE study traces to profile the characteristic workload demands from an edge perspective. Our analysis of the frequency and timing of timestamps reveals that image capture and subsequent computational triggers occur in bursts. Further, approaches to expand a study's geographic footprint affect the magnitude of bursts. Timestamp analysis also revealed the latency window for edge computing systems to make runtime adaptations to improve data quality. We adapted service-level-objectives (SLO) to characterize ADAE study demands a widely used paradigm in cloud computing.

A. ADAE workflow: design, execution, and results

We illustrate the canonical workflows of ADAE studies in Figure 2. The ADAE workflow comprises three phases: design, execution, and results. The design phase consists of establishing the study objective and study parameters. The ADAE study objectives include the location, species of interest, AI methods used, and ADAE hypothesis. The study parameters include the remote sensing hardware used, such as drones or camera traps, the AI sensitivity settings, and the edge resource provisioning strategy. The ADAE execution workflow includes four subphases: (1) animal dynamics, (2) generic image processing, (3) study-specific feature extraction, and (4) runtime adaptations. The final phase produces the results, where the dataset has been collected and is ready for analysis.

The first subphase of ADAE study execution is *animal dynamics*. This includes collecting imagery data with drones or camera traps of the animals of interest and extracting the collected frames for analysis. In this phase, the data arrival rate is dictated by the behavior of the species of interest and its interactions with the camera trap and drone hardware. The second subphase is *generic image processing*. This includes detection and localization computer vision tasks to answer the following questions: Is there an animal in this frame? If so, where is the animal located in the frame? The classification computer vision task may be viewed as a component of the generic image processing tasks if required for a downstream task, like individual identification. Classification may also be considered a study-specific feature extraction if used to complete a biodiversity ADAE study. Commonly used computer vision models for detection, localization, and classification tasks for ADAE include YOLO [39] and PyTorch Wildlife [28].

The third subphase is *study-specific feature extraction*. This includes computer vision tasks to infer information, including the animal's tracks, posture, behavior, and individual identification [4]. These study-specific feature extraction tasks inform the fourth subphase, *runtime adaptations*, which may include camera relocation, adjusting the sampling duration, and updating the edge resource management to respond to the workload demands. The runtime adaptation instructions are returned to the data collection module, and the ADAE study continues execution until sufficient data has been collected.

Unlike traditional field ecological studies, ADAE workloads require computational resources provisioned at the edge. Like traditional studies, ADAE studies can fail because the data is insufficient to support or reject the hypothesis. However, ADAE studies only succeed if the edge platform can make runtime adaptations quickly enough to capture high-quality study-appropriate data. We do not claim that our study is representative of all ADAE studies, but the observed characteristics are well-motivated and will likely generalize to future studies.

B. Required SLOs for runtime adaptations

The latency requirement for the ADAE study is dictated by the service-level-objective (SLOs) of the computer vision pipeline used to gather the data and inform runtime adaptations. The computer vision pipeline for ADAE workloads is illustrated in Figure 2. SLOs for specific ADAE studies are detailed in Table I. An additional component of the SLO is the rate of requests met or the percentage of inference requests that must be met. The required rate of requests met varies depending on the computer vision tasks and study parameters. Depending on the edge hardware available and the study design, some computer vision tasks may have strict SLO. At the same time, other pipeline components may be offloaded to the cloud for post-hoc analysis. The average number of frames, from photos or video, that the computer vision pipeline must process per second determines the SLO requirements. Depending on the study design, the pipeline may include one or more computer vision models to accomplish different tasks. For computer vision pipelines that navigate autonomous drones for ADAE, the SLO will be stricter than studies using camera traps [9]–[11]. For camera trap studies, the SLO is set to handle the real-time data analysis and processing needed to inform runtime adaptations, such as the frequency and sampling duration.

A secondary benefit of the ADAE approach is that it can enable near-real-time ecological insights instead of solely re-

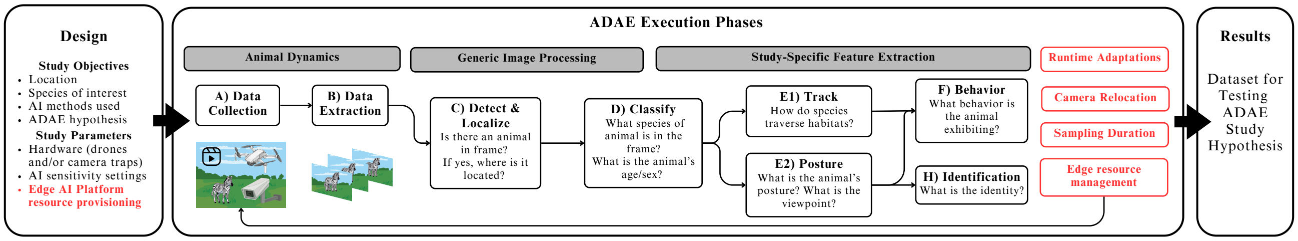
Design
Study Objectives
• Location
• Species of interest
• AI methods used
• ADAE hypothesis
Study Parameters
• Hardware (drones and/or camera traps)
• AI sensitivity settings
• Edge AI Platform resource provisioning
ADAE Execution Phases
Animal Dynamics
Generic Image Processing
Study-Specific Feature Extraction
Runtime Adaptations
A) Data Collection
B) Data Extraction
C) Detect & Localize
Is there an animal in frame?
If yes, where is it located?
D) Classify
What species of animal is in the frame?
What is the animal's age/sex?
E1) Track
How do species traverse habitats?
E2) Posture
What is the animal's posture? What is the viewpoint?
F) Behavior
What behavior is the animal exhibiting?
H) Identification
What is the identity?
Camera Relocation
Sampling Duration
Edge resource management
Results
Dataset for Testing ADAE Study Hypothesis

Fig. 2. Canonical workflow for field AI-driven animal ecology (ADAE) studies.

ID	Location	Hardware	Species	Data	AI Computer Vision Tasks	ADAE Result	Latency sec/frame	SLO Requests Met
1	Yellowstone Park, USA	Single fixed-wing drone	Bison	Photo	Detect, localize	Count of calves in herd	0.4	50%
2	Zimbabwe	Single-fixed wing drone	Multiple species	Photo	Extract frames, detect, localize, classify	Detect endangered species	1.0	99%
3	Kenya	Single quadcopter	Giraffe	Photo	Detect, localize, classify, track	Count by habitat type	1.0	80%
4	Kenya	Quadcopter swarm	Zebra	Video	Extract frames, detect, localize, track	Behavioral time budgets	1.0	80%
5	Conservation Center, USA	Single smart camera trap	African Wild Dogs	Video	Extract frames, detect, localize, track	Behavior by time of day	0.03	95%
6	Columbia	Single smart camera trap	Multiple Species	Photo	Detect, localize, classify	Species distribution and population estimates	180	99%
7	Columbia	Camera trap network	Multiple Species	Video	Detect, localize, classify, track	Behavior & individual identification	0.03	95%

TABLE I
AI-DRIVEN ANIMAL ECOLOGY (ADAE) STUDIES SERVICE-LEVEL OBJECTIVES (SLOS)

lying on post hoc processing techniques. Here, near real-time means completing a request to process an image in minutes to hours (versus days to months) from when the image was first collected. If no edge processing is used, the large volumes of imagery captured must all be processed offline after the study is concluded, which may take months or years to analyze. Moving the detection and localization computer vision tasks to the edge hardware reduces the amount of data that must be analyzed offline. For example, the Orinoquía camera trap dataset contains 20% blank imagery [71]. Current state-of-the-art for near real-time camera trap image processing is an average of 7.35 minutes per image for a network where each camera produced 17 images per day on average [73].

C. Representative ADAE studies

We describe seven representative ADAE studies in Table I. We use the hardware, species of interest, data type, computer vision tasks, and desired outputs to define the SLO requirements for the ADAE studies. ADAE 1 uses a fixed-wing drone to survey bison to count the number of calves present in the herd [16]. Counting the number of young is an important data point to quantify the success of conservation efforts to repopulate this species in the American plains. We assume the fixed-wing drone uses the default settings for a survey mission to generate an orthomosaic image of the herd: nadir-view, 60 m altitude, 1.3 cm/pixel resolution, with a 75% front overlap and 70% side overlap [18]. This flight plan generates approximately one image every 0.2 seconds, however, due to the overlap in images, it is sufficient to analyze 30% of the images received and still be confident that the bison are in view of the drone. ADAE 2 also uses a single fixed-wing drone to detect the presence of endangered wildlife in a conservation area. This approach is similar to the SPOT Poachers in Action study [35], which reported an average latency rate of approximately 1 second per frame with a GPU. Endangered animals with low population levels may be rarely spotted, therefore, the minimum requests met for such studies is high to ensure the frames containing rare or endangered species are not missed.

ADAE 3 uses a single quadcopter drone to count the number of giraffes present in different habitats in Kenya by detecting, localizing, and classifying the animals. The imagery collected by the drone is also used to classify the habitat as open or closed, categorized by the amount of vegetation present. A quadcopter drone was selected for this study because, unlike fixed-wing drones, it can more easily navigate around occlusions from vegetation in closed habitats. Group-living animals may be autonomously tracked with drones using a detection and localization model, such as YOLO, integrated into the control software [44]. This autonomous herd-tracking navigation model requires a 1-second per frame latency and a tolerance of 80%. The frame rate may be adjusted depending on the average speed of the species of interest. If the SLO is violated, the drone may lose sight of the animals, forcing the data collection mission to end prematurely. ADAE 4 scales ADAE 3 by implementing the herd-tracking navigation pipeline with a swarm of multiple quadcopters. The aim of ADAE 4 is to collect videos of zebra herds to study their behavioral differences by the time of day, similar to the methodology used in the KABR study [43]. As this study collects behavior videos of group-living animals instead of photos of a single species, it requires a longer sampling time compared to ADAE 3 but maintains the same SLO.

ADAE 5 represents a single smart-camera trap study that collects video behavior data of a pack of African Wild Dogs at a wildlife conservation center. This study uses a motion-activated camera to trigger video recording if an animal is detected in view. The camera tracks the animal until it is out of sight. This methodology for collecting behavior videos with motion-activated smart camera traps has been successfully used to collect large-scale ape behavior datasets [13], [60]. For ADAE 5, the recording duration is the essential runtime adaptation, which depends on the accuracy of the tracking step (E1 shown in Figure 2). For this study, African Wild Dogs are the only species in the enclosure, so species classification is unnecessary.

ADAE 6 uses a single, smart camera trap to collect photos to estimate species distribution and population estimates for a biodiversity study. It is assumed an average of 20 images are collected each hour, so the SLO for completing the CV pipeline is 3 minutes per image, or 600 seconds per frame, to prevent a queue from being formed. ADAE 7 independently scales the study from ADAE 6 by adding additional smart camera traps distributed geographically to estimate species distributions and populations over a wider area, similar to the Orinoquía Camera Trap study [71], which we obtained from LILA BC repository online [2].

ADAE 8 scales the study from ADAE 6. Instead of placing camera traps distributed geographically, it places additional camera traps in the same spot, which produces correlated scaling. This correlated scaling approach is better suited for behavioral studies (F from Figure 2) and studies requiring individual identification (H from Figure 2) using a tool like WildMe [4]. AI computer vision models to classify behavior and identify individual animals benefit from having access to views of the animal(s) from multiple angles, which requires a correlated scaling approach.

D. ADAE workloads in Edge AI research

Our group and others collected the ADAE traces profiled in this work. However, all ADAE traces profiled contained two essential components. One, imagery data in the form of videos or photos. Two, timestamped arrival rates for the imagery data associated with request arrivals for the ADAE computer vision tasks. Arrival rates for computer vision tasks depend on the specific ADAE study objectives, as described in Table I. An essential contribution of our effort is discovering commonalities that enabled rigorous analysis. Edge systems researchers can leverage the SLOs described in this section to explore new distributed computing techniques designed for use in field ADAE studies. By analyzing the workload patterns revealed by timestamp data, researchers can focus on optimizing SLOs for edge computing systems. This could include developing techniques to predict and manage latency for ADAE-specific computer vision tasks and ensuring runtime adaptations occur within the required time frame for improved data quality.

III. Modeling ADAE Workloads

Our workload characterization of ADAE studies in Section II suggests that their computational demand will exceed the capacity for individual edge devices deployed in remote settings. Meeting latency and throughput goals will require assessing edge configurations before deployment and predicting their performance before resources are provisioned. Evaluating proposed configurations in situ is challenging due to ethical, logistical, and resource constraint considerations; thus, we present a framework to enable offline evaluation.

We model ADAE workloads using the characteristic request arrival rates generated by these studies described in Section II. We describe study features that affect ADAE workload burstiness: ecological factors, camera placement and scaling, and hardware and AI model considerations. We describe our methodology to characterize and quantify these bursty workloads as a time-varying Poisson process. We also provide code to profile and scale real-world ADAE studies along with worked examples here: https://github.com/jennamk14/adae_model.

A. Factors driving burstiness

Computer vision model workloads often exhibit burstiness, with periods of high activity followed by low or no activity intervals, depending on the ADAE study parameters [72]. These study parameters include the type and location of the sensors, the AI models used, and the habitat and behavior of the species of interest. Bursty workloads describe those in which request arrival times are unpredictable, but there is also a high degree of covariance between requests. Previous studies demonstrate that autonomous navigation models that track and monitor animals using drones produce bursty workloads [43]. Bursts of high arrival rates increase the queuing times and processing delays, potentially violating SLO. When determining SLO for ADAE studies, we aim to minimize queuing delays to meet the latency requirements. Thus, the bursty nature of these workloads must be considered when designing systems to meet these requirements.

1) Ecological factors: Various factors can influence burstiness, including species-specific activity patterns, seasonal variations, and inter-species interactions. Species that share the same space may actively interact with each other (e.g., predation or resource competition), neutrally coexist (e.g., mixed-species groups of ungulates in the Serengeti [65]), or actively avoid each other (e.g., tigers and leopards [41]). Active species interactions are rare and require overlapping workloads covering the potential interaction occurrence area. Neutral coexistence leads to overlapping detection and classification model workloads. Finally, active avoidance leads to mostly non-overlapping workloads.

Species-specific activity patterns influence burstiness. By definition, diurnal species are active during the day, and crepuscular species are active during dawn and dusk, generating camera trap captures during different times of day. Thus, quantified using the coefficient of variation metric, diurnal patterns would considered bursty. However, animal ecology patterns are less pronounced and predictable than diurnal patterns in cloud or e-commerce systems.

2) Camera placement and scaling strategies: Placing camera traps and drones significantly shapes the workload dynamics and determines the appropriate scaling strategies. Two common scaling approaches are 1) Independent scaling, which distributes more cameras over a large area, and 2) Correlated scaling, which increases the camera density in specific locations. Independent scaling is typically used to study wide-ranging species, such as wolves or migratory ungulates, to understand their landscape-level movements and habitat preferences [54], or studying the spatial distribution of sympatric species, such as jaguars and pumas [26]. By distributing the camera traps or drones over a large area, independent scaling is suitable for studies focusing on species distribution, habitat use, or landscape-level interactions [59]. Independently scaling the hardware increases the spatial coverage and captures a broader range of animal activities, reducing the burstiness of the workload. However, it may lead to increased workload overlap as different species' territories or movement patterns are more likely to be captured simultaneously.

Correlated scaling by increasing the camera density in specific locations is appropriate for studies focusing on fine-scale animal behavior using drones or camera traps [13], [57], social behavior and group dynamics of species like zebras [43], chimpanzees, or African elephants, which require detailed observations at specific sites [47]. Inter-species interactions, or monitoring hotspots of activity [42], such as water holes or mineral licks where multiple species congregate, allowing for the study of inter-species interactions and temporal partitioning of resources [36]. This approach increases burst intensity during events, as multiple cameras capture the same activity from different angles or close succession.

3) Hardware and AI models: The type of hardware, e.g., smart camera traps or drones, and the AI computer vision models used to analyze the data impact the study's workflow. Camera traps and drones may continuously record data, generating a constant data stream for analysis. Or, more commonly for camera trap studies, use a motion or heat-activated sensor to capture photographs or videos only when an animal is present [24]. Fixed-wing drones are typically deployed to survey extensive, remote areas and capture photographs, which are analyzed to detect and classify the animals [12], [32]. The workflow generated by fixed-wing drone missions depends on the frequency at which the drone captures animals, which is impacted by the habitat and species of interest. Quadcopter drones are smaller and more agile, allowing them to follow groups of animals and quickly navigate to capture a variety of angles, which are particularly effective for behavior studies [43], [45]. The autonomous navigation models used to pilot these drones often exhibit bursty characteristics [44].

Number of Cameras	ADAE Study (Table I)	CoV
Single smart camera trap	5	1.59
Smart camera trap network	7	7.33
Single quadcopter	3	3.01

TABLE II
QUANTIFYING BURSTINESS OF ADAE STUDIES

The workflow of the ADAE study is also impacted by the computer vision tasks performed on the edge to enable the required runtime adaptations, as shown in Figure 2. Adjusting recording duration in camera trap studies can be dynamically adapted based on the species or behavior detected. For drone studies, runtime adaptations include the navigation decisions based on the detected species or behavior [43], [45], [53].

B. Methodology for modeling workloads

We model workloads as time-varying Poisson processes, with rate changes identified at key inflection points. This allows the scaling of traces while maintaining the burstiness characteristics. This method preserves the realism of animal ecology workloads while enabling the testing of different hardware and configurations through workload generation based on real-world traces.

1) Quantifying burstiness: We characterize the burstiness of ADAE workloads by the time each burst arrives, $t \in T$, burst duration μ, and arrival rate within a burst λ. We quantify the burstiness of ADAE using the coefficient of variation, a metric commonly used to characterize bursty arrival rates [5], with results shown in Table II. The burstiness of three representative ADAE studies are visualized in Figure 3, where portions with a relatively larger gradient represent a burst. A single, smart camera trap (ADAE 5) exhibits the least burstiness, as there are only three change points where the arrival rates change dramatically, reflected in the CoV score of 1.59. ADAE 3 with a single quadcopter exhibits comparatively more bursts than ADAE 5, where the gradient

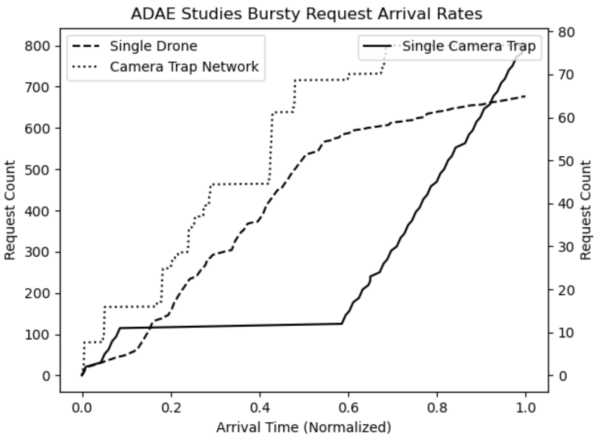

Fig. 3. Modeling the burstiness of ADAE studies. The request count for the Single Drone and Camera Trap network traces are shown on the left y-axis, and the request count for a Single Camera trap is shown on the right y-axis. Arrivals are normalized with respect to time- the single camera trap arrival data has a 24-hour duration, while the drone and camera trap arrival times have a 2-hour duration.

increases more rapidly with a CoV of 3.01. A network of smart camera traps, ADAE 7, exhibits the highest level of burstiness, visualized in Figure 3 as steep gradients where the arrival rate increases rapidly, which is reflected in its CoV of 7.33. In practice, these workloads may scale, for example, from 40% to 80% utility of a single node, due to independent or correlate scaling, as discussed in Section III-A2.

2) Modelling and scaling bursty workloads: We model the bursty workloads as Poisson processes with rate variations at given change points to preserve the characteristic burstiness in the arrival rates. To model the traces as a time-varying Poisson process, we identified the inflection points of the arrival rate gradient, denoted as change points. These change points define trace segments and the average arrival rate λ was calculated for each segment duration. To scale the traces, we multiplied all λ in the trace by a factor to generate the desired average utilization. This approach allows to capture the expected scaling from different animal ecology studies. For example, better cameras with a higher frame rate, larger models, or slower hardware can be modeled while maintaining the shape of our arrival curve. The scaling approach focuses on the relative parametrization, normalizing among realistic animal ecology study characteristics while testing different configurations.

3) Workload generation: The inputs of our bursty workload generator is the total time of the simulation T, and rate R_t, where t is the change point, and $t \in T$. The process to generate bursty arrival times from real-world traces is illustrated in Algorithm 1. For each unique change point value t, we calculate the duration of the R_t, corresponding to t. The arrivals are generated by randomly sampling the Poisson distribution of R_t multiplied by the duration. Next, the arrivals are uniformly sampled for the timestamps for the duration of the rate R_t. These arrivals are sorted and added to the array t_a. Finally, the arrival times for the last change point's interval t are generated similarly.

Algorithm 1 Bursty arrival times modeled as Poisson process with rate variation

```
Data: $T$ (total time of the simulation), $R_t$ (rate), where $t$ are
      the change points, and $t \in T$.
Result: List of bursty arrival times.
$t_s \leftarrow []$ */// start time of current duration
$t_a \leftarrow []$ */// arrival times
for $i$ in $t$ do
    $rate = R_i$
    $duration = i - t_s$
    $arrivals \overset{\text{R}}{\leftarrow} Poisson(rate * duration)$
    $arrivals \overset{\text{R}}{\leftarrow} U(t_s, t_s + duration, arrivals)$
    $t_a \leftarrow$ sort $arrivals$
*/// handle last interval
$rate = R_{-1}$ */// last rate in list
$duration = T - t_s$
 $arrivals \overset{\text{R}}{\leftarrow} Poisson(rate * duration)$
 $arrivals \overset{\text{R}}{\leftarrow} U(t_c, t_c + duration, arrivals)$
 $t_a \leftarrow$ sort $arrivals$
 return $arrival\_times$
```

IV. Framework for Scaling and Replaying Traces from ADAE Studies

Replaying and scaling ADAE studies in situ presents ethical and logistical issues. Ethically, deploying camera traps and drones in the field can disturb natural habitats, discomfort animals, and provide pathways for poachers to victimize protected species. One-off, long-term deployments yielding valuable ecological insights can address these concerns, but throughput and latency tests do not justify the ethical risks. Logistically, ADAE studies are conducted in remote areas away from research labs and electrical power infrastructure. Replaying studies in situ to test edge configurations imposes a significant logistical burden. However, for ADAE studies, under-provisioned edge resources hamper runtime adaptations, leading to inconclusive study outcomes. It is critical to test edge configurations before studies begin under realistic conditions. As discussed in Section II, edge configurations for future studies will likely need to support (1) bursty traffic, (2) multiple, co-located workflows seeking different outcomes (e.g., population counts and behavior profiles), and (3) large and complex computer vision models.

Our framework considers edge environments comprising multiple networked nodes that share computational resources to meet aggregate demand. These edge resources can use distributed inference, i.e., partitioning workflows and placing

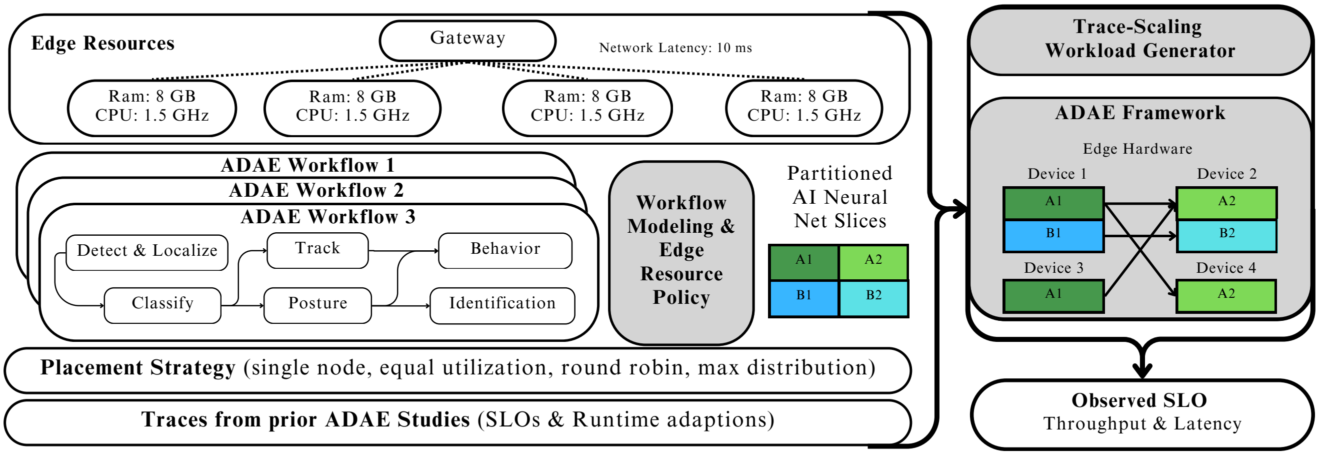

Fig. 4. Framework for characterizing latency and throughput on edge AI platforms for ADAE studies

partitions on specific nodes for execution, to improve performance [30], [33], [37]. Given edge resources and a trace from a prior ADAE study, our framework can set up and test the workflow offline on representative hardware. In addition, our framework can partition and automatically distribute workflows, scale traffic from camera traps and drones, support co-located studies, and test various network latency settings. We also developed a predictive model to forecast workload performance and service-level-objective (SLO) attainment for different edge configurations and placement strategies. The model accurately predicts how workloads will perform. These predictive capabilities enable informed decisions on resource provisioning and system deployment for ADAE studies.

Using representative edge hardware, we developed a framework to scale and replay ADAE traces offline. Figure 4 illustrates our framework. Abstract representations of the edge resources in terms of compute, network latency, memory capacity per node, hosted ADAE workflows, and workflow partitioning and placement strategies are provided as input. Our framework automatically partitions representative workflows using distributed, model-parallel slicing techniques [30], [48]. The workflows are distributed on representative hardware, which receives visual data from a workload generator that replays and scales traces from prior studies. Our framework runs ADAE workflows on actual hardware (mainly because ADAE studies use affordable and accessible devices), but it could be adapted to use virtual resources [68]. Note that our framework is designed only to facilitate the study of ADAE under realistic conditions. State-of-the-art edge simulation platforms provide enhanced features and faster setup and execution [68].

A. Hardware and network infrastructure

We utilize six Raspberry Pi 4B units as our edge devices, each equipped with a quad-core ARM Cortex-A72 CPU (1.5GHz), 8GB LPDDR4-3200 SDRAM, and an integrated GPU for basic acceleration tasks. ADAE studies frequently use Raspberry Pi units [21], [40], [73]. We seek to mirror these setups. We implemented a custom networking framework to emulate the networking environment encountered in field studies. We utilize the Linux Traffic Control (tc) utility to emulate various network conditions, including bandwidth limitations, latency, and packet loss characteristics of cellular and satellite links in remote areas. This allows us to simulate various real-world networking scenarios, from high-bandwidth, low-latency connections to unreliable, high-latency satellite links.

B. Realistic workload generation

We used time-stamped traces for ADAE datasets collected for prior studies. These studies include camera traps for species distributions [71] and drones for monitoring wildlife behavior [43]. Timestamps provide the arrival rate for data from camera traps and drones that trigger ADAE workflow execution. We are interested in how our system performs as these workloads scale; however, naively increasing the arrival rates can significantly alter the workload's critical characteristics. We use correlated or independent scaling, described in Section III-A2, depending on how the specific workload is expected to scale in a real-life deployment. The independent scaling approach involves randomly interleaving bursts and reduces burstiness by increasing the number of independent bursts. The correlated scaling approach maintains burstiness by appending requests during bursts, effectively increasing the send rate without extending the timeframe.

C. Intelligent model splitting and placement strategies

ADAE workloads are bursty, and previous studies have shown such workloads benefit from pipeline parallelism, although these studies have been restricted to homogeneous hardware [49]. ADAE studies primarily rely on heterogeneous hardware; thus, we focused on load-balancing placement techniques that enable pipeline parallelism on heterogeneous hardware. We focus on basic load balancing techniques as a first step to demonstrate that edge computing techniques can be applied to allow for the deployment of ADAE studies.

Edge AI systems designed for ADAE study deployment must possess four characteristics to be effective:

1) Ability to exploit bursty workloads
2) Designed for remote regions with limited compute and memory resources.
3) Support latency-sensitive AI computer vision tasks
4) Run efficiently on diverse edge hardware

We examine four model splitting and placement strategies that accomplish the abovementioned goals, summarized in Table III.

The deployment of AI models in edge computing for ADAE studies often follows a naive approach, which we consider our baseline. In the naive approach, the entire model is placed on individual edge devices without consideration for the specific capabilities of each device or the nature of the workload [73], [80]. This naive approach has several drawbacks: underutilization of resources, inability to handle large models, and lack of adaptability. Some devices may be overwhelmed while others remain underutilized, leading to inefficient use of the overall system resources. This can result in bottlenecks at specific nodes while others sit idle, reducing the system's overall efficiency. Edge devices with limited memory may not be able to accommodate larger, more complex models that could provide higher accuracy. This constraint can force researchers to use simpler, less accurate models, potentially compromising the quality of their ecological insights. Finally, this static placement cannot adjust to changing workload patterns or network conditions. The inability to adapt to dynamic conditions can lead to sub-optimal performance, especially in long-term deployments where environmental and animal behavior patterns may change over time.

To address the limitations of the baseline approach and better meet SLOs, we propose exploring the following strategies: *Naive Round Robin, Utilization-Balanced Model Splitting*, and *Proportional Model Splitting*, shown in Table III. The naive round-robin approach does not split models but instead assigns whole models to available nodes. This approach is simple to implement and can provide load distribution. However, it does not account for heterogeneous device capabilities or varying model sizes. The utilization-balanced model splitting approach balances node utilization through intelligent model layer splitting, using a bin-packing algorithm [30] to determine optimal splitting and placement. To implement proportional model splitting, each model is split into segments proportional to the computational power of the available nodes. This approach may not achieve perfectly balanced node utilization but can provide better throughput for collocated workloads that do not receive traffic simultaneously.

D. Representative computer vision models

Our experiments are conducted using the YOLOv5 [39] suite of models. We evaluated YOLO since it is currently one of the most popular and widely used models for ADAE computer vision tasks including detection, classification, and behavior identification. The YOLOv5 family includes models of varying sizes and complexities, allowing us to evaluate our strategies across a spectrum of computational demands. This choice reflects the common use of YOLO-based models in recent ADAE studies due to their efficiency and accuracy in real-time object detection, localization, and classification tasks [44], [76].

E. Experimental procedure

Our experiment procedure has four steps: establish a baseline, evaluate the strategy, compare to the baseline, and simulate network conditions. We establish baseline performance metrics for each model and workload combination using standard, non-distributed inference. This provides a point of comparison to quantify the improvements achieved by our proposed strategies. We systematically evaluate our three distributed inference strategies: *Naive Round Robin, Utilization-Balanced Splitting*, and *Proportional Splitting*. Each strategy is tested across workload scenarios and network conditions to assess its robustness and adaptability. We implement and compare the three placement strategies across various workload scenarios, focusing on their ability to meet the defined SLO. This comparison helps identify the strengths and weaknesses of each approach under different operating conditions. We simulate different inter-node and intra-node network conditions to evaluate the strategies' performance under various connectivity scenarios typical in animal ecology field studies. This includes testing under ideal conditions and challenging scenarios with high latency and low bandwidth.

F. Performance metrics

We collect comprehensive metrics to evaluate system performance, focused on latency and resource utilization. For latency, we capture end-to-end processing time for individual inference requests, including network transmission delays. This metric is crucial for assessing the system's ability to provide real-time adaptations, which is vital for ADAE. We measure the utilization of CPU, GPU, and memory across all devices in the cluster. These metrics assess the efficiency of our placement strategies in balancing load across heterogeneous resources. We evaluate how well each strategy achieves SLO attainment, which captures the percentage of inference requests meeting predefined latency thresholds under each ADAE from Table I. For resource utilization balance, variance in CPU, GPU, and memory utilization across nodes for each strategy. A low variance indicates more balanced resource utilization, which can lead to better overall system efficiency and reduced bottlenecks. We will

Placement Strategy	Description	Procedure	Advantages	Limitations
Single Node	Baseline approach	One device handles all inference	Simple to implement	Under-utilization of resources, inability to handle large models, lack of adaptability
Round Robin	Assigns whole models to nodes in a round-robin fashion	Each node is assigned one entire model, cycling through the available nodes until all models are placed	Simple to implement and can provide basic load distribution	Does not account for heterogeneous device capabilities or varying model sizes
Equal Utilization	Aims to balance node utilization through intelligently placing splits	Models are split into segments, and these segments are distributed across nodes in a greedy fashion to achieve even utilization	Can lead to reduced queue times and more efficient resource use	Not always optimal if network latency is high or if utilization is low
Max Distribution	Splits each model across as many nodes as possible	Each model is split into segments proportional to the computational power of the available nodes	Can improve performance when load is concentrated on one neural network at a time	Does not scale well with utilization or network latency

TABLE III
EDGE AI MODEL PLACEMENT STRATEGIES

assess how each strategy performs under varying network conditions, measuring the degradation in performance as network quality decreases. This analysis will help identify which strategies are most robust to the challenging and variable network environments often encountered in remote field studies.

G. Characterizing performance of ADAE studies on representative edge hardware

Figure 5 shows the effects of burstiness, network latency, and placement strategy on SLO attainment as the arrival rate of data increases (i.e., normalized traffic). We tested two co-located workloads for all experiments. The first workload comprising 30% of the aggregate traffic is ADAE 1. The other co-located workload is shown in Figure 5. The bottom row shows performance under slow network connectivity at the edge with a 1-second round trip time. As expected, the distributed inference is ineffective in this context, and the worst-performing strategy is the equal-utilization policy. In contrast, the top row shows equal utilization consistently outperforms all other policies under fast-edge networks. ADAE 4 is the most bursty and ADAE 2 is the least. Looking across burstiness in the columns, we observe that burstier workloads magnify the performance gains achieved by the placement strategies.

Figure 6 depicts the effect of increasing edge resources as the arrival rate increases. As expected under slow network latency, the equal utilization placement strategy performed poorly. However, under low-latency network configurations, this approach achieves near-linear scaling. Finally, we also tested how equal utilization placements compare to maximum distribution when animals show avoidant behavior, increasing burstiness. We observed a slight but consistent improvement of roughly 5% for maximum distribution at scale.

V. PERFORMANCE MODELING FOR ADAE STUDIES

We tested four models for estimating the performance gains for a given edge AI system: random forest, XD Gradient, M/D/1 queuing model, and a hybrid random forest, M/D/1 queuing model, shown in Table IV. XD Gradient boost has performed well in previous studies in predicting system performance with few data points. For our study, however, XD Gradient only produced a 35% accuracy in estimating performance gains. We also tested a regular M/D/1 queue, assuming Poisson arrival rates, which also produced a 30% error. The random forest takes the utility level, λ, μ, and expected output latency as inputs, which generated a 24% error. The optimal model for predicting performance gains was a hybrid random forest, M/D/1 queue approach, which produced a 19.6% error. This model predicts expected random forest performance gains and fine-tunes the results with the M/D/1 queue to predict the number of anticipated bursts to overlap for a given workload.

Model	Error Rate
XD Gradient	35 %
M/D/1 Queue	30 %
Random Forest	24 %
Random Forest M/D/1 Hybrid	19 %

TABLE IV
DISTRIBUTED EDGE PROVISIONING TECHNIQUES

We expected to see a reduction in error rates with additional data points. The advantage of this hybrid approach over previous works, such as AlpaServe [49], is that this method allows practitioners to create a scheduling system without requiring historical data, using traces from a single deployed node, optimized for their specific Edge AI system and ADAE.

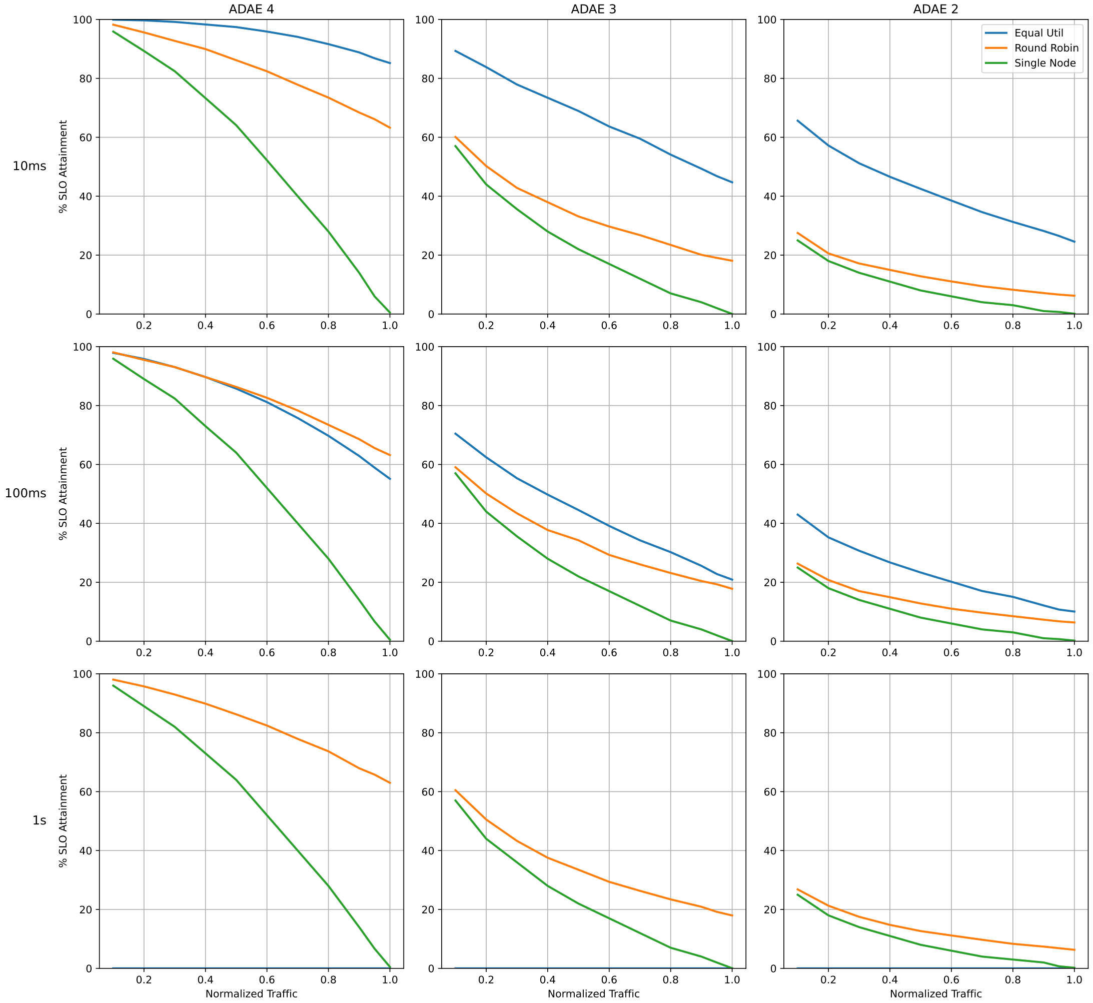

Fig. 5. Performance comparison of different placement strategies (Equal Utilization, Round Robin, Single Node) under various ADAE workloads (each column) and network latencies (each row). The tests were conducted using a 2-node cluster. The x-axis (normalized traffic) is the utilization level of a single node, i.e. the arrival rate of a single node.

VI. Related Work

Edge computing and artificial intelligence are increasingly being applied to ADAE studies, enabling new data collection and analysis approaches. This convergence of technologies, often referred to as Edge AI, has the potential to enable sophisticated processing of data gathered from remote sensing devices such as camera traps and drones. Edge AI systems perform computations at the edge near the source of the data, as opposed to sending data to a centralized cloud server [66]. Edge AI requires massive amounts of data and computing capacity. Still, recent advancements in sensors, hardware, and communication technology like 5G and 6G networks have made this possible on the edge in remote regions [40], [66].

Edge AI is enabled by distributed computing paradigms that allocate tasks across a network of devices. Recent studies have demonstrated how model splitting and co-location can be applied to edge computing paradigms to improve system performance [25], [31], [58]. Model splitting and co-location can reduce latency and utilize system compute more efficiently, i.e., increase the frequency at which the system achieves its SLO [31], [49]. Implementing model splitting and co-location has effectively reduced latency for bursty workloads, although this study focused on homogeneous hardware [49]. Heterogeneous hardware and network conditions are considered in [29], which proposes a deep

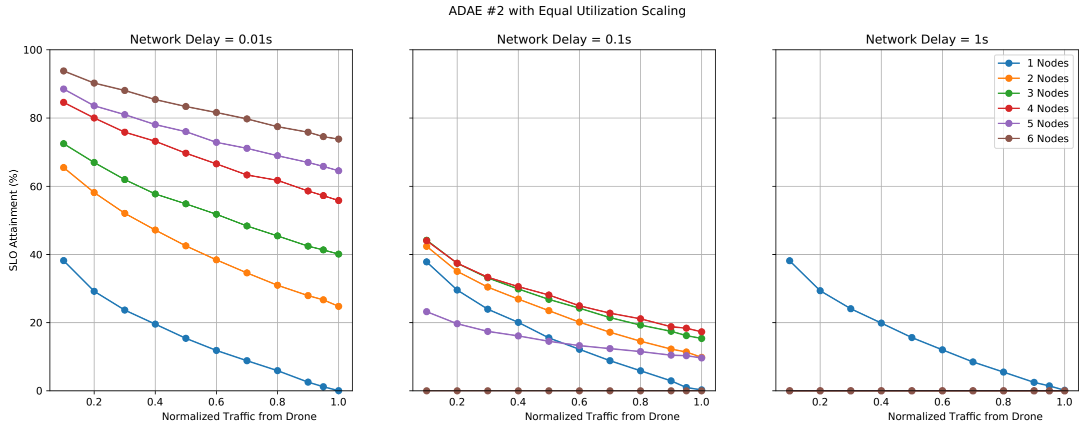

Fig. 6. Scaling performance of the equal utilization placement strategy under different network latencies (0.01s, 0.1s, 1s) for the ADAE 2 workload. The ADAE 2 workload was selected due to its relatively lower SLO attainment on a 2-node setup. Each line represents a different number of compute nodes (ranging from 1 to 6).

reinforcement learning approach to speed up convolutional neural network inference on distributed edge devices. Edge AI system performance can be optimized through model architecture for distributed inference [20], [23], [64].

Optimizing camera placement to improve the performance of computer vision pipelines has been investigated, namely for traffic cameras [75]. Sensor position and orientation dictate the data that cameras can capture, which in turn dictates the accuracy of real-time image analytics. Edge AI enables sensors to be continuously adjusted in real time to maximize workload accuracy under resource constraints. Traffic tasks are similar to animal ecology computer vision tasks. They include detecting objects of interest, counting the number of objects of interest, detection with bounding boxes, and aggregate counting of unique objects of interest [75]. However, this traffic camera study is restricted to a single camera and does not consider request arrival rates. ADAE studies must consider the network of sensors, including drone and camera traps, as well the characteristically bursty arrival rates when designing and implementing runtime adaptations.

As the volume and complexity of ecological data continue to grow, there is an increasing need for efficient computing approaches that can handle the unique challenges posed by wildlife monitoring in remote environments. Recent studies have investigated on-device processing for a more immediate analysis of ecological data. Mobile computing devices, such as laptops, tablets, or custom-built portable units, may also augment in-situ processing capabilities. Such devices could serve as intermediate processing nodes, bridging the gap between data collection points and cloud infrastructure [22]. However, more research is needed to establish their effectiveness in field conditions [40], [73].

Ongoing improvements in the performance of edge processors may enable the deployment of more sophisticated AI models on smart camera traps and drones. Smart camera traps, equipped with on-board computers, can perform initial data processing and filtering, reducing the volume of data that needs to be transmitted or stored for later analysis [6], [17], [21], [52], [73], [80]. Drones equipped with on-board GPU are increasingly available, enabling real-time, on-board processing for autonomous navigation policies [7], [8], [44], [51]. Ecologists have raised concerns about the potential risks of disturbance of wildlife caused by drones [62], which could be reduced by edge-enabled autonomous navigation equipped with safeguards.

The AI and ecology communities have a history of collaboration, applying state-of-the-art computer vision techniques to uncover ecological insights. The CV4Animals: Computer Vision for Animal Behavior workshop, held annually at The Conference on Computer Vision and Pattern Recognition (CVPR), published 40 works this past year alone and featured 18 previously published works on computer-vision-based animal behavioral analysis. As computer vision models grow in size and complexity, we expect models developed for ADAE applications to follow this trend. CNN-based models such as YOLO [38] remain popular for detection, localization, and classification tasks for ADAE studies [43], [44], [76], and YOLO-based models have been tuned to boost performance on ADAE aerial imagery [55]. Recently, vision transformer (ViT) models have proven to perform well, particularly for multi-modal foundation models, such as the species classification models BioClip [67] and Arboretum [77], both based on the OpenCLIP [34] ViT architecture. Increasingly sophisticated models have been developed for more specialized ADAE studies, including inferring animal behavior from video [13] and 3D pose estimation of wildlife from drone footage [63].

VII. Conclusion and Future Work

ADAE studies have the potential to revolutionize animal ecology and biodiversity studies. Unlike traditional ecological studies in the field, ADAE studies leverage edge computing resources to control smart camera traps and autonomous drones, filtering images and adjusting the viewing angles at runtime to improve data quality. Runtime adaptations improve data quality, allowing ecologists to derive insights quickly from their data. They also reduce the time spent parsing data irrelevant to study objectives. Data captured after runtime adaptations can differentiate between datasets that yield insights and inconclusive studies. For these reasons, ADAE studies are a growing edge workload.

An essential contribution of this work is discovering commonalities of ADAE traces that enabled rigorous analysis. Using timestamped traces from prior studies, we observed that (1) the workflows are characterized by interdependent, complex computer vision tasks that transform harvested visual data into ecological datasets; (2) SLO can be repurposed to describe the strict latency demands required for runtime adaptation; and (3) animal dynamics partially explain the bursty workloads observed across many studies.

We replayed ADAE traces offline on representative hardware to understand interactions with edge hardware. We found that workflow partitioning schemes have a complex effect on SLO attainment, especially at scale. We also found that performance modeling approaches using queuing theory and machine learning provide a good starting point to predict SLO attainment.

AI models will likely increase in complexity following current trends. However, ADAE studies are still in the very early stages of adoption. We anticipate simple models, such as YOLO, will be the first to be implemented in real time to inform system adaptations. Thus, we first focused on profiling the implementation of ADAE studies using YOLO. We will expand our approach to implement more complex AI models and workflows in the future. We plan to explore more sophisticated load-balancing approaches tailored to the specific application and available edge devices. We hope others will be interested in investigating this as well.

We encourage others to leverage our findings to propose innovative edge systems for sophisticated ADAE to further our ability to understand and protect our planet's biodiversity. Numerous interdisciplinary innovations have been in computer vision and ecology, but these sophisticated AI models require edge computing innovation to be successfully deployed. ADAE studies offer transformative potential for animal ecology by using edge computing to control smart camera traps and drones, enhancing biodiversity research through advanced edge systems.

VIII. Acknowledgements

The ICICLE project is funded by the National Science Foundation (NSF) under grant number OAC-2112606. The NSF supports the Imageomics Institute under Award No. 2118240.

References

[1] Animl. Animl Camera.

[2] Lila science. LILA BC Labeled Information Library of Alexandria: Biology and Conservation.

[3] Sentinel smart camera trap. https://conservationxlabs.com/sentinel.

[4] Wildme. WildMe.

[5] A. Adegboyega. Quantifying cloud workload burstiness: New measures and models. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, page 987–990, May 2017.

[6] J. A. Ahumada, E. Fegraus, T. Birch, N. Flores, R. Kays, T. G. O'Brien, J. Palmer, S. Schuttler, J. Y. Zhao, W. Jetz, M. Kinnaird, S. Kulkarni, A. Lyet, D. Thau, M. Duong, R. Oliver, and A. Dancer. Wildlife insights: A platform to maximize the potential of camera trap and other passive sensor wildlife data for the planet. *Environmental Conservation*, 47(1):1–6, Mar. 2020.

[7] W. Andrew, C. Greatwood, and T. Burghardt. Aerial animal biometrics: Individual friesian cattle recovery and visual identification via an autonomous uav with onboard deep inference. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, page 237–243, Macau, China, Nov. 2019. IEEE.

[8] W. Andrew, C. Greatwood, and T. Burghardt. Fusing animal biometrics with autonomous robotics: Drone-based search and individual id of friesian cattle (extended abstract). page 38–43, 2020.

[9] M. Bala, T. Eiszler, X. Chen, J. Harkes, J. Blakley, P. Pillai, and M. Satyanarayanan. Democratizing drone autonomy via edge computing. In *2023 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 40–52. IEEE, 2023.

[10] J. Boubin, C. Burley, P. Han, B. Li, B. Porter, and C. Stewart. Marble: Multi-agent reinforcement learning at the edge for digital agriculture. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, pages 68–81, 2022.

[11] J. Boubin, J. Chumley, C. Stewart, and S. Khanal. Autonomic computing challenges in fully autonomous precision agriculture. In *2019 IEEE International Conference on Autonomic Computing (ICAC)*. IEEE, 2019.

[12] J. Boubin, C. Stewart, S. Zhang, N. T. Babu, and Z. Zhang. Softwarepilot. http://github.com/boubinjg/softwarepilot, 2019.

[13] O. Brookes, M. Mirmehdi, H. Kühl, and T. Burghardt. Triple-stream deep metric learning of great ape behavioural actions. (arXiv:2301.02642), Jan. 2023. arXiv:2301.02642 [cs].

[14] A. C. Burton, E. Neilson, D. Moreira, A. Ladle, R. Steenweg, J. T. Fisher, E. Bayne, and S. Boutin. Review: Wildlife camera trapping: a review and recommendations for linking surveys to ecological processes. *Journal of Applied Ecology*, 52(3):675–685, 2015.

[15] C. Chalmers, P. Fergus, S. Wich, S. N. Longmore, N. D. Walsh, P. A. Stephens, C. Sutherland, N. Matthews, J. Mudde, and A. Nuseibeh. Removing human bottlenecks in bird classification using camera trap images and deep learning. *Remote Sensing*, 15(1010):2638, Jan. 2023.

[16] E. Corcoran, M. Winsen, A. Sudholz, and G. Hamilton. Automated detection of wildlife using drones: Synthesis, opportunities and constraints. *Methods in Ecology and Evolution*, 12(6):1103–1114, 2021.

[17] J. S. Dertien, H. Negi, E. Dinerstein, R. Krishnamurthy, H. S. Negi, R. Gopal, S. Gulick, S. K. Pathak, M. Kapoor, P. Yadav, M. Benitez, M. Ferreira, A. J. Wijnveen, A. T. L. Lee, B. Wright, and R. F. Baldwin. Mitigating human–wildlife conflict and monitoring endangered tigers using a real-time camera-based alert system. *BioScience*, 73(10):748–757, Oct. 2023.

[18] DroneDeploy. Dronedeploy flight app.

[19] I. Duporge, M. Kholiavchenko, R. Harel, S. Wolf, D. Rubenstein, M. Crofoot, T. Berger-Wolf, S. Lee, J. Barreau, J. Kline, M. Ramirez, and C. Stewart. Baboonland dataset: Tracking primates in the wild and automating behaviour recognition from drone videos. (arXiv:2405.17698), May 2024. arXiv:2405.17698 [cs].

[20] B. J. Eccles, L. Wong, and B. Varghese. Rapid deployment of dnns for edge computing via structured pruning at initialization. (arXiv:2404.16877), Apr. 2024. arXiv:2404.16877 [cs].

[21] P. Fergus, C. Chalmers, S. Longmore, S. Wich, C. Warmenhove, J. Swart, T. Ngongwane, A. Burger, J. Ledgard, and E. Meijaard. Empowering wildlife guardians: An equitable digital stewardship and reward system for biodiversity conservation using deep learning and 3/4g camera traps. *Remote Sensing*, 15(1111):2730, Jan. 2023.

[22] A. Gholami and J. S. Baras. Collaborative cloud-edge-local computation offloading for multi-component applications. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, page 361–365, Dec. 2021.

[23] L. Giovannesi, G. Proietti Mattia, and R. Beraldi. Targeted and automatic deep neural networks optimization for edge computing. In L. Barolli, editor, *Advanced Information Networking and Applications*, page 57–68, Cham, 2024. Springer Nature Switzerland.

[24] F. Hamann, S. Ghosh, I. J. Martinez, T. Hart, A. Kacelnik, and G. Gallego. Low-power continuous remote behavioral localization with event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18612–18621, 2024.

[25] P. Hao and Y. Zhang. Eddl: A distributed deep learning system for resource-limited edge computing environment. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, page 1–13, Dec. 2021.

[26] B. J. Harmsen, R. J. Foster, S. C. Silver, L. E. T. Ostro, and C. P. Doncaster. Spatial and temporal interactions of sympatric jaguars (panthera onca) and pumas (puma concolor) in a neotropical forest. *Journal of Mammalogy*, 90(3):612–620, June 2009.

[27] P. A. Henrys, T. O. Mondain-Monval, and S. G. Jarvis. Adaptive sampling in ecology: Key challenges and future opportunities. *Methods in Ecology and Evolution*, 15(9):1483–1496, 2024.

[28] A. Hernandez, Z. Miao, L. Vargas, R. Dodhia, P. Arbelaez, and J. M. L. Ferres. Pytorch-wildlife: A collaborative deep learning framework for conservation, 2024.

[29] X. Hou, Y. Guan, T. Han, and N. Zhang. Distredge: Speeding up convolutional neural network inference on distributed edge devices. In *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, page 1097–1107, Lyon, France, May 2022. IEEE.

[30] K.-J. Hsu, K. Bhardwaj, and A. Gavrilovska. Couper: Dnn model slicing for visual analytics containers at the edge. In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, pages 179–194, 2019.

[31] Y. Hu, C. Imes, X. Zhao, S. Kundu, P. A. Beerel, S. P. Crago, and J. P. Walters. Pipeedge: Pipeline parallelism for large-scale model inference on heterogeneous edge devices. In *2022 25th Euromicro Conference on Digital System Design (DSD)*, page 298–307, Maspalomas, Spain, Aug. 2022. IEEE.

[32] A. Hua, K. Martin, Y. Shen, N. Chen, C. Mou, M. Sterk, B. Reinhard, F. F. Reinhard, S. Lee, S. Alibhai, and Z. C. Jewell. Protecting endangered megafauna through ai analysis of drone images in a low-connectivity setting: a case study from namibia. *PeerJ*, 10:e13779, Aug. 2022.

[33] Y. Hui, J. Lien, and X. Lu. Characterizing and accelerating end-to-end edgeai inference systems for object detection applications. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 01–12. IEEE, 2021.

[34] G. Ilharco, M. Wortsman, R. Wightman, C. Gordon, N. Carlini, R. Taori, A. Dave, V. Shankar, H. Namkoong, J. Miller, H. Hajishirzi, A. Farhadi, and L. Schmidt. Openclip. 10.5281/zenodo.5143773, 2021.

[35] E. iot, F. Fang, M. Hamilton, D. Kar, D. Dmello, J. Choi, R. Hannaford, A. Iyer, L. Joppa, M. Tambe, and R. Nevatia. Spot poachers in action: Augmenting conservation drones with automatic detection in near real time. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(11), Apr. 2018.

[36] H. Jamadagni. Experimenting usage of camera traps for population dynamics study of the asian elephant (elephas maximus) in tropical mixed deciduous forests of southern india. 2012.

[37] S. Y. Jang, B. Kostadinov, and D. Lee. Microservice-based edge device architecture for video analytics. In *2021 IEEE/ACM Symposium on Edge Computing (SEC)*, pages 165–177. IEEE Computer Society, 2021.

[38] G. Jocher. Ultralytics yolov5, 2020.

[39] G. Jocher, A. Chaurasia, and J. Qiu. Ultralytics yolov5, 2023.

[40] J. W. Jolles. Broad-scale applications of the raspberry pi: A review and guide for biologists. *Methods in Ecology and Evolution*, 12(9):1562–1579, 2021.

[41] K. U. Karanth and M. E. Sunquist. Behavioural correlates of predation by tiger (panthera tigris), leopard (panthera pardus) and dhole (cuon alpinus) in nagarahole, india. *Journal of Zoology*, 250(2):255–265, 2000.

[42] R. Kays, B. Kranstauber, P. Jansen, C. Carbone, M. Rowcliffe, T. Fountain, and S. Tilak. Camera traps as sensor networks for monitoring animal communities. In *2009 IEEE 34th Conference on Local Computer Networks*, page 811–818, Zurich, Switzerland, Oct. 2009. IEEE.

[43] M. Kholiavchenko, J. Kline, M. Ramirez, S. Stevens, A. Sheets, R. Babu, N. Banerji, E. Campolongo, M. Thompson, N. Van Tiel, J. Miliko, E. Bessa, I. Duporge, T. Berger-Wolf, D. Rubenstein, and C. Stewart. Kabr: In-situ dataset for kenyan animal behavior recognition from drone videos. page 31–40, 2024.

[44] J. Kline, C. Stewart, T. Berger-Wolf, M. Ramirez, S. Stevens, R. R. Babu, N. Banerji, A. Sheets, S. Balasubramaniam, E. Campolongo, M. Thompson, C. V. Stewart, M. Kholiavchenko, D. I. Rubenstein, N. Van Tiel, and J. Miliko. A framework for autonomic computing for in situ imageomics. In *2023 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*, page 11–16, Toronto, ON, Canada, Sept. 2023. IEEE.

[45] J. M. Kline, M. Kholiavchenko, O. Brookes, T. Berger-Wolf, and C. Stewart. Integrating biological data into autonomous remote sensing systems for in situ imageomics: A case study for kenyan animal behavior sensing with unmanned aerial vehicles (uavs).

[46] B. Koger, A. Deshpande, J. T. Kerby, J. M. Graving, B. R. Costelloe, and I. D. Couzin. Quantifying the movement, behaviour and environmental context of group-living animals using drones and computer vision. *Journal of Animal Ecology*, n/a(n/a).

[47] H. S. Kühl, A. K. Kalan, M. Arandjelovic, F. Aubert, L. D'Auvergne, A. Goedmakers, S. Jones, L. Kehoe, S. Regnaut, A. Tickle, E. Ton, J. van Schijndel, E. E. Abwe, S. Angedakin, A. Agbor, E. A. Ayimisin, E. Bailey, M. Bessone, M. Bonnet, G. Brazolla, V. E. Buh, R. Chancellor, C. Cipoletta, H. Cohen, K. Corogenes, C. Coupland, B. Curran, T. Deschner, K. Dierks, P. Dieguez, E. Dilambaka, O. Diotoh, D. Dowd, A. Dunn, H. Eshuis, R. Fernandez, Y. Ginath, J. Hart, D. Hedwig, M. Ter Heegde, T. C. Hicks, I. Imong, K. J. Jeffery, J. Junker, P. Kadam, M. Kambi, I. Kienast, D. Kujirakwinja, K. Langergraber, V. Lapeyre, J. Lapuente, K. Lee, V. Leinert, A. Meier, G. Maretti, S. Marrocoli, T. J. Mbi, V. Mihindou, Y. Moebius, D. Morgan, B. Morgan, F. Mulindahabi, M. Murai, P. Niyigabae, E. Normand, N. Ntare, L. J. Ormsby, A. Piel, J. Pruetz, A. Rundus, C. Sanz, V. Sommer, F. Stewart, N. Tagg, H. Vanleeuwe, V. Vergnes, J. Willie, R. M. Wittig, K. Zuberbuehler, and C. Boesch. Chimpanzee accumulative stone throwing. *Scientific Reports*, 6:22219, Feb. 2016.

[48] Z. Li, L. Zheng, Y. Zhong, V. Liu, Y. Sheng, X. Jin, Y. Huang, Z. Chen, H. Zhang, J. E. Gonzalez, et al. {AlpaServe}: Statistical multiplexing with model parallelism for deep learning serving. In *17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23)*, pages 663–679, 2023.

[49] Z. Li, L. Zheng, Y. Zhong, V. Liu, Y. Sheng, X. Jin, Y. Huang, Z. Chen, H. Zhang, J. E. Gonzalez, and I. Stoica. Alpaserve: Statistical multiplexing with model parallelism for deep learning serving. (arXiv:2302.11665), July 2023. arXiv:2302.11665 [cs].

[50] W. Luo, G. Zhang, Q. Shao, Y. Zhao, D. Wang, X. Zhang, K. Liu, X. Li, J. Liu, P. Wang, L. Li, G. Wang, F. Wang, and Z. Yu. An efficient visual servo tracker for herd monitoring by uav. *Scientific Reports*, 14(1):10463, May 2024.

[51] W. Luo, G. Zhang, Q. Shao, Y. Zhao, D. Wang, X. Zhang, K. Liu, X. Li, J. Liu, P. Wang, L. Li, G. Wang, F. Wang, and Z. Yu. An efficient visual servo tracker for herd monitoring by uav. *Scientific Reports*, 14(1):10463, May 2024.

[52] Z. Ma, Y. Dong, Y. Xia, D. Xu, F. Xu, and F. Chen. Wildlife real-time detection in complex forest scenes based on yolov5s deep learning network. *Remote Sensing*, 16(88):1350, Jan. 2024.

[53] B. McNutt, L. Zhang, A. Carey-Douglas, F. Vollrath, F. Pope, and L. Brickson. Whole-herd elephant pose estimation from drone data for collective behavior analysis.

[54] L. D. Mech and L. Boitani. *Wolves: Behavior, Ecology, and Conservation*. University of Chicago Press, Oct. 2010.

[55] C. Mou, T. Liu, C. Zhu, and X. Cui. Waid: A large-scale dataset for

wildlife detection with drones. *Applied Sciences*, 13(1818):10397, Jan. 2023.

[56] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725, June 2018.

[57] K. Ozogány, V. Kerekes, A. Fülöp, Z. Barta, and M. Nagy. Fine-scale collective movements reveal present, past and future dynamics of a multilevel society in przewalski's horses — nature communications. *Nature Communications*, 14(1):5096, Sept. 2023.

[58] B. Ramprasad, P. Mishra, M. Thiessen, H. Chen, A. da Silva Veith, M. Gabel, O. Balmau, A. Chow, and E. de Lara. Shepherd: Seamless stream processing on the edge. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*, pages 40–53. IEEE, 2022.

[59] F. Rovero and F. Zimmermann. *Camera Trapping for Wildlife Research*. Pelagic Publishing Ltd, June 2016.

[60] F. Sakib and T. Burghardt. Visual recognition of great ape behaviours in the wild. *CoRR*, abs/2011.10759, 2020.

[61] L. Schad and J. Fischer. Opportunities and risks in the use of drones for studying animal behaviour. *Methods in Ecology and Evolution*, 14(8):1864–1872, 2023.

[62] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[63] V. Shukla, L. Morelli, F. Remondino, A. Micheli, D. Tuia, and B. Risse. Towards estimation of 3d poses and shapes of animals from oblique drone imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLVIII-2–2024:379–386, June 2024.

[64] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed. Efficient acceleration of deep learning inference on resource-constrained edge devices: A review. *Proceedings of the IEEE*, 111(1):42–91, Jan. 2023.

[65] A. R. E. Sinclair. Does interspecific competition or predation shape the african ungulate community? *The Journal of Animal Ecology*, 54(3):899, Oct. 1985.

[66] R. Singh and S. S. Gill. Edge ai: A survey. *Internet of Things and Cyber-Physical Systems*, 3:71–92, Jan. 2023.

[67] S. Stevens, J. Wu, M. J. Thompson, E. G. Campolongo, C. H. Song, D. E. Carlyn, L. Dong, W. M. Dahdul, C. Stewart, T. Berger-Wolf, W.-L. Chao, and Y. Su. BioCLIP: A vision foundation model for the tree of life. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[68] M. Symeonides, Z. Georgiou, D. Trihinas, G. Pallis, and M. D. Dikaiakos. Fogify: A fog computing emulation framework. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*, page 42–54, Nov. 2020.

[69] D. Tuia, B. Kellenberger, S. Beery, B. R. Costelloe, S. Zuffi, B. Risse, A. Mathis, M. W. Mathis, F. Van Langevelde, T. Burghardt, R. Kays, H. Klinck, M. Wikelski, I. D. Couzin, G. Van Horn, M. C. Crofoot, C. V. Stewart, and T. Berger-Wolf. Perspectives in machine learning for wildlife conservation. *Nature Communications*, 13(1):792, Feb. 2022.

[70] D. Tulasi, A. Granados, P. Gunawardane, A. Kashyap, Z. McDonald, and S. Thulasidasan. Smart camera traps: Enabling energy-efficient edge-ai for remote monitoring of wildlife. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on AI-driven Spatio-temporal Data Analysis for Wildlife Conservation*, GeoWildLife '23, page 9–16, New York, NY, USA, Dec. 2023. Association for Computing Machinery.

[71] J. Vélez, W. McShea, H. Shamon, P. J. Castiblanco-Camacho, M. A. Tabak, C. Chalmers, P. Fergus, and J. Fieberg. An evaluation of platforms for processing camera-trap data using artificial intelligence. *Methods in Ecology and Evolution*, 14(2):459–477, 2023.

[72] B. G. Weinstein. A computer vision for animal ecology. *Journal of Animal Ecology*, 87(3):533–545, 2018.

[73] R. C. Whytock, T. Suijten, T. van Deursen, J. Świeżewski, H. Mermiaghe, N. Madamba, N. Mouckoumou, J. A. Zwerts, A. F. K. Pambo, L. Bahaa-el din, S. Brittain, A. W. Cardoso, P. Henschel, D. Lehmann, B. R. Momboua, L. Makaga, C. Orbell, L. J. T. White, D. M. Iponga, and K. A. Abernethy. Real-time alerts from ai-enabled camera traps using the iridium satellite network: A case-study in gabon, central africa. *Methods in Ecology and Evolution*, 14(3):867–874, 2023.

[74] WILDLABS. Megadetector on the inventory, 2024. url=https://wildlabs.net/inventory/products/megadetector.

[75] M. Wong, M. Ramanujam, G. Balakrishnan, and R. Netravali. Madeye: Boosting live video analytics accuracy with adaptive camera configurations.

[76] Y. Xie, J. Jiang, H. Bao, P. Zhai, Y. Zhao, X. Zhou, and G. Jiang. Recognition of big mammal species in airborne thermal imaging based on yolo v5 algorithm. *Integrative Zoology*, 18(2):333–352, 2023.

[77] C.-H. Yang, B. Feuer, Z. Jubery, Z. K. Deng, A. Nakkab, M. Z. Hasan, S. Chiranjeevi, K. Marshall, N. Baishnab, A. K. Singh, A. Singh, S. Sarkar, N. Merchant, C. Hegde, and B. Ganapathysubramanian. Arboretum: A large multimodal dataset enabling ai for biodiversity, 2024.

[78] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, and Q. Li. Lavea: Latency-aware video analytics on edge computing platform. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pages 1–13, 2017.

[79] S. Young, J. Rode-Margono, and R. Amin. Software to facilitate and streamline camera trap data management: A review. *Ecology and Evolution*, 8(19):9947–9957, 2018.

[80] I. Zualkernan, S. Dhou, J. Judas, A. R. Sajun, B. R. Gomez, and L. A. Hussain. An iot system using deep learning to classify camera trap images on the edge. *Computers*, 11(11):13, Jan. 2022.