PROGRESS REPORT

You Li
Student# 1004891511
youforest.li@mail.utoronto.ca

Luohong Wang
Student# 1005073313
luohong.wang@mail.utoronto.ca

Devansh Khare
Student# 1004980292
devansh.khare@mail.utoronto.ca

Rose Hyunjie Kim
Student# 1004341134
hjrose.kim@mail.utoronto.ca

1 BRIEF PROJECT DESCRIPTION

The purpose of this project is to create a tool that will be able to recognize the genre of a given piece of music based exclusively on the audio content. The motivation behind the project comes from the need for music recommendations. With the rising number of music streaming services, a personalized recommendation system attracts more users and boosts engagement (Dee, 2022). The platforms are then able to use the data generated by its additional users to further improve their model. This project will attempt to recreate genre classification which is one of the essential parts of music recommendation.

Machine learning is the ideal approach to this problem as there is a very large amount of unprocessed data that comes from various music sources with no specific ruleset for recommendation. A deep learning model can learn from these data points and make a prediction based on its training.

2 INDIVIDUAL CONTRIBUTIONS AND RESPONSIBILITIES

Group 45 is currently practicing organization by scheduling weekly meetings to discuss the agenda, plan internal deadlines, provide updates on work progress, and to distribute work amongst all members. Discord is being used as an online platform to communicate and host the meetings. The team's approach to track the progress is to provide brief verbal summaries of our work progress, and by documenting necessary documents such as codes, csv files, png files, and others in the shared Google Drive folder.

During the meetings, our team can check on each file separately to see what the work looks like. Each member in the team is responsible for completing their assigned tasks within the internal schedule/deadline, participating in regular group meetings, and communicating on a timely basis to provide updates on work progress as well as to address any issues and difficulties with the project if any arises.

In terms of expectations from each member around meeting deadlines, each team member is expected to declare soft and hard deadlines for themselves to the team. Assuming proper team communication has been done, soft deadlines can be flexible within a few days. However, hard deadlines must be submitted on the exact deadline. Hard deadlines include submission for written reports and presentations which must be submitted on the exact deadline. The team is currently making progress in the project by scheduling internal deadlines on the project during meetings and informing each other about hard deadlines.

To work on the tasks, our team splitted into 2 sub teams. 1 sub team worked on data processing and the other subteam worked on the creation of baseline and CNN model. The following items were set down for each team member for up to November 4th. Each team member has completed all their items in this list:

Table 1: Project Tasks With Deadlines - Data Processing team

Project Task	**Name**	
	You Li	Rose Hyunjie Kim
Data Processing: Audio file feature extraction	Oct 30th -Prepare code to extract necessary audio file features to classify the GTZAN data set	
Data Processing: Splitting data to smaller chunks	Oct 31st Enhance the run time for splitting the GTZAN data set	Oct 30th Prepare code to split the GTZAN data set to 3 second intervals
Data Processing: Training and Validation Set	Oct 31st Prepare code to split the the MTZAN data set to training and validation set	
Progress Report	Nov 4th -Data processing section write up	Nov 4th -Individual contributions and responsibilities section write up. -Review and edit data processing section.

Table 2: Project Tasks With Deadlines - CNN team

Project Task	**Name**	
	Luohong Wang	Devansh Khare
Building Baseline Model	October 29th KNN Model Completed	
Training Baseline Model	October 29th First Training Completed and accuracy recorded	
Tuning Baseline Model	October 30th Second training completed and recorded accuracy for multiple scenarios	
Building CNN Model	Nov 2nd CNN Model & helper functions developed	Nov 2nd CNN Model & helper functions developed
Training CNN Model	Nov 2nd First pass of training	Nov 2nd First pass of training
Tuning CNN Model		Nov 3rd First and second pass of tuning completed Final model for progress report completed
Progress Report	Nov 4th - Baseline model write up	Nov 4th - Primary model write up

The completed schedule demonstrates that our team is currently on track with the overall project since the works that will outline the project are completed. Each team member has been contributing to the project on a regular basis.

Until the end of the project, the team plans to schedule weekly deadlines to ensure that the project is being completed on time. The tasks have been divided per each neural network that have to be built. To counter any risks that may prevent our team from completing the tasks, two people have been assigned for each neural network respectively and each team member is expected to assist and support each other internally when needed.

Based on it, our team's future plans is as follows:

Table 3: Project Tasks With Deadlines

Project Task	**Name**			
	Luohong Wang	You Li	Rose Kim	Devansh Khare
Building CNN	November 12th			November 12th
Training CNN	November 12th			November 12th
Tuning CNN	November 12th			November 12th
Building RNN		November 12th	November 12th	
Training RNN		November 12th	November 12th	
Tuning RNN		November 12th	November 12th	
Building CRNN	November 22th			November 22th
Training CRNN	November 22th			November 22th
Tuning CRNN	November 22th			November 22th
Final Presentation		November 25th	November 25th	
Final Project Report		December 2nd	December 2nd	

3 Notable Contribution

3.1 Data Processing

The data was taken from Kaggle, an online public source for users to explore, analyze and share data (kag). The dataset itself is the GTZAN Dataset - Music Genre Classification. It is the most used public dataset for evaluation in machine listening research for music genre recognition and considered a benchmark in the field (Olteanu, 2020)(mac). The dataset includes 10 genres of music, with 100 audios in each genre at 30 seconds each.

The raw audio files were cleaned and processed a few different ways. Firstly, a Mel Spectrogram representation of each of the audio files was generated through the librosa python package. These plots were then saved as PNG files in the group's common Google Drive space. An 85-15 split was done on the image files to split the data into training and validation. An example of the Mel Spectrogram image can be seen below as Figure 1.

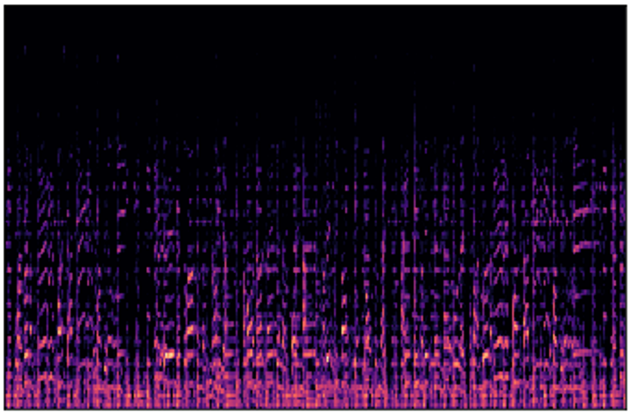


Figure 1: Mel Spectrogram Representation of a Blues Audio

Next, the group decided to split each of the audio files into 3 second clips, increasing the number of total data points by a factor of 10. This ensures that the models will have enough data points to train from. This step was done locally on the members' personal machines to save space on the common drive. Below in Figure 2 is an image of the Mel Spectrogram representation of the same audio clip, but only for the first 3 seconds.

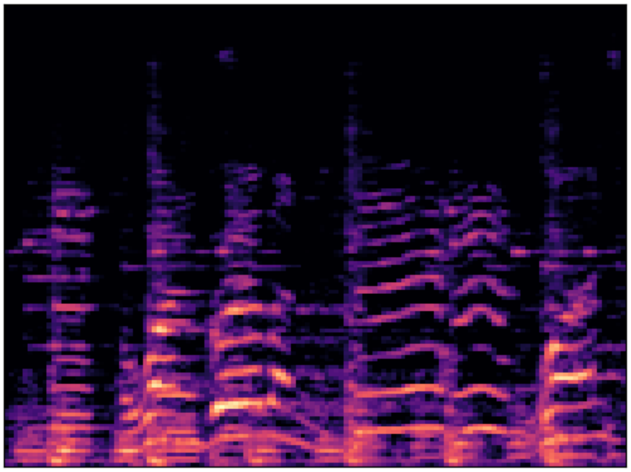


Figure 2: Mel Spectrogram Representation of a 3-second Blues Audio

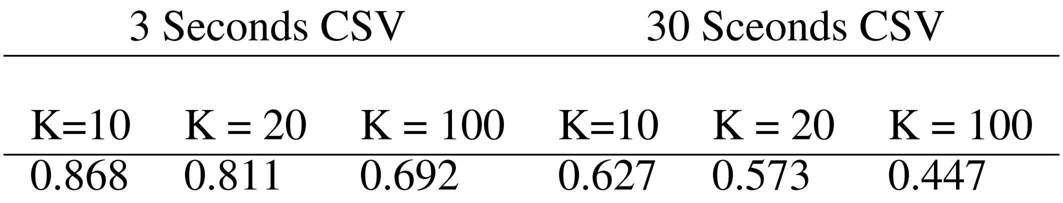
Following the image generation, the team has also decided to extract certain features from each of the audio files. The process is also done through the librosa python package. The features extracted include: Short-time Fourier transform (STFT) of chromagram, root-mean-square (RMS) value, spectral centroid, spectral bandwidth, roll-off frequency, zero-crossing rate, Mel-frequency cepstral coefficients across 20 channels, harmonic elements, percussive elements and tempo. The mean and variance of each of the features were selected in order to normalize the data. Similarly to the earlier Mel Spectrogram image generation, this step was redone for audios split into 3 seconds clip to ensure sufficient data for the model. These features are saved under a CSV format, split into 85-15 for training and validation sets.

To ensure accurate testing of the models, the team has decided to find brand new data that is not part of the GTZAN Dataset taken from Kaggle. As the model needs audio clips from different genres, it is possible to obtain new songs made after the GTZAN Dataset creation date from music streaming services such as YouTube or Spotify. This will ensure that the audio samples selected are not a part of the GTZAN.For the test set, the same number of samples will be gathered as the validation set, which will be 15 samples of 30 second files for each genre of music.

3.2 Baseline Model

The baseline model of our choice is the KNN model, where an object is classified through a vote of its K's nearest neighbours. We think KNN is a good baseline model for Music Recognition since similar music features such as tempo, harmonics, percussive elements, etc, should indicate that those music are in the same genre. The data set used for KNN model is the CSV file. We decided to perform KNN classification on both the 30 second version and the 3 second version, to give us a better idea on how the data works. For each data set, we did 10 neighbours, 20 neighbours and 100 neighbours. Below table shows result of the KNN models.

Table 4: KNN model accuracy



3 Seconds CSV			30 Sceonds CSV		
K=10	K = 20	K = 100	K=10	K = 20	K = 100
0.868	0.811	0.692	0.627	0.573	0.447

As expected, the KNN model done on the 3 second CSV file clearly shows much better accuracy on the validation set than compared to the 30 second model. The KNN model that yielded the best result in nearest neighbour value of 10, with validation accuracy of 0.868.

3.3 Primary Model

3.3.1 Architecture

The purpose of the deep learning model is to take the Mel Spectrogram graphs and predict the genre to a better accuracy than the KNN model.

Given that the Mel Spectrogram graph for the audio file is saved as an image, the team chose a CNN-based architecture to limit the number of parameters in the model. The input images had dimensions of 4 x 288 x 432, and the model outputs were the corresponding music genres for the audio files. The team chose a 10 layered CNN structure for this classification – with 4 convolution operations layers, 4 max pool operations and 2 linear layer connections. ReLU activation was used after every convolution operation. The architecture of the project has been shown in Figure 3 below.

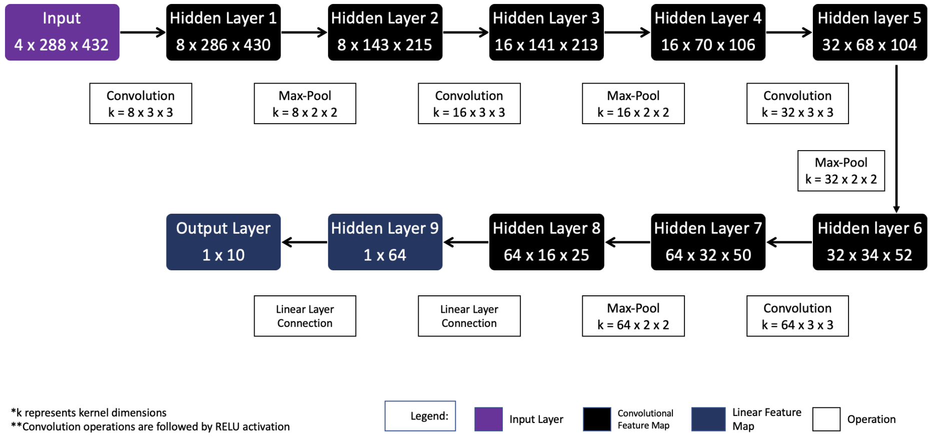


Figure 3: Present Network Architecture

Considering that the problem involves multi-class classification, the team chose to use nn.CrossEntropy to compute the loss function with automatic sigmoid activation. Helper functions were written to obtain data and accuracy for different datasets and batch sizes, and the manual seed for the torch library was set to 1000 to ensure replicable results.

The team incorporated two types of optimizations: fit optimizations and runtime optimizations. To optimize the model's fit, the team tested different values of dropout between linear layers and of L2 regularization in its model. To optimize the model's runtime, the team tested different optimizers with momentum and incorporated early stopping with patience. The team also set up code breaks (minimum accuracy = 0.25) in case the validation accuracy was too low after a fixed number of epochs (epoch 13, out of 18 total epochs) to minimize time spent on unproductive learning.

The following hyperparameters were selected for the architecture to yield the highest validation accuracy so far at 53.33%:

Table 5: Selected Hyperparameters For Best Model So Far

Hyperparameter	Selected Value
Learning Rate	0.001
Batch Size	64
Epochs	18
Optimizer	Adam
Momentum	0.5
Regularizer	0.001
Dropout	0.25

3.3.2 Quantitative Results

The model had a final training loss of 0.0163, final training accuracy of 74.12% and final validation accuracy of 53.33%. The figures depicting the same have been shown below:

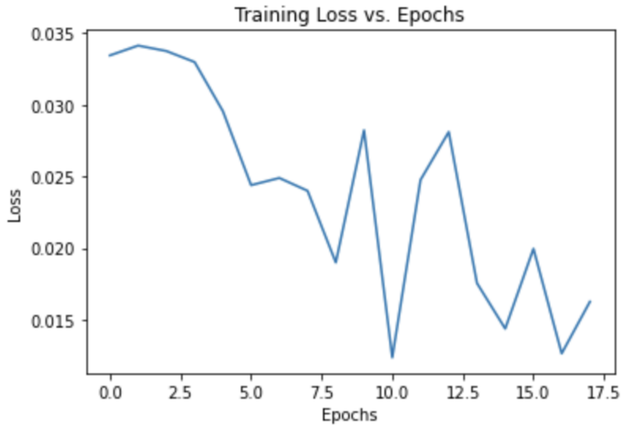


Figure 4: Training Loss vs. Epochs

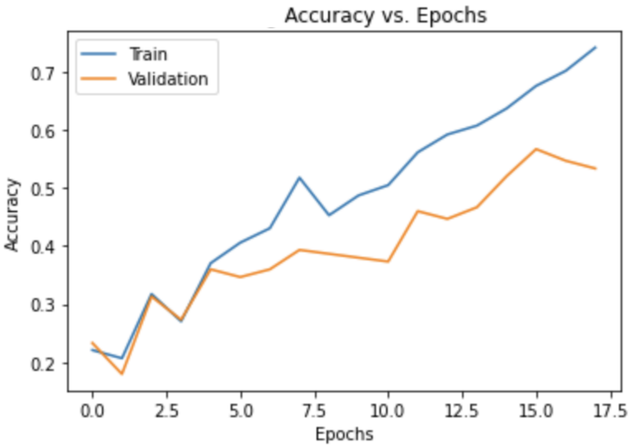


Figure 5: Accuracy vs. Epochs

Three observations need to be noted for the present model:
1) The loss values start oscillating after epoch 8.
2) The validation accuracy of 53.33% is lower than the baseline KNN model presently.
3) The model is probably overfitting, because the training accuracy exceeds validation accuracy by more than 20%.

The ’Challenges’ subsection below elaborates on strategies to tackle these concerns.

3.3.3 Qualitative Results

The qualitative results showed overfitting to the Blues genre in the training set. The model achieved an accuracy of 47% on Blues music, but did not cross 15% accuracy for the other genres.

The ’Challenges’ subsection expands on ways to tackle this trend in the next few weeks.

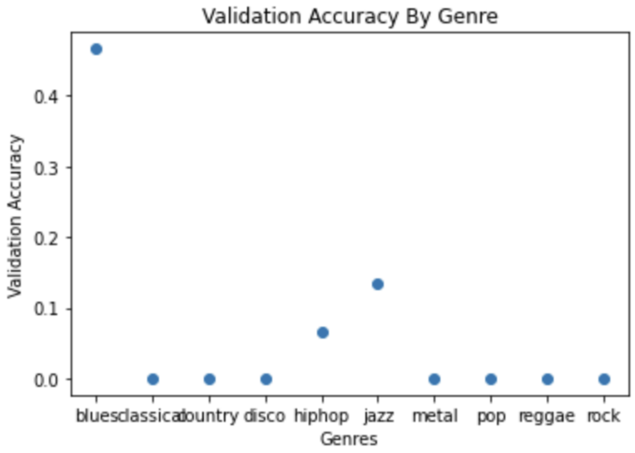


Figure 6: Validation Accuracy By Genres

3.3.4 Challenges and Strategies

The model faces three notable challenges presently - oscillations in the loss, low validation accuracy and overfitting to the Blues genre. Table 4 details the measures that the team will incorporate to overcome them.

Table 6: Challenges and Navigation Strategies

CNN Model Challenges	Navigation Strategy
Oscillations In Loss Values After Epoch 8	1) Increase batch size while training 2) Set up scheduled learning rate decreases after Epoch 8 3) Increase momentum in the optimizer after Epoch 8
Low Validation Accuracy	1) Increase size of training set by collecting more audio samples 2) Increase the depth of CNN model 3) Further tune the hyperparameters listed in Table 3
High Overfitting To Blues Genre	1) Minimize any skew in class distribution in training set 2) Increase regularization penalty for large weights 3) Incorporate heavier dropout across layers

Additionally, the team will also build an RNN model in the upcoming weeks as an alternative with temporal awareness for music classification. This RNN model could be used in combination with the CNN for ensemble learning and predictions to improve accuracy.

4 Link to Colab Notebook

https://colab.research.google.com/drive/1bEnPoQ-BnN6fZegwO2qujM_doygWvkZn?usp=sharing

References

Datasets documentation. URL `https://www.kaggle.com/docs/datasets`.

Gtzan genre dataset. URL `https://datasets.activeloop.ai/docs/ml/datasets/gtzan-genre-dataset/`.

Catherine Dee. How do music recommendations work to boost user engagement?, Jun 2022. URL `https://www.algolia.com/blog/ux/how-does-a-music-recommendations-algorithm-work-to-boost-user-engagement/`.

Andrada Olteanu. Gtzan dataset - music genre classification, Mar 2020. URL https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification.