Comparison of Support Vector Machines, Neural Networks, and Naive Bayes for Classifying Audio Snippets Within Music Genres

Ethan Dibble, Larry Bilodeau, Godwin Ferguson Achaab, Ivan Lopez

August 2024

Abstract – We attempt to classify audio snippets within the music genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, rock. The GTZAN dataset provides 100 samples each per the 10 genres. We extract mel frequency capstone coeficients from the data for use in a Concurrent Neural Network (CNN) for preprocessing and feature reduction. Afterwards, we will compare the effectiveness of SVM, Neural Network, and Naive Bayes classifiers. We also attempt an Autoencoder as an alternative method of feature compression and extraction. We test the limitations of this approach by seeing how the model attempts to classify songs of multiple genres, live recordings, and new versus old songs. This same workflow may also be extended to other domains such as classification of animal sounds and as such may be of interest to general audio classification tasks.

1. Contributions

- Ethan: Training the CNN for feature extraction, SVM and DNN classifiers. Report writing and editing. Flask app that uses the RBF SVC to classify and generate a spectrogram image for uploaded files.
- Larry: Model Research. Training the Autoencoder for alternative feature extraction and DNN classifier.
- Ferguson: Model research. Large contribution to report writing and editing.
- Ivan: Model Research. Naive Bayes classifier and analysis

2. Motivation

Classifying music genres is an interesting problem in machine learning and audio signal processing. There are theoretical and practical ramifications to being able to reliably classify music excerpts into genres like pop, reggae, rock, hip-hop, jazz, metal, blues, pop, country, disco, and so on. In practical terms, precise genre classification may improve user search and discovery processes, allow for more efficient content labeling for streaming services, and improve music recommendation systems. It offers, in theory, a reliable testing ground for developing audio processing methods and pushing the limits of machine learning models.

The GTZAN dataset, a well-established resource in the field, offers a balanced representation of these genres with 100 samples each. By leveraging a Convolutional Neural Network (CNN) for preprocessing and feature reduction, we aim to capture and distill the complex audio features that differentiate genres. CNNs have proven effective in extracting hierarchical features from audio signals, making them well-suited for this task.

We compare the effectiveness of neural networks and support vector machines (SVMs) in classifying the extracted characteristics. This comparison aims to assess these models' advantages and disadvantages in audio categorization, so it's not just an academic exercise. Our focus is on determining how well these models generalize to diverse kinds of audio content, including different genres of music, live recordings with audience noise, and recordings from different eras.

By extending this workflow to other domains, such as animal sound classification or general audio classification tasks, we anticipate contributing to broader advancements in audio analysis. Generalizing these techniques beyond music genres can unlock new applications and insights, making this research valuable for both specialized and general audio classification tasks.

3. GTZAN Dataset

The GTZAN dataset is a widely used benchmark dataset in the field of music genre classification and audio analysis. The genres included in the dataset are: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. Each genre is represented by 100 audio clips, providing a balanced dataset for classification tasks.

The audio clips in the GTZAN dataset are in the WAV format with a sample rate of 22,050 Hz, which is standard for many audio processing applications. This format ensures that the dataset is compatible with various audio analysis tools and techniques.

The dataset's structure facilitates the development and evaluation of music genre classification algorithms. By providing a diverse set of genres with a significant number of samples per genre, the GTZAN dataset allows researchers to assess the performance of their models in distinguishing between various musical styles. Additionally, it enables the exploration of different feature extraction and classification methods, making it a valuable resource for advancing the state of the art in music genre classification.

Despite its utility, the GTZAN dataset has some limitations, including the corruption of jazz.00054.wav, potential issues with label noise and variations in recording quality. Never-

theless, it remains a foundational resource for evaluating and benchmarking audio classification algorithms.

4. Methodology

This project's methodology involves several key audio classification techniques, including CNN preprocessing, SVM classification, Neural Network (Perceptrons) classification, exploring different kernels, and evaluating Naive Bayes. Each step is designed to address specific aspects of the classification task and assess the performance of various models.

4.1. CNN Preprocessing

Convolutional Neural Networks (CNNs) are employed for preprocessing and feature extraction from audio data. This process includes:

- **Data Preparation:** Audio clips are resampled to a consistent sample rate of 22,050 Hz. This ensures that all audio data is uniform and suitable for feature extraction.
- **Feature Extraction:** 40 Mel Frequency Cepstral Coefficients (MFCCs) are extracted from each audio clip. These features are scaled over the timeframes of the audio file and processed through a CNN to capture hierarchical patterns and reduce dimensionality.
- **CNN Architecture:** The CNN consists of several convolutional layers, pooling layers, and dropout layers to effectively learn and extract features from the audio signals.

CNNs are commonly used for image processing for their ability to identify spatially relevant data. Audio files can be represented as a Mel Spectrogram image making audio classification much the same as image classification. MFCC values are calculated using the discrete cosine transform of Mel Log Powers and these make the inputs to the CNN. The CNN is trained using a dense neural network classifier to be able to extract the necessary features.

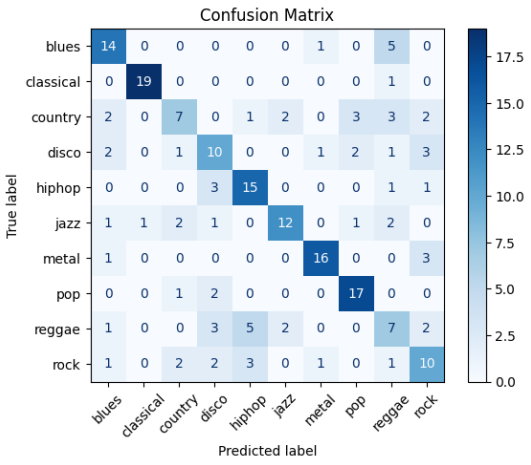

Fig. 1. CNN Feature Extraction; Initial Test Accuracy = 0.635

4.2. Neural Network (Perceptrons) Classification

Neural Networks, specifically Perceptrons, are also employed for classification. This approach includes:

- **Training:** A neural network with one or more layers of perceptrons is trained on the features extracted by the CNN. The network learns to classify audio clips into genres based on the learned features.
- **Evaluation:** The neural network's performance is evaluated similarly to the SVM, using metrics such as accuracy and confusion matrix.

4.3. SVM Classification

Support Vector Machines (SVMs) are used to classify the features extracted by the CNN. The SVM aims to find the optimal hyperplane that separates the data into different genres. Key aspects include:

- **Training:** The SVM is trained using the features obtained from the CNN. Various kernel functions are explored to optimize performance.
- **Evaluation:** The performance of the SVM is assessed using metrics such as accuracy, confusion matrix, and classification report.

4.4. Different Kernels

The SVM classification is tested with different kernel functions to determine the most effective approach for the dataset. Kernels considered include:

- **Linear Kernel:** Assumes a linear relationship between features.
- **Polynomial Kernel:** Captures polynomial relationships between features.
- **Radial Basis Function (RBF) Kernel:** Handles non-linear relationships by mapping features into a higher-dimensional space.
- **Sigmoid Kernel:** Similar to the sigmoid function in logistic regression, handles non-linear and sigmoidal relationships between data points.

4.5. Naive Bayes Classification

Naive Bayes classifiers are also evaluated as an alternative approach. This method assumes feature independence and applies Bayes' theorem for classification:

- **Training:** The Naive Bayes classifier is trained on the same features used for SVM and neural network classification.
- **Evaluation:** Performance is measured using accuracy, confusion matrix, and classification report.

The combined use of these methodologies allows for a comprehensive evaluation of different classification techniques and their effectiveness in music genre classification. Each method contributes to understanding the strengths and limitations of various approaches in handling audio classification tasks.

5. Neural Network

[1] The Neural Network plays a critical role in our approach to classifying music genres within the GTZAN dataset. We employ a deep neural network architecture, specifically lever-

[1] github.com/edibblepdx/bug-free-enigma/tree/main

aging a multi-layer perceptron (MLP) to classify audio features extracted via the CNN preprocessing stage.

a) Architecture:

The architecture of the neural network consists of multiple fully connected layers:

- **Input Layer:** The input layer receives the processed feature vectors, which are the output of the CNN.
- **Hidden Layers:** The network includes several hidden layers, each composed of perceptrons. These layers apply non-linear transformations to the input features, enabling the network to learn and model complex patterns within the data.
- **Output Layer:** The final layer is a softmax output layer that provides probability distributions over the 10 music genres, facilitating multi-class classification.

b) Training Process:

The neural network is trained using the backpropagation algorithm. We use the Adam optimizer, which is known for its efficiency in handling sparse gradients, and categorical cross-entropy as the loss function, which is well-suited for multi-class classification tasks. The model is trained over multiple epochs, with the goal of minimizing the loss function and improving classification accuracy.

c) Overfitting Prevention:

To mitigate the risk of overfitting, which is common in deep neural networks, we employ techniques such as dropout and early stopping. Dropout randomly omits neurons during training, which helps prevent the network from becoming too reliant on specific paths through the network. Early stopping monitors validation performance and halts training when the model's performance ceases to improve.

d) Performance:

The effectiveness of the neural network is evaluated by comparing its accuracy and confusion matrix against other classification methods such as Support Vector Machines (SVM) and Naive Bayes. The network's ability to capture non-linear relationships in the data makes it a powerful tool for this classification task.

e) Advantages:

The neural network's depth and complexity allow it to model intricate relationships within the audio features, making it highly effective for tasks like genre classification, where subtle differences in features can be significant.

f) Challenges:

While powerful, neural networks require careful tuning and significant computational resources, particularly when dealing with large datasets and complex architectures. Additionally, their susceptibility to overfitting necessitates the use of regularization techniques.

5.1. NN Results

We found that the Neural Network performed better on raw features extracted using the python library librosa, rather than those further processed by the CNN. There was an observed accuracy of 69% on the test set when trained on raw features and an accuracy 61.5% when trained on the features processed by the CNN. This was actually lower than the 63.5% initial accuracy on the test set after feature extraction using the CNN. Over the epochs of training, the model also fully overfit on the training set (Figure 3). This could potentially be due to the CNN feature extractor also being trained with a Dense Neural Network classifier to be able to identify the spatially relevant features, which would be used as input into this model. Overall, the Neural Network Classifier Performed worse than expected on this dataset.

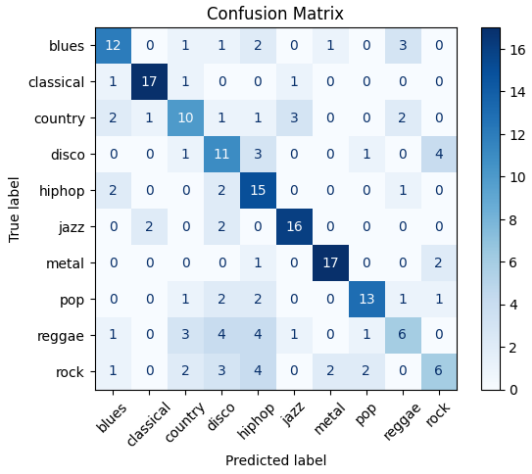

Fig. 2. DNN Kernel; Test Accuracy = 0.615

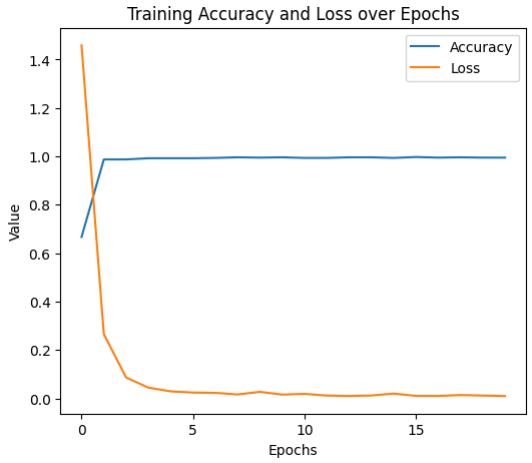

Fig. 3. DNN Training Accuracy and Loss over Epochs

6. Support Vector Machine

[2] The Support Vector Machine (SVM) is a key component in our approach to classifying music genres within the GTZAN dataset. SVMs are powerful, supervised learning models that are particularly effective for classification tasks where the objective is to find the optimal decision boundary between different classes.

[2] github.com/edibblepdx/bug-free-enigma/tree/main

The SVC is a soft-margin SVM that solves the following optimization problem:

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$
$$\text{subject to } y_i \left(w^T \phi(x_i) + b \right) \geq 1 - \zeta_i$$
$$\zeta_i \geq 0, i = 1, \ldots, n$$

a) Feature Extraction:

Before applying SVM, the audio features are extracted using a Convolutional Neural Network (CNN). The CNN reduces the dimensionality of the input data, capturing the most relevant features and thus providing a compact yet informative representation of the audio snippets. These features serve as the input to the SVM classifier.

b) Kernel Trick:

To enhance the SVM's ability to classify non-linear data, we utilize different kernel functions. Kernels transform the input data into higher-dimensional spaces where a linear separator can more easily distinguish between different genres. In this study, we experiment with several kernels:

- **Linear Kernel:** Best for cases where the data is linearly separable and it does not alter the data in any way. The complexity of non-linear data may be lost.
- **RBF (Radial Basis Function) Kernel:** Useful for non-linear classification, as it maps the input features into a higher-dimensional space. The RBF kernel can represent both polynomial and non-linear relationships between data points.
- **Polynomial Kernel:** Captures interactions between features by representing them in polynomial spaces. The polynomial kernel works well with low-dimensional, dense data.
- **Sigmoid Kernel:** Similar to the sigmoid function in logistic regression, useful for comparison with a two-layer perceptron model. The sigmoid kernel can represent non-linear and sigmoidal relationships between data points.

c) Training Process:

The SVM classifier is trained using the features extracted from the CNN. The training process involves identifying the hyperplane that maximizes the margin between different genre classes. SVMs are particularly robust in high-dimensional spaces and can be regularized to prevent overfitting.

d) Parameter Tuning:

To achieve optimal performance, we perform hyperparameter tuning for the SVM. Key parameters include the choice of kernel, the regularization parameter C, which controls the trade-off between achieving a low error on the training data and minimizing the margin, the kernel coefficient γ for non-linear kernels like RBF, and slack variables ζ.

e) Evaluation:

The performance of the SVM classifier is evaluated using metrics such as accuracy and confusion matrix. SVM's strength lies in its ability to find the optimal separating hyperplane, making it well-suited for this multi-class genre classification problem.

f) Advantages:

SVMs are particularly effective in scenarios with high-dimensional feature spaces and are less prone to overfitting when properly regularized. The ability to use different kernels makes SVMs versatile for various types of data distributions.

g) Challenges:

One of the challenges of using SVMs is the computational cost, especially with large datasets and complex kernel functions. Additionally, SVMs require careful tuning of hyperparameters to achieve optimal performance, which can be time-consuming.

The Support Vector Machine (SVM) is a key component in our approach to classifying music genres within the GTZAN dataset. SVMs are powerful, supervised learning models that are particularly effective for classification tasks where the objective is to find the optimal decision boundary between different classes.

h) Feature Extraction:

Before applying SVM, the audio features are extracted using a Convolutional Neural Network (CNN). The CNN reduces the dimensionality of the input data, capturing the most relevant features and thus providing a compact yet informative representation of the audio snippets. These features serve as the input to the SVM classifier.

6.1. SVM Results

The best performing kernel was the Radial Basis Function (RBF) kernel. The worst performing was the polynomial kernel. The polynomial kernel was also more susceptible to biases in classification. Classical music was the most well defined classification overall.

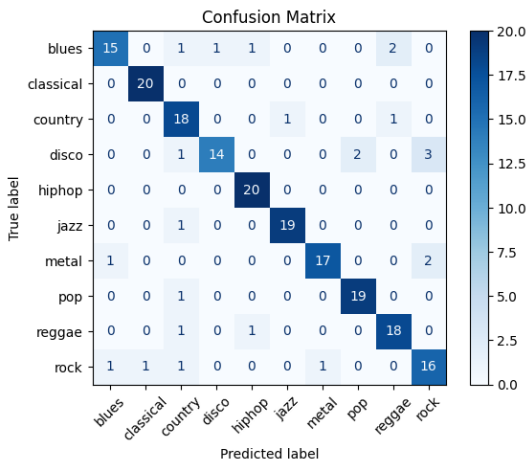

Fig. 4. SVM Linear Kernel; Test Accuracy = 0.88

A test accuracy of 88% using a linear kernel (Figure 4) suggests that the underlying features have some linear separability and that the CNN successfully captured the most important features of the audio files. The greater 91.5% accuracy of the RBF kernel (Figure 5) however, implies that the data still has some complex non-linear relationships.

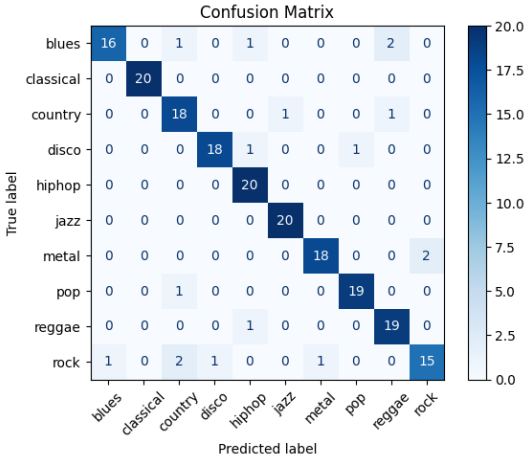

Fig. 5. SVM RBF Kernel; Test Accuracy = 0.915

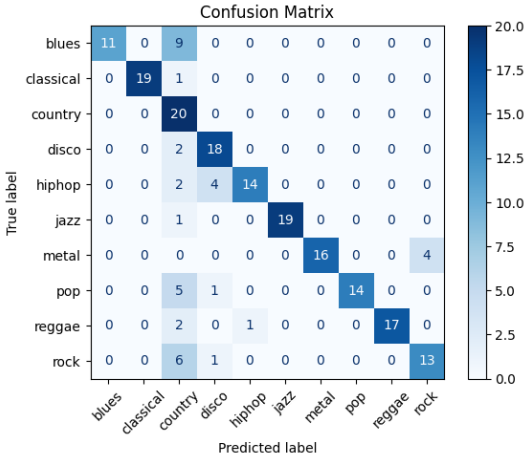

Fig. 6. SVM Polynomial Kernel; Test Accuracy = 0.805

The polynomial kernel (Figure 6) performed the worst with a test accuracy f 80.5%. It also was strongly biased toward country classification. Nevertheless, the trend of classical music being well defined is still present. The poor performance of this model in comparison

The sigmoid kernel (Figure 7) is appropriate for classifying categorical data like music genres. It can also represent non-linear and sigmoidal relationships. The test accuracy of 85% is less than the linear kernel, but suggests some non-linear relationships among the data. It however could not identify those relationships as well as the RBF kernel.

7. Naive Bayes

[3] Naive Bayes classifiers are based on Baye's Theorem. They are statistical models that assume features are independent. They are a key component in classification tasks that require efficiency over being optimal. A Naive Bayes model could be trained and deployed faster than other models and give a fast response to users of that model.

[3] github.com/Lopez9988/ML_Project/tree/main

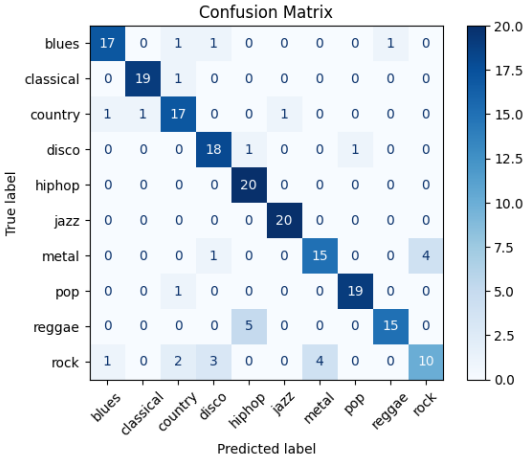

Fig. 7. SVM Sigmoid Kernel; Test Accuracy = 0.85

a) Naive Assumption:

Naive Bayes is naive because it assumes that any pair of features are conditionally independent given the class label. This assumption is rarely true, but a naive bayes model can perform well in practice even when some features are correlated.

b) Advantages:

The naive assumption simplifies the model and makes it more interpretable. They are fast to implement and train in comparison to other many other models and are often significantly less computationally intensive. Predictions are returned quickly efficiently.

c) Challenges:

The naive assumption can oversimplifies the problem and lead to models that don't perform well as they don't capture the dependencies between the data points.

d) Feature Extraction:

Before applying naive bayes, the audio features are extracted using two methods: a Convolutional Neural Network (CNN), and raw feature extraction using the python library librosa. The CNN reduces the dimensionality of the input data, capturing the most relevant features and thus providing a compact yet informative representation of the audio snippets. Librosa allows for the extraction of many different audio features, but these are not preprocessed and may be noisy.

7.1. Naive Bayes Results

The Naive Bayesian model tends to perform better on the features extracted by the CNN compared to the raw feature extraction from the python librosa library. An accuracy of 45% was observed on the raw feature test set and an accuracy of 49% was observed on the CNN feature test set. While the Naive Bayes Classifier indicates a pattern of linear separation of genres (Figure 8), it struggles to classify disco, hiphop, and rock. This could be due to the similarity in the genre music styles causing conflicts between the averages of the extracted features, resulting in misclassification by the Naive Bayes Classifier. Overall, the Naive Bayes Classifier performs worse than the Neural Network and the Support Vector Machine Classifiers.

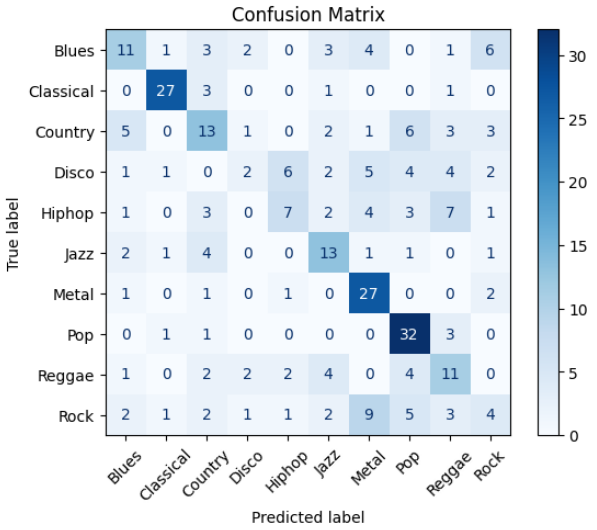

Fig. 8. Naive Bayes; Test Accuracy = 0.49; F1 Score = 0.523

8. Autoencoder

[4] An autoencoder consists of two parts: and encoder and a decoder. The encoding function transforms the input data and the decoder function recreates the data from the encoded output. Autoencoders are a type of unsupervised model and are used to learn efficient encoding of the data. They are often used as generative models but they are key to our research as an alternative method of feature extraction.

a) Advantages:

We can train the autoencoder and only keep the encoder portion of the model. Autoencoders are often simpler than CNNs and they also provide a means of compressing the inputs and reducing features. It can make a model trained on these features as inputs more efficient and space effective.

b) Challenges:

Autoencoders are simpler and more general than CNNs, but CNNs are more more widely used for image recognition which is closely related to audio classification using Mel Spectrograms. Autoencoders may not perform as well as CNNs in the domain of this research.

8.1. Autoencoder Results

The autoencoder tended to overfit the training set. The validation accuracy remained low and the validation loss high. This may suggest an overly complex model. We should try to reduce model parameters or apply regularization techniques such as dropout or early stopping. This may also suggest that an autoencoder is not as sufficient for audio preprocessing compared to a CNN.

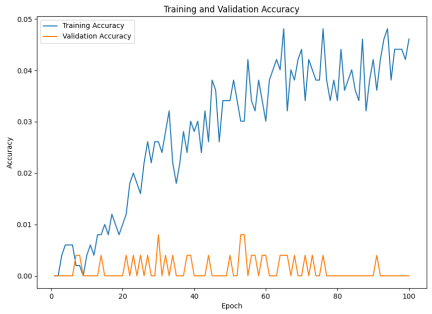

Fig. 9. Autoencoder Accuracy

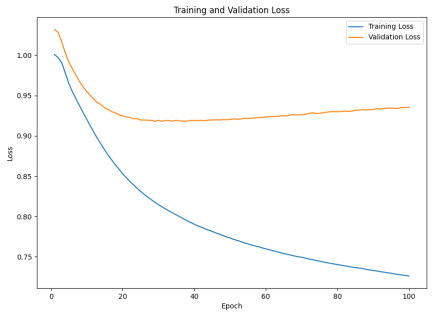

Fig. 10. Autoencoder Loss

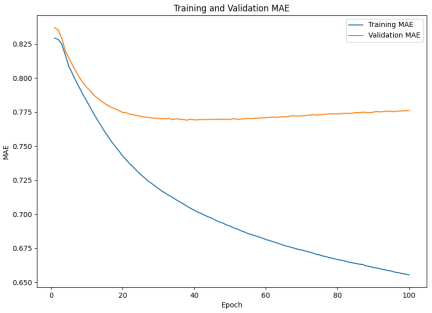

Fig. 11. Autoencoder MAE

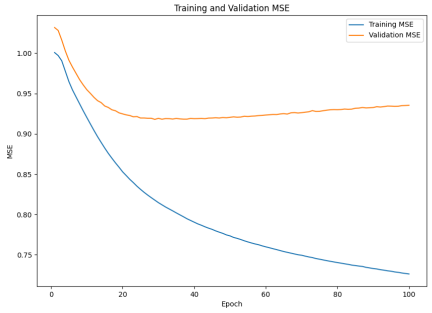

Fig. 12. Autoencoder MSE

[4] github.com/marineGagets/PSU_classes/tree/main/cs445_group_project

9. Comparison of Models

The best model used was the SVC with an RBF Kernel function with a training accuracy of 91.5%. This model was effective in combination with a CNN trained to extract important features from audio files. The other SVC models with different kernels were effective as well, but they failed to identify the complex non-linear relationships that the RBF kernel could.

The DNN model trained using raw features extracted from the python library librosa performed a bit worse than all of the SVM models. It's 69% accuracy still makes it somewhat sufficient for audio classification however. The DNN trained using features extracted with a CNN performed worse that that using raw features. This model significantly overfit the training set and did not generalize well. The accuracy on the test set also dropped below the baseline making use believe that this method not viable for audio classification.

The Worst performing method was Naive Bayes. The test accuracy was 49%; if you picked any some at random your accuracy would be 10%, so there is some independence among the features. The naive assumption however does not lead to a sufficient model for audio classification.

10. Evaluation of Methods

The evaluation of the classification methods used in this study is crucial to understanding their effectiveness in recognizing music genres within the GTZAN dataset. We focus on several key metrics and analyses to assess the performance of the Neural Network, Support Vector Machine (SVM), and other classification approaches.

a) Accuracy:

Accuracy is the primary metric used to evaluate the performance of our models. It is calculated as the ratio of correctly predicted instances to the total number of instances. This metric provides a straightforward indication of how well each model performs in classifying audio snippets into the correct genre.

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$

b) Confusion Matrix:

The confusion matrix offers a more detailed evaluation by displaying the number of correct and incorrect predictions across the various genres. This matrix allows us to identify specific genres that are more challenging for the models to classify correctly. We visualize the confusion matrix to better understand the strengths and weaknesses of each model.

c) Comparison of Models:

We compare the performance of the Neural Network and SVM classifiers directly, analyzing how each model handles the features extracted by the CNN. Additionally, we explore the performance of other models like Naive Bayes to determine which approach best suits the task of genre classification.

d) Limitations and Challenges:

During evaluation, it is important to consider the limitations of each model. For instance, while Neural Networks may excel in handling complex patterns, they require significant computational resources and are prone to overfitting if not properly regularized. On the other hand, SVMs are effective with high-dimensional data but can be computationally expensive with large datasets and complex kernels.

e) Final Thoughts:

The evaluation of these methods provides a comprehensive understanding of their strengths and weaknesses in the context of music genre classification. The insights gained from this analysis are critical for refining the models and improving their performance in future applications.

11. Patterns and Trends

There was a consistent pattern of the genres Classical, Metal, and Pop being the best classified among the 10 genres. The most frequently worst classified was rock. There also appears to be strong similarity between the two genres metal and rock.

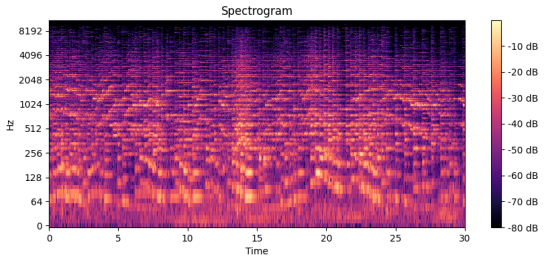

Fig. 13. Classical Sample Spectrogram

Classical (Figure 13) tends to have a low sound intensity at higher frequencies. This attribute is shared somewhat with pop and jazz, but to a lesser extent. Classical appears to be more steady and flatter at higher frequencies. This may have lead to Classical being most commonly the best classified genre.

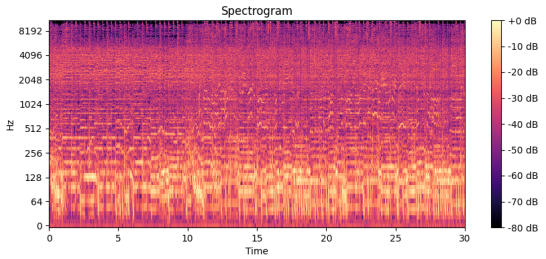

Fig. 14. Metal Sample Spectrogram

Metal (Figure 14) appears to have the highest average sound intensity across all frequencies. It is somewhat ahead of disco in this regard. Rock also had segments in the songs that had a similar shape to Metal which likely lead to metal sometimes being classified as rock.

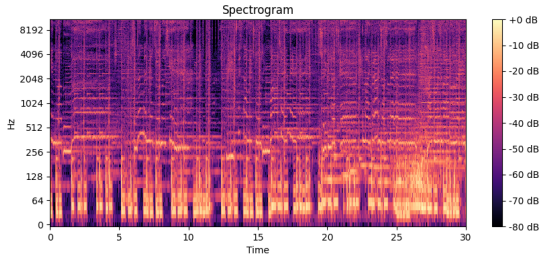

Fig. 15. Pop Sample Spectrogram

Pop (Figure 15) has a very pronounced beat over the time of the song. Hiphop is similar but has a much faster beat than pop. This easy to identify rhythm may have lead to pop being so well classified behind classical.

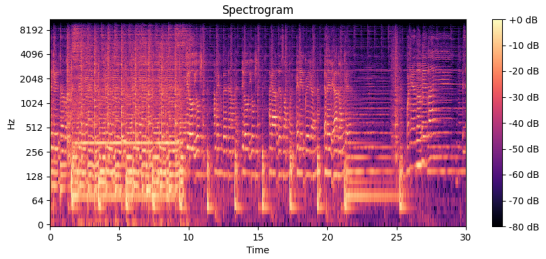

Fig. 16. Rock Sample Spectrogram

The most often worst classified genre was rock (Figure 16). It was also frequently misclassified as metal. And indeed Segments of rock spectrograms appear very similar to the metal genre if you look to the left side of the rock sample spectrogram. It is likely that CNN the model overfit to this feature and learned to extract it from the rock genre leading other models to learn to misclassify the genre. However, it was also observed that while songs were rarely misclassified as rock, when they were it was often metal music suggesting a two-way relationship. There appears to be a strong similarity between the two genres metal and rock.

12. Explored Limitations

Obviously the model is unable to predict genres outside of those that it is trained on, but we should test the limitations of the model by seeing how it attempts to classify songs of multiple genres, subgenress, live recordings with audiences, and new versus old songs. Will a model be able to pick out one of the assigned genres or identify related ones? These tests were ran using the SVC with an RBF Kernel.

We observe that the model has poor performance when attempting to classify subgenres; of the 8 subgenres tested, only 1 was correctly classified within one of its parent genres. Old versus new songs and live recordings did not vary in this regard. We can conclude that the model does not extend well beyond "popular" genres. It had however, often still been able to correctly classify songs that conformed more closely to the genres of that the model was trained on. And it was often the case that misclassifications were between more related genres such as rock and metal, jazz and blues.

Tested Songs:

- *"Walk Away Renee" by The Left Banke 1966* (Baroque Pop: a mixed subgenre of Rock and Pop); misclassified as Country
- *"Dark Was the Night, Cold Was the Ground" by Blind Willie Johnson 1928* (Gospel Blues: a subgenre of Blues); misclassified as Jazz
- *"The Rank Stranger" by Stanley Brothers 1960* (Bluegrass: a subgenre of Country); misclassified as Jazz
- *"What It Is (So So Def Bass allstars)" by Virgo 1996* (Miami bass: a subgenre of Southern Hip Hop and Electronic dance music); misclassified as Pop
- *"Coupe De Ville" by Si Cranstoun 2012* (Boogie Woogie: a subgenre of Jazz, influences of Rock and Blues); misclassified as Disco
- *"Going Down - Live at the Greek Theatre" by Joe Bonamassa 2020* (Blues Rock: a subgenre of Electronic Blues and Rock); correctly classified as Rock
- *"Honey Bucket" by Melvins 1993* (Sludge Metal: a subgenre of Metal, influences of Hardcore punk and Doom metal); misclassified as Blues
- *"Walking On The Moon" by The Police 1979* (Reggae Rock: a subgenre of Reggae and Rock); misclassified as Country

References

[1] Andrada, 2020, "GTZAN Dataset - Music Genre Classification," Kaggle. [Online]. Available: www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification
[2] Mostafa Ibrahim. "An Introduction to Audio Classification with Keras." Weights & Biases. Accessed: Aug. 3, 2024. [Online]. Available: wandb.ai/mostafaibrahim17/ml-articles/reports/An-Introduction-to-Audio-Classification-with-Keras–Vmlldzo0MDQzNDUy
[3] Geek. "How to Choose the Best Kernel Function for SVMs." Geeks for Geeks. Accessed: Aug. 17, 2024. [Online]. Available: https://www.geeksforgeeks.org/how-to-choose-the-best-kernel-function-for-svms/https://www.geeksforgeeks.org/how-to-choose-the-best-kernel-function-for-svms/
[4] S. Alipek, M. Maelzer, J. Moll. "Bat Echolocation Call Analysis with Deep Learning Models." Mendeley Data. [Online] Available: https://data.mendeley.com/datasets/9x2g6dsbtv/1
[5] S. Alipek, M. Maelzer, Y. Paumen, H. Schauer-Weisshahn, J. Moll. "An Efficient Neural Network Design Incorporating Autoencoders for the Classification of Bat Echolocation Sounds." Animals (Basel). Aug. 2023. [Online] Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10451853/
[6] "Mel spectrogram". MathWorks. [Online] Available: https://www.mathworks.com/help/audio/ref/melspectrogram.html
[7] J. Brownlee. "Gentle Introduction to the Adam Optimization Algorithm for Deep Learning." Machine Learning Mastery. [Online] Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/
[8] TensorFlow. [Online] Available: https://www.tensorflow.org/
[9] Keras. [Online] Available: https://keras.io/
[10] Librosa. [Online] Available: https://librosa.org/
[11] Scikit-Learn. [Online] Available: https://scikit-learn.org/