

## **Librosa.effects.hpss**

<https://librosa.org/doc/main/generated/librosa.effects.hpss.html>

This is one I have experimented with a bit already and there may be potential to work with this further, although it would need to be paired with other forms of processing. Essentially, this function takes your source file and provides 2 output files, one with percussive components and one with harmonic components.

The documentation is not the best but the “margin” argument seems to have a massive amount of impact on which components get added to which file.

The main reason I believe this has potential is because Cricket noises are inherently percussive, and UAV sounds should be considered harmonic components.

<https://github.com/timsainb/noisereduce>

The noisereduce python library seems to have potential applications here with its use of spectral gating to denoise a signal.

The caveat is that you need to supply a “typical noise” file along with the source file for the best results. This may be difficult because the noise is not always the same throughout our data.

## **Manual FFT subtraction**

“But since you're basically trying to remove noise, it might be simpler to take an example of the noise itself, and then try to reduce that noise from the original sample. Take the FFT of the noise, subtract the frequency bins of the noise from the frequency bins of the sample, and run an inverse FFT to produce the final filtered result. If you take this approach, make sure to only do FFTs at half the sample rate”

Again, the downside is getting our hands a pure noise file that we can use for approaches like this.

<https://github.com/nussl/nussl>

This is something I have not looked into much, but it appears to be a framework with implementation for deep learning based approaches to audio source separation. Also provides access to pre-trained speech/music separation models.

\*I have been wondering if there are any transfer learning approaches from music/speech separation to our use case. Obviously, our audio has nothing to do with speech or music so it seems a bit more challenging.

<https://dev.to/highcenburg/separate-vocals-from-a-track-using-python-4lb5>

The last approach I will mention for now is this one which utilizes librosa decomposition functions, the primary use case is to separate from vocals from music in songs. In an ideal world, the UAV audio would be the vocals and the noise/crickets would be the background/music.