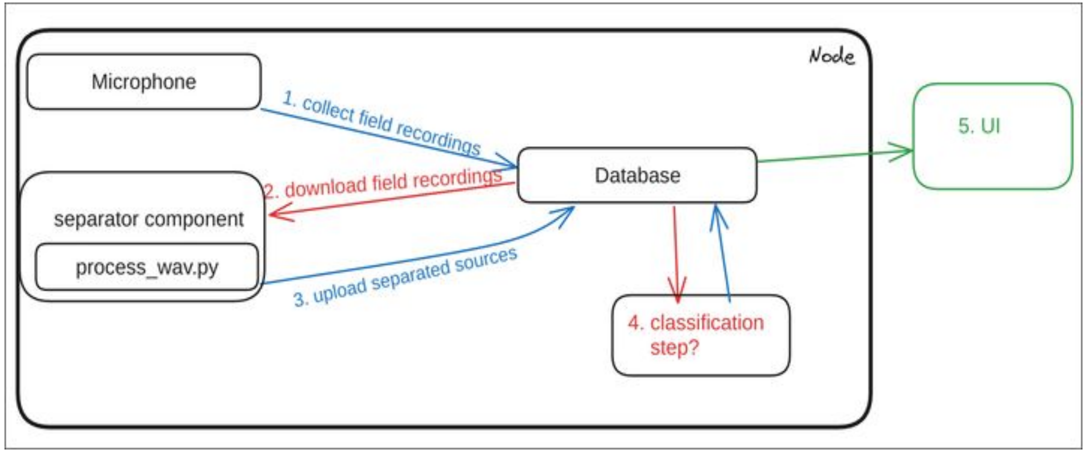
Audio Separator Plugin

Alex Nishio and Michael Szostak

Design

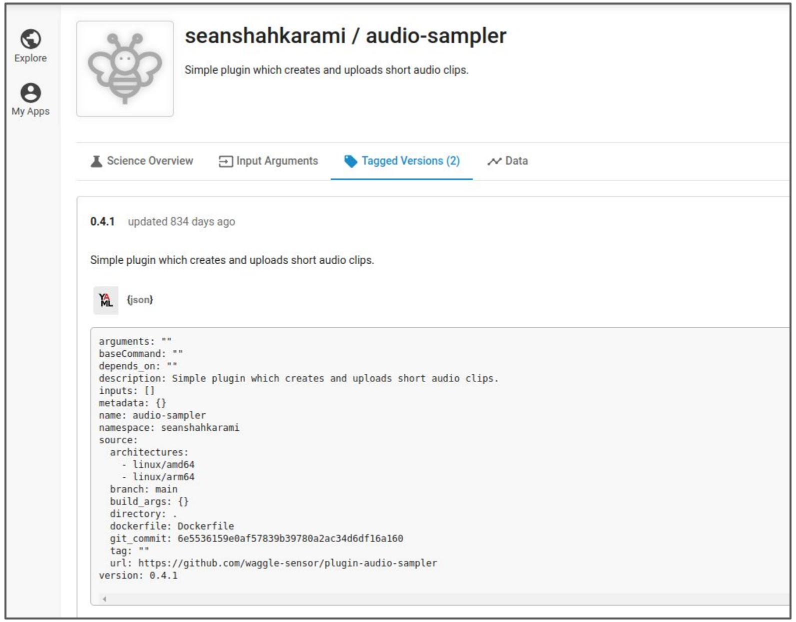


Credit to Michael

Steps 1~3 are uploaded on Michael's GitHub page
Steps 4 and 5 are future improvements

1. Microphone Node is activated by Docker Container
2. Field Recordings are sent to a database
3. An inference model is ran on the database that analyzes the wav files and keeps only the important wav files
4. Classification Model
5. User Interface

Microphone Portion (Step 1 and 2)



seanshahkarami / audio-sampler

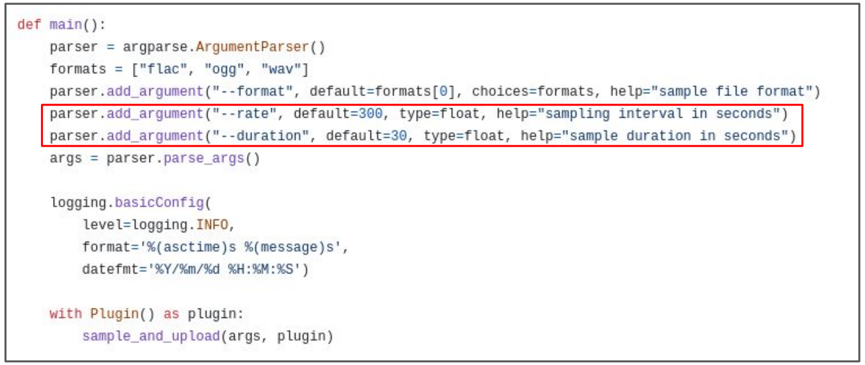
Simple plugin which creates and uploads short audio clips.

0.4.1 updated 834 days ago

Simple plugin which creates and uploads short audio clips.

```yaml
arguments: ""
baseCommand: ""
depends_on: ""
description: Simple plugin which creates and uploads short audio clips.
inputs: []
metadata: {}
name: audio-sampler
namespace: seanshahkarami
source:
  architectures:
    - linux/amd64
    - linux/arm64
  branch: main
  build_args: {}
  directory: .
  dockerfile: Dockerfile
  git_commit: 6e5536159e0af57839b39780a2ac34d6df16a160
  tag: ""
  url: https://github.com/waggle-sensor/plugin-audio-sampler
version: 0.4.1
```

Sean’s Implementation of the Audio Sampler can be directly used

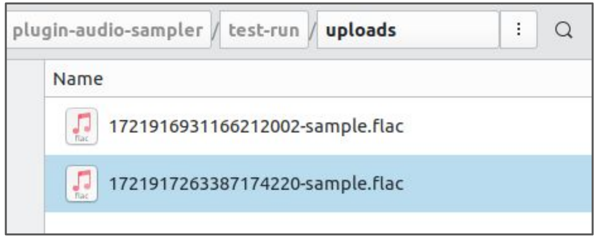


```python
def main():
    parser = argparse.ArgumentParser()
    formats = ["flac", "ogg", "wav"]
    parser.add_argument("--format", default=formats[0], choices=formats, help="sample file format")
    parser.add_argument("--rate", default=300, type=float, help="sampling interval in seconds")
    parser.add_argument("--duration", default=30, type=float, help="sample duration in seconds")
    args = parser.parse_args()

    logging.basicConfig(
        level=logging.INFO,
        format='%(asctime)s %(message)s',
        datefmt='%Y/%m/%d %H:%M:%S')

    with Plugin() as plugin:
        sample_and_upload(args, plugin)
```

Can tune the time intervals and length of clip here



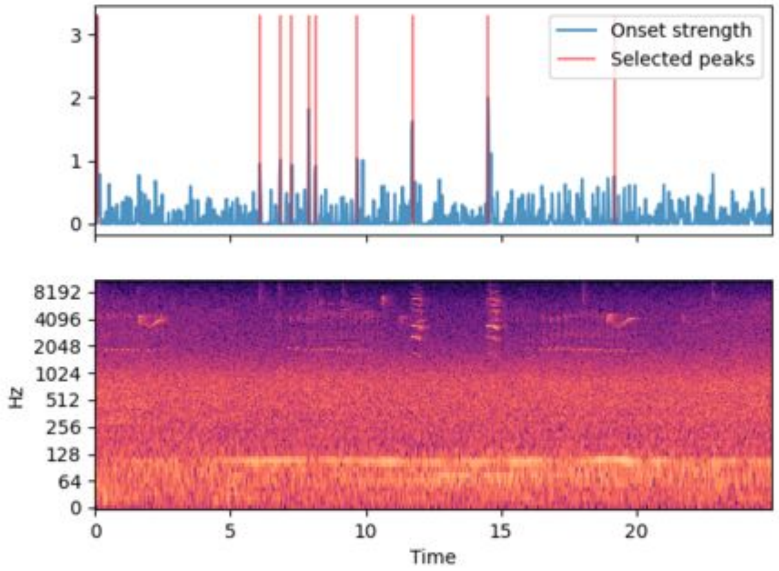
plugin-audio-sampler / test-run / uploads

Name
1721916931166212002-sample.flac
1721917263387174220-sample.flac

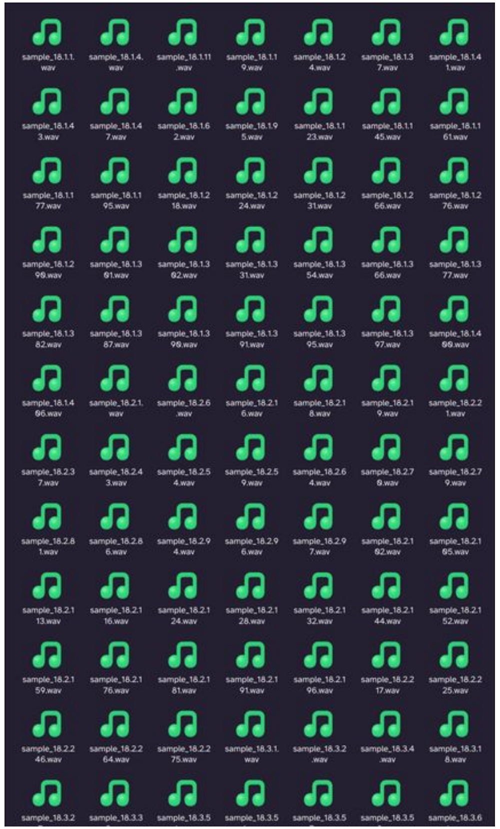
30 second clips recorded in 5 min intervals stored in a file storage system

Results of Peak Finder Script (Step 3)

More information can be found in Michael’s repository



Red lines indicate peak sounds (important sounds)



Proof that the separated files are present

Future Improvements

- Finishing Steps 4 and 5
    - Add a classification model to distinguish audio files
    - Create a UI to make it more user friendly