– Independent Work Report Fall 2021 –

Deep Learning for Automatic Transcription of Human Beatboxing

Oleg Golev
Advisor: Adam Finkelstein

I pledge my Honor that this paper represents my own work in accordance with University regulations. - *Oleg Golev*

Abstract

Related to the problems of Automatic Music Transcription (AMT) and Automatic Drum Transcription (ADT), analysis and transcription of human beatboxing could be performed with similar techniques. A set of phoneme-based human drum imitation utterances belonging to one of four classes (bass drum, snare drum, closed hi-hat, and open hi-hat) was extracted from the Amateur Vocal Percussion (AVP) dataset. The first 28 mel-frequency cepstral coefficients (MFCCs) and their first and second deltas are generated from each utterance. This paper performs an experiment to test each of five proposed hypotheses inspired by the dataset and previous research. In each experiment, we train and test three neural networks for sound classification: a 2-layer fully connected neural network, a simple convolutional neural network, and a modified AlexNet. Each experiment is replicated on AVP's "Fixed" recordings, "Personal" recordings, and all recordings. We explore the effect of phoneme-consistency on classification accuracy, the effect of adding MFCC deltas to MFCCs for training, the dynamics of using some or none of the improvisation recordings for training, as well as the efficacy of our models in classifying sounds made by voices unseen during training.

1. Introduction

1.1. Motivation and Goal

The motivation for this project comes from the difficulty of obtaining an actual drum track, whether for personal projects, music compositions, or fast music prototyping. These are the two most common ways to obtain a custom drum track:

1. Record oneself or someone else playing the desired drum pattern on a drum set or drum pads
2. Have oneself or someone else synthesize the drum track digitally in Digital Audio Workstation (DAW) software such as Ableton Live or FL Studio

While the above two methods take time and require some level of proficiency in either playing the drum set or using a DAW, most people have the ability to phonetically imitate drum sounds (vocal percussion or human beatboxing) and record them using a mobile device. One area of active research is creation of a sophisticated piece of software that can take an audio recording of human beatboxing (Figure 1) and translate it into a recording with the human imitations replaced by real drum set samples (Figure 2).

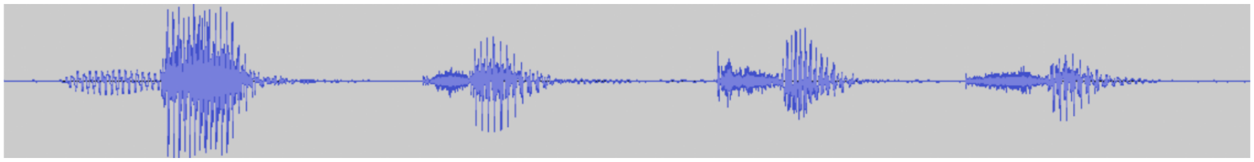


Figure 1: Waveform representation of an audio file. The recording is of the phrase "boots and cats" spoken by an ameteur beatboxer with minimal vocalization. The four non-silent segments are (left-to-right) for the imitations of a kick drum, closed hi-hat, snare drum, and closed hi-hat respectively.

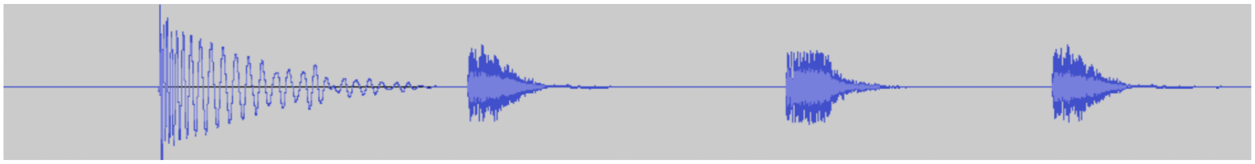


Figure 2: Waveform representation of the recording shown in Figure 1, but with beatbox imitations replaced by samples of a real kick drum, snare drum, and closed hi-hat.

A piece of software designed for beatbox transcription would take a beatbox recording as input and process it in three stages:

1. Location of the different sounds' onsets and offsets, using common methods such as non-negative matrix factorization (NMF) [3] or recurrent neural networks, which provided better results than state of the art NMF systems in multiple studies [13] [15]
2. Feature extraction and classification of the different sounds found in the recording
3. Replacement of the human beatbox sounds with user-specified drum samples

There are both free and paid products on the market that achieve the goal of beatbox transcription to MIDI, yet they are largely unreliable. Free VST plugins like KTDrumTrigger[1] are not accurate at transcribing multiple classes of beatbox sounds in a single recording and require user input to tune several analysis parameters to obtain the desired transcription results. The plugin relies on a frequency-based trigger system, so the tuning of the plugin is not easily generalizable across users, their distinct voices, and their interpretations of how to imitate different drum set sounds. Similar paid plugins like Drumagog[2] perform a similar function of monophonic drum transcription, but still need manual parameter tuning of frequency thresholds for different sounds. Arguably the best automatic "to-MIDI" beatbox transcription software is currently Ableton Live's "Convert Drums to New MIDI Track" function[3], yet it detects a lot of extraneous events, achieving low F_1-scores of 0.518, 0.470, and 0.297 in detection of the kick drum, snare drum, and hi-hat imitations in beatbox recordings [10].

At this moment, there does not exist a generalized plugin, feature, or separate piece of software that accurately and reliably automates transcription of amateur beatboxing audio. There has also been substantially little research exploring the applications of deep learning models in beatbox transcription. However, a lot of research in Automatic Music Transcription (AMT) and Automatic Drum Transcription (ADT) [16] can be directly applicable to the problem of Vocal Percussion Transcription (VPT). With recent ADT research showing basic CNN models achieve 97% classification

[1] http://koen.smartelectronix.com/KTDrumTrigger/

[2] https://www.drumagog.com/

[3] https://www.ableton.com/en/manual/converting-audio-to-midi/

accuracy, outperforming k-NN's and random forests, in classifying among the real kick drum, hi-hat, and snare recordings [5], there seems to be a lot of potential in applying deep learning to beatbox sound classification.

The goal of this paper is thus to explore three neural network architectures and see whether they provide better testing accuracy on a relatively new dataset of beatboxing audio than what was accomplished by previous research.

1.2. Related Work

The most recent complete study on designing a more effective beatbox transcription software was undertaken by Ramires as part of his Master's thesis, where he created a new Ableton Live plugin LVT which can be used to translate beatboxing into drum transcriptions [11]. In his study, he collected two beatbox recordings from each of twenty amateur participants who imitated the kick drum, snare drum, and hi-hat in a vocalized fashion. The high frequency content (HFC) algorithm (as implemented by the *aubioonset*[4] command-line tool) was used for onset detection in the recordings, and the k-nearest neighbours algorithm was used for classification of the sounds between the three classes. The input to the k-NN algorithm included a large set of both temporal and spectral (including MFCCs) features. LVT achieved much higher accuracy and F-measure results than LDT and Ableton Live's built-in "Convert Drums to New MIDI Track" feature. While LVT achieved an F-measure of 0.9114 in transcribing kick drum imitations recorded on a professional AKG microphone, the transcription accuracy of snare drums and hi-hats were less than ideal, at 0.691 and 0.802 respectively. This seems to be a limitation of the k-NN algorithm and the dataset, based on the size of the dataset and the drum sound imitations being differently interpreted across participants.

In an older 2005 study, Hazan faced a similar difficulty as Ramirez in retaining accuracy when faced with cross-participant differences in drum sound interpretations [6]. The researchers attempted a similar system to what is covered in this project: sound separation, feature extraction, and classification. Using a self-made dataset of 304 total drum imitation utterances (bass drum, snare

[4] https://aubio.org/manpages/latest/aubioonset.1.html

drum, open hi-hat, or closed hi-hat) extracted from recordings by 4 participants, the study achieved 90.0% classification accuracy using the C4.5 algorithm with bagging on a test set of 62 instances. In terms of features, Hazan used both temporal and spectral features, where the zero-crossing rate, attack and decay durations, kurtosis, second and fourth MFCCs were the most significant in the task. The researchers found that once the test set included drum imitations that are characterized by different phonemes than the ones used in training, the accuracy dropped quickly. This is important as for the production of an industrial-level software that can transcribe beatboxing to drums, the software might have to classify sounds made by users who have different interpretations for the imitations of different drum set components.

In the same year, a dataset that ensured consistent interpretations of drum sounds across participants was made and used by Sinyor et al. to assess classification accuracy of SVM (83.8%) and AdaBoost with C4.5 (93.37%) on a corpus of five beatbox sound classes: kick drum, p-snare, k-snare, closed hi-hat, and open hi-hat [12]. When the classes were reduced only to the kick drum, snare, and hi-hat, AdaBoost with C4.5 achieved an accuracy of 98.15% after feature selection. It is important to note that the dataset was only made from five participants, three of which were trained beatboxers, and all were told to enunciate each sound in an unpitched manner. That is, the dataset was not phoneme-based, but rather boxeme-based (the beatboxing equivalent of phonemes), with the sounds most closely resembling drums rather than spoken syllables. For amateurs, more drum-like beatboxing sounds are hard to produce, but the success of this study shows great promise in the possibilities of transcribing more professional beatbox audio, where the rhythmic patterns may be faster, include a higher variety of inter-woven sounds, and are harder to replicate using the drum set and DAWs.

Kapur et al. produced a similar application to Ramires's LVT and trained a neural network classifier that achieved 97.3% testing accuracy (kicks, snares and hi-hats) using the zero-crossing rate in their features [14]. However, the dataset only consisted of 75 sound utterances made by two participants and is therefore hardly generalizable. Ending up with similar limitations, Ramanathan created a dataset with samples of kick drum, snare drum, open hi-hat, and closed hi-hat imitations only using his own voice [9]. While his SVM and neural network achieved classification accuracies of 95.7% and 96% respectively, the accuracy of the models on utterances made by another individual are unknown.

Given that amateur beatboxing is often vocalized, the efficacy of speech recognition systems was assessed by Evain et al. for beatbox sound classification [4]. The researchers adapted the Kaldi ASR toolkit's HMM-GMM speech recognition approach to analyze recordings made by two beatboxers, one professional and another amateur. With 80 classes of boxemes, their models achieved 86.53% boxeme recognition accuracy. Picart et al. conducted a similar study, using the HTK toolkit's HMM speech recognition model [8]. On a custom dataset produced by two professional beatboxers, the model achieved a 9% word error rate (WER) differentiating between five non-pitched drum imitations (cymbal, hi-hat, kick, rimshot and snare). Both studies indicate speech recognition systems' potential in the task of beatbox sound classification (given the large boxeme and phoneme pools in the dataset of both studies), but not their generalizability due to their limited number of participants.

1.3. Amateur Vocal Percussion (AVP) Dataset

Previous studies that consider beatbox sound classification (whether vocalized or non-vocalized) generally present non-generalizable results due to a small number of participants involved in making the dataset. Additionally, the participants varied significantly in their skill levels across studies, with some being trained beatboxers who could imitate drum sounds in a more authentic unpitched way while others were amateurs who vocalized their drum imitations. In an attempt to provide a consistent dataset that contained recordings reflective of amateur beatbox performances, the Amateur Vocal Percussion (AVP) dataset was created. The dataset contains 9780 phoneme-based utterances meant to imitate one of four drum set sounds: kick drum (kd), snare drum (sd), closed hi-hat (hhc) and open hi-hat (hho) [2]. Most importantly, the recordings were made by 28 participants who all had little to no experience beatboxing.

The dataset is split into two sections: *Personal* and *Fixed*. For each section, every participant recorded five tracks: for each sound, a track with multiple utterances of just that sound, and an improvisation track, where the participants could use any of the four sounds to create their own rhythmic patterns.

In the *Personal* section of the dataset, the participants were free to provide their own interpretations for each of the requested sounds. That is, this portion of the dataset is perfect for a classification models' robustness when classifying drum imitations that are not phoneme-consistent between participants, as this was a concern directly raised by Ramires [11] and Hazan [6] in their respective studies. We call these recordings *personalized*.

For the *Fixed* section of the dataset, the participants were told the precise phoneme to use to represent each of the four classes of drum sounds. That is, this portion of the dataset is perfect to gauge the efficacy of classification models in distinguishing between drum imitations that are phoneme-consistent between participants. A beatbox transcription software may do something similar: ask the user to use a specific phoneme for each drum sound. We call these recordings *non-personalized*.

Each recording is provided in a *.wav* file and has a corresponding *.csv* file which contains the labels and onsets for each beatbox sound in that recording (Figure 3).

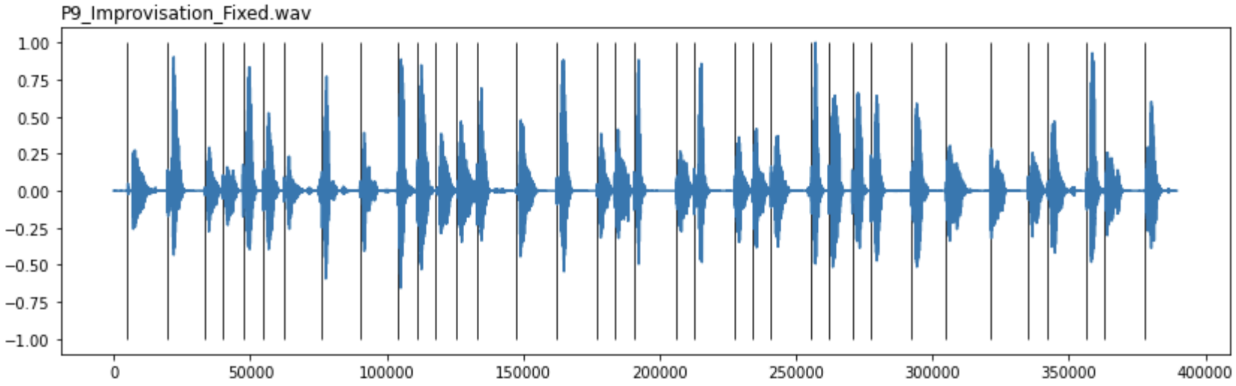


Figure 3: Waveform representation of the improvisation done by the 9th participant for the *Fixed* section of the AVP dataset. The vertical black bars mark the onsets for each separate beatbox sound utterance.

1.4. Related Work with the AVP Dataset

There are currently two papers available that use the Amateur Vocal Percussion dataset for the purposes of human beatbox sound classification.

In his Master's thesis, Mukhutdinov [7] uses two feature sets, one that consists of the first 20 MFCCs computed from the first 4096 frames of each utterance, and the other being the same feature set used by Ramires [11]. Mukhutdinov performs deep feature extraction using convolutional autoencoders (CAEs) for use in a k-NN model and a 3-layer fully-connected neural network. The CAEs were trained on features from a combination of beatbox and real drum samples, and the neural network was trained on a subset of non-improvisation utterances from the AVP dataset and samples from the *beatboxset1*[5] dataset. While Mukhutdinov achieves 97.65% accuracy using the k-NN classifier using one of the CAE types, it is unknown whether the testing set consists of other non-improvisation utterances from the AVP dataset or some other subset of his data pool. Generally however, the neural network classifiers underperformed in comparison to k-NN classifiers. Besides these, Mukhutdinov made several other important findings:

1. Training on a single person's utterances produced better transcriptions for that person than training on a joint set made from utterances of several participants.
2. Integration of a language model (DrumsRNN) was detrimental to transcription accuracy, likely due to lack of consistent rhythmic structure in the AVP dataset participants' improvisations. However, we suggest that the use of language models would be more useful for more professional beatbox recordings.
3. Classification accuracy is higher on the non-personalized than personalized recordings, likely due to cross-participant phoneme consistency.

While these findings are in-line with previous research, Mukhutdinov's testing sets are ambiguous, and his classification models lack variety. To explore deep learning models specifically, the creators of the AVP dataset released a comparative study assessing their efficacy concurrently with the course of this paper [1].

[5] https://archive.org/details/beatboxset1

The researchers used the non-improvisation utterances for their training set and improvisation utterances for their testing set. For features, they used the first 13 MFCCs, the zero-crossing rate, and a set of spectral descriptors. The features were selected using random forests and fed into a set of traditional classification models, as well as three different types of deep learning models: a multi-layer perceptron (MLP), a variety of CNN's, and a triplet-loss CNN. They found the regular CNN to perform best on both the original (77.7%) and augmented datasets (82.2%), although these accuracies were not much higher than of other deep learning models or random forests. None of the deep learning models were sophisticated either (although optimized), so it is also possible to improve upon their findings.

2. Statement of Purpose

The purpose of this research is to assess the efficacy of deep learning models in classifying human beatbox sounds and to test a set of hypotheses inspired by the AVP dataset and issues raised by previous research.

3. Approach and Implementation

3.1. Data Processing and Tools

All code was written in Python in Google Colab Notebooks. The audio files from the AVP dataset were normalized and then enhanced using SpeechBrain's mimic-loss-trained model. Every recording was then split into clips, where each clip contains only a singular utterance of one of the four drum imitations: kick drum, snare drum, open hi-hat, or closed hi-hat.

The start of each clip was determined by the onset provided in the dataset. The end of each clip was either 500ms after the start, the end of the recording, or at the onset of the next utterance in the recording, whichever came first. We used 500ms because the average duration of any utterance is "0.456 s for AVP Fixed and 0.433 s for AVP Personal" [7]. Likewise, most information required for classification seems to be contained at the very beginning of the clip, so we were not worried about cutting some longer utterances short.

The first 28 MFCCs (28x28 matrix) and their first- and second-order derivatives were extracted from each clip using the Librosa library. The deep learning models were written using Keras. Numpy was used to process and store all of the multi-dimensional data. Matplotlib was used to produce model training graphs (for tracing by-epoch behavior) and visualize MFCCs. The confusion matrices were constructed using Matplotlib, seaborn, and sklearn libraries.

3.2. MFCCs

The term *MFCCs*[6] is short-hand for *mel-frequency cepstral coefficients* which make up the mel-frequency cepstrum (MFC), a representation of an audio recording's power spectral density. The idea is that each sound class is expected to have a general MFCC representation that is distinct from those of other sound classes (Figure 4). To test one of our hypotheses, we also extracted the first and second MFCC deltas, which show how the MFCCs change between audio frames.

In this study, we use MFCCs because they have been consistently shown to be one of the most deterministic features in sound classification both in language models and for transcription purposes [6] [11]. Mihmudinov [7] later found that using MFCCs for classification either matched or outperformed the feature set defined by Ramires.

3.3. Experimental Design and Hypotheses

This paper conducts an experiment and assesses its results for each of the following five hypotheses. For each experiment, we also conduct three trials: one using only the personalized recordings, one using only the non-personalized recordings, and one using the entire dataset (mixed).

3.3.1. Experiment #1

In the personalized recordings, different participants were free to interpret the requested drum sounds as they saw fit, with some participants using different phonemes for the same drum sound, or similar phonemes for different drum sounds. Thus we expect the classification accuracy of all models to be noticeably lower for the personalized recordings than the non-personalized recordings. This has been supported by previous studies [6] [7] [11], and we explore this difference again in

[6] https://en.wikipedia.org/wiki/Mel-frequency$_cepstrum$

this paper. All improvisation and non-improvisation clips were pooled and separated into training, validation, and testing sets with a 70-15-15 percent split.

3.3.2. Experiment #2

We are interested in exploring whether adding the first and second MFCC deltas to the feature set of MFCCs would improve classification accuracy. Using MFCC deltas is common in speech recognition applications and was used previously by Picart et al. [8]. The AVP dataset contains phoneme-based drum imitations, so classifying these sounds may be similar to classifying recordings of syllables in a spoken language. For this reason, we suspect that using MFCC deltas alongside the MFCCs may significantly improve classification accuracy, especially when dealing with personalized recordings. All improvisation and non-improvisation clips were pooled and separated into training, validation, and testing sets with a 70-15-15 percent split.

3.3.3. Experiment #3

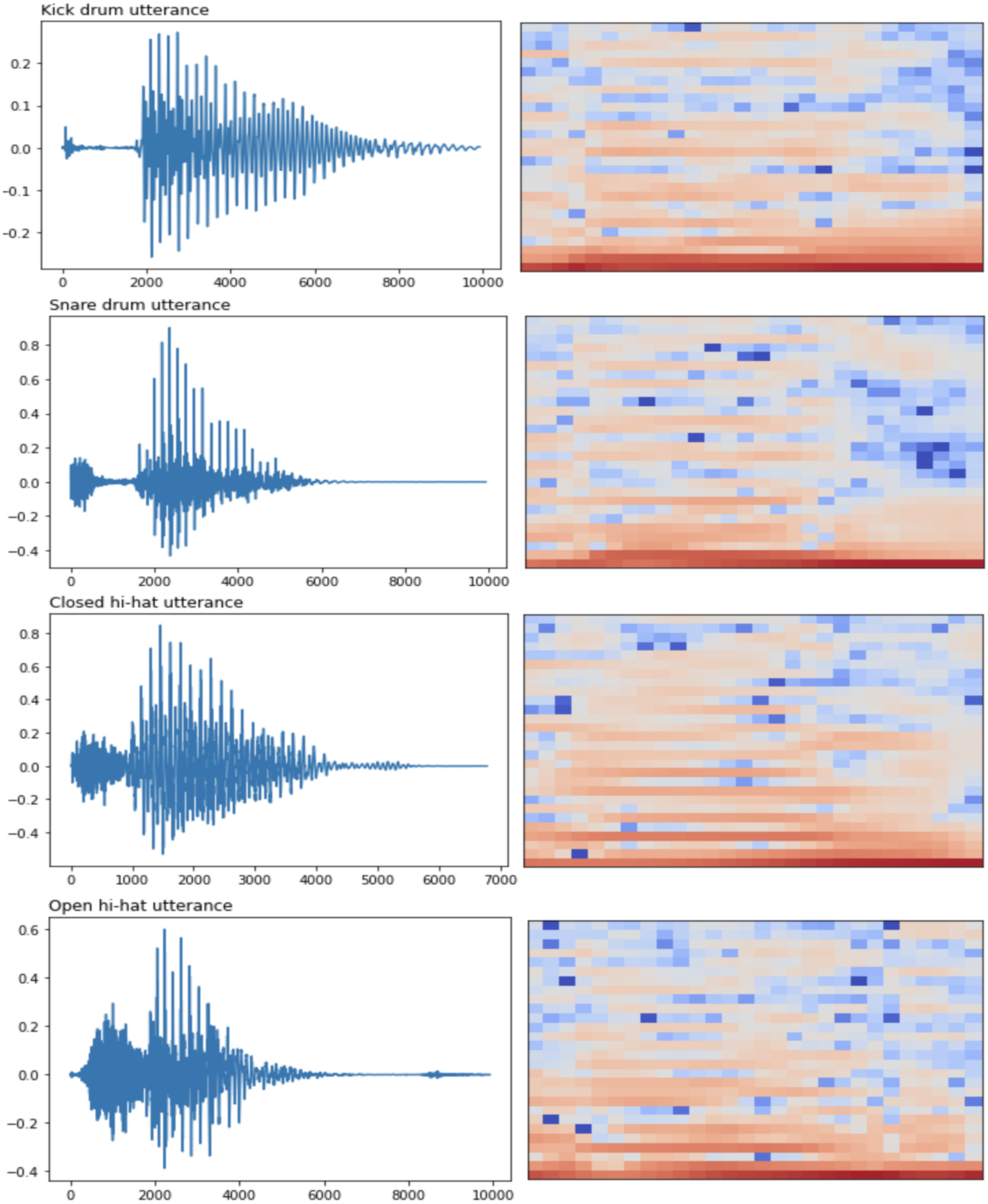
The AVP dataset contains two types of audio files: recordings that contain repeated utterances of a specific drum sound imitation, and a rhythmic improvisation that uses all four classes of drum sound imitation. When thinking about a piece of software that can transcribe a user's beatbox recording, it is reasonable to consider the possibility of the software asking the user to produce their own imitation utterances of each desired drum sound to train a classification model before the user can improvise. For this reason, we want to explore the classification accuracy that our models can achieve when using utterances from non-improvisation recordings for the training set, and utterances from improvisation recordings for the validation and testing sets (50-50 percent split), similar to what was done by the creators of the dataset themselves [1]. We expect the accuracy to drop somewhat but insignificantly from what we would achieve by training and testing the models on a mix of both improvisation and non-improvisation utterances.

3.3.4. Experiment #4

To make the above an ablation experiment, it would be interesting to see how much the classification accuracy changes if we use a third of the improvisation utterances in the training set, leaving a third for validation and the other third for testing. We expect that the accuracy would increase substantially, nearly to the levels achieved when training and testing on both improvisation and non-improvisation utterances, like in the first experiment.

3.3.5. Experiment #5

Finally, we explore whether the models would be robust to classifying utterances made by participants who do not have any of their utterances included in the training set. Specifically, we use all clips from the first 24 participants for the training set and from the last 4 participants (2 male and 2 female) for the testing set. We expect that classification accuracy will be lowest in this experiment as each voice is unique, and 28 participants are not enough to account for the wide range of vocal characteristics that can cause misclassification. However, we still expect higher classification accuracy for the non-personalized than the personalized recordings.



Kick drum utterance

Snare drum utterance

Closed hi-hat utterance

Open hi-hat utterance

Figure 4: A waveform representation and its 28x28 MFCC representation (time on the x-axis, frequency on the y-axis, and deeper colors corresponding to higher intensity) of a selected utterance for each sound class. The utterances were extracted from the 9th participant's improvisation audio file in the *Fixed* section of the AVP dataset.

3.4. Models

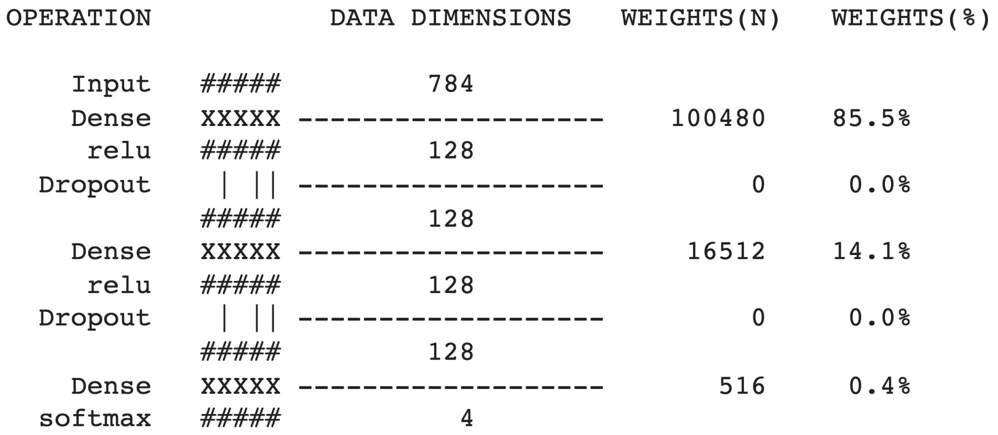
After processing the data into clips and extracting features from each clip, three models were trained and tested. A *keras_sequential_ascii*-generated summary for each model is provided below (Figures 5, 6, 7). These models were kept consistent between experiments. The final activation layer in each model uses the softmax function, such that the *i*'th element of the 4-dimensional output vector is given by:

$$\sigma(\vec{z}_i) = \frac{e^{z_i}}{\sum_{j=1}^{4} e^{z_j}}$$

All other convolutional and dense layers use ReLU activation, unless they are followed by a batch normalization layer.

For the second experiment, the models were modified to take data in the shape [28, 28, 3], and the number of nodes in the dense layers of the DNN and CNN models were increased from 128 to 512.

3.4.1. Fully-Connected Neural Network (DNN)

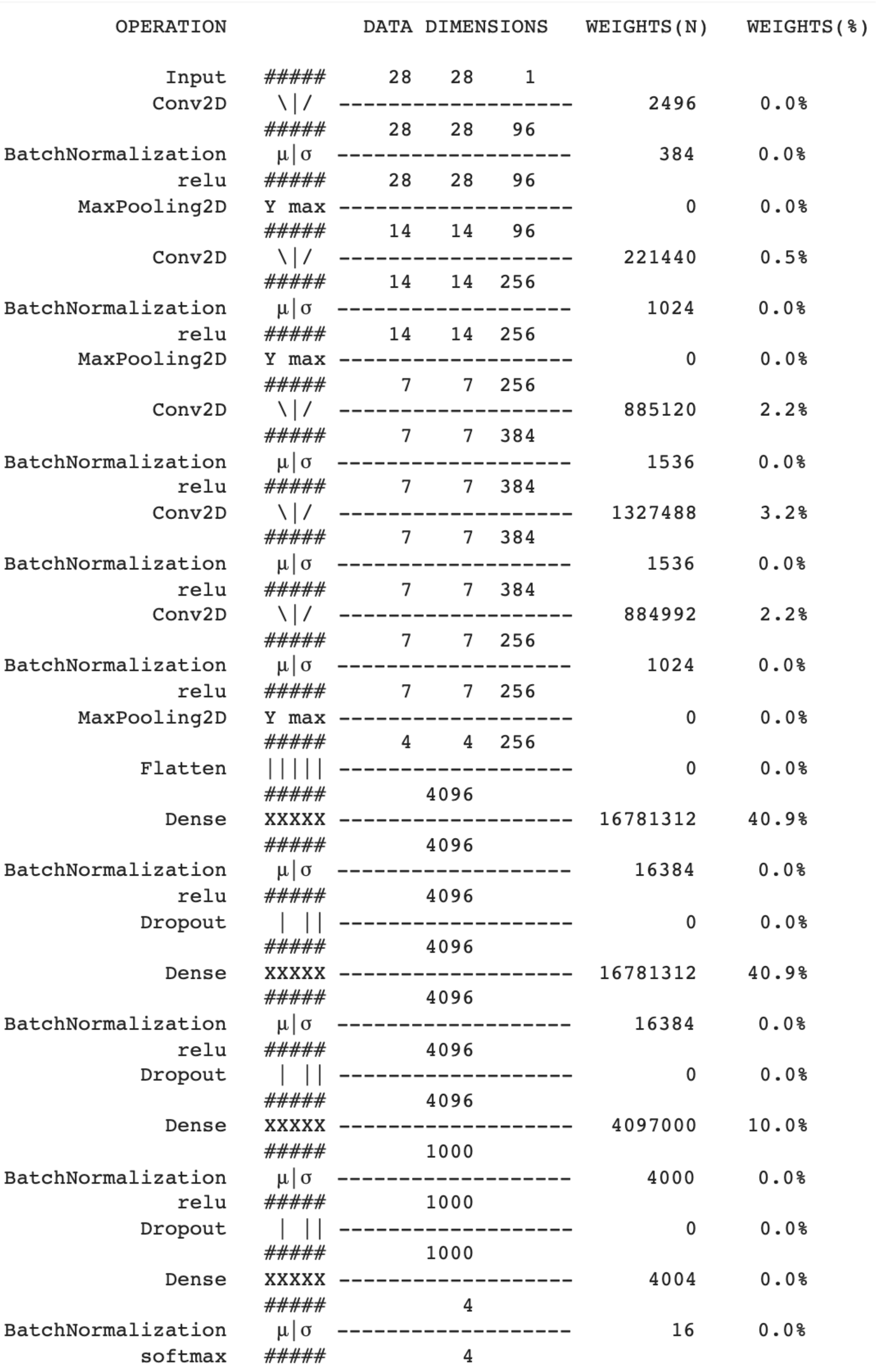


```
OPERATION           DATA DIMENSIONS   WEIGHTS(N)   WEIGHTS(%)

    Input   #####         784
    Dense   XXXXX --------------------    100480    85.5%
     relu   #####         128
  Dropout    | || --------------------         0     0.0%
            #####         128
    Dense   XXXXX --------------------     16512    14.1%
     relu   #####         128
  Dropout    | || --------------------         0     0.0%
            #####         128
    Dense   XXXXX --------------------       516     0.4%
  softmax   #####           4
```

Figure 5: A simple dense neural network architecture with two hidden layers.

3.4.2. Modified AlexNet (AlexNet)

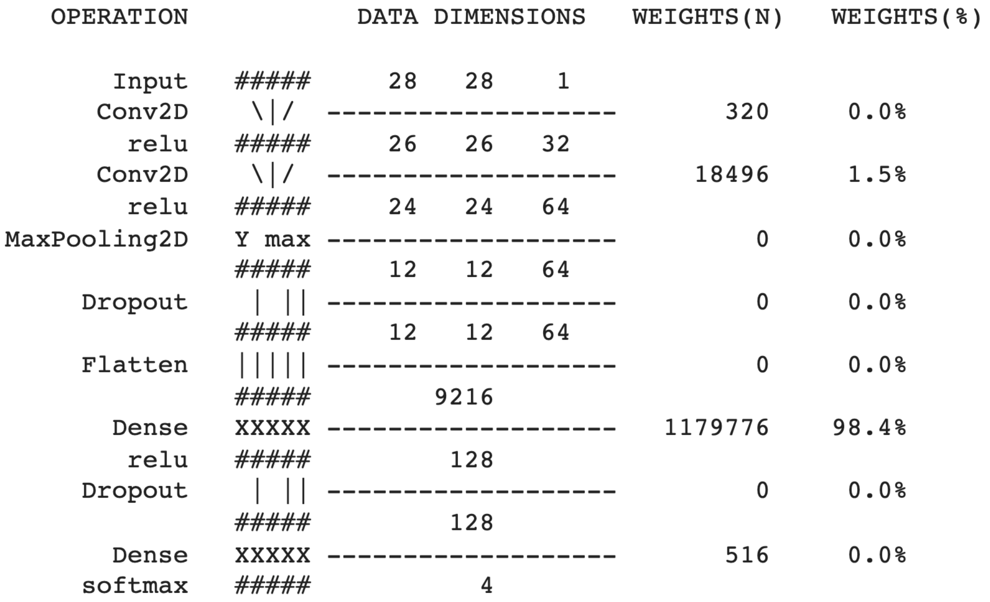


```
           OPERATION           DATA DIMENSIONS   WEIGHTS(N)   WEIGHTS(%)

               Input   #####     28   28    1
              Conv2D    \|/  -------------------      2496     0.0%
                       #####     28   28   96
  BatchNormalization    μ|σ  -------------------       384     0.0%
                relu   #####     28   28   96
        MaxPooling2D   Y max -------------------         0     0.0%
                       #####     14   14   96
              Conv2D    \|/  -------------------    221440     0.5%
                       #####     14   14  256
  BatchNormalization    μ|σ  -------------------      1024     0.0%
                relu   #####     14   14  256
        MaxPooling2D   Y max -------------------         0     0.0%
                       #####      7    7  256
              Conv2D    \|/  -------------------    885120     2.2%
                       #####      7    7  384
  BatchNormalization    μ|σ  -------------------      1536     0.0%
                relu   #####      7    7  384
              Conv2D    \|/  -------------------   1327488     3.2%
                       #####      7    7  384
  BatchNormalization    μ|σ  -------------------      1536     0.0%
                relu   #####      7    7  384
              Conv2D    \|/  -------------------    884992     2.2%
                       #####      7    7  256
  BatchNormalization    μ|σ  -------------------      1024     0.0%
                relu   #####      7    7  256
        MaxPooling2D   Y max -------------------         0     0.0%
                       #####      4    4  256
             Flatten   ||||| -------------------         0     0.0%
                       #####        4096
               Dense   XXXXX -------------------  16781312    40.9%
                       #####        4096
  BatchNormalization    μ|σ  -------------------     16384     0.0%
                relu   #####        4096
             Dropout    | || -------------------         0     0.0%
                       #####        4096
               Dense   XXXXX -------------------  16781312    40.9%
                       #####        4096
  BatchNormalization    μ|σ  -------------------     16384     0.0%
                relu   #####        4096
             Dropout    | || -------------------         0     0.0%
                       #####        4096
               Dense   XXXXX -------------------   4097000    10.0%
                       #####        1000
  BatchNormalization    μ|σ  -------------------      4000     0.0%
                relu   #####        1000
             Dropout    | || -------------------         0     0.0%
                       #####        1000
               Dense   XXXXX -------------------      4004     0.0%
                       #####           4
  BatchNormalization    μ|σ  -------------------        16     0.0%
             softmax   #####           4
```

Figure 6: AlexNet with the kernel size and stride adjusted for the 28x28 MFCC input.

3.4.3. Convolutional Neural Network (CNN)



```
   OPERATION           DATA DIMENSIONS   WEIGHTS(N)   WEIGHTS(%)

       Input   #####     28   28    1
      Conv2D    \|/  -------------------       320     0.0%
        relu   #####     26   26   32
      Conv2D    \|/  -------------------     18496     1.5%
        relu   #####     24   24   64
MaxPooling2D   Y max -------------------         0     0.0%
               #####     12   12   64
     Dropout    | || -------------------         0     0.0%
               #####     12   12   64
     Flatten   ||||| -------------------         0     0.0%
               #####        9216
       Dense   XXXXX -------------------   1179776    98.4%
        relu   #####         128
     Dropout    | || -------------------         0     0.0%
               #####         128
       Dense   XXXXX -------------------       516     0.0%
     softmax   #####           4
```

Figure 7: A simple convolutional neural network architecture reminiscent of a VGG block.

3.4.4. Training

All models were trained on MFCCs and their first and second deltas for the second experiment and just on MFCCs for all other experiments. All models were trained for a maximum of 50 epochs with a batch size of 32 and a learning rate of 0.0001 for the DNN, and 0.00001 for the CNN and AlexNet. To avoid overfitting in several experiments, some models had their dropout rate increased in the dropout layers, learning rate adjusted, and/or batch size increased to 64 or 128. All models used the Adam optimiser and either categorical cross-entropy or sparse categorical cross-entropy loss. The number of trainable parameters for each model is provided below (Table 1).

Model	#Parameters
DNN	117,508
CNN	1,199,108
AlexNet	41,006,308

Table 1: The number of trainable parameters in the selected models.

Given the number of trainable parameters, training was the longest for AlexNet, followed by the CNN, with the DNN taking the shortest time to train. To justify the much longer time cost of training AlexNet, it should perform substantially better compared to the CNN and DNN models.

3.5. Evaluation

For each experiment, we report the testing set accuracy and the highest validation accuracy achieved during training to better contextualize the performance of our non-optimized models.

Confusion matrices, training loss and accuracy plots were generated to help adjust parameters and compute the accuracy measures. The confusion matrices are provided in Appendix A and discussed throughout the next section to help us interpret our results.

4. Results and Discussion

We present and analyze the results of the experiments cumulatively. For each experiment, we report FTA = Final Testing Accuracy and HVA = Highest Validation Accuracy and then explain what we learn from all results up to that point.

4.1. Experiment #1

In the first experiment, we split all utterances into training, testing, and validation using a 70-15-15 percent split. The intent was to observe the overall performance of the selected classification models with similar composition of the training and testing sets. The results are presented in Table 2.

Model	Personalized		Non-personalized		Mixed	
	FTA(%)	HVA(%)	FTA(%)	HVA(%)	FTA(%)	HVA(%)
DNN	85.31	85.99	94.86	97.02	87.50	89.88
CNN	**91.11**	92.17	96.86	97.97	92.07	92.94
AlexNet	90.57	**92.31**	**97.86**	**98.11**	**93.56**	**93.48**

Table 2: Testing and highest validation accuracy scores obtained for the deep learning models in the first experiment. Best scores for each dataset partition are in bold.

We hypothesized that accuracy would generally be better when trained and tested on non-personalized recordings than personalized recordings. Looking at the accuracies achieved by all models, we confirm our hypothesis as we achieved substantially higher accuracy scores when using the non-personalized recordings than the personalized or mixed recordings. The phoneme consistency in the non-personalized section of the dataset played a crucial role in reducing the number of misclassifications. The highest accuracy achieved was the highest validation accuracy seen when training AlexNet: 98.11%. This is higher than what was seen in most previous research and on par with accuracy achieved by Sinyor et. al (98.15%) [12], although they used a different dataset with three instead of four classes. Another observation is that while AlexNet performed better than either DNN and CNN (except for the test accuracy on the personalized dataset, with 91.11% by the CNN vs. 90.57% by AlexNet), the difference was not substantial enough to warrant the much longer training time. While AlexNet took hours to train, the CNN could be trained in only around 10 minutes. Finally, the confusion matrices (A.1) reveal the general difficulty of the DNN to distinguish between the snare drum, closed hi-hat, and open hi-hat in personalized and mixed recordings, compared to the CNN and AlexNet, although these two models had a similar but less prominent misclassification pattern.

4.2. Experiment #2

In the second experiment, the training, validation, and testing set were set up exactly the same as in the first. However, we trained the models on the first and second MFCC deltas in addition to the MFCCs we use in all other experiments. The intent was to see whether training on deltas in addition to MFCCs would substantially improve classification accuracy compared to what we achieved in the first experiment. The results are presented in Table 3.

Model	Personalized		Non-personalized		Mixed	
	FTA(%)	HVA(%)	FTA(%)	HVA(%)	FTA(%)	HVA(%)
DNN	89.82	90.07	95.42	96.99	90.57	92.07
CNN	88.25	91.82	97.14	**97.54**	92.39	93.98
AlexNet	**91.64**	**92.41**	**97.77**	97.26	**93.67**	**94.33**

Table 3: Testing and highest validation accuracy scores obtained for the deep learning models in the second experiment. Best scores for each dataset partition are in bold.

We hypothesized that since the recordings are phoneme-based and thus similar to syllables of spoken languages, we would see significant improvements in classification accuracy, specifically when working with personalized recordings. Compared to previous results in Table 2, adding the MFCC deltas most noticeably increased the classification accuracy of the DNN across the board (testing accuracies rose from 85.31%, 94.86%, and 87.50% to 89.82%, 95.42%, 90.57% for the personalized, non-personalized, and mixed recordings respectively). The largest accuracy increase was the 4.51% jump for the DNN with personalized recordings, as hypothesized. However, the CNN and AlexNet saw little to no improvement. Given that adding MFCC deltas to the feature set increases training time significantly for those models, doing so for the CNN and AlexNet is impractical. The DNN however trains really quickly and therefore sees the most benefit from adding these features. While feature set selection was not the focus of this paper, it is a good area of exploration, as well as model optimization, to see to what extent the classification accuracies could be maximized. Looking at the confusion matrices (A.2), we see patterns similar to what was revealed in the first experiment. One other observation is that a noticeable number of snare drum utterances were misclassified as kick drum utterances by the CNN for personalized recordings. This is not as big of a problem in neither non-personalized nor mixed recordings, and no other model produced the same behavior.

4.3. Experiment #3

In the third experiment, the training set consisted of only non-improvisation utterances, and the testing set consisted of only improvisation utterances. This experiment was performed by Delgato et al. [1] in their latest paper, and we replicate it here in hopes of achieving better results (Table 4). The intent is to see whether improvisation and non-improvisation instances are so distinct that the testing accuracy would drop significantly when the models are trained and tested this way.

Model	Personalized		Non-personalized		Mixed	
	FTA(%)	HVA(%)	FTA(%)	HVA(%)	FTA(%)	HVA(%)
DNN	61.02	61.50	88.40	89.27	72.88	74.85
CNN	**67.19**	66.34	89.79	90.28	**76.53**	**78.12**
AlexNet	66.59	**67.68**	**90.42**	**90.78**	75.97	76.76

Table 4: Testing and highest validation accuracy scores obtained for the deep learning models in the third experiment. Best of scores for each dataset partition are in bold.

Delgado et al. achieved the highest accuracy of 77.7% using a CNN trained on utterances from mixed recordings. This is comparable to our 76.53% and 78.12% accuracies achieved by our CNN. Unfortunately, AlexNet did not perform as well as the CNN on mixed recordings, and performed very similarly to the CNN on all partitions, more generally, making it impractical due to longer training times. We hypothesized that the accuracies would not drop significantly from our results in the first experiment. Unfortunately, our hypothesis seems to hold up only for the non-personalized recordings (the testing accuracy on AlexNet "only" dropped by 7.44%, from 97.86% to 90.42%, and on DNN and CNN by 6.46% and 7.07%). The difference in testing accuracies achieved by the DNN, CNN, and AlexNet between the personalized and non-personalized partition is vast, 27.38%, 22.6%, and 23.83% respectively, making us reject our hypothesis. This way of training and testing the model only amplifies the significance of phoneme consistency when recording for a beatbox transcription system. The confusion matrices (A.3) are a bit more ambiguous here. For personalized recordings, the closed hi-hat was often misclassified as an open hi-hat, and the snare drum had a substantial number of misclassifications as the other three sound classes. For non-personalized recordings, a lot of misclassifications arose between the closed and open hi-hat utterances (likely because both phonemes “ti” and “ta” used for the imitations both start with a “t”), as well as closed hi-hat utterances being misclassified as the snare drum (likely due to the similar tightness of sound). Kick drum utterances are consistently classified with high accuracy as in other experiments, only sometimes being misclassified as snare drums.

4.4. Experiment #4

In the fourth experiment, we wanted to build on the third experiment and reintroduce a third of the improvisation recordings into the training set, leaving the other two thirds evenly split between validation and testing sets. The intent was to see how much the classification accuracy of the models would increase from the results given in Table 4 towards the results given in Table 2. The results of this experiment are given below in Table 5.

Model	Personalized		Non-personalized		Mixed	
	FTA(%)	HVA(%)	FTA(%)	HVA(%)	FTA(%)	HVA(%)
DNN	72.10	75.09	95.65	91.48	80.82	80.44
CNN	77.17	76.00	93.38	93.75	83.78	83.69
AlexNet	**83.70**	**84.73**	**96.22**	**94.32**	**85.91**	**85.82**

Table 5: Testing and highest validation accuracy scores obtained for the deep learning models in the fourth experiment. Best of scores for each dataset partition are in bold.

The introduction of a portion of improvisation instances substantially increased the achieved accuracies compared to Table 4, as we hypothesized. The magnitudes of the increase are given in Table 6 and shown against the total drop in accuracy from the first to the third experiment. The goal is to see how many of the percentage points of that drop were "gained back" by reintroducing a third of the improvisation utterances into the training set.

Model	Personalized		Non-personalized		Mixed	
	FTA(%)	HVA(%)	FTA(%)	HVA(%)	FTA(%)	HVA(%)
DNN	11.08/24.29	13.59/24.49	7.25/6.46	2.21/7.75	7.94/14.62	5.59/15.03
CNN	9.98/23.92	9.66/25.83	3.59/7.07	3.47/7.69	7.25/15.54	5.57/14.82
AlexNet	17.11/23.98	17.05/24.63	5.8/7.44	3.54/7.33	9.94/17.59	9.06/16.72

Table 6: Percentage point difference between the results in Table 5 and Table 4 out of the total percentage point difference between the results in Table 2 and Table 4.

Generally, the models' accuracies varied the least on non-personalized recordings. The magnitude of accuracy improvement was generally the highest for personalized recordings, then for mixed recordings, then for non-personalized recordings, across the board, although the accuracies did not increase nearly to the levels in Table 2 as we hypothesized. For the trial using personalized recordings, AlexNet saw the largest (by absolute and relative magnitude) increase in accuracy with the introduction of improvisation utterances into the training set. The confusion matrices (A.4) reveal patterns similar to those in experiment three, although the magnitude of misclassifications is lower, since the training set looks a bit more like the testing set in this experiment.

4.5. Experiment #5

In the fifth experiment, we wanted to explore how well the models could classify human beatbox utterances produced by participants excluded from the training set. In other words, we explore the robustness of the models to classify previously unseen voices. The models were trained on the utterances from the first 24 participants and tested on utterances from the last 4 participants (2 male and 2 female). The results of this experiment are given below in Table 7.

Model	Personalized		Non-personalized		Mixed	
	FTA(%)	HVA(%)	FTA(%)	HVA(%)	FTA(%)	HVA(%)
DNN	**51.95**	54.15	85.89	84.39	63.46	63.66
CNN	49.27	49.51	88.32	88.05	**64.92**	68.54
AlexNet	48.05	**60.24**	**88.81**	**89.51**	64.80	**70.61**

Table 7: Testing and highest validation accuracy scores obtained for the deep learning models in the fifth experiment. Best of scores for each dataset partition are in bold.

Unexpectedly, the DNN outperformed the CNN on personalized recordings in this experiment. More generally, the results on the personalized recordings were odd and subpar. Training the models became very difficult to avoid overfitting, and all models continued to only achieve around 50% final testing accuracy on repeated runs. Across all partitions, the final testing accuracies were very similar between the CNN and AlexNet. The models' accuracies on non-personalized recordings were still very high (∼85-90%). Given that the training set only contained 24 voices, this is good news: as long as unseen participants use the same phonemes for their drum imitations, the models' performance remains high. This means that the combination of phoneme inconsistency and introduction of new voices leads to a disastrous drop in accuracy for our models. The confusion matrices (A.5) reveal that in the personalized recordings, the majority of closed hi-hat utterances were misclassified as either a kick drum or snare drum, and the majority of open hi-hat utterances were misclassified as snare drums. For non-personalized recordings, we see a familiar pattern of closed and open hi-hats being misclassified as one another. The model performed much better when working with mixed rather than personalized recordings, but still had general difficulty differentiating between the non-kick drum sounds.

4.6. General Observations

In all experiments, the models achieved the highest accuracies on non-personalized recordings. The lowest accuracies were on the personalized recordings, with the accuracies on mixed recordings being the middle ground. We attribute this, as hypothesized, to the fact that personalized recordings were inconsistent across participants in the phonemes used for drum imitation. For the purposes of beatbox transcription software, if the software trains a distributed online model based on the users' recordings, it would be best to provide the users with a sample that they would replicate, such that the imitations are more phonetically consistent across users. Testing accuracy of $\sim$90% was achieved by all three models on the non-personalized recordings in the third experiment, indicating good potential for this approach. Alternatively, the piece of software could ask the users to provide multiple utterances of each class of drum imitations before the software attempts to transcribe their improvisations.

Between the three models, AlexNet performed the best across most experiments, although the difference in performance between AlexNet and CNN was marginal in most cases. Since AlexNet takes a much longer time to train than the DNN or CNN, its use may be more justified for something like the fourth experiment, as AlexNet performed substantially better (83.70%) on the personalized recordings than the DNN (72.10%) and CNN (77.17%). Otherwise, the hours' difference in training time between AlexNet and CNN would not always be justified if the payoff is only one or two percentage points in accuracy.

Generally, the kick drum sound, out of all sound classes, was the most accurately classified across all experiments. The models sometimes misclassified the open and closed hi-hats for one another in non-personalized recordings (likely due to both starting with the consonant sound "t"), and had general trouble differentiating non-kick drum utterances in personalized recordings (likely due to phoneme inconsistencies between participants).

5. Suggestions for Future Research

This paper primarily focused on deep learning models, and based on the findings of Delgado et al. [1], a CNN achieves higher accuracies than traditional classification models on both the original and augmented datasets. Further exploration of maximizing classification accuracy on an augmented dataset could be useful to come up with the best CNN architecture for the task of classifying improvisation utterances after training on non-improvisation ones.

Based on the results of the fifth experiment, our models do not generalize well to unseen voices, especially in classifying utterances extracted from the personalized recordings. It is reasonable to assume that some audio features may be more robust in classifying new voices than others, so a different study can be done to select a feature set that provides that robustness. Likewise, it would be useful to assess how much data augmentation (pitching, bending, and editing the utterances in the training set) can help increase classification accuracy on unseen voices as well.

This dataset is also a perfect opportunity to replicate our third experiment but on a participant-by-participant basis. In each section of the dataset, every participant has an improvisation recording and four non-improvisation recordings, each with around 25 utterances of a specific class of drum sound imitation. Training the models on only a single participant's non-improvisation utterances and then testing the efficacy of transcribing their improvisation is a good way to replicate one of the commonly proposed beatbox transcription systems, where the software requests the user to provide examples of each sound before the users can record for transcription purposes.

Beyond this dataset, future research in Vocal Percussion Transcription (VPT) should involve introducing more classes of sounds for classifications, transcribing beatbox tracks that contain both vocal percussion and vocalized sound (e.g. humming), and transcribing more professional beatbox tracks with fast sound patterns.

6. Conclusion

In this paper, we extracted MFCCs from short clips of human beatbox utterances to train a fully-connected neural network, a convolutional neural network, and AlexNet to classify between human imitations of the kick drum, snare drum, open hi-hat, and closed hi-hat. We performed five separate experiments on personalized, non-personalized, and mixed recordings from the AVP dataset to test hypotheses enabled by the dataset or raised by previous research. While AlexNet was often the best-performing model across all experiments, the training time was often too long to justify the performance increase over what was achieved by our simple CNN. Best model selection thus would ultimately be dependent on the context of where beatbox transcription is required.

In this paper, we found that (1) phoneme-consistency in non-personalized recordings ensures consistently high classification accuracy across all experiments, with models achieving lower accuracies on mixed recordings, and lowest on personalized recordings; (2) using first and second MFCC deltas in addition to MFCCs produced a noticeable performance increase for the DNN but trivial for the CNN and AlexNet; (3) using improvisation utterances only for testing yielded poor performance on all models when dealing with personalized recordings, (4) indicating that there is a substantial difference in same-class utterances in improvisation and non-improvisation recordings when the users are free to use their own interpretation(s) of the sounds, inconsistent with other participants' interpretations; (5) the current models and MFCCs are not robust to testing on unseen voices, dramatically so when dealing with personalized recordings.

These findings have important implications for future research in VPT and provide a good basis for further exploration and optimization of presented and suggested methods.

7. Acknowledgements

I would like to extend sincere gratitude to my advisor Professor Adam Finkelstein from Princeton University and Dr. Zeyu Jin for their advice. I would also like to thank Pranay Manocha for taking time to aid me in data processing and experimental design, and for reviewing the draft of this paper.

References

[1] A. Delgado, S. McDonald, N. Xu, C. Saitis, and M. B. Sandler, "Learning models for query by vocal percussion: A comparative study," *CoRR*, vol. abs/2110.09223, 2021. [Online]. Available: https://arxiv.org/abs/2110.09223

[2] A. Delgado, S. 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*, Sep 2019. [Online]. Available: http://dx.doi.org/10.1145/3356590.3356844

[3] C. Dittmar and D. Gärtner, "Real-time transcription and separation of drum recordings based on nmf decomposition," in *DAFx*, 2014.

[4] S. Evain, B. Lecouteux, D. Schwab, A. Contesse, A. Pinchaud, and N. Henrich Bernardoni, "Human beatbox sound recognition using an automatic speech recognition toolkit," *Biomedical Signal Processing and Control*, vol. 67, p. 102468, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1746809421000653

[5] N. Gajhede, O. Beck, and H. Purwins, "Convolutional neural networks with batch normalization for classifying hi-hat, snare, and bass percussion sound samples," in *Proceedings of the Audio Mostly 2016*, ser. AM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 111–115. [Online]. Available: https://doi.org/10.1145/2986416.2986453

[6] A. Hazan, "Towards automatic transcription of expressive oral percussive performances," in *Proceedings of the 10th International Conference on Intelligent User Interfaces, IUI 2005, San Diego, California, USA, January 10-13, 2005*, R. S. Amant, J. Riedl, and A. Jameson, Eds. ACM, 2005, pp. 296–298. [Online]. Available: https://doi.org/10.1145/1040830.1040904

[7] D. Mukhutdinov, "Deep feature extraction and music language modelling for amateur vocal percussion transcription," Ph.D. dissertation, 2020.

[8] B. Picart, S. Brognaux, and S. Dupont, "Analysis and automatic recognition of human beatbox sounds: A comparative study," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4255–4259.

[9] A. Ramanathan, "Beatboxing to drums using support vector machines," 2017.

[10] A. Ramires, R. Penha, and M. E. P. Davies, "User specific adaptation in automatic transcription of vocalised percussion," *CoRR*, vol. abs/1811.02406, 2018. [Online]. Available: http://arxiv.org/abs/1811.02406

[11] A. F. S. Ramires, "Automatic transcription of drums and vocalised percussion," Ph.D. dissertation, 2017.

[12] E. Sinyor, C. McKay, R. Fiebrink, D. McEnnis, and I. Fujinaga, "Beatbox classification using ace." 01 2005, pp. 672–675.

[13] C. Southall, R. Stables, and J. Hockman, "Automatic drum transcription using bi-directional recurrent neural networks," in *ISMIR*, 2016.

[14] G. Tzanetakis, A. Kapur, and M. S. Benning, "Query-by-beat-boxing: Music retrieval for the dj," in *ISMIR*, 2004.

[15] R. Vogl, M. Dorfer, and P. Knees, "Recurrent neural networks for drum transcription," in *ISMIR*, 2016.

[16] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller, and A. Lerch, "A review of automatic drum transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 9, pp. 1457–1483, 2018.

A. Appendix: Confusion Matrices

A.1. Experiment #1

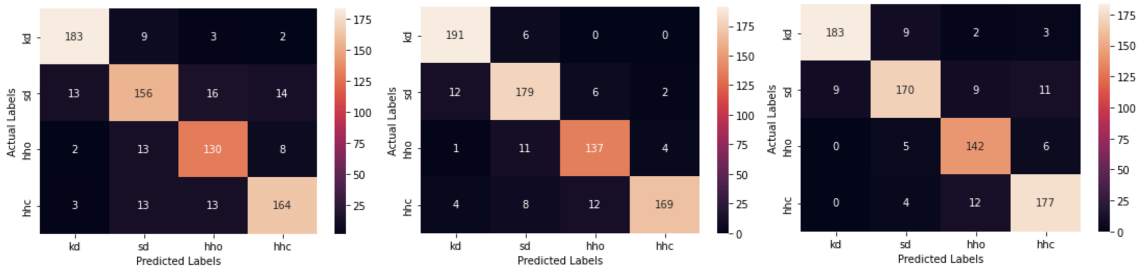


Figure 8: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **personalized** recordings.

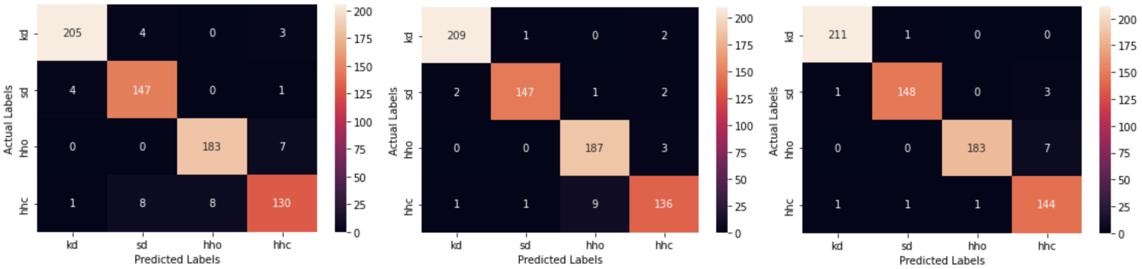


Figure 9: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **non-personalized** recordings.

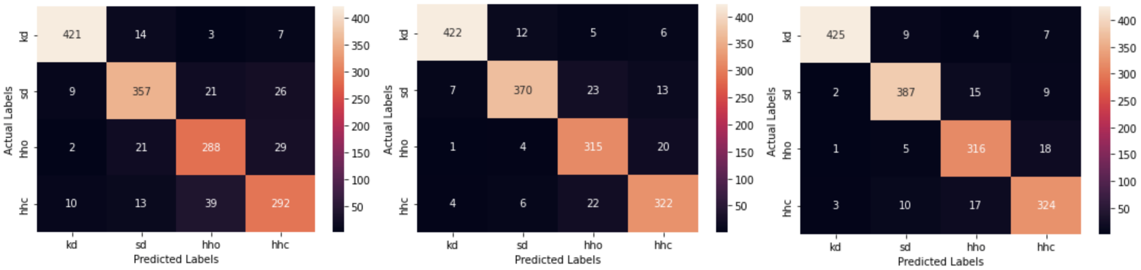


Figure 10: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **mixed** recordings.

A.2. Experiment #2

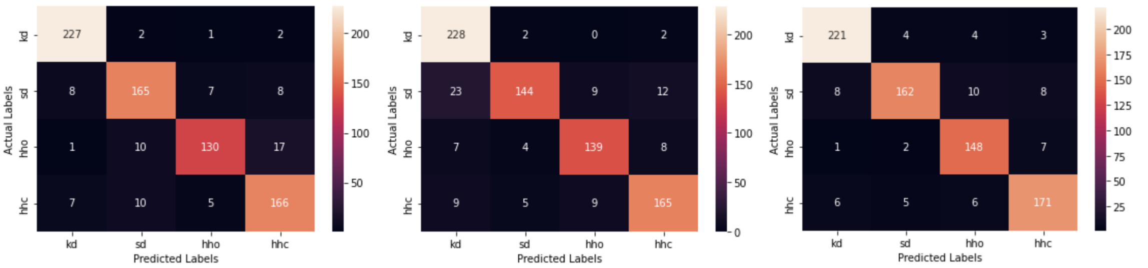


Figure 11: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **personalized** recordings.

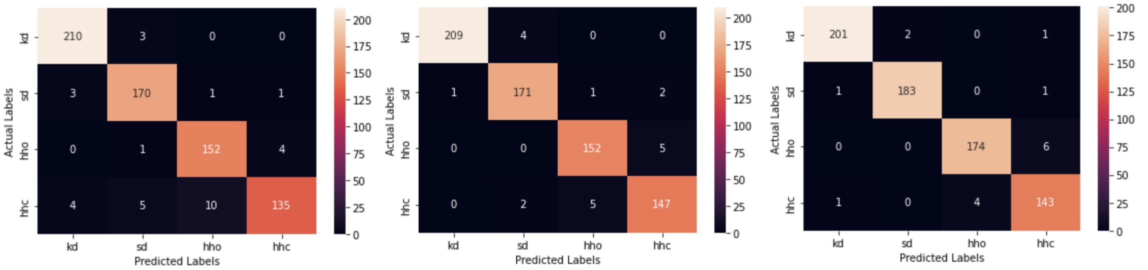


Figure 12: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **non-personalized** recordings.

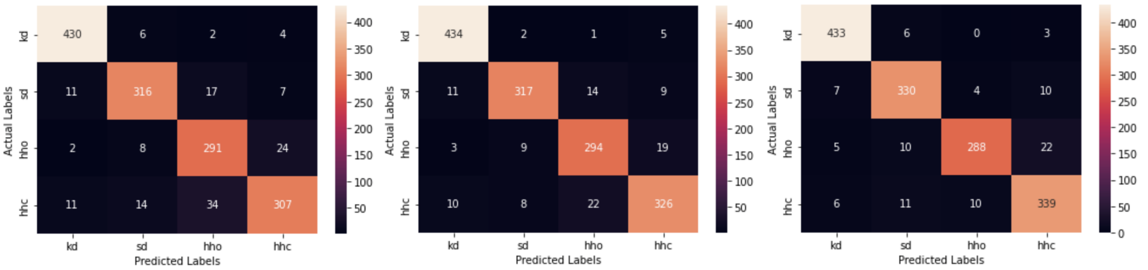


Figure 13: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **mixed** recordings.

A.3. Experiment #3

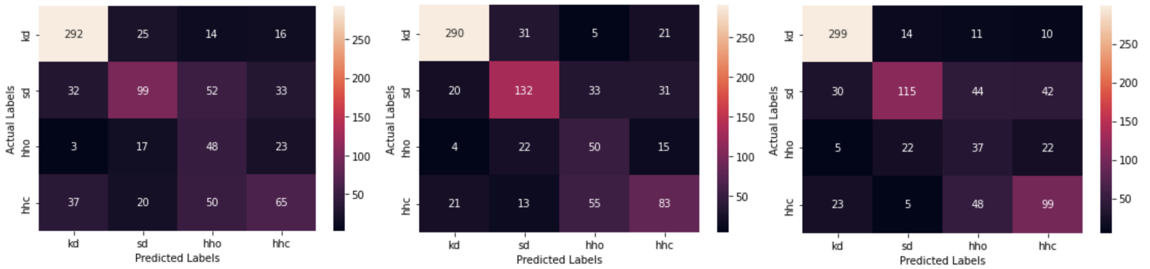


Figure 14: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **personalized** recordings.

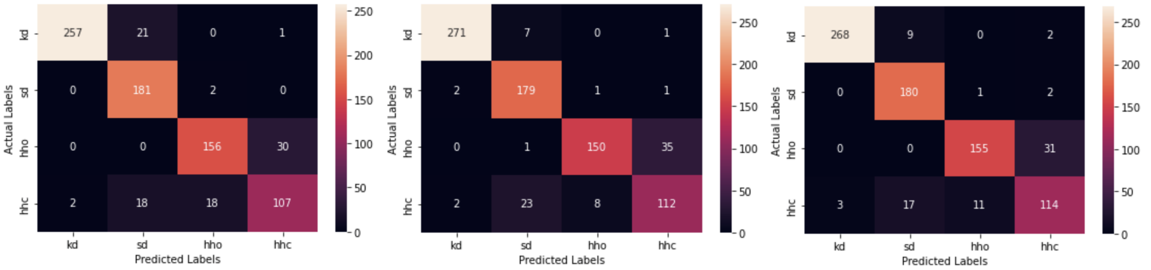


Figure 15: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **non-personalized** recordings.

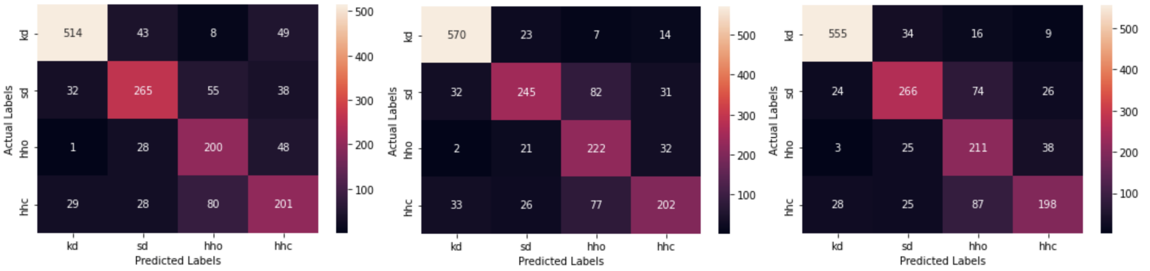


Figure 16: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **mixed** recordings.

A.4. Experiment #4

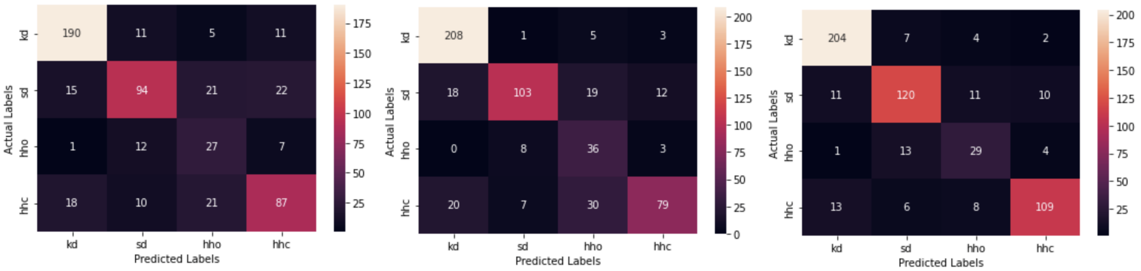


Figure 17: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **personalized** recordings.

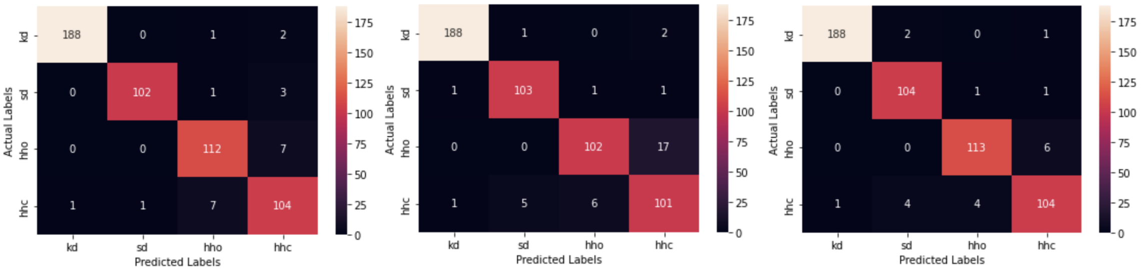


Figure 18: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **non-personalized** recordings.

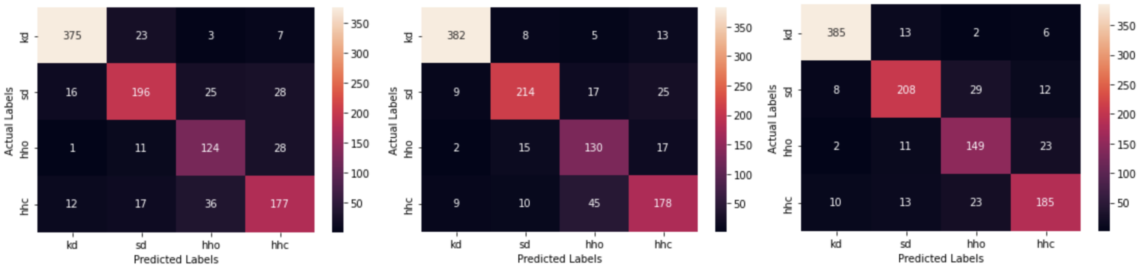


Figure 19: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **mixed** recordings.

A.5. Experiment #5

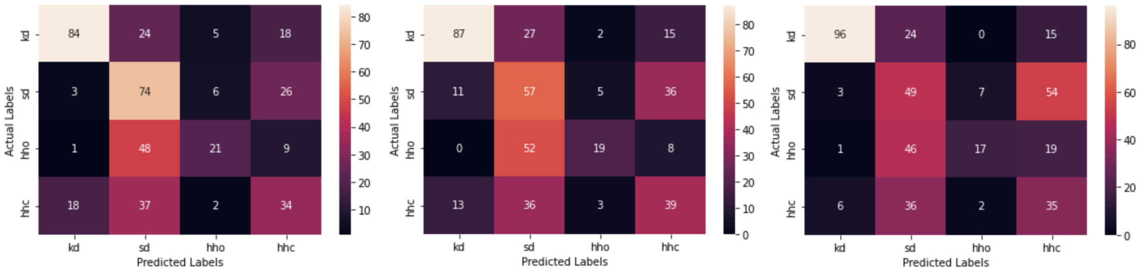


Figure 20: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **personalized** recordings.

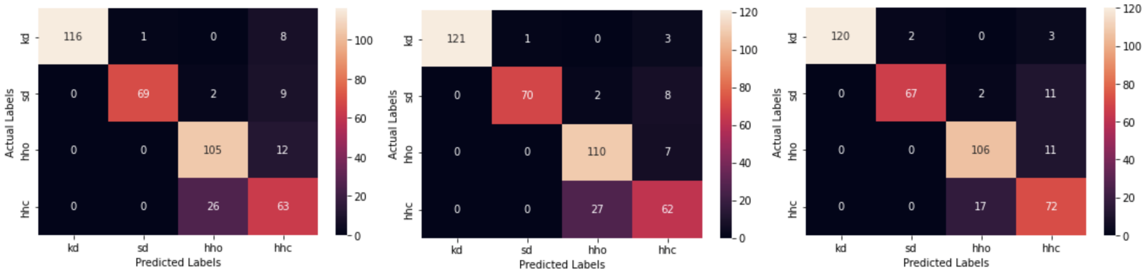


Figure 21: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **non-personalized** recordings.

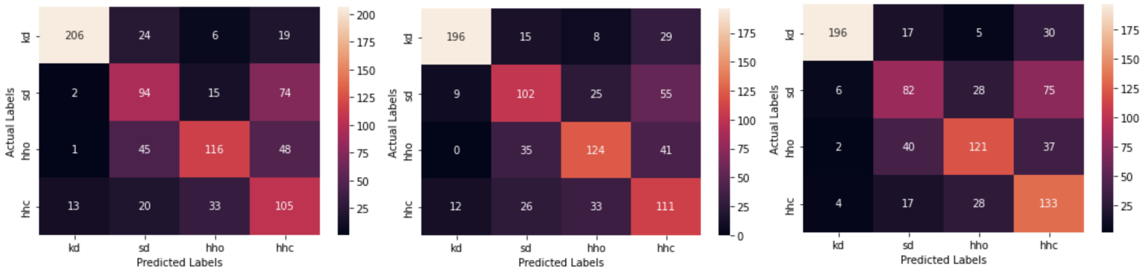


Figure 22: Confusion matrices generated from the testing set evaluation of the DNN (left), CNN (middle), and AlexNet (right) on the **mixed** recordings.