| **Title: Music Genre Classification Project Proposal** | |
| --- | --- |
| **Project Team: 32** | |
| **Oluwasina Olowookere** | **1006648068** |
| **Tyler Wong** | **1006999134** |
| **Anna Szatan** | **1006995209** |
| **Blythe Huang** | **1007418153** |

**Abstract**

Automating the classification of music genres based on sound data has the potential to be applied to many music labeling and streaming companies for indexing and recommendations. The team aims to design and train a neural network model that can accurately classify a song’s music genre. The project plan requires the following steps: data collection, data processing/cleaning, feature extraction, model design and construction, training, validation, and testing. The type of network architecture to be applied is a Convolutional Neural Network (CNN). CNNs have been successful in pattern recognition and are often used for speech recognition, lyric transcription, and image recognition. The level of accuracy in the model will be determined using the frequency in which the network classifies the song according to previously known and accepted genre labels. Furthermore the neural network will be compared to a baseline model of nearest neighbor classification. The CNN must outperform this baseline model to a considerable degree to be considered successful. Some risks associated with the project include biased data, overfitting, and copyright issues, however these will be mitigated according to certain contingency plans. In addition, ethical considerations such as informed consent and privacy, have also been considered. The team has broken down the steps to constructing an accurate model, and associated each step with internal and formal deadlines. The project's outcome will be a trained CNN model that has the ability to accurately classify large sets of audio data into music genres.

1. **Introduction**

With the increased popularity of music streaming apps and new artists, the number of published songs has been increasing significantly [1]. Companies such as Spotify or Apple Music would benefit from automated, and accurate music genre classification to organize and recommend their libraries better for their users [2]. The goal of this project is to create a neural network that accurately classifies music genres from audio content. The milestones in this project include collecting data, building and training the neural network, optimizing its parameters, and reporting on its accuracy. The team’s interest in the topic stems from all team members having some connection to music such as through learning to play different instruments or through frequent listening. Deep learning can improve user experiences through recommendations and organization of music, in addition to being faster and able to handle large amounts of data [3]. While artists may classify their own music as belonging to a specific genre, a deep learning model can provide other genres a song may apply to.

1. **Illustration / Figure**

The aspired classification model will require several stages in order to process and correctly categorize the music it is provided. The diagram below provides an overview of the different components of the project and how they will work.

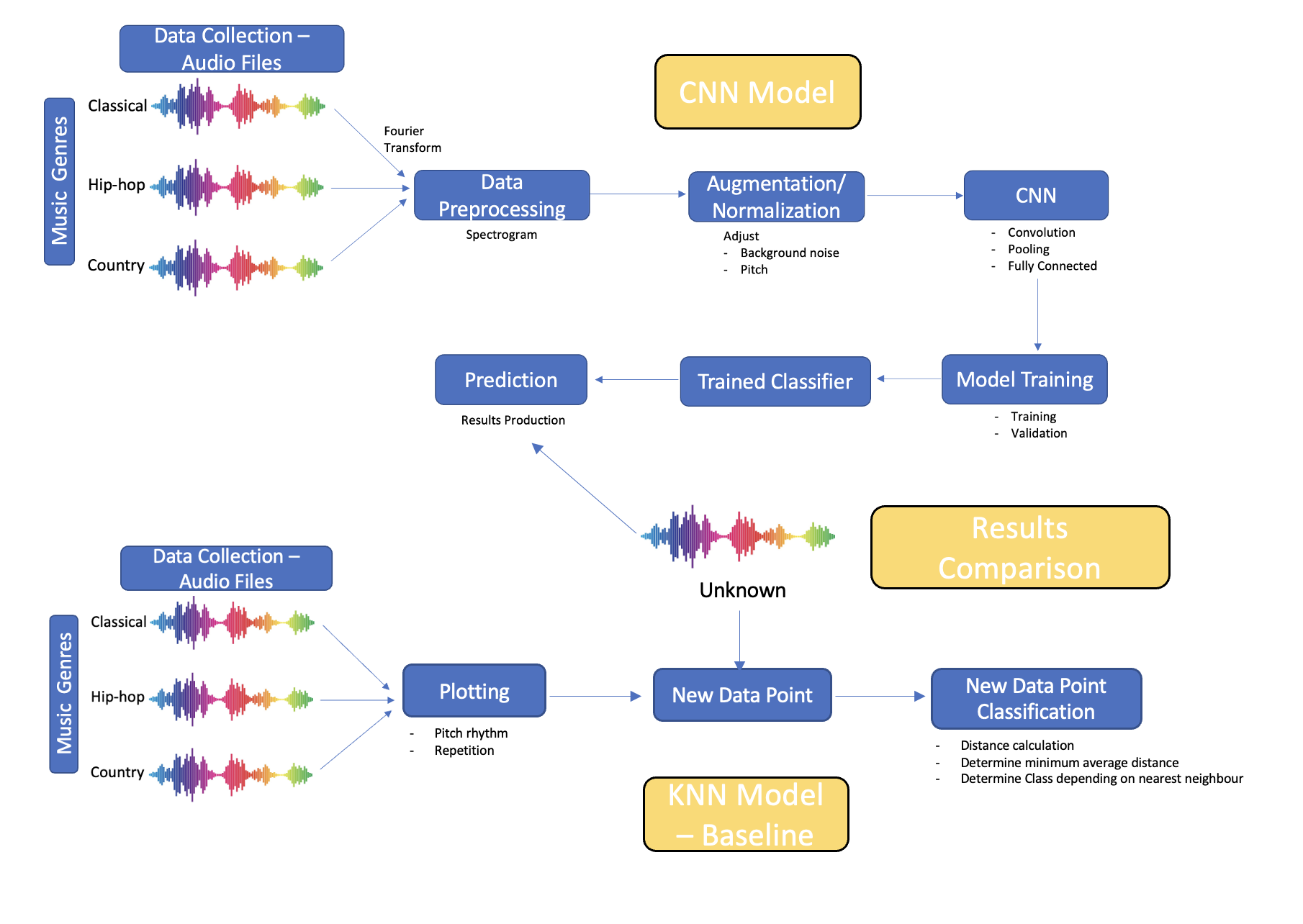


Figure 1. Diagram depicting the components of the project

1. **Background & Related Work**

Music genre recognition is a significant subset of the research field related to Music Information Retrieval (MIR) [4]. While most are familiar with music identification applications like *Shazam*, which can recognize songs from short samples, genre recognition is a more complex and subjective process [5]. Music genre classification is also a form of music label classification, with other forms being writer, emotion, and region classification, among others [6]. Some potential applications of music genre recognition include music organization and storage management within databases, internet search engine queries, recommendation systems within streaming services, and tools for music inspiration and creation [6].

Since the 1990s, genre recognition has been studied by many data and computer scientists [6]. This work is predominantly conducted using MP3 and WAV audio files [6]. In 2006, researchers in Singapore developed the first application of a “deep confidence network” to music genre classification [6]. This was done by using a multilayer classification system with Support Vector Machine (SVM) learning [6, 7]. One of the main methods of feature selection used was through using beat spectrum analysis, which automatically classifies rhythm and tempo [7]. To develop predictions, the research team utilized the theory that a subsection of a piece of music can be approximated by a linear combination of other related music samples [7].

In 2011, another group of researchers developed feature vectors for their genre recognition machine learning model through the use of a two-dimensional feature map which accounted for pitch and rhythm [6]. The feature vector coefficients generated were then used as approximations for melody [6]. This method of identifying feature vectors led to an 81% accuracy rate with the model [6].

1. **Data Processing**

In order to train the model, a significant amount of data will be required such that the prediction results are accurate. The steps involved in the our Music Genre Classification data processing will be as follows:

1. Data Collection:

In order to produce a model that will output accurate information, we need a large dataset that has various classes of music with already labeled elements [8]. For these reasons, we will be using the GTZAN dataset, which is referred to as the MNIST dataset for music [8]. This dataset is suitable because it contains 10 genres, each represented by 100 tracks [8]. The tracks are all 22050 Hz Mono 16-bit audio files in .wav format [8]. The dataset has various genres such as ‘rock’, ‘jazz’, and ‘classical’, among others [8]. This allows for easy splitting of the data in order to train our model. The presence of a variety of genres could potentially be useful when classifying songs that may fit more than one genre. An alternative dataset that was considered was the Million Song Dataset (MSD), however, we decided against this dataset because it did not contain any audio files, but instead contained the metadata of the songs in the set [9]. The file format was also in HDF5 which is slightly difficult to work with [9].

1. Data Preprocessing:

The data comes in audio form as .wav files. This is not suitable to be input into a model such as a Convolutional Neural Network (CNN). The means of addressing this issue would be to create visual representations of the audio files [10]. This can be done by using spectrograms [10]. Spectrograms are visual representations of the spectrum of frequencies of a signal as it varies with time [10]. This would involve applying a Fourier transform to the raw audio data, then dividing it into small time frames and applying a windowing function to each frame [11]. The resulting spectrogram images can be used as inputs to a CNN. This can be done using PyTorch audio transform methods [12].

1. Augmentation and Normalization:

The dataset can be augmented to further extend the data used to train the model. This can be done by using several methods, including adding background noise, or changing the pitch. This is done in order for the training to be extended, improving the accuracy of the model. For example, the data can be split into 700 samples for training, 150 samples for validation and 150 samples for testing. The training set can then be augmented by changing a specific property of the audio files as previously indicated. After this, the data should be normalized to have zero mean and unit variance, which helps to improve the stability during the training process [13].

After performing these operations, the data is ready to be fed into the model and then the next steps would be to test its accuracy and grade it accordingly. The goal is to be able to test the model and have it perform well using data generated by the team. This would be in the form of recording one another playing an instrument or possibly using songs from one of our personal playlists. The model should still maintain high accuracy when fed data that it has never encountered before.

1. **Architecture**

For the purposes of this project, the team will be building a music genre classification model using a convolutional neural network (CNN). CNNs are types of artificial neural networks that contain convolutional layers [14]. Each layer contains a set of filters which transforms (convolves) the data such that a prediction can be made as to what the data is depicting [14]. CNNs are commonly used for image classification due to their ability to detect patterns [14]. This pattern recognition shows promise for audio files, as the network’s detection capabilities could be used to detect rhythm, pitch, and other aspects of music [14]. In order for the CNN to use the data, the audio files will need to be converted to images in the form of spectrograms as described in the previous section [15]. An example of a spectrogram of Blues music is provided below.

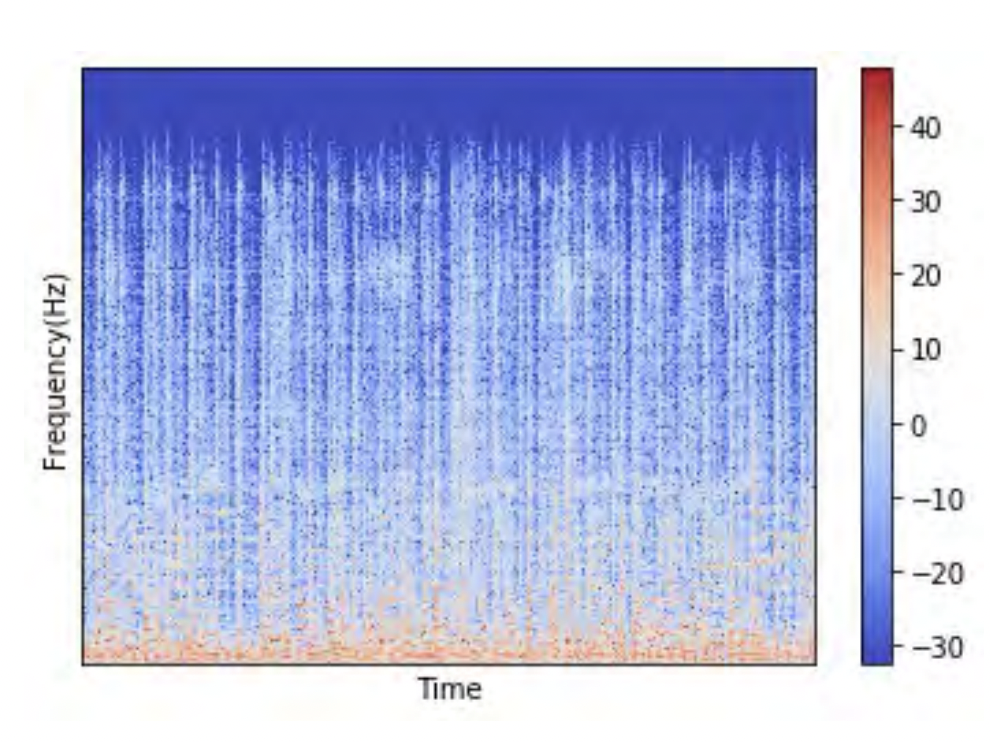


Figure 2: An example of the spectrogram from the Blues genre. [15]

CNNs have been used frequently in the past for other music classification projects [15]. For example, a study at Gazi University in Turkey used GTZAN data to train a CNN model, obtaining a testing accuracy rate of over 90% [15]. Further, a group of researchers in the Faculty of Economics at the Institute of Information Economics and Marketing (IISM) in Germany, were able to carry out the same task with the same data, with relatively similar results [15]. Research projects such as these indicate the viability of the chosen model.

After an audio file has been converted to an spectrogram image, the following CNN process will be implemented. The input image will first be transformed with convolution layers, where filters are shifted over the input image to obtain feature maps [15]. These feature maps contain pixel values representing the image with their specified filters [15]. Second, pooling layers will be applied to reduce the size of the feature maps and reduce the number of parameters, however at the same time, preserve the most important features [15]. Third, the fully connected layer will be applied to obtain the final feature vectors by specifying the number of neurons in the last layer with respect to the number of classes to be classified [15]. A simplified diagram of the process can be seen in Figure 3 [15].

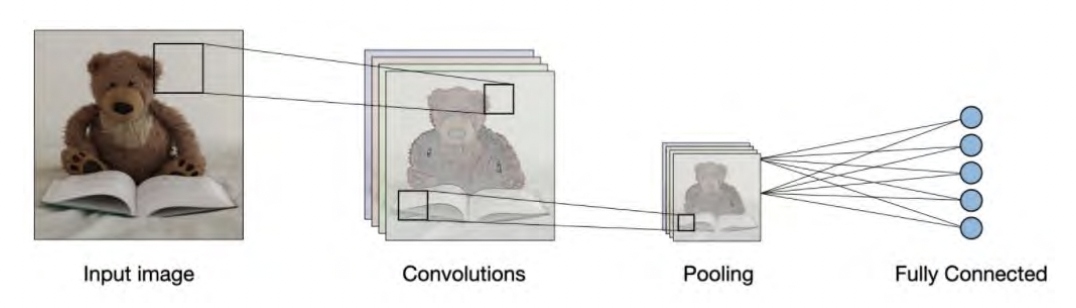
****

Figure 3: A CNN structure layers [15]

In our model, different parameters will be investigated to optimize the results. For example, for the convolution layers, the team can explore the filter numbers, kernel size, activation function, and padding. For max pooling, the team can explore the pool size, strides, and padding. In addition, with regards to other hyperparameters, the team can investigate epochs, batch size, and learning rate, among others.

1. **Baseline Model**

The goal of this project is to perform a classification task. As such, a baseline model by which the proposed model can be compared would be a model utilizing a K-Nearest Neighbour algorithm (KNN). KNN is a regression and classification algorithm that sorts data into categories [16]. The theory behind the algorithm is that if a sample has the same characteristics as the majority of samples within a certain class, the sample must be part of that class [17]. This algorithm has been used to assess the capabilities of other music classification models [7].

The algorithm is first initialized by plotting training data points based on specific properties as denoted by plot axes (i.e., pitch rhythm, repetition, etc.) [18]. These training points are labeled based on their classification and should form clusters by class on the plot [18]. When using the algorithm, an unclassified data point is introduced to the model [18]. The new data point is classified based on the training group with the highest representation among *k* nearest training samples [17]. If more than one class has the same proportion of representation, the class with training points having the minimum average distance from the new data point is chosen [17].

Detailed steps to reproduce this KNN models are outlined below for any reproduced purpose: [19]

1. Import Required Libraries (np, pd, wav, mfcc, TemporaryFile, os, math, pickle, random, and operator). [19]
2. Define a function that calculates distance between feature vectors and find neighbours. To elaborate further, this function should find each point’s distance with another point in the training data set, find all the nearest *k* neighbours, and return all neighbours to calculate the distance of two points. [19]
3. Define a function to identify the class that has the maximum neighbours count. Therefore, we can store the class and its respective count of neighbours. [19]
4. Define a function to evaluate the model and check accuracy and performance of the algorithms. [19]
5. Conduct feature extraction including high-level features such as chords, rhythms, melody; mid-level features such as beat level attributes, pitch-like fluctuation patterns, and MFCCs; low-level features such as energy and a zero-crossing rate. Note: MFCC helps to extract mid-level and low-level features [19].
6. Clean up samples including dividing audio files to 20-40 ms long, separate linguistic frequencies from the noise through discrete cosine transform (DCT) of the frequencies. [19]
7. Train the classifier through the use of a training data set, a validation data set, and a testing data set.

The creators of these guidelines developed a KNN model that has achieved approximately 70% of the accuracy [19]. The team will use the code as inspiration while developing our baseline model.

1. **Ethical Considerations**

The ethical considerations in data collection and impact of using the model are as follows:

1. Informed Consent:

Informed consent is a key factor because the dataset contains audio data that was created by an individual. As such, it is important to ensure that the creators are informed about their music being used to compile a dataset, how their music will be used, and obtain permission to use their work. This is more to prevent intellectual theft and copyright infringements. This also implies us as collectors being transparent with our data collection means and making the collected data replicable.

1. Bias and Fairness:

It is imperative that the dataset be representative of a variety of genres and styles to avoid bias. Selection of music should be based on a clear and objective criteria, not personal biases or preferences. A consequence that should be avoided is publishing a false classification as this could harm the sales of the artists whose music is processed by the model.

1. Privacy:

The music creators or rights holders should be assured that their personal information, such as their name and contact information, will not be shared or disclosed without their consent. They should also be informed about how their music will be stored, secured, and used to ensure their privacy is protected.

1. **Project Plan**

The team has devised a plan by which we will follow in order to complete the project in a timely manner. The team plan includes a meeting schedule, modes of communication, code overwriting prevention planning, an outline of major deliverables, and a task breakdown.

1. Meetings

The team will meet **weekly from 6:00 - 7:00 pm on Monday**. This meeting can either be virtual or in-person depending on the assigned tasks. This meeting time will be used to provide any weekly updates from each team member, track project progress, distribute work, and discuss any difficulties with assigned tasks. If additional meetings are required, the team will schedule them based on everyone’s availability using the When2Meet platform.

1. Mode of Communication

The team will communicate predominantly through a WhatsApp group chat. This ensures if team members have any questions or concerns outside the meeting time, other team members can be reached directly.

1. Code Collaboration

The team will use a shared Google Colab file to prevent overwriting each other's code. In addition, clear tasks and expectations will be outlined to prevent overlapping in responsibilities.

1. Official Deliverables, Deadlines and Internal Deadlines

The following table outlines the major deliverables for this project.

Table X. List of Project deliverables and their deadlines

| Project Deliverables | Deadlines | Internal Deadlines | Descriptions |
| --- | --- | --- | --- |
| Project Proposal | Friday,  Feb 17 | Thursday, Feb 16 | The deliverable demonstrates the initiation of the project and covers the goals, motivation, training dataset plan, rough idea for the neural network, related work, measures of success, and team charter. |
| Project Progress Report | Friday, March 17 | Thursday, March 16 | This deliverable demonstrates the team is on track on the project. At this point, the team should have collected all required data, produced a baseline model, and produced at least one result from training the neural network model. |
| Project Presentation | Friday,  April 7 | Wednesday, April 5 | This deliverable requires the creation of a final video presentation of at most 7 minutes in length covering the project goals, the deep learning model, and a live demonstration of the model results. |
| Project Final Report | Friday,  April 14 | Wednesday, April 12 | This deliverable covers the results of the project and contains an addendum that covers each team member’s personal contribution to the project. More details will be provided by the teaching team. |

1. Team Tasks Internal Deadline

This section breaks down the major deliverables as outlined in the previous section into smaller, more specific tasks.

Table X. Task Breakdown

| Tasks | Internal Deadlines | Descriptions |
| --- | --- | --- |
| Data Collection | Feb 27th | Data with different categories of labeled audio are collected in the .wav format. |
| Data Processing | Feb 27th | Audio files are transformed into spectrograms using fourier transform theory. |
| KNN Baseline Model | Mar 11th | The KNN baseline model is complete. |
| KNN Baseline Results | Mar 12th | Sample data is fed to KNN baseline model and at least 2 results are produced. |
| CNN - Feature Extraction | Mar 5th | CNN model has a framework with functioning feature extraction tools. |
| CNN - Filter Generation | Mar 11th | CNN model has the filter generation components including weights, kernel size, layers (convolutions/ pooling), epochs, batch size, and learning rate integrated. |
| CNN Model Results | Mar 12th | Sample data is fed to the CNN model and at least 2 results are produced. |
| Progress Report | Mar 16th | All the respective sections for progress report are completed including discussions related to data collection, data processing, KNN baseline model, CNN model, results comparison, next steps, etc. Additional sections can be referred to the progress report guideline. |
| CNN Model Training | Mar 27th | CNN model is adjusted to optimize hyperparameters by training with testing and validation data set. |
| CNN Model Testing | Mar 31st | The accuracy of our CNN model is evaluated against a testing data set. |
| Final Presentation - Slides | April 4 th | All the respective sections for the final presentation are completed and put on the slides. Full instructions and required sections can be seen in the final presentation guidelines. |
| Final Presentation - Rehearsal | April 5th | The team completes the rehearsal for the project and makes any required final edits. |
| Final Report | April 12th | All the respective sections for the final report are completed. Full instructions and section requirements can be seen from the final report guidelines. |

1. Team Members Specific Tasks for Project Deliverables

The following table outlines the task distribution among all team members

| Team Members | Tasks | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| Project Proposal | | Project Progress Report | | Project Presentation & Final Report | |
| Alex | * Data Processing * Ethical Considerations * Research | | * Collect the data * Process data to spectrograms * Report Discussion with data collection and spectrograms | | * Produce all data results * Adjust hyperparameters for accuracy * Test the accuracy of the CNN model * Report writing * Presentation preparation | |
| Anna | * Background & Related Work * Architecture * Baseline Model * Edit Document Coherence * Research | | * Producing the baseline model * Produce at least 2 results from the baseline model * Report Discussion with the baseline model | | * Produce all data results * Adjust hyperparameters for accuracy * Test the accuracy of the baseline model * Report writing * Presentation preparation | |
| Blythe | * Illustration   /Figure   * Project Plan * Baseline Model * Edit Document Coherence * Research | | * Producing the CNN model * Produce at least 2 results from the CNN model * Report Discussion with the CNN model | | * Produce all data results * Adjust hyperparameters for accuracy * Test the accuracy of the CNN model * Report writing * Presentation preparation | |
| Tyler | * Introduction * Risk Register * Abstract * Research | | * Producing the CNN model * Produce at least 2 results from the CNN model * Report Discussion with the baseline model | | * Produce all data results * Adjust hyperparameters for accuracy * Test the accuracy of the CNN model * Report writing * Presentation preparation | |

1. **Risk Register**

The following section outlines the potential risks the team may face throughout the course of this project.

1. Biased Data:

**Description:** The collected data could come from the same source of genre classifier to train and validate the model. Therefore the model could be biased towards a specific method and taste of classifying music genres.

**Likelihood:** Low

**Contingency:** Using a set of diverse datasets, the music is more likely to have been classified by numerous sources as well as cover a wider variety of genres.

1. Overfitting:

**Description:** If the model performs exceptionally well on the training set, but poorly on the validation or testing set, then the model is overfitted. Thus the model has not learned how to classify new data since its memorized patterns specific to the training set [20].

**Likelihood:** High

**Contingency:** The model can apply cross validation, reducing the number of epochs, and regularization to prevent overfitting [20].

1. Accuracy:

**Description:** If the model is not optimized and is inaccurate, the music genres will be classified incorrectly and user satisfaction will decrease.

**Likelihood:** Moderate

**Contingency**: Applying a large, accurate data set to train the model on will allow the model to learn more. In addition, parameters should be optimized for the lowest validation error and loss [21].

1. Copyright

**Description:** Larger online data sets may be limited to music that is free from copyright which does not accurately represent the most relevant music.

**Likelihood:** Moderate

**Contingency:** The team can obtain data using an account to legally download music from streaming sites, and ensure the music data will be only used for research and not distribution purposes.

1. Rushed or Unfinished work

**Description:** If the team starts work late, there is a possibility of missed deadlines, and incomplete/unedited sections as the project will be rushed. This will result in a poorly constructed and documented neural network.

**Likelihood**: Unlikely

**Contingency**: The team will set many internal deadlines and check-ins to ensure work is completed gradually. This will give time to ask questions to the TAs or Instructors, edit work, and propose new ideas.

**X.**  **Project Link**

The model will be developed using Google Colab and will be accessible through the following link:

**XI. Conclusion**

Due to personal interest in the topic, as well as the range of potential applications for both data organization, and recommender systems, our team has decided to develop a machine learning model that can classify music by genre. The model will be developed using CNN, and will be compared to a baseline model made using KNN. The data will be collected as audio files and will be converted to spectrogram form for processing. The project plan has been developed, and accounts for contingency plans in the event of issues.

**XII. References**

[1] L. K. Puppala, S. S. R. Muvva, S. R. Chinige, and P. Selvi. Rajendran, “A Novel Music Genre Classification Using Convolutional Neural Network,” *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, Jul. 2021, doi: https://doi.org/10.1109/icces51350.2021.9489022.

**‌**

[2] Adiyansjah, A. A. S. Gunawan, and D. Suhartono, “Music Recommender System Based on Genre using Convolutional Recurrent Neural Networks,” *Procedia Computer Science*, vol. 157, pp. 99–109, 2019, doi: https://doi.org/10.1016/j.procs.2019.08.146.

‌

[3] “Machine Learning,” *Spotify Research*. https://research.atspotify.com/machine-learning/ (accessed Feb. 17, 2023).

‌[4] K. Vaidya, “Music Genre Recognition using Convolutional Neural Networks (CNN) — Part 1,” *Medium*, Dec. 09, 2020. https://towardsdatascience.com/music-genre-recognition-using-convolutional-neural-networks-cnn-part-1-212c6b93da76

‌

[5] K. Kosina, “MUSIC GENRE RECOGNITION,” *music.mcgill.ca*, 2002. http://www.music.mcgill.ca/~ich/classes/mumt611\_07/similarity/KosinaMusicGenreRec.pdf (accessed Feb. 17, 2023).

‌

[6] Q. He, “A Music Genre Classification Method Based on Deep Learning,” *Mathematical Problems in Engineering*, vol. 2022, no. 9668018, pp. 1–9, Mar. 2022, doi: https://doi.org/10.1155/2022/9668018.

‌

[7] C. Xu, N. C. Maddage, X. Shao, F. Cao, and Q. Tian, “Musical genre classification using support vector machines,” in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03).*, Apr. 2003, vol. 5, p. V–429. doi: 10.1109/ICASSP.2003.1199998.

‌

[8] “gtzan | TensorFlow Datasets,” *TensorFlow*. https://www.tensorflow.org/datasets/catalog/gtzan (accessed Feb. 17, 2023).

‌

[9] K. Choi, “Datasets — Music Classification: Beyond Supervised Learning, Towards Real-world Applications,” *music-classification.github.io*, 2021. https://music-classification.github.io/tutorial/part2\_basics/dataset.html#million-song-dataset-2011 (accessed Feb. 17, 2023).

‌

[10] K. Bharath, “Audio Classification with Deep Learning,” *Paperspace Blog*, Jul. 01, 2022. https://blog.paperspace.com/audio-classification-with-deep-learning/ (accessed Feb. 17, 2023).

‌

[11] M. Muller, S. Balke, and F. Zalkow, “Short-Time Fourier Transform and Chroma Features,” *www.audiolabs-erlangen.de*, Apr. 2020. https://www.audiolabs-erlangen.de/content/05-fau/professor/00-mueller/02-teaching/2020s\_apl/1\_stft/stft\_and\_chroma\_features.html (accessed Feb. 17, 2023).

‌

[12] “torchaudio.transforms — Torchaudio 0.13.1 documentation,” *pytorch.org*. https://pytorch.org/audio/stable/transforms.html (accessed Feb. 17, 2023).

‌

[13] M. Riva, “Batch Normalization in Convolutional Neural Networks | Baeldung on Computer Science,” *Baeldung*, Oct. 29, 2020. Accessed: Feb. 17, 2023. [Online]. Available: https://www.baeldung.com/cs/batch-normalization-cnn#:~:text=Batch%20Norm%20is%20a%20normalization

‌

[14] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a Convolutional Neural Network,” in *2017 International Conference on Engineering and Technology (ICET)*, Antalya, Turkey, Aug. 2017, pp. 1–6. doi: https://doi.org/10.1109/icengtechnol.2017.8308186.

‌

[15] H. C. Ceylan, N. Hardalaç, A. C. Kara, and F. Hardalaç, “Automatic Music Genre Classification and Its Relation with Music Education,” *World Journal of Education*, vol. 11, no. 2, p. 36, Apr. 2021, doi: https://doi.org/10.5430/wje.v11n2p36.

‌

[16] B. Mahesh, “Machine Learning Algorithms - A Review,” *International Journal of Science and Research (IJSR)*, vol. 9, no. 1, 2020, Accessed: Feb. 17, 2023. [Online]. Available: https://www.researchgate.net/profile/Batta-Mahesh/publication/344717762\_Machine\_Learning\_Algorithms\_-A\_Review/links/5f8b2365299bf1b53e2d243a/Machine-Learning-Algorithms-A-Review.pdf?eid=5082902844932096

‌

[17] A. Kataria and M. Singh, “A Review of Data Classification Using K-Nearest Neighbour Algorithm,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 6, pp. 354–360, Jun. 2013, Accessed: Feb. 17, 2023. [Online]. Available: https://www.researchgate.net/profile/Aman-Kataria/publication/353306410\_A\_Review\_of\_Data\_Classification\_Using\_K-Nearest\_Neighbour\_Algorithm/links/60f28968fb568a7098b61001/A-Review-of-Data-Classification-Using-K-Nearest-Neighbour-Algorithm.pdf

[18]O. Harrison, “Machine Learning Basics with the K-Nearest Neighbors Algorithm,” *Medium*, Sep. 10, 2018. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

‌

‌‌[19] R. Agrawal, “Music Genre Classification Project Using Machine Learning Techniques,” *Analytics Vidhya*, Mar. 21, 2022. https://www.analyticsvidhya.com/blog/2022/03/music-genre-classification-project-using-machine-learning-techniques/ (accessed Feb. 17, 2023).

‌

[20] P. Baheti, “What is Overfitting in Deep Learning and How to Avoid It,” *www.v7labs.com*, Feb. 02, 2023. https://www.v7labs.com/blog/overfitting (accessed Feb. 17, 2023).

‌

[21] J. Dsouza, “How to Improve the Accuracy of Your Image Recognition Models,” *freeCodeCamp.org*, Nov. 29, 2021. https://www.freecodecamp.org/news/improve-image-recognition-model-accuracy-with-these-hacks/#:~:text=Increase%20Epochs (accessed Feb. 17, 2023).

**Attribution Table**

Key:

* RS - Research
* WD - Wrote Draft
* MR - Major Revision
* ET - Edited
* FP - Final Proofread

| Section | Name | | | |
| --- | --- | --- | --- | --- |
| Abstract | Blythe Huang | Oluwasina Olowookere | Anna Szatan | Tyler Wong |
| Introduction |  |  | RS, ET |  |
| Illustration/Figure | WD, MR |  |  |  |
| Background & Related Work | ET |  | RS, WD, ET |  |
| Architecture | RS, MR, ET |  | RS, WD, ET |  |
| Baseline Model | RS, WD, ET |  | RS, WD, ET |  |
| Ethics |  |  |  |  |
| Project Plan | WD, MR, ET |  | ET |  |
| Risk Register |  |  | ET |  |
| Project Link |  |  |  |  |
| Conclusion |  |  | WD, ET |  |
| References |  |  | WD |  |
| LaTeX | ET |  | ET |  |
| Final Proofread | FP |  | FP |  |

‌