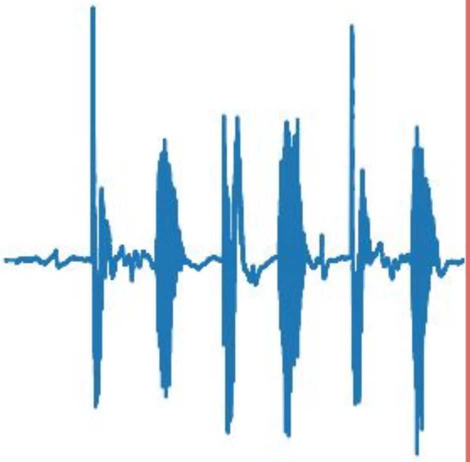
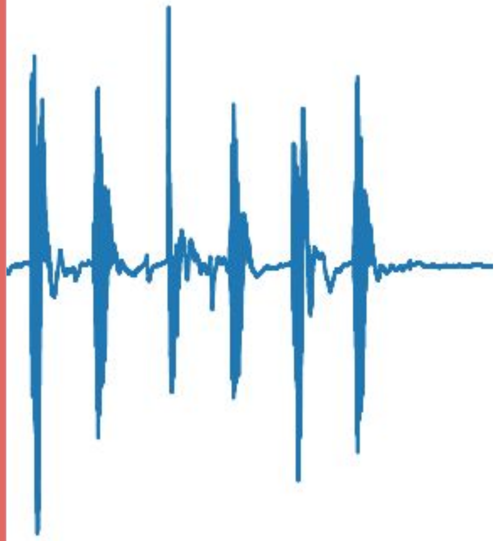


Beatbox to Analog Drum Translation Using a Convolutional Neural Network

Joe Bumpus, Noah Miller, & Julia Weinstock
ECE 408 - The Art of Machine Learning
University of Rochester



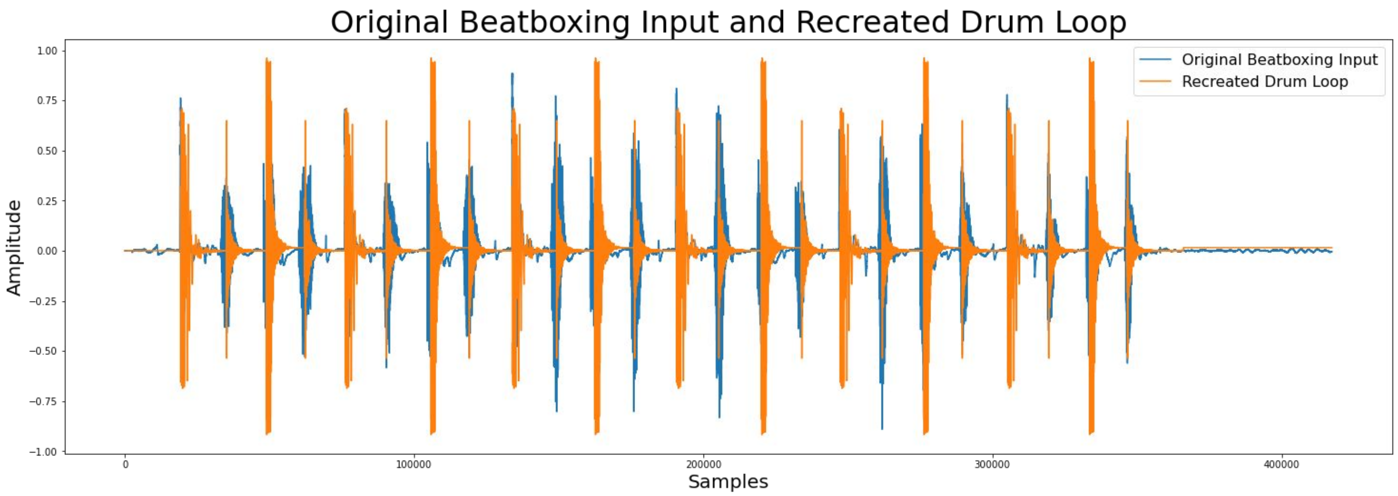
Presentation Roadmap

- Problem Statement & Purpose
- Mel Spectrograms
- Dataset & Data Preparation
- CNN Model Architecture & Results
- Sample Triggering
- Future Considerations

Project Statement & Purpose

Project Statement

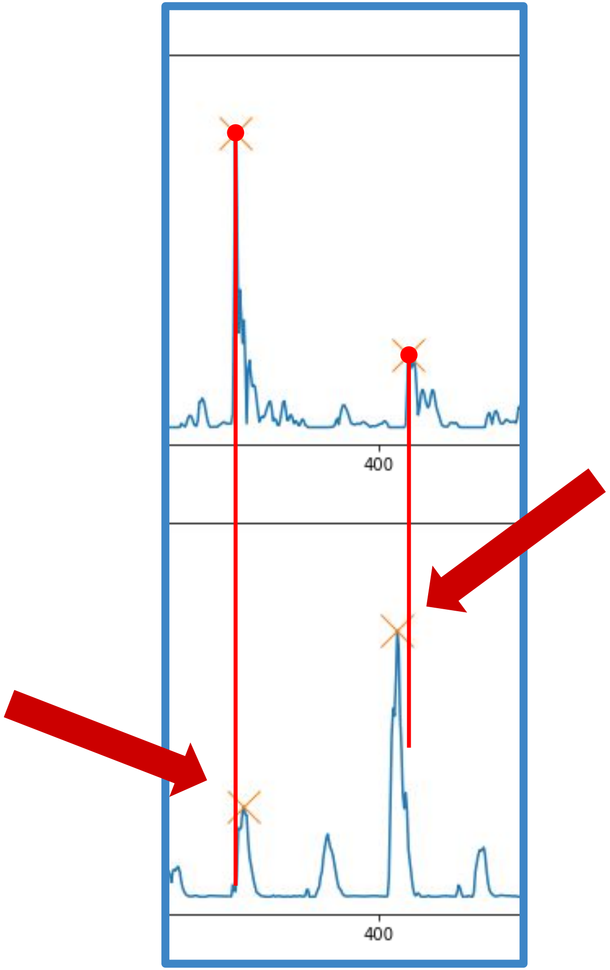
Utilize a Convolutional Neural Network (CNN) to translate a beatboxing recording into a corresponding audio track, matching the sound and timing of a performance using preloaded drum samples.



Project Purpose

- Facilitate quick form musical idea creation in a way that can be natural, intuitive, and inspiring for musicians
- Former NMF based solution relied heavily on analytically chosen parameters

  Train a CNN that will result in a more efficient, accurate, and robust translation



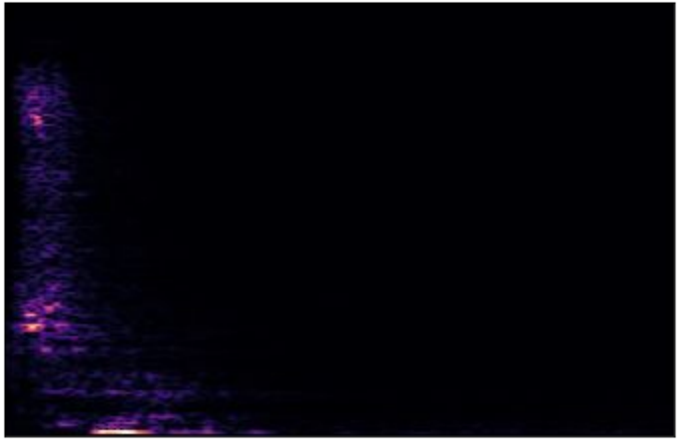
Mel Spectrograms

CNN Input:
Mel Spectrograms

Traditional Spectrograms

- Result of taking Fourier Transform of audio signal
- Visual representation of the frequency content of the audio signal
- Allows us to treat the audio information like any other image recognition task

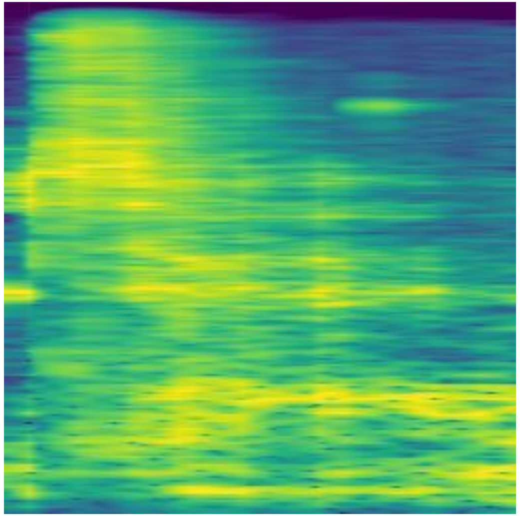
Hi-hat Closed



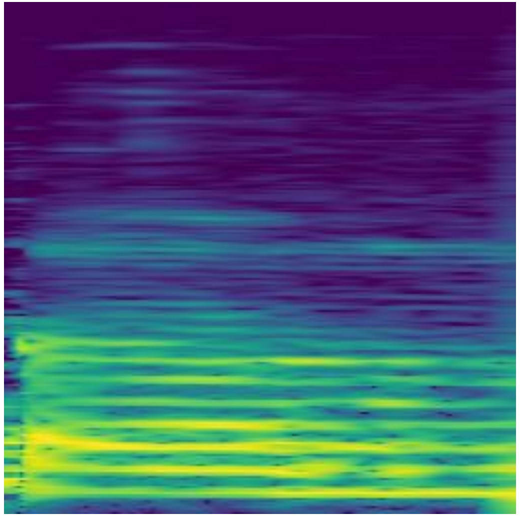
Kick

CNN Input:
Mel Spectrograms

- Represents the frequency information more closely to how humans actually perceive sound
- Focusing on the most important aspects
- More meaningful representations
- More visually distinct



Hi-hat
Closed



Kick

Dataset & Data Preparation

Dataset

- Amateur Vocal Percussion (AVP) dataset [1]
- Contains 28 Participants with the following .wav files:
  - High Hat Open
  - High Hat Closed
  - Kick
  - Snare
  - Improvisation
- Each .wav file 10-12 seconds long, with about 30 percussive sounds
- Each has accompanying .csv file which contains the onset time, instrument, and phenome of each sound

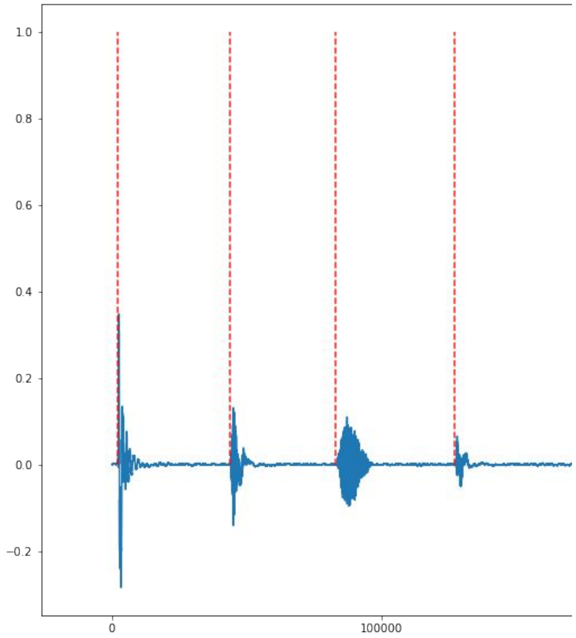
[1] A. Delgado, S. K. T. McDonald, N. Xu, and M. Sandler, “A new dataset for amateur vocal percussion analysis,” *Proceedings of the 14th International Audio Mostly Conference: A Journey in Sound*, 2019. DOI: 10.1145/3356590.3356844

Data Extraction

- Each of the sounds extracted from the .wav files using the corresponding onset

- sound[i]= audio[(onset[i]-1000) : onset[i+1]]
    - Where onset is the sound location in samples, and audio is the .wav file
    - Buffer the beginning of the sample by 1000 to prevent smearing in the mel spectrogram

- Each sound is then fixed to a length of 11025 samples



Data Augmentation

- To increase robustness of the network, the dataset is augmented with samples that have added noise, pitch shift, time shift, and gain change
- This doubles the amount of data from 4873 samples to 9746 samples

```python
def apply_augmentation(samples: np.array):

    gaussian_noise = AddGaussianNoise(
        min_amplitude=0.001,
        max_amplitude= 0.015,
        p=0.5
    )
    time_shift = Shift(
        min_fraction=-0.2,
        max_fraction=0.2,
        rollover=False,
        fade=True,
        p=0.5
    )
    pitch_shift = PitchShift(
        min_semitones=-0.5,
        max_semitones=0.5,
        p=0.25
    )
    gain = Gain(p=0.5)
    augmenter = Compose(
        [time_shift, gain, pitch_shift, gaussian_noise])
    return augmenter(samples=samples,
                     sample_rate=44100)
```

Data Preparation

- For each file:
    1. Extract each hit using the onset locations in the .csv
    2. Apply augmentation
    3. Get mel spectrogram of the augmented audio
    4. Add corresponding label, spectrogram, and audio to the dataset
- After data extraction and preparation, the mel spectrograms are saved into a DataLoader to avoid redundant preparation
- Training, validation and testing is a 3:1:1 split

CNN Model Architecture & Results

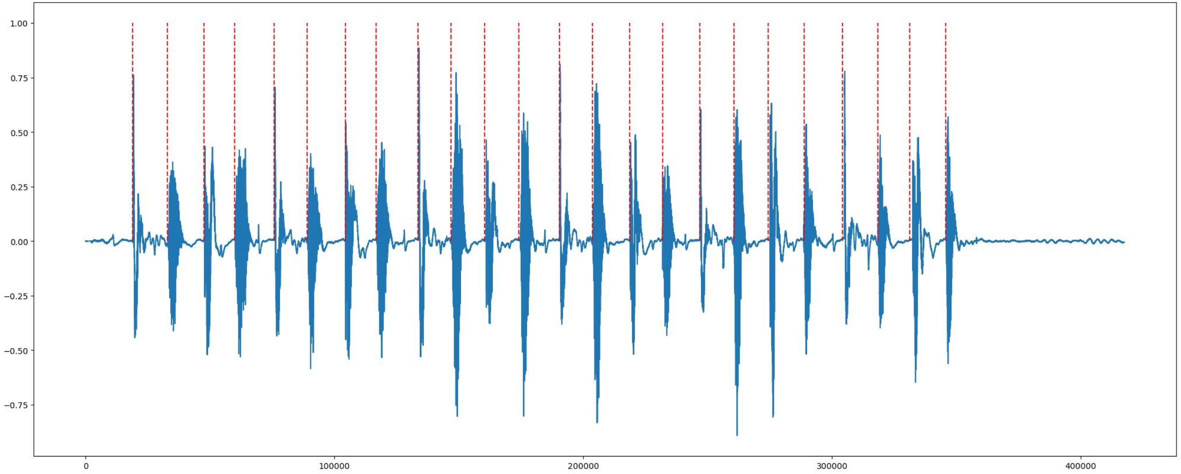
CNN Architecture and Training

- Input size of (N, 3, 256, 256)
- Peak training accuracy of ~97%
- Validation accuracy ~94%

```
==========================================================================================
Layer (type:depth-idx)                   Input Shape               Output Shape
==========================================================================================
Network                                  [1, 3, 256, 256]          [1, 4]
├─Sequential: 1-1                        [1, 3, 256, 256]          [1, 4]
│    └─Conv2d: 2-1                       [1, 3, 256, 256]          [1, 32, 252, 252]
│    └─MaxPool2d: 2-2                    [1, 32, 252, 252]         [1, 32, 126, 126]
│    └─ReLU: 2-3                         [1, 32, 126, 126]         [1, 32, 126, 126]
│    └─Conv2d: 2-4                       [1, 32, 126, 126]         [1, 64, 122, 122]
│    └─Dropout2d: 2-5                    [1, 64, 122, 122]         [1, 64, 122, 122]
│    └─MaxPool2d: 2-6                    [1, 64, 122, 122]         [1, 64, 61, 61]
│    └─ReLU: 2-7                         [1, 64, 61, 61]           [1, 64, 61, 61]
│    └─Flatten: 2-8                      [1, 64, 61, 61]           [1, 238144]
│    └─Linear: 2-9                       [1, 238144]               [1, 200]
│    └─ReLU: 2-10                        [1, 200]                  [1, 200]
│    └─Dropout2d: 2-11                   [1, 200]                  [1, 200]
│    └─Linear: 2-12                      [1, 200]                  [1, 4]
==========================================================================================
Total params: 47,683,500
Trainable params: 47,683,500
Non-trainable params: 0
Total mult-adds (M): 965.08
==========================================================================================
Input size (MB): 0.79
Forward/backward pass size (MB): 23.88
Params size (MB): 190.73
Estimated Total Size (MB): 215.40
==========================================================================================
```

Sample Triggering

Input Onset Detection & Separation

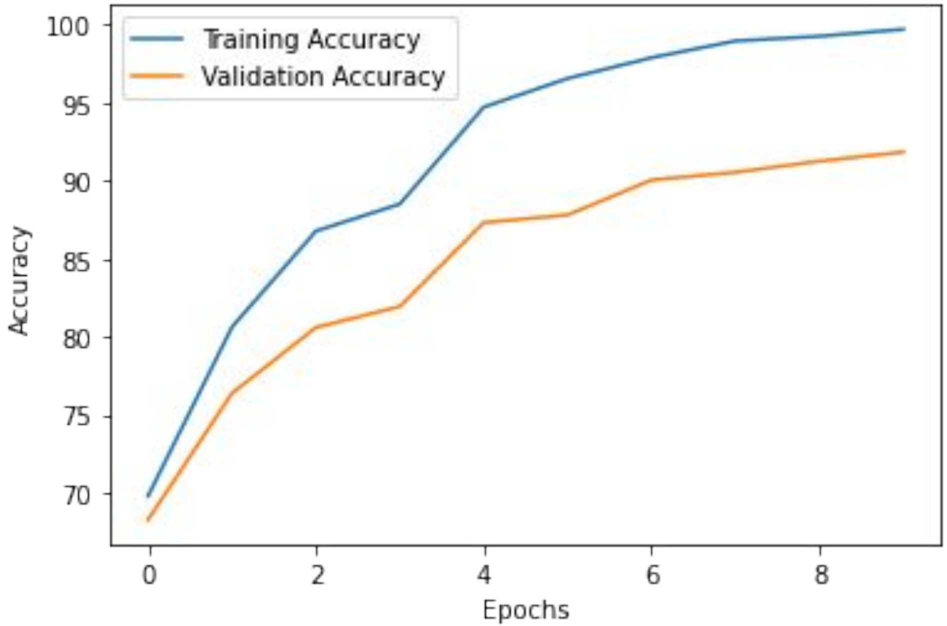


- Separate individual drum sounds using Librosa onset_detection function
  - Can encode detected onsets in time, frames, or samples
- Predict label of individual sound using CNN
- Based on label and detected onset, trigger sample

Future Considerations

Future Considerations

- Accuracy still improving at 10 epochs
    - Shows promise for expanded training
        - More epochs
        - More training data (more diverse, more augmentations)
- Product consideration - continued learning with user feedback



Questions?