What’s in the noise? Advancing Musical Genre Classification using Neural Networks

Group Members: Katherine Xu - katherinewxu, Tamish Pulappadi - tamish01, Krystal Li - krysli, Eliska Peacock - eliska

Abstract

Music genres have demonstrated increasing fluidity, and their categorization demonstrates increasing complexity. To navigate this diverse realm, we explore several machine learning techniques in the creation of an innovative classification model for music genres. Utilizing the GTZAN Genre Collection dataset, we preprocess audio segments into mel spectrograms and employ a VGG neural network architecture. Through experimentation, we optimize hyperparameters such as learning rate and batch size to enhance model performance. Additionally, we implement data augmentation strategies to improve robustness. Our proposed model indicates potential to improve facilitation of tailored music recommendations and efficient organization of music libraries.

Introduction

The modern landscape of music is marked by a rich tapestry of genres, each characterized by its unique blend of musical elements and cultural influences. However, as music continues to evolve and diversify, the traditional boundaries between genres have become increasingly blurred, presenting a significant challenge for music enthusiasts and platforms alike. In light of this, our project aims to tackle the problem of music genre distinction, leveraging machine learning techniques to develop a robust classification model capable of identifying genres from audio files.

The importance of this endeavor lies in its potential to revolutionize the way users engage with music, particularly in the digital era where vast libraries of songs are readily accessible. By accurately classifying music genres, our model can empower music platforms to offer more personalized recommendations and enhance the organization of music libraries, thereby improving the overall user experience. Motivated by the complexities of modern music and the growing demand for tailored music recommendations, we have embarked on this project with a clear vision: to develop a classification model that not only accurately identifies music genres but also adapts to the nuanced characteristics of contemporary music.

The input to our algorithm is an audio file extracted from the GTZAN dataset, containing music samples spanning various genres. We then utilize a VGG (Visual Geometry Group) neural network architecture to process the audio features extracted from the spectrogram representation of the audio file. The neural network learns to analyze the spectral patterns and temporal dynamics inherent in the audio data, ultimately outputting a predicted music genre classification.

Related Work

In addressing the challenge of music genre classification, several approaches have been proposed in existing literature, encompassing a range of technical methods and learning algorithms. In this section, we categorize these approaches based on their methodologies and discuss their strengths, weaknesses, and relevance to our work.

Feature-based methods rely on extracting relevant features from audio signals and using them to train classification models. One prominent example is the work of Tzanetakis and Cook (2002), who introduced a set of audio features including timbral texture, rhythmic content, and pitch content to classify music genres. While feature-based approaches offer interpretability and computational efficiency, they may struggle to capture complex temporal and spectral patterns present in audio signals, limiting their classification accuracy, especially for genres with subtle distinctions.

Deep learning techniques, particularly convolutional neural networks (CNNs), have gained popularity for music genre classification due to their ability to automatically learn hierarchical representations from raw audio data. Salamon and Bello (2016) demonstrated the effectiveness of CNNs for environmental sound classification, highlighting the potential of deep learning in capturing intricate patterns in audio signals. By leveraging CNNs, our work aligns with this approach, aiming to develop a robust classification model capable of handling the complexities of modern music genres. However, these approaches have a high computational cost and a need for large labeled datasets for training, which may pose challenges in practical implementation.

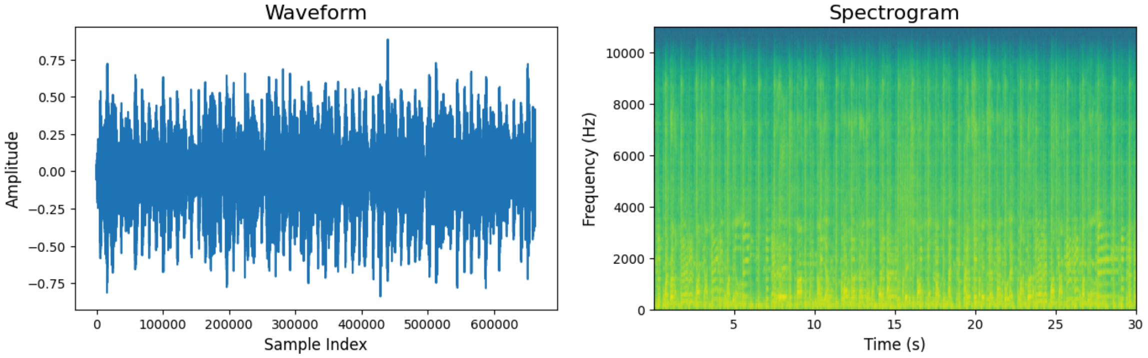
Hybrid approaches combine traditional feature-based methods with deep learning techniques to achieve improved classification performance. For example, Li et al. (2010) proposed a hybrid approach that combines feature extraction with a support vector machine (SVM) classifier, achieving competitive results in music genre classification tasks. By integrating the strengths of both feature-based and deep learning methods, hybrid approaches offer a promising avenue for enhancing classification accuracy while maintaining computational efficiency.

Dataset

Our dataset consists of music samples obtained from the GTZAN Genre Collection, a widely used benchmark dataset for music genre classification tasks. The dataset contains audio files spanning ten different genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock.

We preprocessed the audio data by segmenting each audio file into non-overlapping 2-second segments. For each segment, we compute the mel spectrogram using librosa library functions, which represents the frequency content of the audio signal over time. We used a hop length of 256 samples, an FFT size of 512 samples, and computed 128 Mel frequency bands to generate the spectrogram. Additionally, we augment the dataset by considering 14 different segments from each audio file to increase the diversity of training examples. The resulting dataset consists of tuples containing the mel spectrogram representation of each audio segment along with its corresponding genre label.

Furthermore, we split the dataset into training, validation, and test sets with approximately 10,000, 2,000, and 2,000 examples, respectively. We shuffled the dataset to ensure randomness in the distribution of examples across the splits. The images generated from the spectrogram have a resolution of 64x173 pixels (64 Mel frequency bands, 173 time frames). Each image represents a segment of 2 seconds from the audio file. We apply one-hot encoding to the genre labels to convert them into a categorical format suitable for classification tasks. Below is an example of a waveform and spectrogram visualization for a randomly selected audio file from the blues genre:

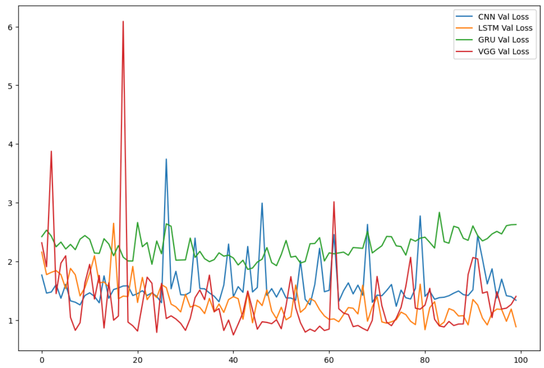


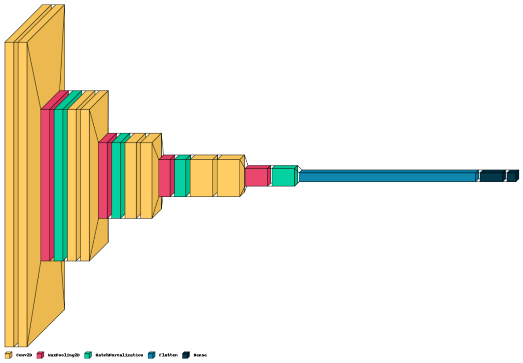
In summary, our dataset comprises preprocessed audio segments represented as mel spectrograms, with corresponding genre labels, obtained from the GTZAN Genre Collection dataset. This dataset serves as the foundation for training, validating, and testing our music genre classification model.

Methods

In our project, we experimented with various machine learning algorithms to determine the most suitable approach for music genre classification. CNNs are widely used for image processing tasks but have also shown efficacy in analyzing audio data. They work by applying convolutional filters over input data to extract hierarchical features, capturing local patterns. In our case, 2D convolutional layers were utilized to process the mel-spectrogram representations of audio files. Initially, we tested four different models: Constructed Generic Convolutional Neural Networks (CNNs), Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Visual Geometry Group (VGG) networks.

1. LSTMs are a type of recurrent neural network (RNN) designed to capture long-term dependencies in sequential data. They use memory cells to retain information over time, enabling them to model temporal relationships in audio sequences. However, LSTMs may face challenges in capturing complex audio features due to their sequential nature.
2. Gated Recurrent Unit (GRU): GRUs are another variant of RNNs similar to LSTMs but with a simpler architecture. They use gating mechanisms to control the flow of information, allowing them to capture temporal dependencies more efficiently. While GRUs are computationally less expensive than LSTMs, they may struggle with modeling long-term dependencies.
3. Visual Geometry Group (VGG) networks: VGG networks are deep CNN architectures known for their simplicity and effectiveness in image classification tasks. In our project, we adapted a VGG architecture to process mel-spectrogram images of audio files, leveraging its ability to learn hierarchical representations of data.





Based on our evaluation results, we found that the VGG network exhibited the best performance in terms of training accuracy for music genre classification. Inspired by the success of VGG architectures in image classification tasks, particularly in the domain of sound classification, we decided to continue with this model for further experimentation and refinement. The model consists of multiple convolutional layers followed by max-pooling layers to downsample the spatial dimensions of the input feature maps while increasing their depth. Batch normalization layers are added to stabilize and accelerate the training process by normalizing the activations of each layer.

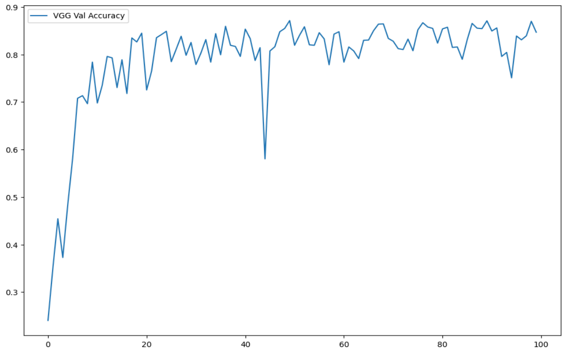
In our implementation, the VGG model comprises several pairs of convolutional layers with increasing filter sizes, followed by max-pooling layers to reduce spatial dimensions. Each convolutional layer is activated using the rectified linear unit (ReLU) activation function to introduce non-linearity into the network. The final layer of the model is a fully connected dense layer with softmax activation, which outputs the probability distribution over the classes. During training, the model minimizes the categorical cross-entropy loss using the Adam optimizer with a learning rate of 0.001.

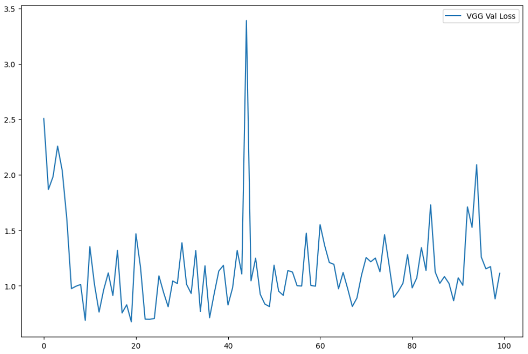
The VGG model learns to classify music genres from mel spectrogram images of audio files by iteratively adjusting its weights through backpropagation. The convolutional layers extract hierarchical features from the input spectrogram representations, capturing patterns indicative of different genres. The subsequent fully connected layers combine these features to make predictions about the genre of the input audio. By optimizing the categorical cross-entropy loss, the model learns to accurately classify music genres based on the learned representations. Through this process, the VGG model leverages its depth and hierarchical structure to effectively model the complex relationships between audio features and genre labels. The below figure visualizes the layers of the model.

Experiments/Results/Discussion

The learning rate is arguably the most crucial hyperparameter in training DLMs. It determines the size of the steps the optimizer takes during gradient descent as it seeks to minimize the loss function. If set too high, the learning rate can cause the model to overshoot the minimum of the loss function, leading to erratic and unstable training behavior. Conversely, if set too low, the model may take too long to converge, or it might get stuck in a local minimum.

When we were experimenting with learning rates for our model, we chose an initial learning rate of 0.001 for its proven effectiveness in various deep learning tasks, ensuring steady convergence without overshooting the minima. To adapt to the changing gradients during training, we applied learning rate optimization methods and settled on the Adam optimizer for optimal results, enhancing the model's ability to fine-tune its weights in the latter stages of training. In audio classification, where the input data and feature spaces can be highly complex and variable, fine-tuning the learning rate proved to be especially beneficial. As seen in the figures, the validation accuracy and loss were both optimized through this training.

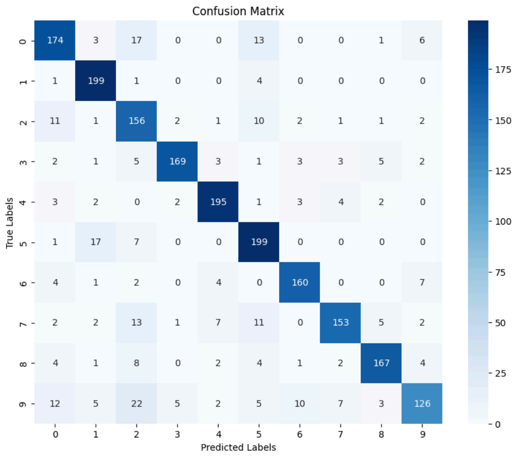




Next in our experiments was batch size optimization. The batch size determines the number of training examples used to estimate the gradient during each optimization step. It has a direct impact on the training dynamics and model performance. In terms of batch size, after several experiments, a size of 64 was selected. This size offered a good compromise between computational efficiency and the model's ability to generalize from the training data, reducing the variance in the gradient estimates. For our audio data, which involved large files and complex preprocessing, finding the right balance for the batch size was crucial. It affected not only the computational efficiency but also how well the model can generalize from the training data to accurately classify unseen audio samples. The validation accuracy and loss were both optimized through this training.

To experiment with improving accuracy metrics we implemented data augmentation on the audio files. This technique is particularly important for audio classification, where we wanted our model to be robust for a wide variety of sounds and recording conditions. For this implementation, we had a three part audio data augmentation pipeline. The first step in the pipeline was adding noise. We generate a noise signal, which is random numbers with the same length as the input audio file, and scale this noise by our noise level. This scaled noise is then added to the original audio signal. Next, we modulated the pitch of the audio signal without changing its duration by first increasing or decreasing the pitch and then time stretching it to return it to its original tempo. Finally, we added a randomized audio data augmentation that combines the two previously mentioned augmentations. It randomly decides whether to add noise and whether to change the pitch and speed of the audio signal, based on a preset probability.

The confusion matrix shows the performance of a classification model on a dataset with 10 classes labeled 0 to 9. The diagonal elements represent correctly classified instances, with the model performing well on classes 0, 1, 3, 4, 5, and 8. Off-diagonal elements indicate misclassifications; for example, 17 instances of class 0 were misclassified as class 2, and 13 as class 5. Class 7 appears to be the most challenging, with relatively low correct predictions and high misclassifications, especially as classes 2 and 5. Class 9 also exhibits some confusion, with instances misclassified as classes 2 and 6. While the model performs reasonably well on several classes, there are notable confusions between certain class pairs.



	0	1	2	3	4	5	6	7	8	9
0	174	3	17	0	0	13	0	0	1	6
1	1	199	1	0	0	4	0	0	0	0
2	11	1	156	2	1	10	2	1	1	2
3	2	1	5	169	3	1	3	3	5	2
4	3	2	0	2	195	1	3	4	2	0
5	1	17	7	0	0	199	0	0	0	0
6	4	1	2	0	4	0	160	0	0	7
7	2	2	13	1	7	11	0	153	5	2
8	4	1	8	0	2	4	1	2	167	4
9	12	5	22	5	2	5	10	7	3	126

Conclusion/Future Work

This paper has showcased the potential of machine learning techniques, particularly deep learning, in addressing the complex task of music genre classification. By leveraging the GTZAN Genre Collection dataset and employing a combination of feature-based methods, deep learning architectures, and data augmentation strategies, our model has demonstrated advancements in accurately discerning music genres from audio files. Through meticulous experimentation and optimization we have achieved promising results, laying the groundwork for enhanced music recommendation systems and digital music library organization. The success of the VGG network can be attributed to its ability to learn hierarchical representations of data and capture intricate patterns present in audio signals. Unlike LSTMs and GRUs, which may struggle with modeling long-term dependencies, the VGG network excels in processing spectrogram images and extracting relevant features for classification. Furthermore, the VGG architecture's simplicity and effectiveness in image classification tasks make it well-suited for music genre classification from spectrogram representations.

The insights from this project support further research and development in the field of music information retrieval. Future endeavors could explore additional data augmentation techniques, refine model architectures, and investigate the integration of user feedback to continuously improve the accuracy and relevance of the systems. Ultimately, the goal is to empower users with personalized music experiences, fostering deeper engagement and enjoyment of the vast world of music in the digital era.

Contributions

Katherine:

- Integrated the GTZAN dataset and preprocessed it for neural networks
- Tested four Convolutional Neural Networks (CNNs) architectures, providing insights for the best to use for genre classification
- Analyzed and plotted model output data for further data analysis
- Managed the project timeline, ensuring team adherence to deadlines and maintaining project momentum

Krystal:

- Established the project's GitHub repository
- Refined the milestone paper: integrating TA feedback, incorporating model performance results, explanation of next steps and potential strategies for advancing the model's capabilities
- Designed & printed project poster, adapted write-up for poster presentation
- Managed the project timeline, ensuring smooth execution and timely project completion

Tamish:

- Ideated different music classification based problems to solve.
- Researched datasets and helped integrate GTZAN dataset for classification project.
- Assisted with music technology knowledge and overall implementation details.

Eliska:

- Edited milestone paper
- Contributed to and edited final writeup

References

Li, Q., Burges, C. J., Downs, T., Li, Y., Platt, J. C., Joachims, T., & Witten, I. H. (2010). A hybrid SVM based decision tree. *Pattern Recognition.* https://www.sciencedirect.com/science/article/pii/S0031320310003067

Salamon, J., & Bello, J. P. (2016). Deep convolutional neural networks and data augmentation for Environmental Sound Classification. *arXiv.org*. https://arxiv.org/abs/1608.04363

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing, 10(5), 293-302.* https://doi.org/10.1109/TSA.2002.800560

Tzanetakis, G., & Cook, P. (2002). GTZAN genre collection. Retrieved from http://opihi.cs.uvic.ca/sound/genres.tar.gz