Auditory Grouping: Using Machine Learning to Predict Locations of Groups in Music Clips

Yang Chen, Cheng-Hsuan (Sean) Jan, William McDonald

March 7th, 2023

A Major Qualifying Project submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the degree of Bachelor of Science

Authors:

Yang Chen

Cheng-Hsuan (Sean) Jan

William McDonald

Date:

May 7th, 2023

Report Submitted to:

Professor Scott Barton

Professor Gillian Smith

Worcester Polytechnic Institute

Acknowledgments

We would like to thank our project advisors Scott Barton, James Doyle, and Gillian Smith for giving us the opportunity to work on this project. We would also like to thank our other partners: Zachary Wagner for helping us throughout the project, setting up the experiment for the project, and writing part of the paper and William McDonald, who helped with the logistics of the experiment, management of data, and also helped us write part of the paper.

Abstract

Humans perceive a variety of features from an auditory stream, such as our acoustic sensors can detect frequency, pitch, dynamics, etc. We can process music in several different ways based on these features. It’s tough for machines, however, to do the same. Some previous research models already can obtain state-of-the-art performance in predicting acoustic boundaries, but machine perception for audio segmentation based on a human perspective remains to be accomplished. Our project aims to use machine learning algorithms to build a model that makes machines able to separate music into segments as humans do. The machine learning model we built allowed for clear grouping distinction for audio clips of the same musical genre we trained the data on, but generalized poorly to other genres. We believe that the model can be improved by having more training data of a larger scope and increasing the quality of grouping boundaries labels for the data.

Executive Summary

Humans recognize things in many different ways, it varies by person but has a lot of similarities in general. Machine perception seeks to develop techniques to let machines similarly predict acoustic boundaries as humans. The machine should be able to group continuous music into different chunks, represented by a start time and duration in the audio. The research aimed to hybrid human and machine perceptions into a model that can tell the differences and similarities between the two.

To collect human perception auditory grouping data, we built a survey and published it on both Qualtrics(for SONA Systems) and Amazon Mturk. We aimed to use the collected information and feed that into the machine learning models to train them to recognize the same grouping cues. The survey also includes demographic and other sorting questions to let us do some traditional data analysis.

To address some problems from our input data like inconsistent output data points, for example, one audio clip could have 2 groups, while another could have 5 groups. We choose the models below to fix that problem.

- Convolutional Neural Network
- Recurrent Neural Network
- Convolutional Recurrent Neural Networks
- VIT(Vision Transformer)

Because the large size of the raw data makes the training slow, we transformed the data into spectrogram images including Mel-spectrogram, a chromogram, a Mel-frequency cepstrum (MFCC), and a tempogram. This changes the original 1-D convolutional input into a 2-D convolutional input which makes the training much faster. The other method that would increase our model performance or accuracy is listed below.

- Supervised Pre-Training
- Google cloud/TPU's
- Fuzziness
- Left/Right aligning

For our result models, the convolutional neural network models had clear boundaries for each group. We tried the methods mentioned above to improve this model's performance. Some failure models like RNN, CNNs & RNNs, and VIT had the common problem that the model cannot distinguish the groups. This may be caused by the vanishing gradient problem that RNN has. In the final model we added more convolutional layers to it and finished with the dense layer which gave us some ground truth labels for the same genres of the music clips, but generalized poorly to other genres.

Table of Contents

Introduction

The field of machine perception seeks to develop techniques for machines to perceive and interpret stimuli in ways similar to humans. In the field of machine perception, auditory perception refers to the set of tendencies and mechanisms related to how humans perceive sound, like music. A key problem in auditory perception is auditory grouping which is the way human perception breaks a continuous stream of perceived sound into chunks, or groups[1]. In the context of machine perception, the auditory grouping problem requires taking in an arbitrary piece of audio, specifically music, and predicting the groups, represented by a start time and a duration, present in that audio. Additionally, grouping grammars can specify categories of auditory groups, and models over those grammars can predict the classifications of identified groups[1].

Human beings categorize information in several different ways, contingent upon the type of stimulus experienced and the nuances detected within. Applied to the perception of sound, especially in the context of music, this phenomenon is described as auditory grouping. Because sounds are not just one constant, unchanging waveform, our brains can categorize and encode audio information differently, as a result of various components of music. With how humans encode this information being highly complicated and not understood in its entirety, unique challenges are presented when attempting to create machines that can perform the same grouping tasks. While this body of research is valuable for gathering information about our experiment, we as researchers intend to hybridize the human and machine components of auditory grouping into a comprehensive model.

Motivation

The potential applications of an accurate auditory grouping model are numerous. A model which can break a piece of music into perceived chunks could be used by musicians to better compose music by supporting algorithmic tools which operate on perceived phrases of notes as a unit. Other applications include big-data analysis of large auditory repositories, where individual reviews of pieces are out of scope. A music streaming service, for example, might prefer to shard their audio streams along group boundaries so temporal anomalies due to low-bandwidth stream latency occur during natural grouping breaks. Another application could be to support a new class of algorithms that operates on auditory data with an understanding of human perception of that data; a compression algorithm could preserve information about groups as individual units, and deprioritize the noise between groups. It also has applications in robotics. For a musical robot to be able to interact with a musician, it needs to be able to interpret the expressions of the musician. Part of that interpretation is auditory grouping.

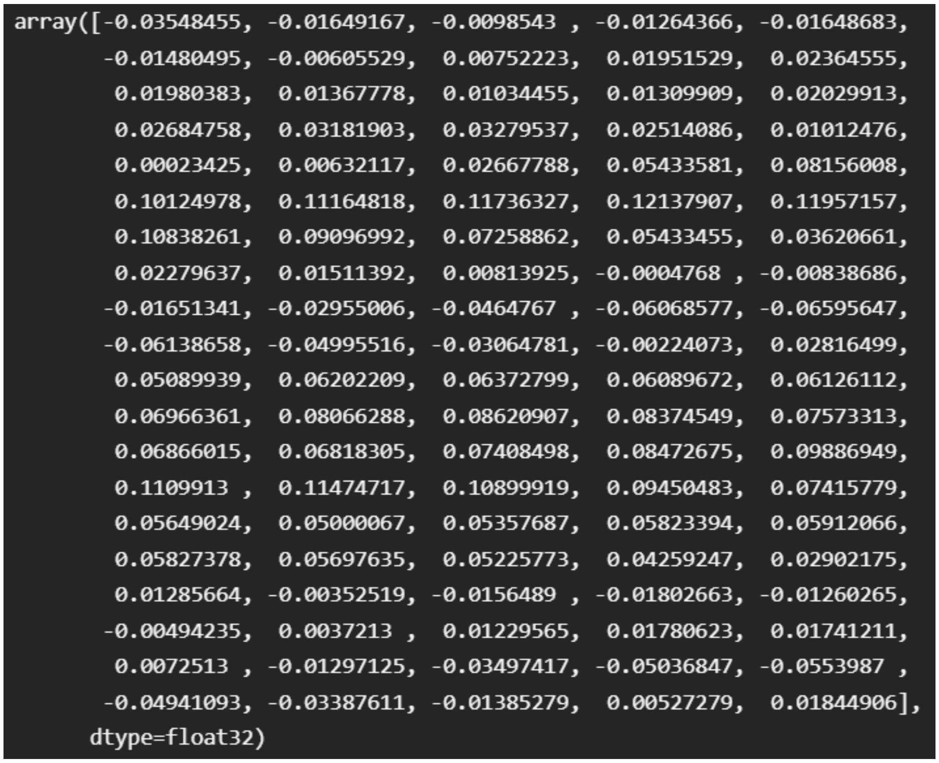
Background

There is existing research on the auditory grouping problem in machine perception [2], [3]. Existing techniques primarily focus on trying to construct a model of human perception of auditory grouping, either prescribing to Gestalt [4] theory, where we group similar entities like sounds, or intentionally in refutation of it, and then constructing a machine to interpret supplied waveforms towards the specification of that model [5]. Most of these models are unsupervised models, which is to say they do not test themselves against human-labeled data [2]. They instead analyze waveforms to predict gaps, melody changes, and other musical features. Using unsupervised models for this task may not be the best approach for this task as, without human input, the model would have to try and replicate human behavior through algorithms alone.

Our approach to this problem involves using artificial neural networks to learn the human behavior of auditory grouping. Neural networks have become an industry standard in machine learning due to their powerful ability to extract features without preprocessing. Various kinds of machine learning methods such as convolutional neural networks and recurrent neural networks have been used in audio machine learning experiments[6].

Convolutional Neural Networks

Convolutional neural networks excel at extracting spatial information from data [7] which can be powerful for analyzing audio clips. Audio clips are stored as a sequence of samples, which can be represented as a 1-D long array. While using a 1-D convolution on the raw audio would work as demonstrated later on, transforming the data into an image spectrogram, which is also demonstrated later on would allow us to better utilize the power of convolutions.



```
array([-0.03548455, -0.01649167, -0.0098543 , -0.01264366, -0.01648683,
       -0.01480495, -0.00605529,  0.00752223,  0.01951529,  0.02364555,
        0.01980383,  0.01367778,  0.01034455,  0.01309909,  0.02029913,
        0.02684758,  0.03181903,  0.03279537,  0.02514086,  0.01012476,
        0.00023425,  0.00632117,  0.02667788,  0.05433581,  0.08156008,
        0.10124978,  0.11164818,  0.11736327,  0.12137907,  0.11957157,
        0.10838261,  0.09096992,  0.07258862,  0.05433455,  0.03620661,
        0.02279637,  0.01511392,  0.00813925, -0.0004768 , -0.00838686,
       -0.01651341, -0.02955006, -0.0464767 , -0.06068577, -0.06595647,
       -0.06138658, -0.04995516, -0.03064781, -0.00224073,  0.02816499,
        0.05089939,  0.06202209,  0.06372799,  0.06089672,  0.06126112,
        0.06966361,  0.08066288,  0.08620907,  0.08374549,  0.07573313,
        0.06866015,  0.06818305,  0.07408498,  0.08472675,  0.09886949,
        0.1109913 ,  0.11474717,  0.10899919,  0.09450483,  0.07415779,
        0.05649024,  0.05000067,  0.05357687,  0.05823394,  0.05912066,
        0.05827378,  0.05697635,  0.05225773,  0.04259247,  0.02902175,
        0.01285664, -0.00352519, -0.0156489 , -0.01802663, -0.01260265,
       -0.00494235,  0.0037213 ,  0.01229565,  0.01780623,  0.01741211,
        0.0072513 , -0.01297125, -0.03497417, -0.05036847, -0.0553987 ,
       -0.04941093, -0.03387611, -0.01385279,  0.00527279,  0.01844906],
      dtype=float32)
```

Figure 1: Approximately 4 milliseconds of audio represented as 100 samples in an array

In addition to convolutions, there are many other layers used commonly in convolutional neural networks that can be used to further improve performance such as max-pooling layers, dropout layers, batch normalization layers, etc. Max pooling layers reduce the dimensionality of an output layer by taking a section of numbers and reducing it down to 1 number by taking the max value which can help reduce noise for the outputs. Dropout layers randomly cancel out neurons to reduce reliance on specific neurons during training which can help with generalization. Batch Normalization normalizes the data between inputs which can help speed up training.

Recurrent Neural Networks

Recurrent neural networks are neural networks that process inputs sequentially allowing them to exhibit temporal behavior. Due to its ability to analyze inputs sequentially, it has been used for various auditory machine learning models[6]. Vanilla recurrent neural networks have mostly faded out of usage due to their issues with training such as exploding or vanishing gradient, but other recurrent layers overcome this issue such as LSTMs[8]. LSTMs come with a feature called a cell state that carries information throughout inference creating some kind of residual connection throughout the LSTM.

Another variant of RNNs is the Convolutional LSTMS [9]. Convolutional LSTMs analyze chunks of data sequentially by running a convolution through chunks and using that as part of the hidden states. This RNN layer could prove especially powerful for spectrograms where we can analyze slices of spectrograms at a time.

Convolutional Recurrent Neural Networks

Convolutional Recurrent Neural Networks utilize both convolutional layers and recurrent layers in the neural network. The data would first pass through the convolutional layers and then the recurrent layers. This architecture allows us to utilize the power of convolutions and recurrency in our models which may prove powerful for auditory-related machine learning tasks[10].

Residual Neural Networks/Skip connections

Modern neural networks use hundreds of layers in their models, allowing earlier layers to better extract more complicated and useful features. Training many layers in a neural network result in a vanishing gradient, which is when the weights of the model become too small to effectively learn. Residual networks or skip connections mitigate this problem by allowing information to skip potentially faulty layers[11]. While our model won't nearly have as many layers as state-of-the-art models, skip connections will allow us to increase model depth without compromising performance.

Vision Transformers (VIT)

Vision Transformers (VIT) use a transformer network, which historically is used for natural language processing tasks[12], to process images[13]. Vision Transformer models have seen success in many image recognition benchmarks and are the current state-of-the-art model for image recognition. While vision transformers are a fairly new technology in the machine learning community, vision transformers have been used for auditory machine learning[14].

Opportunity

We believe that for a task so fundamentally oriented towards subjective human perception, training (and then validating) a model on human-generated data is critical. This validation against real perceptual actors has been beneficial in verifying other domains of auditory perception, like the Iambic-Trochaic law [15], [16]. Thus, we offer our specific contribution to the auditory grouping problem: applying supervised, contemporary, deep-learning convolutional models to human-produced auditory grouping data.

Methodology and Design

There are two components of the design of our project, the experimental design for data collection, and the exploration space for the machine learning modeling work. Because of the relatively unknown domain of supervised auditory grouping, there doesn't exist an easily accessible dataset online containing human grouping data. Thankfully our advisors had a dataset of such grouping data that we used for training. The data however only contained 1718 samples, which we theorized was not enough for training, hence the need for designing an experiment for data collection.

Experimental Design

For our study, aimed to collect data from participants performing auditory grouping tasks, to then feed this information directly to our algorithm to train them to recognize the same grouping cues.

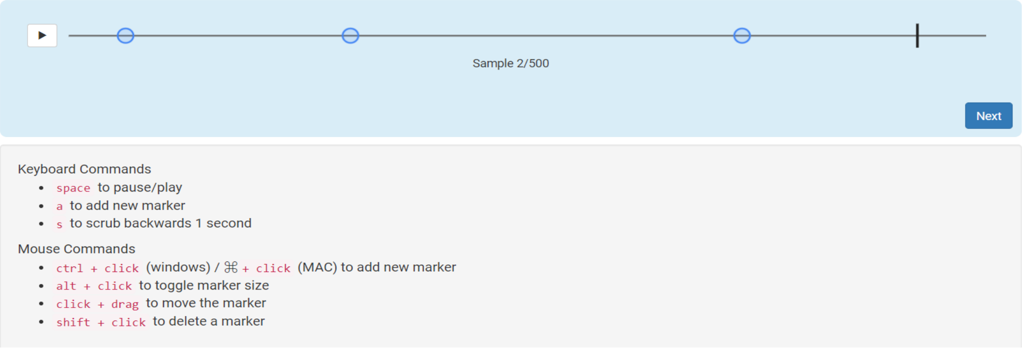
Materials and Methods

To collect data, we have decided upon a survey format conducted through SONA Systems and Amazon MTurk, with a financial incentive. Our materials included 20 audio clips we have decided upon taken from music spanning the baroque, classical, and romantic periods. The independent variable here is the auditory clip stimulus given to a participant, as well as any specific meta-data aspects of the sound bite. Our dependent variable is the way in which the clip has been grouped, first without any consideration for if it was “correct” or not.

Experimental Design

The experimental design of this study is a hybridization of data collection, to serve a primary and secondary goal. The primary goal, consisting of the bulk of the experiment, will seek to gain consistency, repetition, and quantity of data from each individual participant to be fed to a machine-learning algorithm. The secondary goal, consisting of a preliminary section of the survey focusing on demographic and other sorting questions, will seek to provide a basis by which traditional data analysis can be performed for different groups.

The bulk of the survey designed to collect masses of data for the algorithm will work very similarly to those facilitated by CogLab. Participants will be given 100, 10-second audio clips and be asked to click when they think they recognize a specific grouping. This will be visualized by a slider on a flat audio form that DOES NOT reflect the clip the participant is hearing. These experiment pages can be set to automatically move on to the next audio clip once the 10 seconds are through. This will result in a total survey time of approximately 16.6 minutes, not including the time spent on the preliminary questions page, to be used for the secondary goal's traditional analysis.



Keyboard Commands

- `space` to pause/play
- `a` to add new marker
- `s` to scrub backwards 1 second

Mouse Commands

- `ctrl + click` (windows) / ⌘ `+ click` (MAC) to add new marker
- `alt + click` to toggle marker size
- `click + drag` to move the marker
- `shift + click` to delete a marker

Figure 2: The experiment interface used to collect data

The secondary goal of the research will be achieved by adding a section of demographic questions to the end of the survey. The questions will include gender, age, music experience, music affiliation, and other such relevant factors for later sorting data. These questions will serve as a basis for doing manual data analysis on the grouping data. In this way, the survey accomplishes its primary goal of a large quantity of data for training the algorithm, while also yielding a great deal of raw information to be later grouped and analyzed in accordance with the answers in the demographic section. The survey is listed in appendix A.

Model Design

Output structuring

Due to each audio clip having a variable amount of groups for each corresponding audio clip, figuring out how to structure our output such that our model can effectively output data was a challenge. One audio clip could have 2 groups, while another could have 5 groups. This meant traditional supervised learning strategies like classification or regression won't work, as the number of outputs varies.

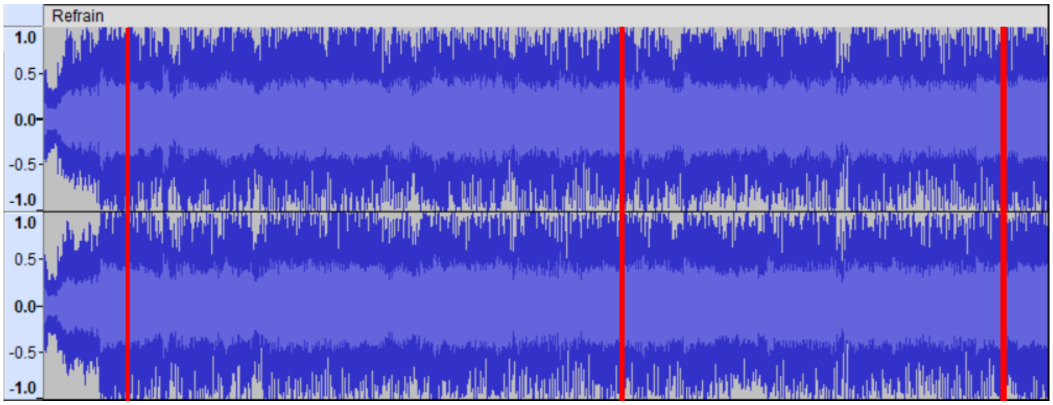


Figure 3: Waveform with potential groups marked in red

While a few strategies to overcome this problem were proposed, like using a recurrent layer as our output layer, we settled on encoding groupings on an array of fixed lengths. Each element in our output represents a time within the audio clip. For example, if the audio clip is 5 seconds long, and we had an output of length 50, the first index in the array would represent the time 0-0.1 seconds, the second index would represent the time 0.1-0.2 seconds, and so on. When a group happened at a certain time, a 1 would be put in the corresponding index. By encoding the groups as 1's on an array of fixed lengths, we can treat this problem as a multi-class, multi-label classification problem.

Budget

The two components of cost for our project are data collection and model training. For data collection, we reached out to groups on campus to reduce monetary costs. For data collection, the costs for mTurk (or comparable tools like prolific) totaled approximately $500 at a rate of $3/participants and a target of 150 participants. For training our machine learning model, we used Google Colab's machine learning infrastructure and Kaggle's machine learning infrastructure which allows us to use powerful hardware for free.

Methods

Spectrogram transformation

MP3 files encode audio in samples, where each sample represents an instance in time. The sample rate of a clip loaded in Librosa, the python library we used to parse data, is 22050. The clips we use to analyze groupings are exactly ten seconds long, meaning the data has a total of 220500 samples. Going through the samples using a dense neural network, a recurrent layer, or even a 1d-convolution would take a long amount of inference time. Transforming the data into a spectrogram, an image representation of a clip, would allow us to decrease inference time and also allow us to use more powerful layers like a 2d-convolution.

The spectrograms we used were a Mel spectrogram, a chromogram, a Mel-frequency cepstrum (MFCC), and a tempogram. All of the spectrograms were created with 128 bins, and transformed from power to decibels. Creating each of the spectrograms resulted in a 431 by 128 image. The images were then layered on top of each other to create a 431 x 128 x 4 array

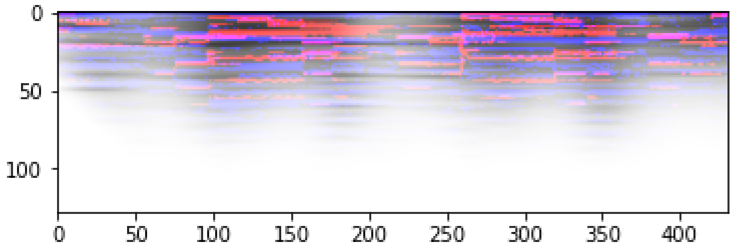


Figure 4: Spectrogram image used as training data

Supervised Pre Training

Due to the small size and limited scope of our dataset, we opted to try supervised pretraining on a larger dataset. The dataset we chose is a small section of the free music archive (FMA)[17]. The section we chose for the FMA contains 8000 music clips from 8 genres. The music clips are around 30 seconds long and each audio clip was split into 3 clips of 10 seconds. A model was trained to classify these music pieces into their various musical genres. This task was chosen because the field of musical genre detection is well established and easy to find examples for. After training was done, the fully connected layers were removed from the model, and a new set of fully-connected layers were appended to resume training on the auditory grouping problem.

Google cloud/TPUs

The datasets for supervised pretraining and auditory groupings were too large to fit in RAM. This meant that data had to be loaded through chunks during training. Normally this would be simple to do on CPU or GPU training, but data requires data to be stored on Google cloud to train on TPUs.

Tensor Processing Units, usually shortened to TPUs, are custom-made circuits meant to quickly do tensor arithmetic. TPUs were designed with deep learning in mind and accelerates training with neural networks.

Because training on TPUs is significantly faster than training on GPUs and CPUs, it was necessary to get the training done in a reasonable time. In addition to having to store our data on the cloud, Google Cloud only allows a certain amount of files to be read from the cloud at a given time. If we store each data instance as a file, our training would be throttled by Google cloud’s reading speed. To overcome these issues, we first encoded the spectrograms as RGBA images, then we stored the images in TFRecord files. TFRecord files allow for multiple instances of training datum to be stored in a single file, allowing the TPU to get multiple instances of training datum with a Google cloud call.

Fuzziness

Instead of having the ground truth label be represented by exactly 1 number, we would have the surrounding values also equal 1. For example, if a label looks like (0,0,0,0,1,0,0,0), after fuzziness, it would look like (0,0,0,1,1,1,0,0). A similar neural network to detect musical onsets used this strategy [18]. The rationale for this strategy is that group beginnings could start slightly later or earlier depending on the participant's reaction time and other factors, so instead of giving a fixed index for a group beginning, we gave a small range.

Left/Right aligning

We arbitrarily chose to analyze audio clips that were 10 seconds long, so we created our model to accept audio clips of 10 seconds. All of our training data however were around 8 seconds long. To overcome this, we first aligned the audio clips with the start of the 10 seconds and padded the end with silence. We then aligned the audio clip with the end of the 10 seconds and padded the beginning with silence. Left/Right aligning gives us the benefit of doubling our training data and also allows us to train without unfairly biasing one end of the audio clips over the other.

Results

Successful Models

Convolutional Neural Networks(Raw Data)

Our first iteration of the models was a simple 1-D convolutional neural network on the raw mp3 files for the data. The raw data was a 1-D array of samples, so 1-D convolutional layers were used to sample data. It consisted of a convolutional layer followed by a dense layer. The result below showed an actual group of three. The model provided a clear group of four, and close to the actual groups. This model had relatively less noise but would take a long time to train as the large size of the 1-D array.

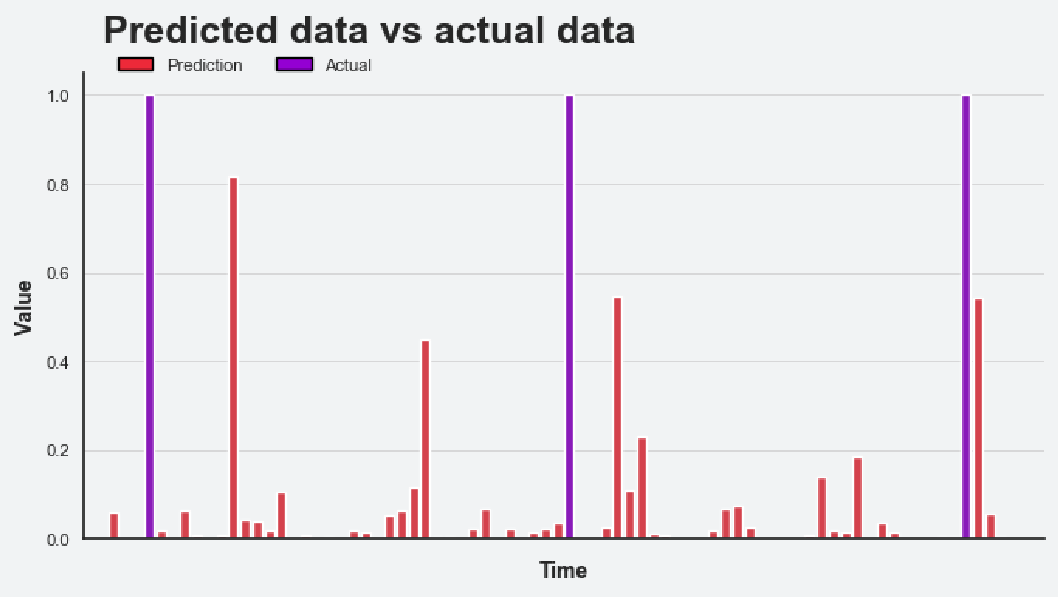


Figure 5: An example output of the CNN(Raw Data) testing model

This graph shows us the ground truth labels of one participant of the beginning of groups in purple, and the predictions of the beginning of groups in red. The X-axis is the time in seconds in which a group beginning happens and the y-axis is the certainty that a group happens at that time. The model predictions can be interpreted as a probability distribution for group beginnings over time. While this approach gave clear group startings for our data, training was slow and there was little we can do to improve model complexity. For these reasons, we tried 2 convolutions.

Convolutional Neural Networks (Spectrogram Image Data)

We used the same structure for convolutional neural networks but changed the input to Mel spectrograms. The same strategies used for image supervised learning were used here and resulted in similar results to the convolutional neural network on raw audio. This method made the training much faster.

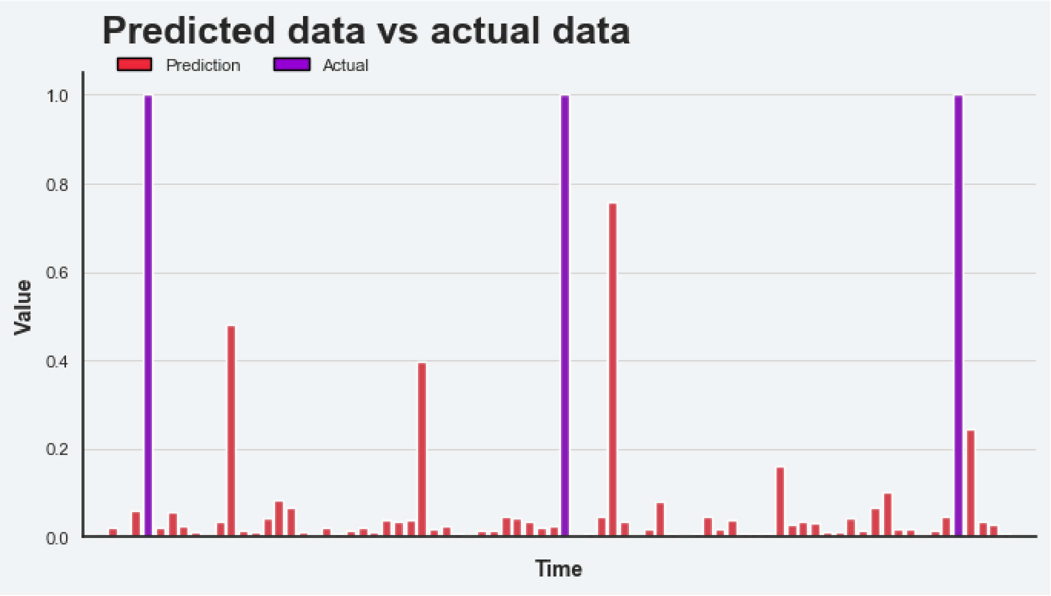


Figure 6: An example output of the CNN(Spectrogram Image Data) testing model

Even though this model had a similar result to the first model, it showed us that the model can work on spectrogram images, allowing us to use more kinds of strategies like vision transformers. It's worth noting that the model appears to place the groups in the same location as the CNN on raw data. Despite the groupings being slightly misaligned to the ground truth labels, this could mean that the machine thinks these locations for groups make more than the ground truth labels.

Convolutional Neural Networks (Fuzziness)

The last big improvement we made to our models was changing our output to have fuzziness. We hope that by including fuzziness, the model can have more ground truth labels to work with and give us better results. We used our first CNN model to test this strategy.

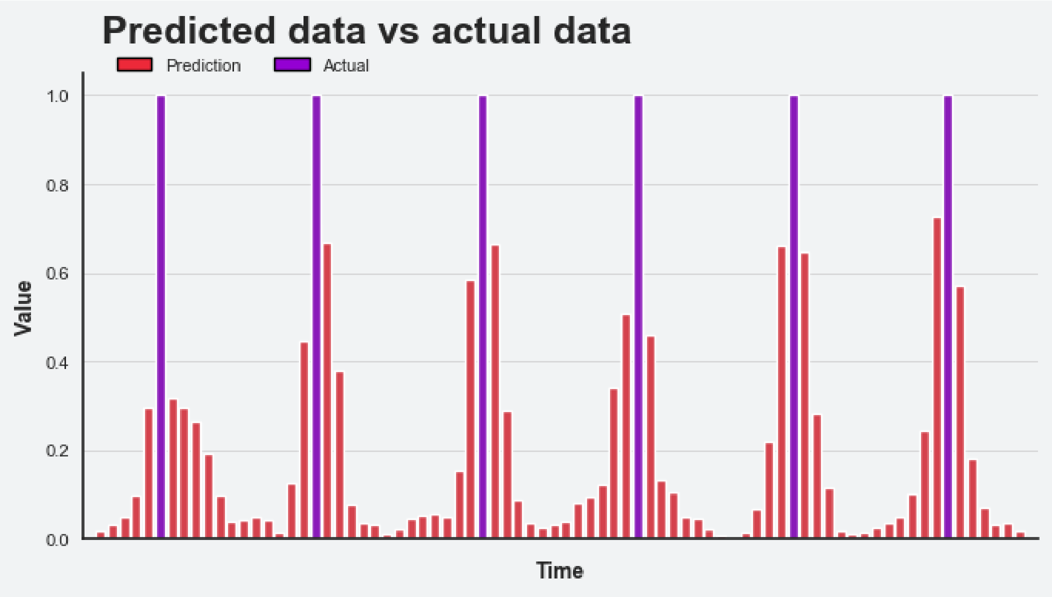


Figure 7: An example output of the CNN(Fuzziness) testing model

The model showed a great advantage in classifying more groups of data and performance was consistent across all examples. It also averages out the noise within the dataset and gives us much nicer distributions.

Unsuccessful Models

Recurrent Neural Networks

First, we tried the LSTM(Long short-term memory) model, but the results didn’t predict any groupings. This may be caused by the gradient vanishing problem that all RNNs have. It means in each iteration, the current weight would change due to the previous weight for an error function. When the gradient is vanishingly small, effectively preventing the weight from changing its value. That’s why it didn't show significant differences between each group.

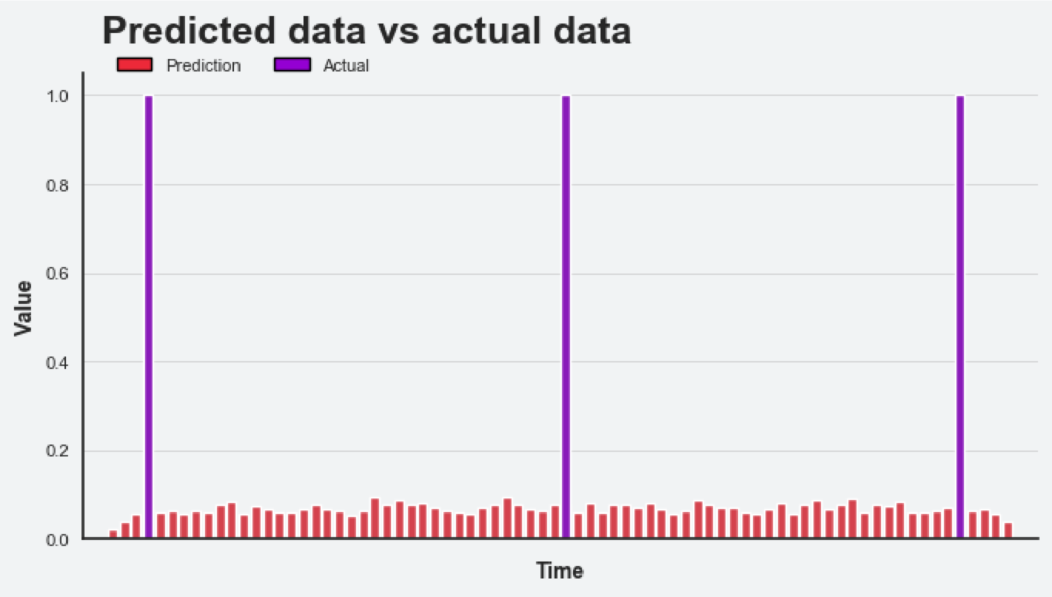


Figure 8: An example output of the RNN LSTM testing model

Second, we tried GRU(Gated Recurrent Unit) instead of LSTM, but the result was similar to LSTM. This is reasonable because GRU and LSTM were all based on the basic RNN architecture, so both of them would have gradient vanishing problems as all RNN models have.

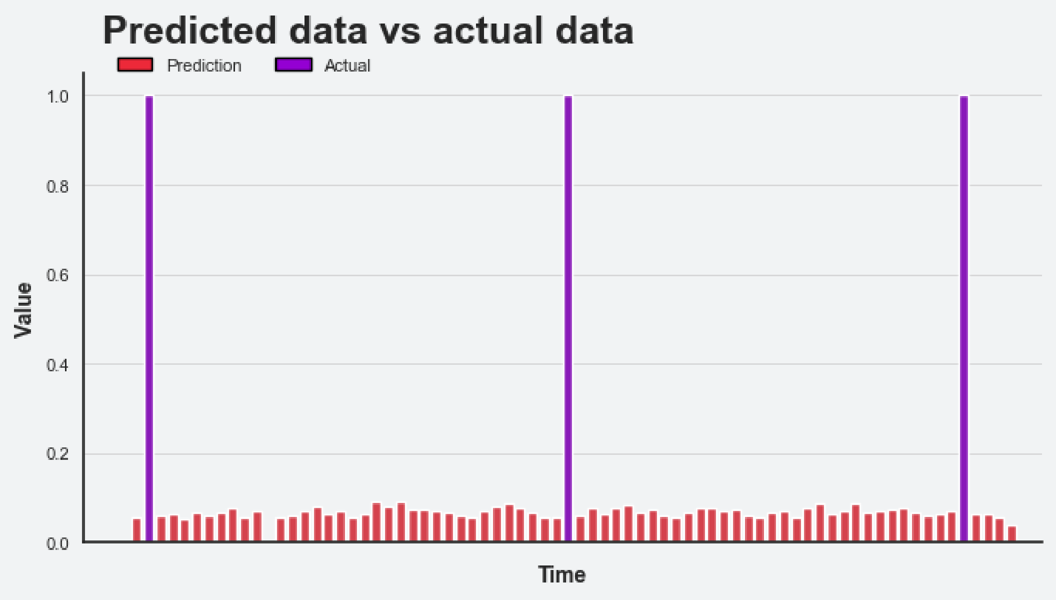


Figure 9: An example output of the RNN GRU testing model

Convolutional Recurrent Neural Networks

We thought that we may have to extract a few features from the spectrogram before using the recurrent layers, so we attempted to use a convolutional recurrent neural network. The data would first pass through the convolutional neural network for feature engineering, then through the recurrent neural network to better extract the temporal features. For our implementation, we used a few convolutional layers followed by an LSTM layer, then a Dense layer.

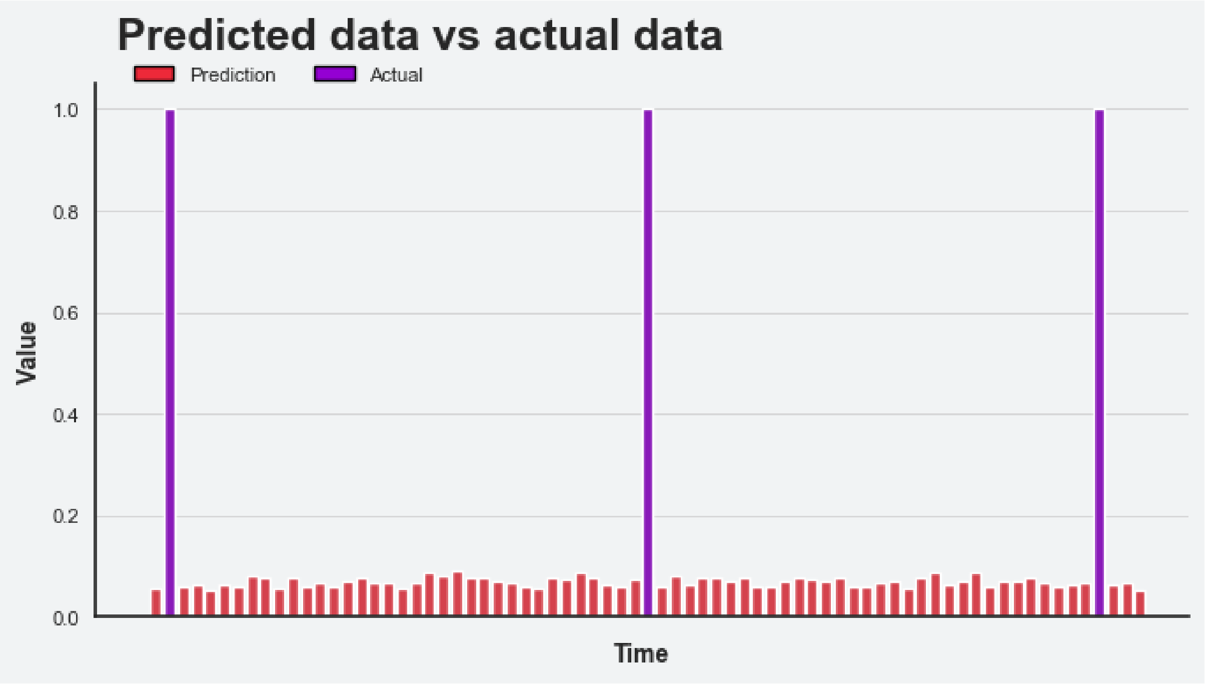


Figure 10: An example output of the CRNN testing model

The model result looked much similar to the pure RNN model. Because the results look so similar, we thought the issues were due to the RNN layers.

VIT(Vision Transformers)

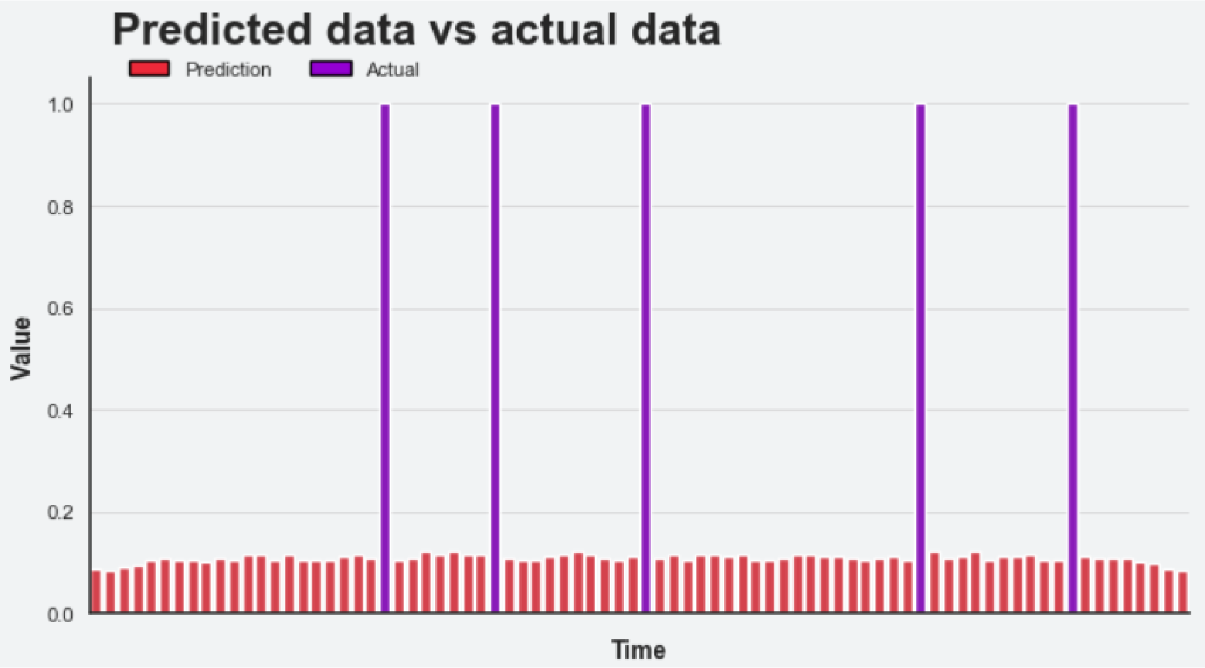
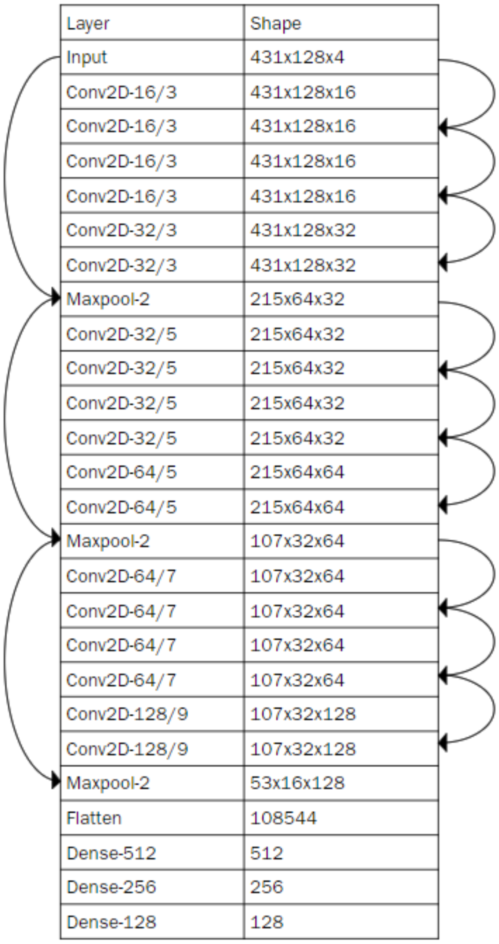


Figure 11: An example output of the VIT testing model

The result of the model was similar to the RNN model. Due to our limited understanding of VITs, we're not quite too sure why it doesn't work. Perhaps implementing the patches in such a way that would work well for image recognition doesn't translate well for image spectrograms.

Final model

In our final model we used a convolutional neural network with same padding, no strides, and skip connections throughout the model to mitigate the potential vanishing gradient issue. We increased the number of layers total to 25 layers which will increase the complexity of the model to allow for it to fit better.



Layer	Shape
Input	431x128x4
Conv2D-16/3	431x128x16
Conv2D-16/3	431x128x16
Conv2D-16/3	431x128x16
Conv2D-16/3	431x128x16
Conv2D-32/3	431x128x32
Conv2D-32/3	431x128x32
Maxpool-2	215x64x32
Conv2D-32/5	215x64x32
Conv2D-32/5	215x64x32
Conv2D-32/5	215x64x32
Conv2D-32/5	215x64x32
Conv2D-64/5	215x64x64
Conv2D-64/5	215x64x64
Maxpool-2	107x32x64
Conv2D-64/7	107x32x64
Conv2D-64/7	107x32x64
Conv2D-64/7	107x32x64
Conv2D-64/7	107x32x64
Conv2D-128/9	107x32x128
Conv2D-128/9	107x32x128
Maxpool-2	53x16x128
Flatten	108544
Dense-512	512
Dense-256	256
Dense-128	128

Figure 12: Final model definition

One thing worth noting is that the convolutional layers are separated into blocks of 6 convolutional layers. Within each block, residual connections connect every other convolutional layer and the first layer within each block has a residual connection to the output of the block. While we hoped that increasing the complexity of the model and implementing the above methods would improve the performance of the final model. The model, however, doesn't generalize well to the testing data. The results are shown in figure 13.

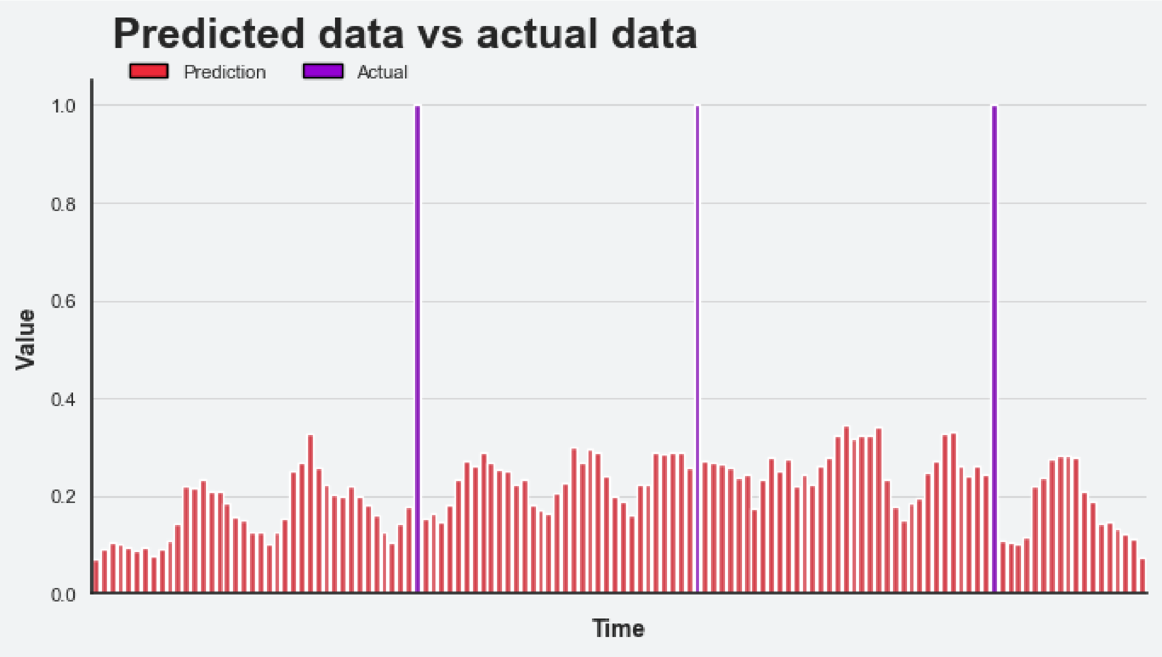


Figure 13: The output of the final model

Discussion

While the model did well on music clips of the same genre as shown in figures 5-7, the model generalized poorly to music clips of other genres as shown in figure 13 above. We believe the main cause of the models' poor performance on testing data is the training data. With more training data on a larger variety of music, we should be able to overcome the problem of poor generalization. We've identified other potential causes for poor performance, such as the model architecture. We only used convolutional layers in our model, but perhaps temporal information cannot be adequately extracted from convolutional layers alone, so giving RNNs another chance, perhaps by changing the axis in which the model parses information temporarily in, would yield better performance. Vision transformers are also another model that could be worth reconsidering, if we were to change our approach of turning the data into patches, then the transformer could yield better results.

Even though the groupings are hard to discern in the figure above, it does provide a general guideline as to where the model predicts groupings. There are certain "mountains" in the predictions and with more familiarity with groupings and experience with this distribution, a human could predict group beginnings based on this distribution alone.

Conclusion

This project focused on training the machine learning model to output groupings of music clips. We used surveys to collect human perceived audio grouping data as our model's input.

Because of the uniqueness of the data type we collected, we explored several ways to process the data like 1-D convolutions and 2-D convolutions. These methods significantly improved our model performance and accuracy.

The most important technical decision in this project was to choose the right machine learning model. The effort was made on several models which include convolutional neural networks, recurrent neural networks, convolutional recurrent neural networks, and VIT(vision transformers) models. Among all the models, convolutional neural networks showed success in clearing boundaries of grouping and less noise. Other models had issues with the vanishing gradient problem, especially the RNN-based models, leading to indecisive grouping boundaries.

Our final model was a convolutional neural network containing 25 layers with residual connections. We theorized that increasing the complexity of the model by adding more layers would increase the model performance. The model was able to produce interpretably significant groupings on music clips of the same musical genre as training data, but generalized poorly to music clips of other genres. We theorized that this was mainly caused by the limited amount of training data.

While the results produced by the model didn't achieve a standard of automated grouping adequate to enable usage by the algorithmic applications considered in the background, the results are still valuable. The positive grouping gestures of the CNN model are indicative of a successful application of a supervised machine learning approach for the auditory grouping problem. This success indicates that future work in this problem space can utilize rapidly developing supervised machine learning techniques to achieve its goals. This is especially relevant as state of the art machine learning techniques are very sensitive to data quantity, training resources, and algorithmic developments. We expect that a more resourced study of hyperparameter optimization with a model similar in form to the CNN model we developed would find qualitative successes in auditory grouping.

The primary shortcomings of the model with regards to the auditory grouping problem were in generalizing results from training genres to external genres. This report posits that this weakness could be resolved by including music from a wider distribution of genres. For validation purposes of a pilot integration, this project utilized many clips from the same song in its labeling scheme. Now that the capabilities of CNN models to model groupings of audio are better understood, a future project can sample against a much wider spread of audio sources.

Specifically, we recommend that future labeling schemes prioritize getting a high number of label passes on each clip and labeling clips from a wide variety of sources. We found high variance in the grouping labels between labellers, likely due to different “thresholds” of what latent partitions designate a group. By collecting high multiplicity label data, the grouping boundaries could be better understood against a probabilistic hierarchy of “tiers” where the most common grouping boundaries indicate higher order partitions of the sample.

Additionally, while our samples were generated by slicing a relatively small set of specifically curated songs into short clips, we find that this was ultimately unnecessary, and likely counterproductive with regards to overfitting against genre. There's no need for high multiplicity on a per-clip level outside of having multiple grouping results for each clip from a single survey session. A more canny scheme would be to collect samples from a wide corpus of audio sources and have each survey participant operate on a randomized subset of those clips. This approach would allow for the validation benefits we received from our clipping scheme without overfitting individual song qualities, like genre. Given that these changes would require recollecting data, they were outside the scope of this project's charter.

While the results of our final model did not reach the standards of grouping necessary to indicate fitness for application, they were adequate to justify future inquiry. Based on the comparative performance of our modeling strategies, we expect further research into the application of convolutional neural networks to the auditory grouping problem to achieve relative success. Additionally, we recommend that future research focuses on a wider distribution of sample sources to remedy the generalization problems we encountered.

References

[1] F. Lerdahl and R. S. Jackendoff, *A Generative Theory of Tonal Music*. Cambridge, MA, USA: MIT Press, 1982.

[2] M. T. Pearce, D. Müllensiefen, and G. A. Wiggins, “Melodic Grouping in Music Information Retrieval: New Methods and Applications,” in *Advances in Music Information Retrieval*, Z. W. Raś and A. A. Wieczorkowska, Eds. Berlin, Heidelberg: Springer, 2010, pp. 364–388. doi: 10.1007/978-3-642-11674-2_16.

[3] E. Cambouropoulos, “Musical Parallelism and Melodic Segmentation: A Computational Approach,” *Music Percept.*, vol. 23, no. 3, pp. 249–268, Feb. 2006, doi: 10.1525/mp.2006.23.3.249.

[4] M. WERTHEIMER and K. Riezler, “GESTALT THEORY,” *Soc. Res.*, vol. 11, no. 1, pp. 78–99, 1944.

[5] Y. Hiraga, “Structural Recognition of Music by Pattern Matching,” University of Library and Information Science, 1997. [Online]. Available: https://quod.lib.umich.edu/cgi/p/pod/dod-idx/structural-recognition-of-music.pdf?c=icmc;idno=bbp2372.1997.113;format=pdf

[6] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, “Deep Learning for Audio Signal Processing,” *IEEE J. Sel. Top. Signal Process.*, vol. 13, no. 2, pp. 206–219, May 2019, doi: 10.1109/JSTSP.2019.2908700.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” in *Advances in Neural Information Processing Systems*, 2012, vol. 25. Accessed: Apr. 28, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html

[8] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W. Wong, and W. WOO, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” in *Advances in Neural Information Processing Systems*, 2015, vol. 28. Accessed: Apr. 28, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/07563a3fe3bbe7e3ba84431ad9d055af-Abstract.html

[10] K. Choi, G. Fazekas, M. Sandler, and K. Cho, “Convolutional recurrent neural networks for music classification,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 2392–2396. doi: 10.1109/ICASSP.2017.7952585.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *ArXiv151203385 Cs*, Dec. 2015, Accessed: Apr. 28, 2022. [Online]. Available: http://arxiv.org/abs/1512.03385

[12] A. Vaswani *et al.*, “Attention is All you Need,” in *Advances in Neural Information Processing Systems*, 2017, vol. 30. Accessed: Apr. 28, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[13] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” *ArXiv201011929 Cs*, Jun. 2021, Accessed: Apr. 28, 2022. [Online].

Available: http://arxiv.org/abs/2010.11929

[14] P. Verma and J. Berger, “Audio Transformers:Transformer Architectures For Large Scale Audio Understanding. Adieu Convolutions,” *ArXiv210500335 Cs Eess*, May 2021, Accessed: Apr. 28, 2022. [Online]. Available: http://arxiv.org/abs/2105.00335

[15] J. S. F. Hay and R. L. Diehl, “Perception of rhythmic grouping: Testing the iambic/trochaic law,” *Percept. Psychophys.*, vol. 69, no. 1, pp. 113–122, Jan. 2007, doi: 10.3758/BF03194458.

[16] M. Spierings, J. Hubert, and C. ten Cate, “Selective auditory grouping by zebra finches: testing the iambic–trochaic law,” *Anim. Cogn.*, vol. 20, no. 4, pp. 665–675, Jul. 2017, doi: 10.1007/s10071-017-1089-3.

[17] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, “FMA: A Dataset For Music Analysis,” *ArXiv161201840 Cs*, Sep. 2017, Accessed: Apr. 19, 2022. [Online]. Available: http://arxiv.org/abs/1612.01840

[18] J. Schlüter and S. Böck, “Improved musical onset detection with Convolutional Neural Networks,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6979–6983. doi: 10.1109/ICASSP.2014.6854953.

Appendix

Appendix A: Survey demographic data

Machine Perception-Auditory Grouping MQP for MTURK

Start of Block: Block 1

Q1 Informed Consent:

Investigators: Zachary Wagner, William McDonald, Yang Chen, Cheng-Hsuan Jan Contact Information: zwagner@wpi.edu, ychen18@wpi.edu, cjan@wpi.edu, and wbmcdonald@wpi.edu

Title of Research Study: Machine Perception Auditory Grouping MQP Advisors: Professor Scott Barton (sdbarton@wpi.edu), Professor James Doyle (doyle@wpi.edu), Professor Gillian Smith (gmsmith@wpi.edu)

You are being asked to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation. The purpose of our study is to collect data on how human beings group pieces of music. To fulfill this goal, you will be listening to 20 different audio clips from various songs, with each recurring 5 times as to allow certainty with the groupings. Each audio clip is ten seconds long and the participant has full discretion to make as many or as few groupings as they want. The volume can be controlled on your device and there will be no risks to the participant.

By participating in this research, you will be aiding in the scientific understanding of music perception, as well as emerging technologies that could benefit from the data. Your responses will be completely confidential, and no data about you is collected other than that which is provided through the music grouping and survey tasks. Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators, the sponsor or it's designee and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identify you by name. Any publication or presentation of the data will not identify you.

If you are participating through WPI's SONA Systems, the appropriate study credit will be applied to you account a few days after study completion. If you are participating through Amazon MTURK, you will receive monetary compensation of 3$.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact: Professor Scott Barton (sdbarton@wpi.edu), Professor James Doyle (doyle@wpi.edu), Zachary Wagner (zwagner@wpi.edu), Yang Chen (ychen18@wpi.edu), Cheng-Hsuan Jan (cjan@wpi.edu), and William McDonald (wbmcdonald@wpi.edu) Also, please feel free to reach out to the IRB Manager (Ruth McKeogh, Tel. 508 831- 6699, Email: irb@wpi.edu) and the Human Protection Administrator (Gabriel Johnson, Tel. 508-831-4989, Email: gjohnson@wpi.edu). Your participation in this research is voluntary. Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. The project investigators retain the right to cancel or postpone the experimental procedures at any time they see fit. By clicking "continue," you agree to understanding all of the above information.

Q2 Do you agree to participate in the survey?

○ Yes (1)

○ No (2)

End of Block: Block 1

Start of Block: Block 2

Q3 In order to listen to and group the songs, we need to redirect you to a separate link for the first part of our study.

Please open this link in a **new, seperate tab**. (Link on the next page).

At the end of the grouping task, you will return to this study portal. You will be asked to copy and paste a unique code before continuing in this main portion of the study.

Please be prepared to copy the unique code.

Page Break

Q4 Please click HERE to be taken to the grouping task.

(Note: Make sure to do this in a new tab)

End of Block: Block 2

Start of Block: Block 3

Q5 Please input the unique ID you received at the end of the grouping tasks.

____

End of Block: Block 3

Start of Block: Block 4

Q6 Did you find anything particularly difficult about the grouping tasks?

○ Yes (1)

○ No (2)

Q7 If so, please specify.

____

Q8 What type of device did you use for the survey and grouping tasks?

○ Computer (1)

○ Phone/Tablet (2)

○ Other (3)

Q9 Please take a moment to describe in more detail how you decided to group each music clip. (percussion, melody, instrumentals, etc.)

____

Q10 What is your gender?

○ Man (1)

○ Woman (2)

○ Non-Binary (3)

○ Prefer not to disclose (4)

○ Prefer to describe (5)

Display This Question:

If What is your gender? = Prefer to describe

Q11 Please describe.

Q12 What is your race?

○ White/Caucasian (1)

○ Black/African American (2)

○ Asian (3)

○ Hispanic (4)

○ Pacific Islander (5)

○ Other (6)

○ Multiracial (7)

○ Prefer Not to Say (8)

Q13 What is your age category?

○ Under 18 (4)

○ 18-25 (5)

○ 26-33 (6)

○ 34-41 (7)

○ 42-49 (8)

○ 50-59 (9)

○ 60-69 (10)

○ 70+ (11)

Q14 Are you currently a college student?

○ Yes (10)

○ No (11)

○ Graduated College (12)

○ Currently In Graduate School (13)

○ On Leave from College (14)

○ Past Enrollment in College (15)

○ Other (16)

Q15 Have you ever played an instrument for a period longer than 1 continuous year?

○ Yes (1)

○ No (2)

Q16 Do you currently play an instrument?

○ Yes (1)

○ No (2)

Display This Question:

If Do you currently play an instrument? = Yes

Q17 If so, how many years of experience do you have?

____

Q18 On a scale from 1-7, how much would you say LISTENING to music plays a role in your life?

None at all A little A moderate amount A lot A great deal

1 2 3 4 5 6 7

Drag slider ()	

Q19 On a scale from 1-7, how much would you say PLAYING music plays a role in your life.

1 2 3 4 5 6 7

Drag Slider ()	

Q20 What is your favorite genre of music?

____

Q21 Do you generally listen to music from outside your own culture?

○ Yes (1)

○ No (2)

Q22 Do you generally listen to instrumental music?

○ Yes (1)

○ No (2)

Q23 What genres are common to your favorite playlists? (select all that apply)

□ Latin (1)

□ Reggae (2)

□ American Pop (3)

□ American Rap (4)

□ Europop (5)

□ Country (6)

□ Pop Country (7)

□ Classic Rock (8)

□ Hard Rock (9)

□ Glam Rock (10)

□ Swing (11)

□ Big Band (12)

□ Classical (13)

□ Rap (Non-American) (14)

□ EDM (15)

□ Dubstep (16)

□ Folk Music (Any Culture) (17)

□ K-Pop (18)

□ C-Pop (19)

□ J-Pop (20)

Q29 What genres are common to your favorite playlists? (select all that apply)

□ Latin (1)

□ Reggae (2)

□ American Pop (3)

□ American Rap (4)

□ Europop (5)

□ Country (6)

□ Pop Country (7)

□ Classic Rock (8)

□ Hard Rock (9)

□ Glam Rock (10)

□ Swing (11)

□ Big Band (12)

□ Classical (13)

□ Rap (Non-American) (14)

□ EDM (15)

□ Dubstep (16)

□ Folk Music (Any Culture) (17)

□ K-Pop (18)

□ C-Pop (19)

□ J-Pop (20)

End of Block: Block 4

Start of Block: Block 4

Q24 **MTURK CODE: 98765432**

Debriefing Statement: Thank you so much for completing our survey today. The true purpose of the survey is to gather data on how people group audio clips, to then feed this data to a deep learning algorithm. Our survey was a part of an MQP that is seeking to determine if there are patterns in how human beings group music, that can then be applied to machine learning models to perform the same groupings. We hope that the data we collected will be sufficient to train an algorithm to group audio in the same way humans do. If you have any questions regarding your participation today in this survey, please contact the researchers at zwagner@wpi.edu, ychen18@wpi.edu, cjan@wpi.edu, and wbmcdonald@wpi.edu

End of Block: Block 4