AI FOR FOREST ELEPHANTS 2

Final presentation

Cornell University

ELEPHANT LISTENING PROJECT

FruítPunch AI

Today, you'll hear from

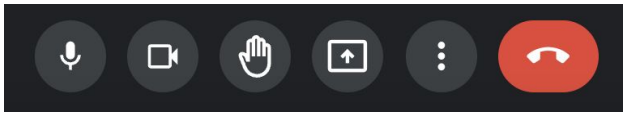
Dorian: Challenge introduction

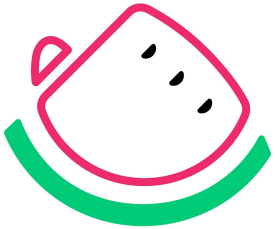
Team 1: Modelling

Team 2: Application

Daniela: Final words

If you have any questions during the presentation raise your hand!

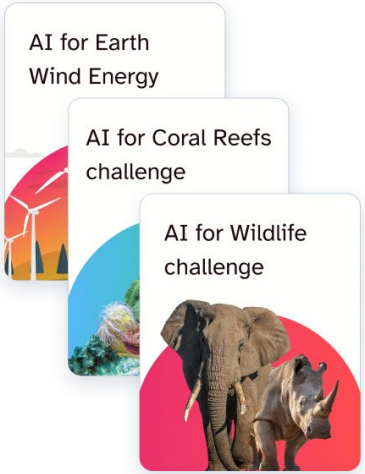




FruítPunch AI

The Global AI for Good Community

To solve **humanity's greatest challenges**
we need to **educate AI engineers at scale**
with an **ethical & sustainable mindset**



Develop AI skills with real-world challenges

Learn peer-to-peer in a global community

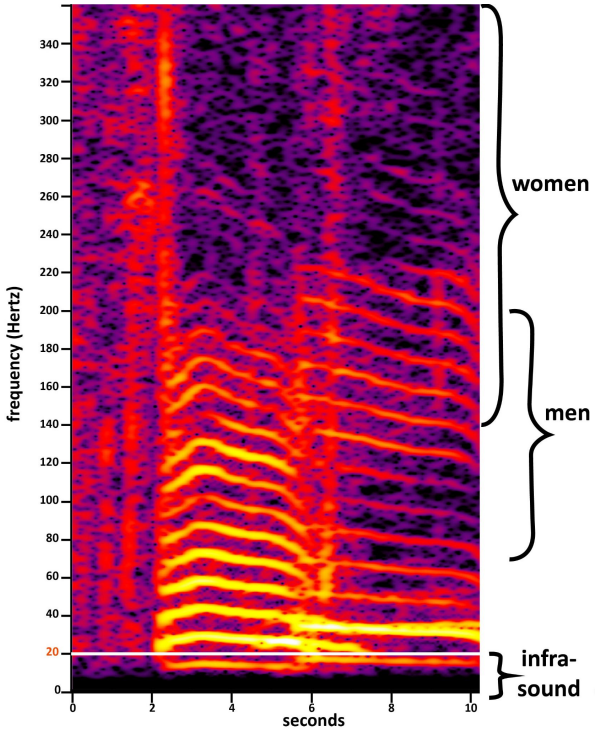
Elephant monitoring and automated audio analysis

- 60% of Forest Elephants were lost in the last decade
- >12,000 are killed each year for their ivory

They live in vast areas making it difficult to monitor and protect them.
Here audio monitoring comes in handy

Elephant rumbles

Infrasound makes monitoring over large distances possible



Goals for this Challenge

Main Goal: Detect Elephants' Rumbles and Gunshots in 24-hour Audio recordings fast and accurately (fast in inference time)

Sub Goals:

- Fast parallel processing of spectrograms
- Detect Gunshots and Elephant rumbles
- Build an easy-to-use pipeline that inputs large datasets and outputs rumbles and gunshots in the desired format.

10

weeks of hard work

2

working groups

21

AI for Good engineers

1600+

hours of engineering work

Modelling team

Meet the team!

Guy	Gea	Davide
Pablo	Ron	Rohit
Remi	Thor	Uri
Arthur	Anton	Gerson
Liubov	Sonny	William

We can add a * to the names of the people that will be presenting

Preliminaries

Data

Recordings from Dzanga and PNNN, testing subset. We took this part of the dataset because of the better annotation quality.

Approaches

1) Image-based: Converting recordings into spectrograms
    - Object detection
    - Image classification

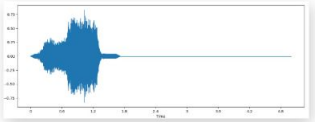
2) Audio-based: Working directly with the audio
    - Audio classification
    - Rumble detection

Object detection model

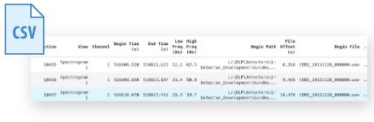
Model: YOLOv8 (You Only Look Once version 8) is originally an object detection model used in computer vision.

Input: spectrograms of 60s long clips, containing several rumbles or none at all.

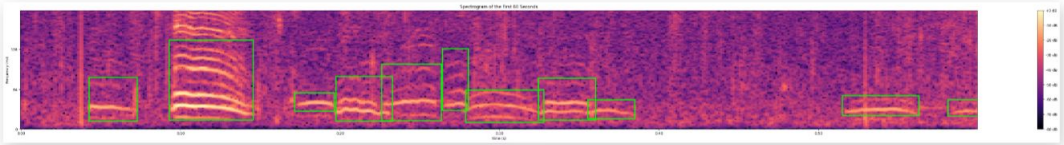
Output: spectrograms with bounding boxes around the areas that are identified as rumbles.



Raw audio files



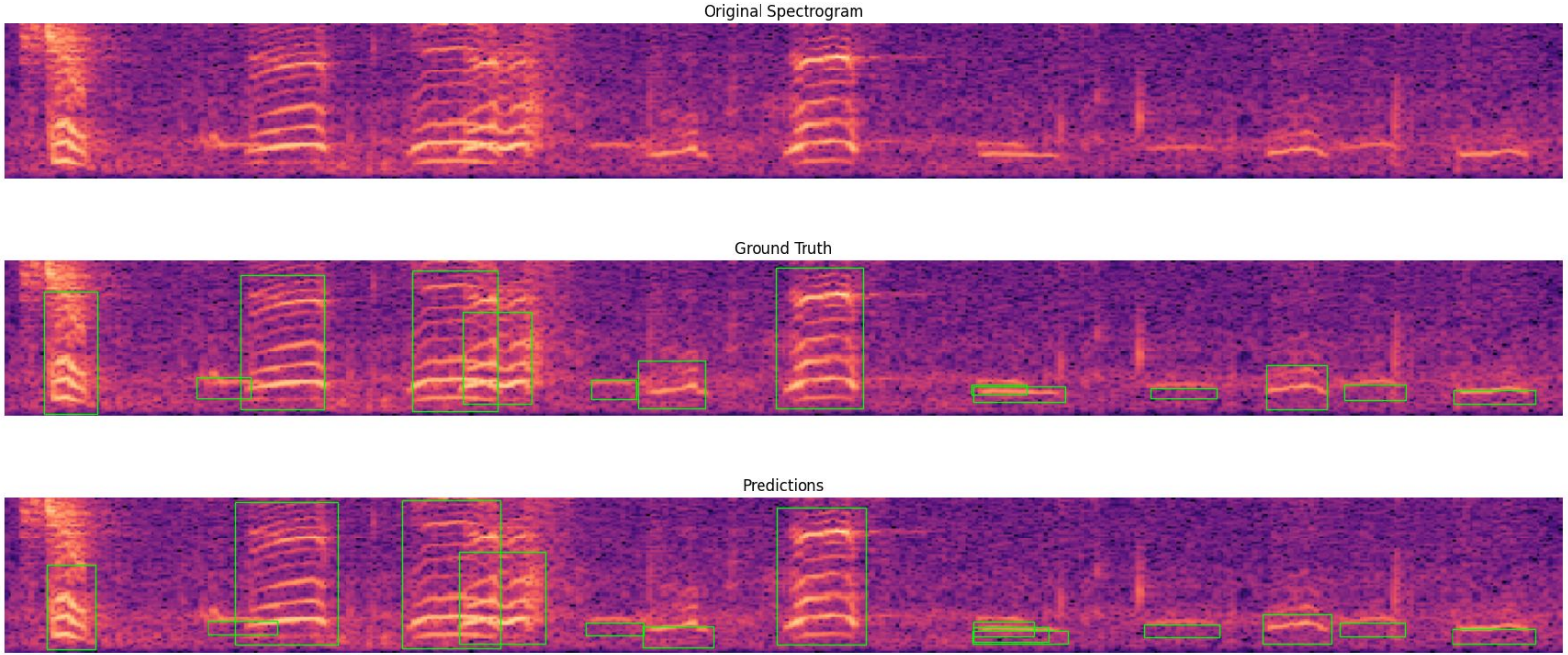
Rumble annotations from Raven Pro



Annotated spectrograms

duration: 60s
frequency range: (0.0, 250.0)

Qualitative Results



Quantitative evaluation

Quantitative evaluation is harder to perform

- Metrics for object detection differ from the standard classification metrics.
- Annotation of the bounding boxes labeling in the dataset is inconsistent
- There are overlapping rumbles which complicates the evaluation even more

Future plans

To get a clean and consistently annotated dataset, then we can fairly easily get the precision, recall, F1-score at 0.5 mIoU for instance.

Processing

CPU

Processing a 24-hour audio file on an 8-core CPU takes approximately 35 seconds in total:

- Loading the audio file: ~4 seconds
- Generating spectrograms: ~11 seconds
- Running model inference: ~19 seconds
- Miscellaneous tasks: ~1 second

GPU + CPU

Processing a 24-hour audio file using a GPU (T4) and an 8-core CPU takes approximately 20 seconds in total:

- Loading the audio file: ~4 seconds
- Generating spectrograms: ~11 seconds
- Running model inference: ~4 seconds
- Miscellaneous tasks: ~1 second

Image classification

Model: DinoV2 for transfer learning

Training data: spectrograms of 5s clips, containing full rumbles, parts of rumbles or no rumbles. Using 10s or longer clips gives poor results with F1-score 0.7

Input: spectrograms of 5s clips

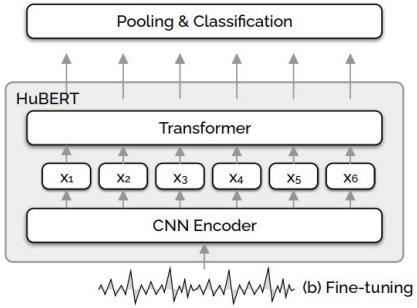
Output: prediction ‘rumble’ or ‘non-rumble’

Results

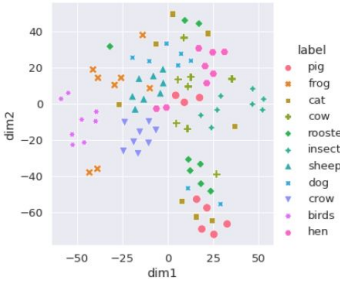
Class	Precision	Recall	F1	Total
Non-rumble	0.84	0.89	0.86	501
Rumble	0.84	0.78	0.81	531
Accuracy	0.84			1032
Macro	0.84	0.83	0.83	1032
weighted	0.84	0.84	0.84	1032

Non-spectrogram based audio detector/classifier

- Rumble detector and short-timescale-classifier tool based on [AVES: Animal Vocalization Encoder based on Self-Supervision]
  - AVES is based on [HuBERT], a self-supervised transformer model for human speech
  - Pre-trained on a wide-range of unlabeled biological sounds
  - Proven track record when fine tuned for specific animals (cows, crows, bats, whales, mosquitos, fruit-bats, ...)

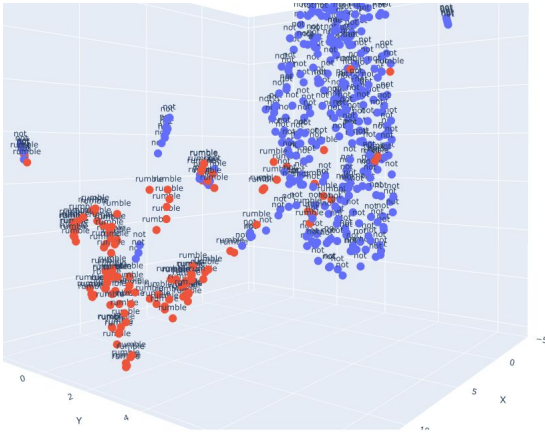


(b) Fine-tuning



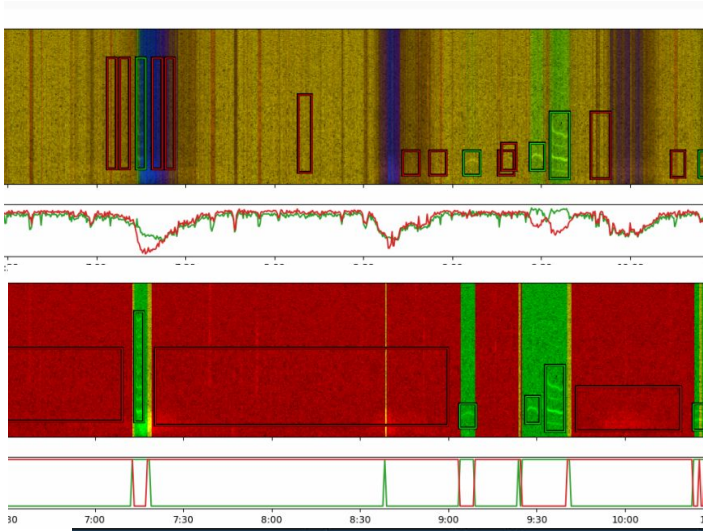
(c) Plot of learned representations

- Modified to work with Elephant Rumbles by training a small companion model.
  - Trained on every ⅓-of-a-second of "01. Data/cornell_data/Rumble/Training" that was labeled as a rumble in the Raven Selection Files, and a similar duration of “not a rumble” labels created from the same files.
  - Tested against every ⅓-of-a-second rumble in "01. Data/cornell_data/Rumble/Test" and a similar duration of sounds that were not labeled as rumbles.
  - Selection of the negative “not-a-rumble” training dataset had a huge impact on model quality. For this approach, the “negative” training dataset needs a wide variety of other animals and background sounds that may overlap with the elephant sounds.



Non-spectrogram based audio detector/classifier - Tech

- AVES generates “embedding vectors” for each fraction of a second of audio.
  - Similar to what HuBERT does for each syllable of human speech.
  - The picture on the upper right visualizes AVES embeddings using cosine-similarity against our training data. It doesn’t know elephants, so isn’t useful out-of-the-box.
  - Tweaked slightly to work on different frequency ranges by upshifting and resampling elephant sounds into human-hearing-ranges. Upshifting 3 or 4 octaves works great. (thx Arnoud!)
  - Tested at both ⅓ second and ⅔ second time-scales.
- Our trained Elephant Rumble classifier classifies each AVES embedding vector as “rumble” vs “not rumble”
  - The middle picture shows how the classifier classifies that clip.
  - Correctly identifies over 95% of the labeled rumbles as “rumble”
  - Correctly identifies over 90% of the background sounds as “not”
  - More metrics and graphs shown here.
  - Complete training notebook and validation tests here.
    - That notebook includes: preparing the data, creating the test/train datasets, creating negative “not-a-rumble” labels, testing entirely on the training data, and generating those metrics purely on the test-data.

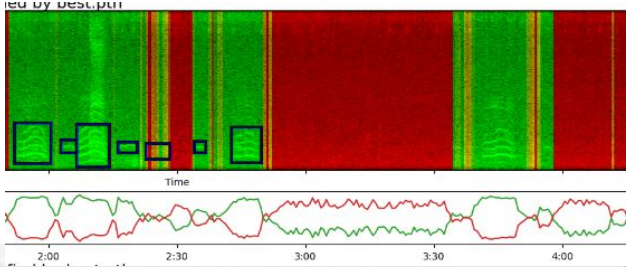


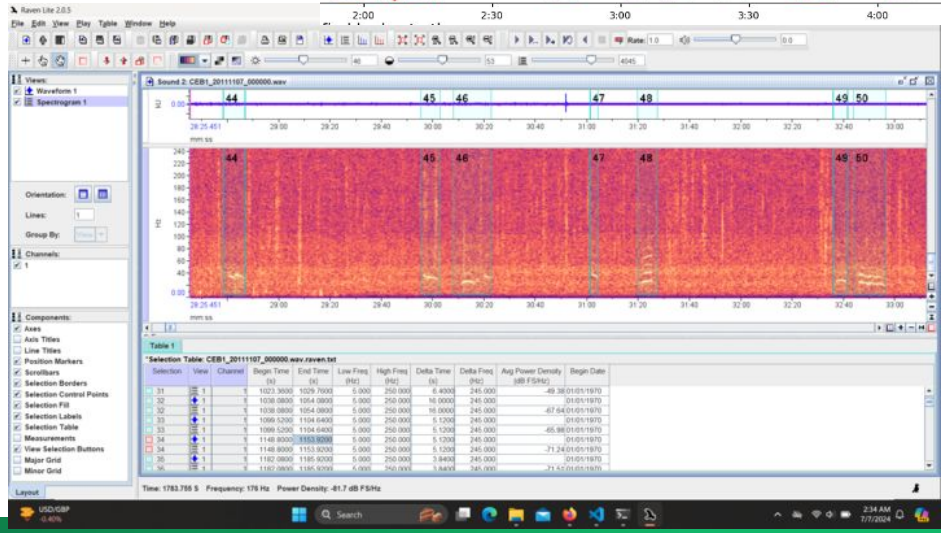
	precision	recall	f1-score	suppor
not	0.97	0.92	0.94	6625
rum	0.92	0.97	0.94	6625
accuracy			0.94	13250
macro avg	0.94	0.94	0.94	13250
weighted avg	0.94	0.94	0.94	13250

Non-spectrogram based audio detector/classifier - Results

- Seems to find rumbles (green blocks) that were not labeled in the original Raven files (blue outlines), mostl in the Training Dataset. More examples here.
- Fast (on a 2060 GPU) - 20 seconds for a 24-hour audio file.
- Command-line utility to create raven files and visualizations of rumbles it finds.
- Demonstration installing and running the tool here.
- Raven Lite showing the output of this tool ——————————>
- Easy to use. These statements should generate raven selection files for every .wav file in a folder.

```
> pip install git+https://github.com/ramayer/elephant-rumble-inference@v0.2.1
> elephant-rumble-inference ./data/*.wav --save-raven
```





Side-note:

- Some unlabeled possible-rumbles were difficult to see in the spectograms with default settings.
- I'm using a technique similar to "Per Channel Energy Normaliziation" that makes some of those hidden possible-rumbles more visible to our eyes.
- If Raven doesn't have PCEN or similar, perhaps they should consider adding it. Cornell has papers showing PCEN's usefulness for other animal sounds..

Perch: Agile Modeling for Bioacoustics

Overall Approach:

- Perch provides a framework for rapid, interactive development of custom classifiers
- Combines transfer learning, similarity search, and active learning
- Designed for scenarios with limited labeled data

Core Components:

- Pre-trained Bird Vocalization Classifier for embedding generation
- Similarity search in embedding space
- Human-in-the-loop labeling
- Simple linear classifier training

Workflow:

- Embed unlabeled dataset and query samples
- Perform similarity search to find potential matches
- Human expert labels small set of examples
- Train linear classifier on embeddings
- Use classifier to find more potential matches
- Iterate to improve classifier

Key Differences from AVES:

- Uses supervised pre-training vs. AVES' self-supervised learning
- Incorporates active learning and human expertise in the loop
- Focuses on rapid development of task-specific classifiers

Advantages:

- Minimal labeled data required
- Quick iteration and classifier improvement
- Leverages human expertise efficiently
- Can adapt to specific research needs rapidly

Use Case:

- Ideal for researchers needing to quickly develop classifiers for specific species/ call types/ sounds within large unlabeled datasets

High performance achieved with minimal active learning iterations.

Metric	Score
Accuracy (ACC)	0.76
AUC-ROC (overall)	0.9
cMAP	0.84

Class-specific AUC-ROC:

class	score
rumble	0.9
unknown	0.9

Embeddings

Object detection, gunshots

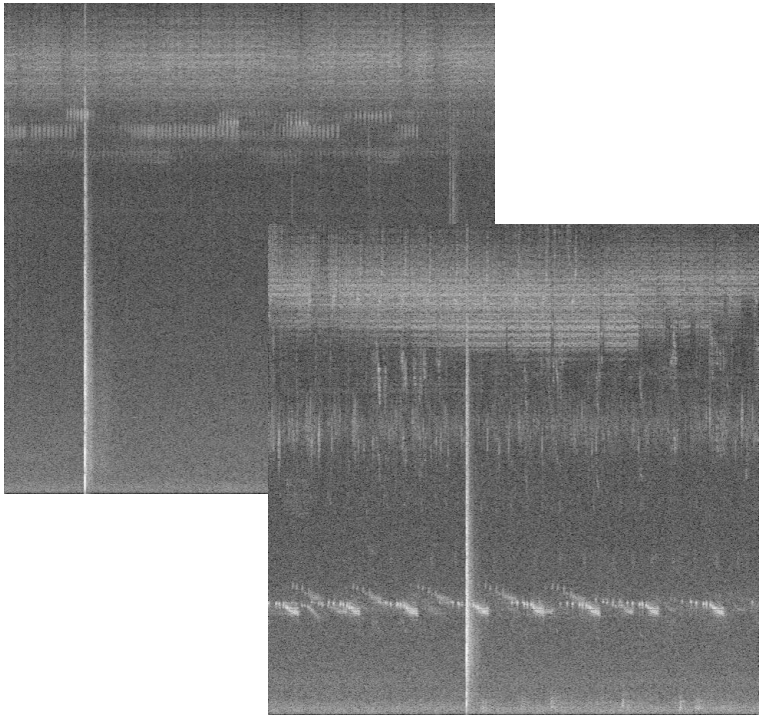
Model: YOLOv8 (You Only Look Once version 8) is originally an object detection model used in computer vision.

Data: 969 clips containing gunshots (split based on date)

- Clips of 160 seconds with 10 second overlap and sample rate of 4096.
- Optimal settings for the spectrograms:
  - n_fft=4096
  - hop_length=1024

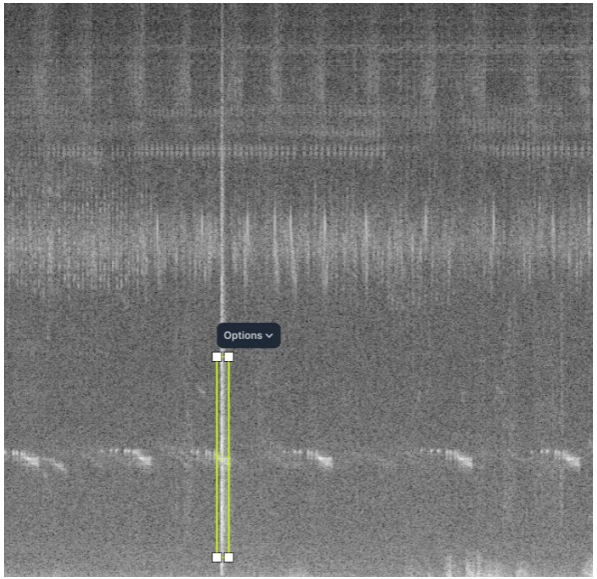
Restrictions: Not all gunshots are annotated correctly

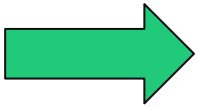
Primary results: F1-score ~0.6

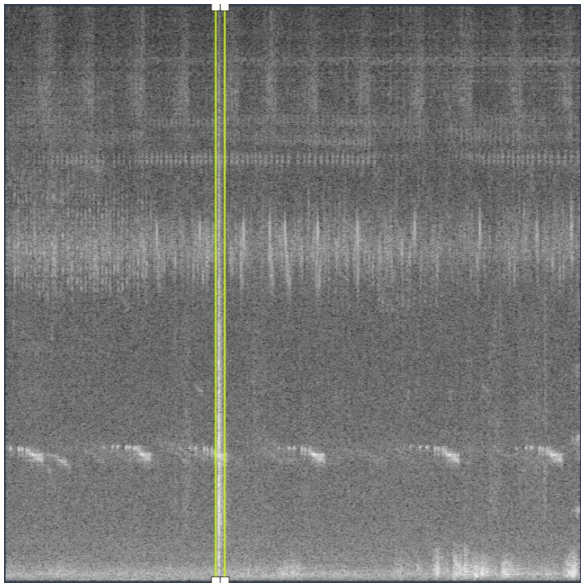


Gunshots data: relabelling

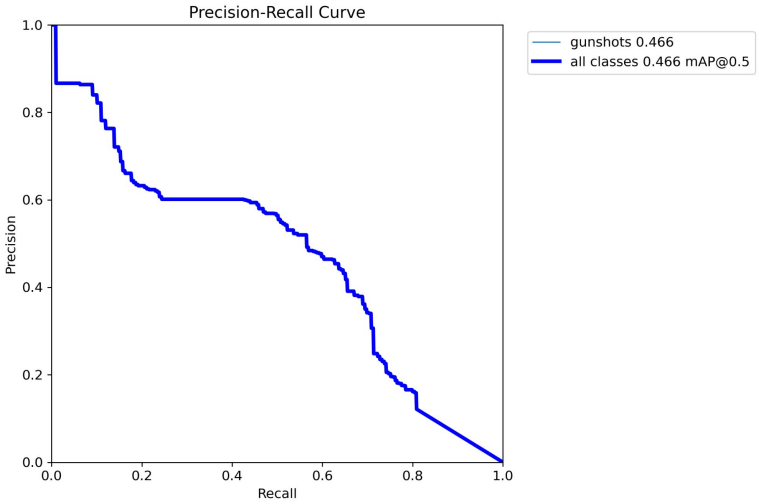
Our team improved the annotations for around 1000 of clips containing gunshots.

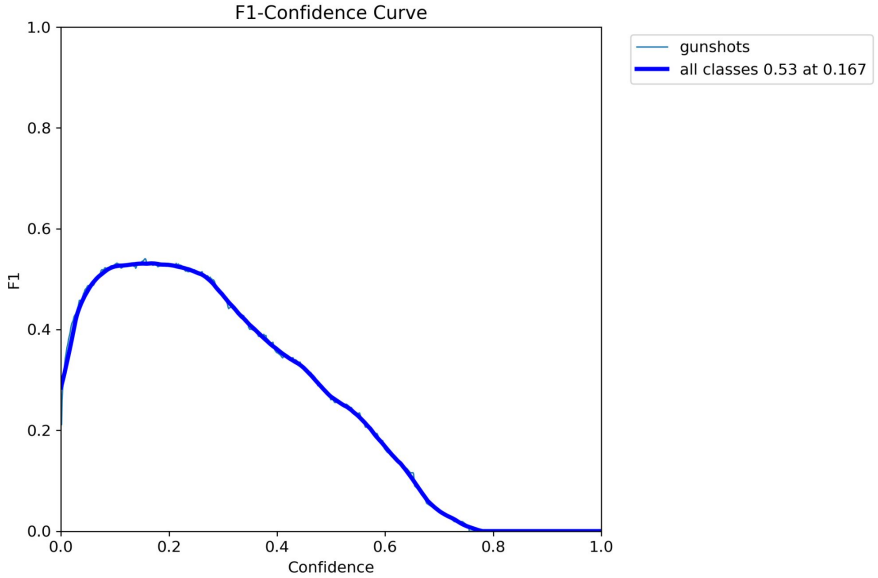




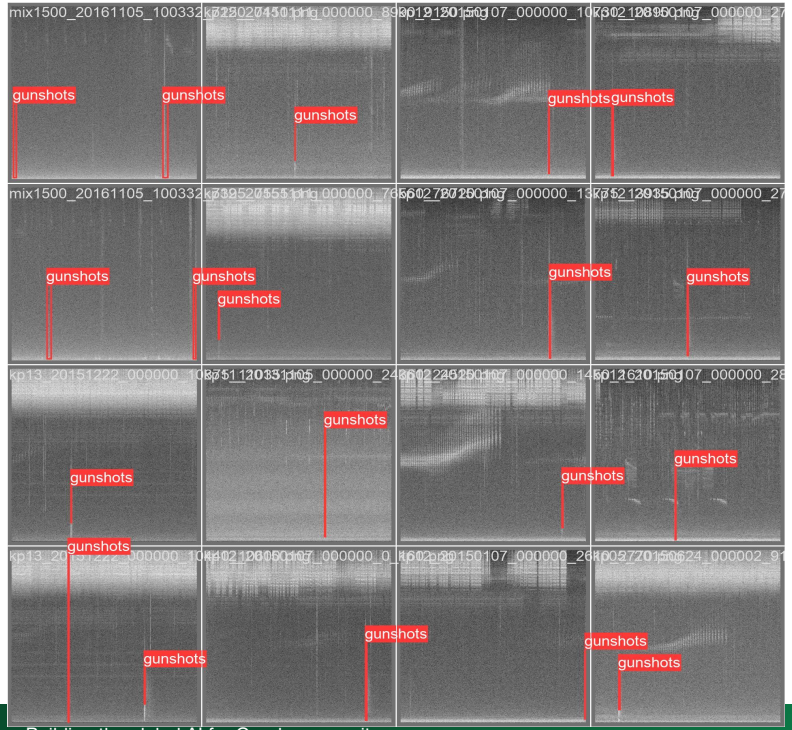


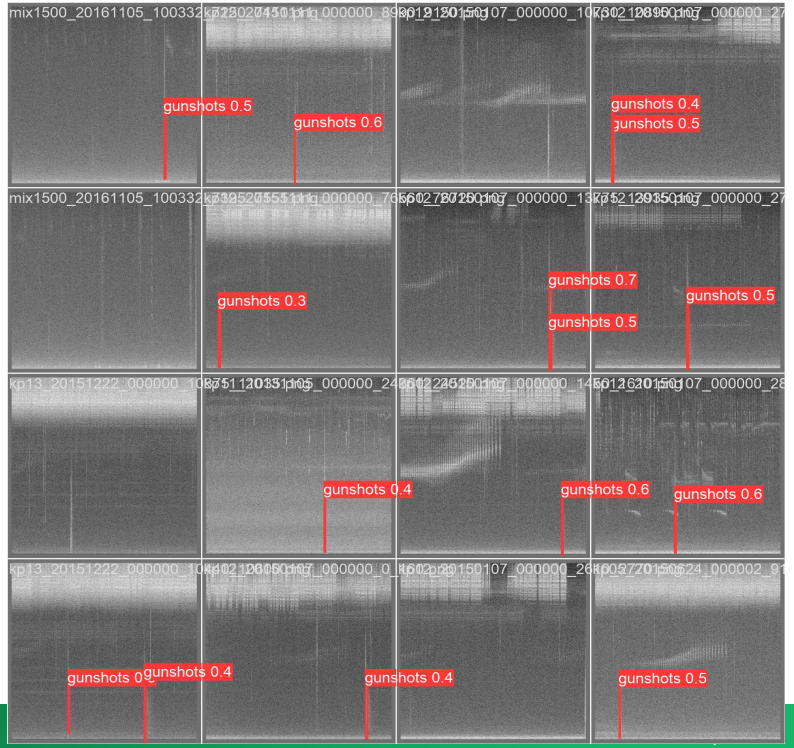
Results


Precision-Recall Curve
gunshots 0.466
all classes 0.466 mAP@0.5
Precision
Recall
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


F1-Confidence Curve
gunshots
all classes 0.53 at 0.167
F1
Confidence
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Labels vs predictions





Summary

We made 5 different experiments for rumbles and gunshots identification.

Task	Model	Performance
Object detection, rumbles	Yolov8	Good results based on qualitative assessment
Image classification, rumbles	Dinov2	0.84 accuracy
Audio classification, rumbles	AVES	0.98 accuracy
Audio classification, rumbles	Perch	0.76 accuracy
Object detection, gunshots	Yolov8	0.6 F1-score

Q&A