Application of Machine Learning on Music Genre Classification

Group Member: Ethan Chandra, Qihang Shi

1. Introduction

Music genre classification is a fundamental task in audio analysis, with applications in recommendation systems, music organization, and intelligent media retrieval. Although human listeners can intuitively distinguish genres based on rhythm, timbre, harmony, and instrumentation, enabling machines to do so reliably remains a challenging problem due to the complexity and diversity of musical structures. Recent advances in machine learning provide two major pathways for addressing this challenge: image-based deep learning using spectrograms and feature-based traditional machine learning using engineered audio descriptors.

Understanding the strengths and limitations of these approaches is increasingly important as modern music platforms rely heavily on automated tagging, large-scale content indexing, and personalized recommendation algorithms. With millions of tracks uploaded across streaming services, manual annotation is no longer feasible, and the quality of genre classification directly affects user experience, search relevance, and playlist generation. A more accurate and robust classification system enables platforms to better capture user preferences, discover emerging trends, and deliver meaningful music suggestions. Therefore, studying how different modeling strategies perform on music genre classification tasks offers valuable insights

In this project, our goal is to compare the effectiveness of spectrogram-based convolutional neural networks (CNNs) against traditional machine learning models trained on handcrafted audio features such as MFCCs, spectral centroids, chroma vectors, and temporal descriptors. By designing two independent classification pipelines, one that treats music as an image classification problem and another that represents audio through engineered statistical features. We aim to understand how different representations influence classification accuracy, generalization, and computational efficiency.

2. SOTA

Recent advances in music genre classification have been driven primarily by deep learning models that treat audio as a time–frequency signal. State-of-the-art methods typically convert raw audio into spectrograms or mel-spectrograms and apply convolutional neural networks

(CNNs) to learn discriminative patterns from these representations. Architectures such as VGG-style CNNs, ResNet variants, and attention-based models have achieved strong performance by capturing local rhythmic structures and global harmonic textures within the spectrogram. In parallel, traditional machine-learning pipelines using handcrafted acoustic features combined with classifiers like SVMs, k-NN, and gradient-boosted trees continue to provide competitive baselines with much lower computational cost. Together, these two directions form the basis of modern SOTA systems, with spectrogram-based deep models generally achieving the highest accuracy.

3. Dataset

The dataset used in this project covers 10 distinct music genres, includes blues, classical, country, disco, hiphop, jazz, metal, pop, reggae and rock. Each genre containing 100 samples, for a total 1,000 tracks. Every track is represented in two complementary forms: (1) spectrogram image capture the time-frequency of the audio. (2) a row in a handcrafted feature CSV file containing more than fifty engineered audio descriptors.

3.1 Spectrogram Image Dataset

The primary dataset used in this project consists of pre-generated **mel-spectrogram images**, each representing a musical track as a 2D time–frequency visualization. A spectrogram converts an audio waveform into an image where x-axis represents time, y-axis represents frequency (mel scale) and pixel intensity represents signal energy (Figure 1). These spectrograms capture musical characteristics such as timbre, harmonic structure, rhythm, and frequency transitions. Because CNNs naturally excel at learning patterns from images, spectrograms allow us to treat music classification as an image recognition problem.

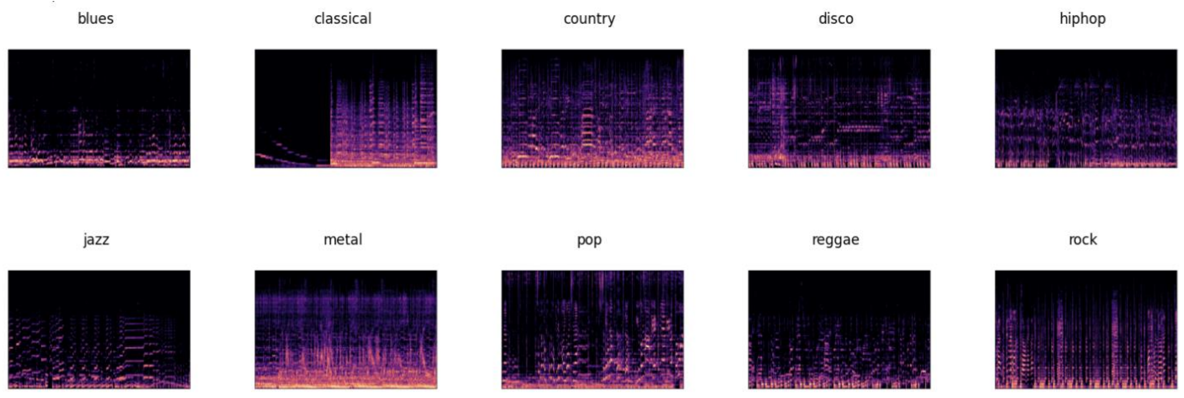
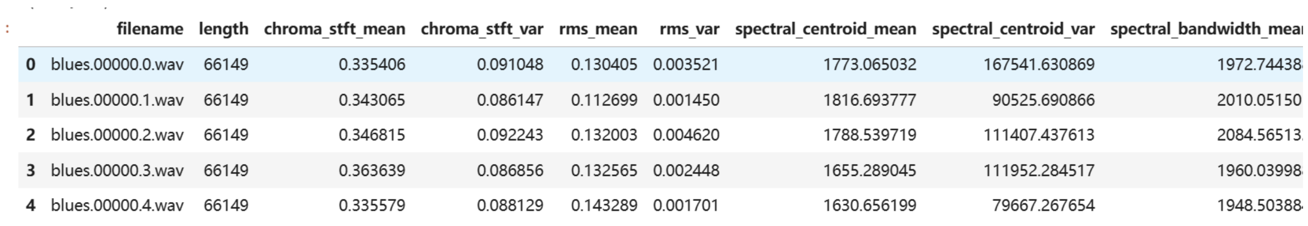


Figure 1: Sample of Spectrogram image for each genre of music in the dataset

3.2 Music feature dataset

In addition to spectrogram images, we also utilize a structured feature dataset extracted from the raw audio files. Each track is represented by a row in a CSV file containing over 50 handcrafted audio descriptors, for example, Temporal Features like RMS energy and zero-cross rate, spectral features like spectral centroid, spectral bandwidth and spectral roll-off. These dataset supports traditional machine learning models such as KNN and MLP by providing explicit numerical representation of musical characteristics. (Figure 2)

As the result, the feature dataset and the spectrogram dataset together enable a two path evaluation framework comparing image based deep learning and feature based ML approaches



	filename	length	chroma_stft_mean	chroma_stft_var	rms_mean	rms_var	spectral_centroid_mean	spectral_centroid_var	spectral_bandwidth_mea
0	blues.00000.0.wav	66149	0.335406	0.091048	0.130405	0.003521	1773.065032	167541.630869	1972.74438
1	blues.00000.1.wav	66149	0.343065	0.086147	0.112699	0.001450	1816.693777	90525.690866	2010.05150
2	blues.00000.2.wav	66149	0.346815	0.092243	0.132003	0.004620	1788.539719	111407.437613	2084.56513
3	blues.00000.3.wav	66149	0.363639	0.086856	0.132565	0.002448	1655.289045	111952.284517	1960.03998
4	blues.00000.4.wav	66149	0.335579	0.088129	0.143289	0.001701	1630.656199	79667.267654	1948.50388

Figure 2: Sample of Music Feature dataset

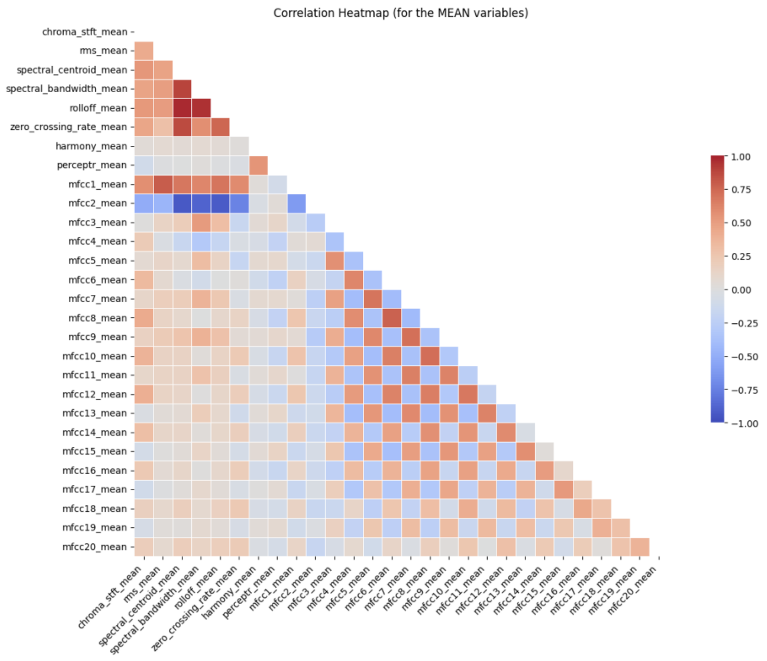


Figure 3: Correlation Heatmap of Music Features

To better understand the structure and redundancy within the engineered audio features, we computed a correlation heatmap using only the mean-valued descriptors (e.g., MFCC means, spectral centroid mean, rolloff mean). The resulting matrix reveals clear patterns of both strong positive and negative correlations, particularly among adjacent MFCC coefficients, which is expected given their sequential representation of spectral envelope information. High correlation clusters indicate feature redundancy, while low-correlation regions suggest complementary information useful for classification. This analysis provides insight into the internal relationships of the handcrafted features.

4. Dataset preprocessing

To prepare the dataset for model training, two separate preprocessing pipelines were designed, one for the spectrogram image dataset and one for the handcrafted feature dataset. Although both originate from the same set of audio tracks, the preprocessing objectives differ: the image preprocessing focuses on standardizing visual representations for CNNs, while the feature preprocessing ensures numerical consistency for traditional machine learning models.

4.1 Spectrogram Image Preprocessing

The preprocessing approach varies significantly across different deep learning architectures due to their distinct training protocols on ImageNet. For the simple CNN model, standard normalization is applied by scaling pixel values to the range [0, 1] through division by 255.0, which provides a baseline preprocessing approach. Transfer learning models, however, require specific preprocessing aligned with their original ImageNet training. ResNet50 utilize caffe-style preprocessing, subtracting channel-wise mean values (103.939, 116.779, and 123.68 for RGB channels respectively) from the original pixel values, resulting in a mean-centered representation with values approximately in the range [-127, 127]. In contrast, EfficientNet employs a two-step normalization process: first scaling pixels to [0, 1], then applying standardization by subtracting ImageNet channel means ([0.485, 0.456, 0.406]) and dividing by standard deviations ([0.229, 0.224, 0.225]), producing values roughly in the range [-2.5, 2.5]. These preprocessing differences are critical for leveraging pre-trained weights effectively, as each model expects inputs in the same format it was trained on; using incorrect preprocessing can result in severe performance degradation, potentially reducing accuracy to random-guess level

4.2 Music Feature Dataset Preprocessing

The second dataset consists of engineered audio descriptors. To prepare these features for machine learning models, irrelevant columns such as filenames and non-informative metadata were removed, and all genre labels were standardized and numerically encoded from 0 to 9. The dataset was then inspected for missing or invalid values, which were corrected or dropped to ensure data integrity. All numerical features were normalized using StandardScaler to prevent features with large numeric ranges from disproportionately influencing distance-based algorithms. Finally, the processed feature matrix was split into training, validation, and testing sets using stratified sampling to maintain balanced class distribution across all subsets.

5. Model

To capture the rich time–frequency patterns captured in mel-spectrograms, we use CNN based architectures. These models learn hierarchical visual features directly from the spectrogram images. In parallel, we build traditional machine learning models using the engineered audio feature dataset. By exploring both deep image-based learning and feature-based classical models, we can compare their performance and gain insights into how different representations of music signals contribute to genre classification.

5.1 CNN Based Model

Baseline CNN Model

The CNN model serves as a baseline architecture for music genre classification from spectrograms. It consists of four convolutional blocks with progressively increasing filter sizes (32, 64, 128, and 256), each followed by batch normalization and max pooling layers for feature extraction and dimensionality reduction. The convolutional layers learn hierarchical features from the spectrogram images, with early layers capturing low-level patterns such as edges and textures, while deeper layers capture more complex, genre-specific features. After feature extraction, the model uses two fully connected dense layers (512 and 256 units) with dropout regularization (0.5 and 0.3) to prevent overfitting, before the final softmax layer outputs predictions for the 10 music genres. This architecture is trained from scratch without pre-trained weights, making it computationally efficient but it requires more data to achieve optimal performance.

VGG 16 model

We employ transfer learning by loading VGG16 pre-trained on ImageNet with approximately 138 million parameters, utilizing its learned feature representations while freezing the convolutional base to prevent overfitting on the relatively small spectrogram dataset. We remove the original fully connected layers and replace them with a custom classification head: a flatten layer to convert the feature maps into a 1D vector, followed by two dense layers (512 and 256 units) with dropout regularization (0.5 and 0.3), culminating in a softmax layer for 10-class genre prediction.

ResNet 50

The architecture consists of 50 layers organized into residual blocks, with each block containing multiple convolutional layers (1×1, 3×3, and 1×1) that learn residual mappings rather than direct mappings, enabling the training of much deeper networks without performance degradation. For this project, we leverage ResNet50 pre-trained on ImageNet (approximately 25 million parameters), freezing the convolutional base and replacing the top layers with a custom classification head that uses global average pooling followed by two dense layers (512 and 256 units) with dropout regularization, and a final softmax layer for 10-

class prediction. ResNet50 requires preprocessing that subtracts ImageNet channel-wise means (103.939, 116.779, 123.68) from input images.

EfficientNet

The base model, EfficientNetB0, contains only 5.3 million parameters, approximately 5 times fewer than ResNet50 and 26 times fewer than VGG16, while maintaining competitive or superior accuracy. The architecture utilizes mobile inverted bottleneck convolution (MBConv) blocks with squeeze-and-excitation optimization, which efficiently captures spatial and channel-wise relationships in feature maps. For this project, we employ EfficientNetB0 pre-trained on ImageNet, freezing the base layers and adding a custom classification head with global average pooling, two dense layers (512 and 256 units) with dropout, and a softmax output for 10 genres. EfficientNet requires a distinct two-step preprocessing: first normalizing pixel values to [0, 1], then standardizing using ImageNet means ([0.485, 0.456, 0.406]) and standard deviations ([0.229, 0.224, 0.225]).

5.2 Traditional Model

Multi-Layer Perceptron (MLP) Model

The MLP architecture we implemented consists of a feedforward neural network with an input layer matching the feature dimension, followed by three fully connected hidden layers (256, 128, and 64 neurons) with ReLU activation functions that learn non-linear relationships between features and genres. Dropout regularization (0.4 and 0.3) is applied after the hidden layers to prevent overfitting, and the final softmax output layer produces probability distributions over the 10 music genres. The model is trained using the Adam optimizer with sparse categorical cross-entropy loss and early stopping (patience=8) to halt training when validation performance peak, ensuring optimal generalization on unseen data.

K-Nearest Neighbors (KNN) Model

We implement KNN using the same extracted audio features as the MLP model. To optimize the model's performance, we employ GridSearchCV with 5-fold cross-validation to systematically search through hyperparameter combinations, testing different values for the number of neighbors (n_neighbors: 3, 5, 7, 9, 11, 15, 21), weighting schemes (uniform or distance-based), distance metrics (Minkowski, Manhattan, Cosine), and the Minkowski power parameter (p: 1 or 2). This comprehensive grid search evaluates 420 candidate configurations to identify the optimal combination that maximizes classification accuracy. The best parameters and corresponding cross-validation accuracy are selected to build the final model, providing a classical machine learning baseline for comparison with deep learning approaches.

<u>XGBoost Model</u>

We implement XGBoost with key hyperparameters including 100 estimators (trees), a learning rate of 0.1 to control the contribution of each tree, a maximum tree depth of 5 to prevent overfitting, and multinomial logistic loss (mlogloss) as the evaluation metric for multi-class classification. XGBoost is particularly well-suited for tabular feature data due to its efficiency, built-in regularization to handle overfitting, and ability to capture complex feature interactions through its boosting framework, making it a strong baseline for comparing traditional machine learning approaches with deep learning methods on handcrafted audio features.

6. Result

Model	Accuracy	Precision	Recall	F1
Baseline CNN	0.11	0.12	0.10	0.02
VGG16	0.65	0.69	0.70	0.68
ResNet 50	0.67	0.71	0.69	0.69
EfficientNet	0.63	0.64	0.62	0.61

Table 1: Result for CNN based model

The experimental results of CNN based model in Table 1 reveal significant performance differences across the four tested architectures for music genre classification from spectrograms. The baseline CNN model achieved extremely poor performance with only 11.88% accuracy, precision, and recall, barely above random guessing (10% for 10 classes), indicating that training a deep network from scratch on this small dataset of 1,000 images is insufficient for learning meaningful genre-specific features. In stark contrast, all three transfer learning models demonstrated substantially superior performance, highlighting the critical importance of leveraging pre-trained ImageNet weights for this task.

ResNet50 emerged as the top performer with 67% accuracy, 71% precision, 69% recall, and 69% F1-score, demonstrating that its residual learning architecture with skip connections effectively captures the hierarchical features present in music spectrograms. VGG16 achieved 65% accuracy with 69.33% precision, 70% recall, and 68% F1-score, performing slightly below ResNet50 despite having significantly more parameters (138M vs 25M). Surprisingly, EfficientNetB0, despite being theoretically superior with its compound scaling method and significantly fewer parameters (5.3M), achieved the lowest performance among transfer learning models with 63% accuracy, 64% precision, 62% recall, and 61% F1-score.

The superior performance of ResNet50 can be attributed to its residual connections that facilitate gradient flow and enable deeper feature learning, which appears particularly beneficial for spectrogram analysis. VGG16's slightly lower but competitive performance demonstrates that its simple, uniform architecture with stacked 3×3 convolutions remains effective at extracting regular, repetitive patterns present in spectrograms. The underperformance of EfficientNet suggests that its architectural innovations, which is compound scaling and mobile inverted bottleneck convolutions designed primarily for natural images and computational efficiency, may not provide the same advantages for spectrogram-based audio classification tasks. These results demonstrate that transfer learning improves accuracy by approximately 6x over the baseline CNN, and that architectural innovations like residual learning are more beneficial than parameter efficiency alone for this specific task.

Model	Accuracy	Precision	Recall	F1
KNN	0.67	0.67	0.68	0.67
MLP	0.72	0.70	0.73	0.70
XGBoost	0.70	0.70	0.70	0.70

Table 2: Result for traditional model

Based on Table 2, the Multi-Layer Perceptron (MLP) achieved the best results among traditional approaches with 72% accuracy, 70% precision, 73% recall, and 70% F1-score, demonstrating that a simple feedforward neural network can effectively learn non-linear relationships between extracted audio features and music genres. XGBoost followed closely with balanced performance across all metrics at 70% accuracy, precision, recall, and F1-score, showcasing the effectiveness of gradient boosting on tabular feature data. K-Nearest Neighbors (KNN) achieved 67% accuracy with 67% precision, 68% recall, and 67% F1-score, performing slightly below the other two methods but still demonstrating reasonable classification capability.

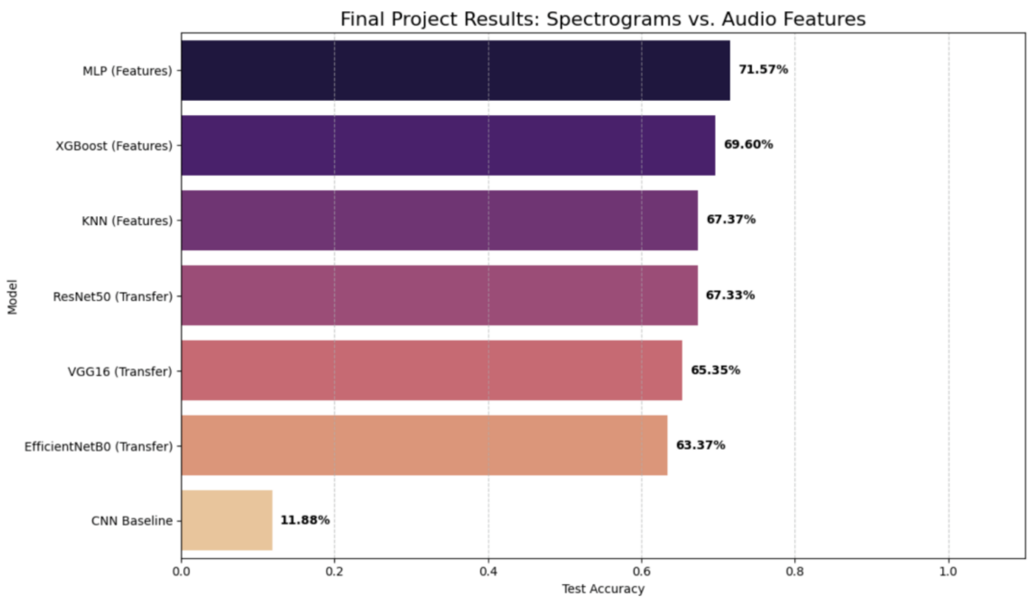


Figure 4: Overall Accuracy Result

Based on the Figure 4, notably, the MLP model (71.57%) outperformed even the best CNN-based approach ResNet50 (67.33%), suggesting that for this specific dataset, carefully engineered audio features combined with a relatively simple neural network architecture can be more effective than complex deep learning models operating on spectrogram images. This result highlights the continued relevance of feature engineering and traditional machine learning methods, particularly when working with small datasets where deep CNN models may struggle to learn robust visual representations.

7. Conclusion

This project systematically evaluated multiple machine learning approaches for music genre classification, comparing traditional feature-based methods with deep learning models operating on spectrograms. The results revealed that both paradigms achieved competitive performance, with the traditional MLP model (72% accuracy) surprisingly outperforming all CNN-based approaches. These findings suggest that for small-scale music classification tasks, handcrafted audio features combined with traditional machine learning methods can be as effective, if not more so, than complex deep learning architectures. The success of transfer learning models highlights the value of pre-trained ImageNet weights, though the superiority of feature-based MLP indicates that domain-specific feature engineering remains highly relevant. For the future work, we could explore hybrid approaches combining both feature extraction and deep learning, larger datasets to better leverage CNN capacity, data augmentation techniques for spectrograms, and ensemble methods combining predictions from multiple models. Additionally, investigating two-stage fine-tuning for transfer learning models and exploring other pre-trained architectures could further improve classification performance.

8. Reference

[1] Dieleman, S., & Schrauwen, B. (2014). End-to-end learning for music audio.

In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 6964-6968).

[2] Piczak, K. J. (2015). Environmental sound classification with convolutional neural networks.

In 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP) (pp. 1-6).

[3] Choi, K., Fazekas, G., Sandler, M., & Cho, K. (2016).

Convolutional recurrent neural networks for music classification. arXiv:1609.04243.

Use of AI

[1] Tool: Chatgpt

Prompt: give me the function for the preprocessing of the dataset for the Resnet50 model

Output:

```
def load_and_preprocess_resnet(path, label):
    img = tf.io.read_file(path)
    img = tf.image.decode_png(img, channels=3)
    img = tf.image.resize(img, [img_size, img_size])
    img = preprocess_input(img)   # ResNet50 specific preprocessing
    return img, label
```

[2] Tool: Chatgpt

Prompt: give me the function for the preprocessing of the dataset for the EfficientNet

Output:

```
def load_and_preprocess_efficientnet(path, label):
    img = tf.io.read_file(path)
    img = tf.image.decode_png(img, channels=3)
    img = tf.image.resize(img, [img_size, img_size])
    img = efficient_preprocess(img)   # EfficientNet specific preprocessing
    return img, label
```

[3] Tool Chatgpt

Prompt: I want to create the correlation heat map of the music feature data, give me the code

Output:

```
import seaborn as sns
```

```
mean_cols = [col for col in df.columns
                        if "mean" in col and col not in ["filename", "label"]]
df_means = df[mean_cols]
corr = df_means.corr(
mask = np.triu(np.ones_like(corr, dtype=bool))
plt.figure(figsize=(14, 10))
sns.heatmap(corr, mask=mask, cmap="coolwarm", vmin=-1, vmax=1, center=0,
square=True, linewidths=0.5, cbar_kws={"shrink": .5})
plt.title("Correlation Heatmap (for the MEAN variables)")
plt.xticks(rotation=45, ha="right")
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```