Wildlife Detection using YOLOv8 for Analytics with Amazon Location Services

DATA/MSML 650 Group Project Proposal

Vivek Ediga, Aditya Sastry, Prakhar Tiwari, Yajat Uppal, Ateeq Ur Rehman

Problem statement

As inhabitants of Earth, we have long observed numerous changes affecting our planet. Warmer temperatures, rising sea levels, and extreme weather events are becoming more prevalent. However, we must not forget that we are not the only species being impacted. Since the 1970s, thousands of wildlife populations and Earth's overall biodiversity have declined due to these environmental changes, as well as human-driven activities such as deforestation, urban development, and pollution (Ritchie, 2022).

Conserving all of Earth's biodiversity is essential to our survival. Wildlife supports healthy and resilient ecosystems, which in turn sustain human health (Shaw, 2024). One of the most effective ways to support biodiversity conservation is by understanding where species live and how they are distributed. This knowledge allows us to identify key habitats for protection and support resource management (NatureServe)..

Species distribution data often comes from occurrence records, which can be captured using edge devices such as trail cameras. However, these devices often have limitations with storage capacity and vulnerability to data loss through damage or tampering. Latency is another major concern, as significant events could go unnoticed for days (Yu, 2024). These events might include rare wildlife behavior or the presence of poachers and other anthropogenic threats to wildlife.

To ensure all data is reliably captured and available to support species distribution, conservation planning, and protection, a more robust and resilient data pipeline must be established. Therefore, we are proposing a cloud-based pipeline that facilitates the ingestion and offloading of data from edge-devices, such as trail cameras, for wildlife or threat detection, analytics, and visualization.

High-Level Approach

Baseline:

Our baseline would be a traditional monitoring approach where images are retrieved and are reviewed manually or with offline machine learning (we will use the latter). Results will be saved, and reporting and visualization will be done after the fact. There will be no notification of significant or threat events.

Proposed Solution:

- **Stage 1: Simulated Ingestion**
  - Using the Spatiotemporal Wildlife Dataset, we will simulate edge device behavior locally by "streaming" data from our dataset one file at a time to our to and S3 bucket. With this simulation, we can also add latency to mimic unreliable networks and delays, which would be a common problem with actual trail cameras. Once uploaded to S3, we can have an event-driven function that logs the streamed and ingested data and notifies a user of a detection.
- **Stage 2: Data Processing & Object Detection using Machine Learning**
  - Using YOLOv8 in SageMaker, load the image from S3 to classify and run inference on one image at a time. Results will be saved to DynamoDB.
- **Stage 3: Alerts & Notifications**
  - Analyze ML output using Lambda to check for species of interest or possible threats (i.e, human or vehicle). Then send an alert via email.
- **Stage 4: Storage & Metadata Management**
  - Store raw images, detection metadata, and visualizations in S3 and DynamoDB.
- **Stage 5: Visualization & Analysis**
  - Create a QuickSight report showing interesting trends
  - Integrate Amazon Location Web Map
  - **Novelty**: Using Amazon Location Services to create a simple web map of detections.

Considerations:

- Ideally, this solution would allow for real-time inference as soon as an image is uploaded; however, that is not possible with the Free-Tier of AWS, so we will be doing scheduled inference that occurs frequently, but with the caveat that the pipeline could be refactored to be real-time if higher tiers of AWS are used.
- The Spatiotemporal Wildlife Dataset is quite comprehensive. To ensure the creation of a robust pipeline, we may start with a simple detection that is only looking for a small subset of species to classify, with the caveat that it could be extrapolated to include the detection of other species if the model were trained on them.

State-of-the-Art:

State-of-the-art systems for wildlife monitoring leverage **fully automated, event-driven, and cloud-integrated architectures** that enable near real-time detection on the edge, classification, and alerting. Modern implementations combine advanced object detection models (e.g., YOLOv8, DETR, or MegaDetector) with scalable cloud services like AWS Kinesis, SageMaker, Lambda, and DynamoDB to create end-to-end pipelines. The system's pipeline automatically triggers whenever new data arrives, logs results in distributed databases, and fully automated event handling pipelines facilitate instant notifications for high-priority detections such as endangered species or human intrusions. Additionally, they integrate fully interactive visualizations and analytics tools like QuickSight or custom dashboards that display detection trends. Examples of SOTA implementations could be Wildbook / WildMe, which utilizes a computer vision pipeline deployed on Azure to assist with tracking individual animals in wildlife populations.

Planned Contribution:

Note*: This is the planned contribution scheme for the project… the actual contribution may differ from the proposed.

Prakhar Tiwari – Data Ingestion & Edge Simulation Lead

Stage: 1 (Simulated Ingestion)
Main Tasks:

- Simulate edge device behavior locally (Python script to "stream" data one file at a time).
- Implement a latency simulation to mimic unreliable networks.
- Set up an S3 bucket and write upload automation (boto3 or AWS CLI).
- Configure **S3 event trigger** (Lambda or SNS) to log ingestion events.
- Test event-driven pipeline (data → S3 → event trigger).
  Deliverables: Data ingestion code, latency simulation script, ingestion logs.

Yajat Uppal – Machine Learning & Model Integration Lead

Stage: 2 (YOLOv8 in SageMaker)
Main Tasks:

- Set up the YOLOv8 environment in **Amazon SageMaker Notebook**.
- Preprocess incoming images from S3 for inference.
- Run **YOLOv8 inference** on each image.
- Store results (bounding boxes, labels, confidence) into **DynamoDB**.
- Optimize model loading and inference for batch or real-time mode.
  Deliverables: SageMaker script, inference pipeline, DynamoDB write function.

Vivek Ediga – Backend Automation & Alerts Engineer

Stage: 3 (Alerts & Notifications)
Main Tasks:

- Build an **AWS Lambda function** to analyze ML output (from DynamoDB).
- Define logic for “species of interest” or “threats” (human, vehicle).
- Integrate **SNS or SES** for alert emails to users.
- Write alert message templates and notification formatting.
- Test alert triggering end-to-end (inference → alert email).
  Deliverables: Lambda alert code, SNS/SES setup, test logs.

Aditya Sastry – Data Storage & Metadata Management Lead

Stage: 4 (Storage & Metadata Management)
Main Tasks:

- Organize raw image storage in **S3 with metadata tagging** (e.g., timestamp, camera ID, species).
- Design **DynamoDB schema** for storing detection metadata and ML results.
- Ensure data consistency between S3 objects and DynamoDB entries.
- Implement versioning and lifecycle policies for storage optimization.
  Deliverables: S3-DynamoDB schema design, metadata script, documentation.

Ateeq Rehman – Visualization & Analytics Lead

Stage: 5 (Visualization & Analysis)
Main Tasks:

- Use **Amazon QuickSight** to create dashboards showing detection trends (time, species frequency, location).
- Integrate **Amazon Location Service (Web Map)** for geospatial visualization of detections.
- Display camera locations and detected species on a dynamic web map.
- Present insights such as “species activity by time of day” or “location heatmap.”
  Deliverables: QuickSight dashboard, map integration demo, report visuals.

Integration Point	**Who Coordinates**	**Who Supports**
S3 setup and trigger config	Prakhar	Aditya
ML inference output (SageMaker → DynamoDB)	Yajat	Vivek
Alert pipeline (DynamoDB → Lambda → Email)	Vivek	Ateeq
Data consistency checks	Aditya	Prakhar
Visualization with live data	Ateeq	Yajat

Implementation Tools

- S3 Buckets
- YOLOv8 in SageMaker
- DynamoDB
- Lambda
- SNS
- QuickSight
- Amazon Location Services

References

Hannah Ritchie, Fiona Spooner, and Max Roser (2022) - “Biodiversity” Published online at OurWorldinData.org. Retrieved from: 'https://ourworldindata.org/biodiversity' [Online Resource]

“Documenting Species and Ecosystems.” *NatureServe*, NatureServe, www.natureserve.org/documenting-species-ecosystems#:~:text=Knowledge%20of%20where%20species%20and,comparability%20of%20data%20among%20programs. Accessed 3 Oct. 2025.

Shaw, Julie. “Why Is Biodiversity Important?” *Conservation.Org*, Conservation International, 15 Oct. 2024, https://www.conservation.org/blog/why-is-biodiversity-important#:~:text=Species%20are%20often%20integral%20to,and%20nature%20is%20generally%20overlooked. Accessed 3 Oct. 2025.

Yu, Hui, et al. "Edge computing in wildlife behavior and ecology." *Trends in Ecology & Evolution* 39.2 (2024): 128-130.