EXPLORING CNNS AND MEL SPECTROGRAMS FOR MUSIC GENRE CLASSIFICATION: AUGMENTATION STRATEGIES AND FINE-TUNING

Luka Blašković, Goran Oreški

Juraj Dobrila University of Pula, Faculty of Informatics, Pula, Croatia
Author 1 Email: lblaskovi@unipu.hr
Author 2 Email: goreski@unipu.hr

ABSTRACT

Sound, as a multi-dimensional signal, has gained increasing attention in the realm of deep learning. The richness and complexity of sound offer extensive potential in various applications, from recommendation systems to text-to-speech technologies. Music, a specific category of sound, has multiple critical representations for its analysis. Among these, mel spectrograms stand out for their capacity to encapsulate both time and frequency dimensions in a manner aligned with human auditory perception. This paper presents enhancements to a CNN model for music genre classification, utilizing Mel spectrograms. We expanded GTZAN dataset through data augmentation techniques, applying four distinct methods to generate synthetic data, which includes both images and raw audio files. We found that converting these images to grayscale significantly improved both training speed and model accuracy. This study also explores the effectiveness of fine-tuning and transfer learning techniques on established neural networks. Notably, the fine-tuned MobileNet architecture achieved the highest accuracy, with a score of 80.13% on our Spotify test dataset, which includes over 400 samples. Importantly, the Spotify dataset was not used for training, emphasizing the robustness of our approach in real-world applications.

***Index Terms*—** deep learning, neural networks, machine learning, music information retrieval, CNN, music, music genre classification, spectrograms, mel-scale, audio transofrmation, GTZAN dataset, SpecAugment, transfer learning, comparative study

1. INTRODUCTION

In the realm of digital information processing, a 'signal' is typically understood as a variation in a certain quantity over time [1]. Signals, particularly in the form of sound, are a common part of our daily lives. We encounter various sounds every day, from the noise of leaves to city sounds and music. Sound, in essence, is a pressure wave traveling through a medium, often air. When it comes to representing sound in the digital domain, it is sampled at discrete intervals and quantized, resulting in a sequence of numerical values [2]. Preparing audio data for deep learning necessitates a deeper dive into understanding the inherent characteristics of sound signals. One such foundational concept is the 'spectrum'. A spectrum encapsulates the set of frequencies combined to produce a signal. For instance, in a piece of music, different instruments playing together can generate a complex sound wave. The spectrum of this music would reveal the individual frequencies and their respective amplitudes. Delving further, there's a distinction between time and frequency domains in signal representation. In the time domain, signals (like audio) are represented as waveforms—depicting how the signal varies over time [3]. However, while waveforms capture temporal variations, they don't explicitly convey the frequencies that constitute the signal. That's where the frequency domain comes into play, and 'spectrograms' shine. A spectrogram is a visual representation of the spectrum of frequencies in a sound signal as they vary with time. It provides an insightful fusion of both time and frequency domains, making it an invaluable tool for many audio processing tasks [4].

Deep learning models, with their power to extract and learn intricate patterns, have revolutionized numerous domains, and audio processing is no exception. Tasks such as music generation, music transcription, text-to-speech, speech-to-text, music recommendation, and voice recognition are now addressed with an efficiency previously deemed unattainable. Among the deep learning architectures, **Convolutional Neural Networks** (CNNs) stand out for their proficiency in handling structured grid data, like images. Given that a spectrogram can be seen as a 2D grid (time vs. frequency) with intensity values, CNNs naturally align as an apt choice for spectrogram-based tasks [5] [6] [7] [8]. These networks inherently capture local patterns and hierarchically combine them, mimicking the manner in which we perceive layered nuances in sound. This makes CNNs particularly effective for tasks like music genre classification using spectrograms.

The paper is structured as follows: Section 2 provides a background on audio signal processing and the theoretical foundations of spectrograms, including a detailed explanation

of why spectrograms are a superior representation for audio analysis, supported by short literature review on evolution of music genre classification. Section 3 outlines our methodology, detailing the dataset used, data augmentation techniques employed, the architecture of our model, and the phases of training. Section 4 presents the results of our experiments. In Section 5, we draw conclusions from these results and discuss their implications. Finally, Section 6 outlines potential avenues for future work in enhancing our model and exploring new applications.

2. AUDIO SIGNAL PROCESSING AND THE THEORY OF SPECTROGRAMS

Audio signal processing is the use of digital computation and algorithms to analyze, manipulate, or transform audio signals [9]. These signals, when represented digitally, can be described as $x[n]$, where n is the sample index. Two fundamental terms essential to this realm are 'sample rate' (f_s) and 'bit-depth' (b).

The **sample rate** f_s denotes how frequently the audio signal is sampled in a given time frame and is usually expressed in Hertz (Hz). A common sample rate is $f_s = 44.1$ kHz, which means 44,100 samples per second (s^{-1}). Mathematically, the Nyquist rate, which is the minimum rate at which the signal can be sampled without loss of information [10], is $f_{\text{Nyquist}} = 2f_{\text{max}}$, where f_{max} is the maximum frequency present in the signal.

Bit-depth b describes the resolution of each sample and is defined as [2]:

$$x[n] = Q\left(A \cdot \sin(2\pi f n T)\right), \tag{1}$$

where Q is the quantization function, A is the amplitude, f is the frequency, and T is the sample period ($T = 1/f_s$). The number of possible amplitude values a sample can assume is 2^b. For instance, a bit-depth of 16 bits provides $2^{16} = 65,536$ possible amplitude values.

To understand the significance of these specifics, it's essential to consider human auditory perception. The **Mel scale**, denoted as $M(f)$, is a perceptual scale that reflects the human ear's response to different frequencies [4] and is defined as [11]:

$$M(f) = 2595 \log_{10}\left(1 + \frac{f}{700}\right). \tag{2}$$

Additionaly, the **decibel scale for amplitude** is defined as [2]:

$$A_{\text{dB}} = 20 \log_{10}\left(\frac{A}{A_0}\right), \tag{3}$$

where A_0 is a reference amplitude.

Mel spectrograms employ the Mel scale on the frequency axis and the decibel scale for amplitude. This representation is more in line with how humans perceive sound and is valuable for applications like audio recognition or music genre classification [11].

Audio signal processing is not just about manipulating numbers; it's about tailoring techniques to align with human perception. Through concepts like the Mel scale and decibel measurement, we bridge the gap between raw audio signals and human listeners.

2.1. Why do Mel Spectrograms perform better?

In audio classification tasks involving deep learning, the choice of input representation is crucial for model efficacy. Mel spectrograms have gained prominence over **Short-Time Fourier Transform** (STFT) spectrograms largely due to their alignment with human auditory perception. STFT is an extension of the Fourier Transform, designed to analyze the frequency components of a signal in a localized manner. The STFT divides the continuous signal into smaller, overlapping segments and applies the Fourier Transform to each of these segments [12]. This yields a time-frequency representation of the signal, offering insights into the temporal evolution of its frequency content. Figure 1 depicts the STFT spectrogram, where the frequency information is linearly scaled.

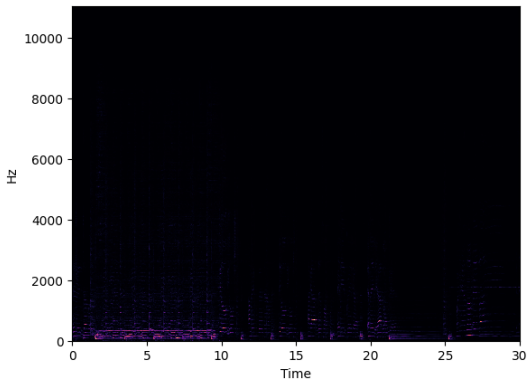

Fig. 1. Short-Time Fourier Transform (STFT) Spectrogram

On the other hand, Figure 2 shows a Mel spectrogram with non-linear frequency scaling designed to mimic the human ear's sensitivity to different frequencies. This focus on lower frequencies is particularly relevant for audio classification tasks such as speech recognition and music genre classification. The logarithmic scale is used to display the power or amplitude of the frequency components.

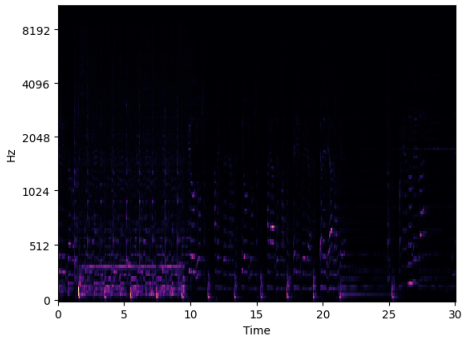

Fig. 2. Mel Spectrogram with a logarithmic scale for frequency intensity

The Mel scale is a nonlinear transformation of the frequency axis, but the actual values can be converted back to Hz for labeling. Mel spectrograms usually undergo additional transformations like logarithmic compression and cosine transforms, enhancing their suitability for pattern recognition tasks [13]. Figure 3 displays a Mel spectrogram where amplitude is represented in specific type of logarithmic scale - decibels (dB), aligning closely with how humans perceive changes in loudness [14].

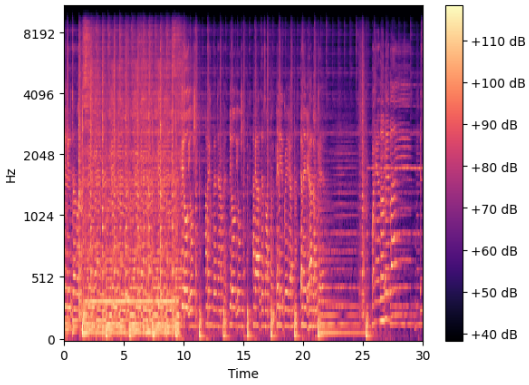

Fig. 3. Mel Spectrogram with a decibel scale for frequency intensity

In contrast, STFT spectrograms may include extraneous data due to their linear frequency scaling, leading to inefficiencies in training time and model performance. Mel spectrograms also facilitate dimensionality reduction, resulting in computationally efficient models. Therefore, their non-linear frequency representation and adoption of a decibel scale for amplitude make Mel spectrograms a preferred choice for deep learning applications in audio classification [13] [15]. Figure 4 illustrates the relationship between frequency, Mel scale, and amplitude. This graph emphasizes how Mel scale and amplitude variations map to actual frequencies, offering a more intuitive and human-aligned representation.

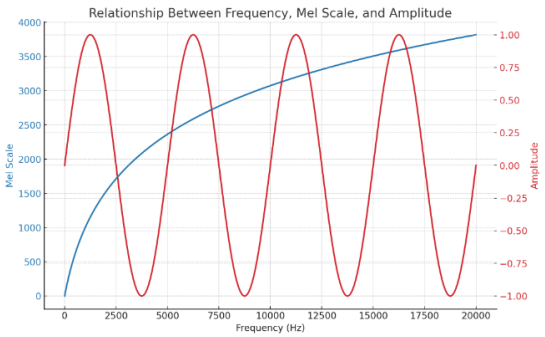

Fig. 4. Relationship Between Frequency, Mel Scale, and Amplitude [16]

2.2. Related Work

Music genre classification has significantly evolved from traditional machine learning techniques to advanced deep learning strategies. Early applications, like Shazam, initially focused on identifying artist and song titles from audio clips. Shazam has since expanded its capabilities to include genre, instrument identification, and user preference prediction [17].

Traditional methods such as kNN, Random Forests, and SVMs were once prevalent, as evidenced by studies utilizing the GTZAN dataset [18]. However, the landscape shifted with the adoption of deep learning techniques. For instance, Bahuleyan's 2018 study contrasted CNN models based on spectrograms with classifiers trained on hand-crafted features, finding the former to be more accurate [19]. Similarly, Feng's research demonstrated the superiority of Deep Belief Neural Networks over conventional neural networks when using augmented data sets [20].

Recent advancements have introduced sophisticated models that blend various neural network architectures. Prabhakar and Lee's 2023 research highlighted the use of innovative approaches like the Bidirectional Long Short-Term Memory with Graphical Convolution Network, achieving remarkable accuracy levels [21]. Other notable contributions include Allamy & Lameiras Koerich's use of 1D residual CNNs, adapting image processing techniques for audio data analysis [22], and the integration of multi-dimensional feature extraction by Pattanaik & Jaiswal [23]. Despite these advancements, the practical applicability of these models remains a concern, especially in real-world scenarios such as analyzing streaming service data. Most studies, including those utilizing the GTZAN dataset, tend to evaluate models on predefined validation sets, which may not accurately reflect the complexity and variability of real-world audio. Furthermore, the GTZAN dataset itself has known limitations, such as mislabeled tracks and a small sample size [18].

2.2.1. *Link to Our Research*

Building on this comprehensive review, our upcoming research will focus on employing straightforward CNN networks while exploring fine-tuning and transfer learning strategies using pretrained models like VGG16 and MobileNet. We will also incorporate extensive data augmentation and crucially test our models on unseen data sourced from Spotify clips. This approach aims to bridge the gap between theoretical accuracies and practical usability, ensuring our models are robust and reliable in real-world conditions. This direction is essential for advancing music genre classification in dynamic and diverse music environments.

3. METHODOLOGY

3.1. Dataset

In the empirical component of this research, the GTZAN dataset serves as the foundation for experimentation. This dataset comprises 1,000 audio samples, evenly distributed across **10 music genres**: blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. Each audio file in this dataset is in WAV format, sampled at a rate of 22,050 Hz, and is mono-channel. Alongside the original audio files, the dataset includes pre-generated spectrograms. However, for this study, custom spectrograms were generated using the Librosa Python library [24], as the original images did not utilize a decibel scale and featured undesirable white horizontal padding at the top and bottom, which adversely affected the results. Each custom spectrogram has dimensions of 500 x 200 pixels and was initially in the RGB color scheme. For the purpose of enhancing model accuracy and expediting training times, these spectrograms were converted to grayscale, effectively reducing the image channels from four (RGB plus alpha) to one. This modification was pivotal in achieving superior model performance due to the reduced computational complexity.

To conclude the empirical methodology of this research, the process begins by converting the raw audio files from the GTZAN dataset into Mel spectrograms using the Librosa Python library. This conversion focuses on emphasizing the relevant frequencies for music processing. Following spectrogram generation, image preprocessing is implemented, which includes normalization through min-max scaling by dividing pixel values by 255. Additionally, each spectrogram is converted from RGB to grayscale to streamline the input data and reduce computational load. The processed images are then fed into a convolutional neural network (CNN), which extracts features and identifies patterns specific to the different music genres. Finally, the CNN outputs are connected to fully connected layers, culminating in a softmax activation layer that provides the probability distribution across the 10 music genres. This methodology illustrated in Figure 5 is designed to enhance accuracy in genre classification.

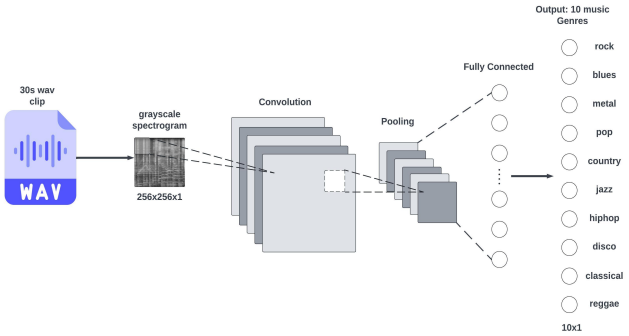

Fig. 5. Methodology

Additionally, a dataset from Spotify, containing 30-second MP3 audio clips publicly available via the official Spotify API [25], was incorporated into this research. As the files lack genre metadata, playlists popularly associated with specific genres were downloaded by querying the most popular playlists after typing a genre name in the Spotify search bar. In total, 453 songs were obtained and subsequently converted into spectrogram representations. These spectrograms were also standardized to grayscale to maintain consistency with the GTZAN dataset's spectrograms. Table 3.1 shows the distribution of songs across different genres in this dataset.

Genre	Number of Songs
Blues	40
Classical	42
Country	40
Disco	66
Hip-Hop	37
Jazz	24
Metal	67
Pop	44
Reggae	37
Rock	56
Total	**453**

Table 1. Distribution of songs in the Spotify dataset by genre

To ensure the uniqueness of the dataset, duplicates were removed post-conversion to spectrograms, including cross-genre duplicates, which is crucial given the subjective nature of music classification. To address this subjectivity in genre classification, the *SparseTopKCategoricalAccuracy* metric was introduced alongside the standard accuracy metric, with k=2, providing a more robust assessment of model accuracy. *SparseTopKCategoricalAccuracy* metric computes how often integer targets are in the top *k* predictions.

The Spotify data, originally in mp3 format with a sample rate of $44.1kHz$, was downsampled to $22.05kHz$ and converted to mono to align with the GTZAN training set. This preprocessing step was essential for ensuring uniformity across datasets and optimizing the conditions for comparative

analysis. Figure 6 shows an example of grayscale mel spectrogram used for training.

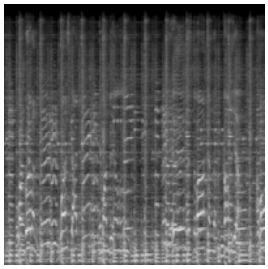

Fig. 6. Example of a spectrogram used for training

3.2. Data Augmentation Techniques

This study has implemented a comprehensive data augmentation strategy using both raw audio augmentation and mel spectrogram augmentation, also commonly referred to as SpecAugment [26]. This section details the specific techniques employed and their relevance to enhancing the performance of machine learning models in this domain.

Raw Audio Augmentation Two primary techniques were utilized to augment the raw audio data from the GTZAN dataset:

- Time Stretching: This technique involves altering the speed of an audio clip without affecting its pitch. It is particularly useful in music genre classification as it simulates variations in tempo, which can be characteristic of certain genres. For example, a slower tempo might be more common in blues or jazz, whereas a faster tempo is typical of genres like disco or metal. To enrich our dataset, we applied stretch factors of 1.25 to speed up the clips and 0.85 to slow them down, resulting in an additional 1986 data instances. This expansion allows for a more comprehensive analysis of how tempo variations within the same music piece can influence genre classification.

- Pitch Shifting: This involves changing the pitch of the audio file without altering the tempo. Pitch shifting helps the model to generalize across different keys, which is important in genre classification because the key can vary widely within the same genre, depending on the artist or specific song style. We implemented pitch shifting by adjusting the audio clips by $n = 2$ and $n = -1$, which shifts the clips up by 2 semitones for a higher pitch, and down by 1 semitone for a deeper sounding clip. This manipulation resulted in an additional 1986 clips, further diversifying our dataset and enhancing the robustness of our genre classification model.

Mel Spectrogram Augmentation For the spectrograms data augmentation, the following techniques were applied:

- Frequency Masking: This method involves randomly masking a range of consecutive frequency channels in the mel spectrogram, a technique particularly useful for genre classification tasks. By selectively obscuring parts of the frequency spectrum, frequency masking compels the classification model to derive insights from less prominent frequency components. This enhances the model's ability to generalize across different musical pieces, even when dominant instruments or vocal pitches vary within the same genre. This robustness is critical in handling the wide array of sound characteristics encountered in diverse music genres. In practice, frequency masking is applied as follows: a specified number of frequency channels, determined by a random value up to a maximum F, are set to a constant value (either zero or the mean of the spectrogram) across all time frames. This is done multiple times per spectrogram as specified by $numMasks$. For instance, if $F = 30$, up to 30 consecutive frequency channels may be masked in each iteration. This randomness in the selection of starting points and the number of channels masked ensures that the model cannot merely memorize specific frequency patterns, thereby improving its ability to focus on broader and potentially more abstract features relevant to music genre classification.

- Time Masking: This technique involves masking consecutive time steps in a spectrogram. Similar to frequency masking but oriented along the time axis, time masking reduces the model's dependence on the exact timing of musical elements. By making the model less sensitive to variations in the timing of notes or beats, which are common within a single genre, this approach enhances the robustness of genre classification. Masking parts of the temporal structure encourages the model to focus on more consistent, genre-defining features rather than specific temporal patterns, thereby strengthening its ability to generalize across similar music types and enhancing training dataset diversity.

The combination of these augmentation techniques significantly expanded the training dataset. Table 2 represents training dataset comprised of 6,537 entries, distributed as follows:

Data Type	**Number of Samples**
GTZAN Training Samples	591
Time Stretching Augmentation	1,986
Pitch Shifting Augmentation	1,986
Frequency Masking Augmentation	987
Time Masking Augmentation	987
Training Total	6537

Table 2. Distribution of the Data Samples

Each augmentation technique was selected based on its demonstrated efficacy in prior studies [27] [28] [29] [30]. The application of these techniques is particularly relevant for music genre classification, as they introduce realistic variability that a model might encounter in real-world scenarios, thereby improving the model's accuracy and generalization capability. By training on this diversified dataset, we observed noticeable improvements in model performance, further underscor-

ing the effectiveness of our chosen data augmentation strategies. The data distribution and splits for training, validation, and testing, including Spotify test data and the data augmentation techniques applied, can be viewed in Figure 7.

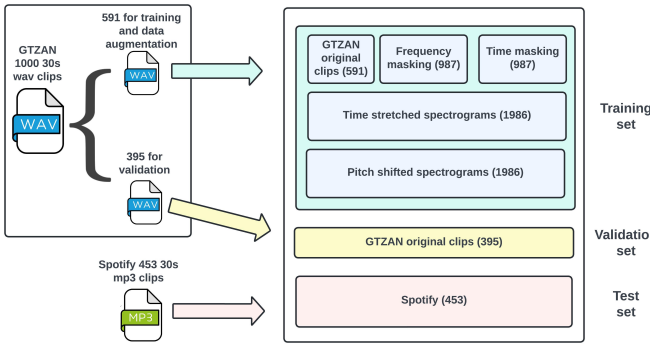

Fig. 7. Data distribution and splits for training, validation, and testing

Additionally, to illustrate the impact of these augmentation techniques, we present in Figure 8 the original spectrogram of a single blues song (A) alongside four other images (B, C, D, E), each corresponding to one of the data augmentation techniques applied.

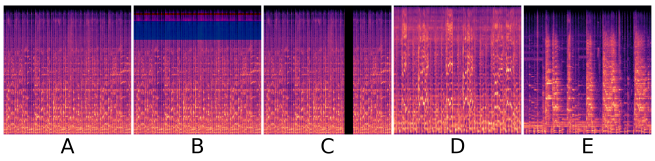

Fig. 8. Spectrograms of a same blues song undergoing various audio transformations: (a) Original, (b) Frequency Masking, (c) Time Masking, (d) Pitch Shift of +2 Semitones, (e) Time Stretch of 1.25x original duration

3.3. Model Architecture

This section outlines the proposed architecture of a sequential convolutional neural network (CNN) designed specifically to effectively recognize patterns within the mel spectrogram representations of audio tracks. Our model employs a layered approach that includes convolutional layers, pooling layers, dropout layers, and dense layers. These components are standard in designing CNNs for both image and audio processing tasks where spectrogram images are fed as input instead.

Input and First Convolutional Block:

- The input layer accepts mel spectrogram images with dimensions of 256x256x1, representing the height, width, and the single grayscale channel of the spectrograms.

- The first layer is a 2D convolutional layer with 32 filters, each of size (3, 3), and uses the ReLU activation function [31]. This layer is designed to capture initial features like edges from the spectrogram.

- Following the convolution, a max pooling layer with a window of (2, 2) is applied to reduce the spatial dimensions of the feature maps, which helps in reducing the computational load and overfitting by providing an abstracted form of the features [32].

- A dropout layer with a rate of 0.35 follows the pooling layer to prevent overfitting by randomly setting a fraction of input units to 0 at each update during training time [33].

Second Convolutional Block:

- Another convolutional layer with 64 filters of size (3, 3) follows, also employing the ReLU activation function. This layer further processes the features extracted by the previous layers, focusing on more complex patterns.

- This is again followed by a max pooling layer with the same dimensions and a dropout layer, repeating the pattern to effectively distill and abstract features while controlling overfitting.

Flattening and Dense Layers:

- The flatten layer converts the 2D feature maps into a 1D feature vector, necessary for the fully connected layers that follow [34].

- A dense layer with 64 units is used next, with ReLU activation to introduce non-linearity.

- A dropout layer with a dropout rate of 0.35 is used after the dense layer to further mitigate the risk of overfitting.

Output Layer:

The final layer is a dense layer with 10 units corresponding to the 10 music genres. This layer uses the softmax activation function [35] to output a probability distribution over the 10 classes, each probability signifying the likelihood of the input spectrogram belonging to a specific genre. The model architecture we have developed is detailed in Table 3, which includes the output shapes and the number of parameters for each layer.

Layer (type)	Output Shape	Params #
Input	(256, 256, 1)	0
Conv2D-32 (3x3)	(254, 254, 32)	320
MaxPooling2D (2x2)	(127, 127, 32)	0
Dropout (0.35)	(127, 127, 32)	0
Conv2D-64 (3x3)	(125, 125, 64)	18,496
MaxPooling2D (2x2)	(62, 62, 64)	0
Dropout (0.35)	(62, 62, 64)	0
Flatten	(246,016)	0
Dense-64	(64)	1,574,496
Dropout (0.35)	(64)	0
Dense-10 (softmax)	(10)	650

Table 3. Our proposed CNN

The total parameters of our custom CNN is 15,764,554, which include trainable and non-trainable parameters inte-

grated throughout the convolutional, dense, and dropout layers.

3.4. Training

The convolutional neural network (CNN) model was trained using a dataset comprising 6,537 grayscale images, each formatted as 256x256 pixels with a single channel. This training was conducted over 12 epochs with a batch size of 32. The choice of grayscale images helps focus the model's learning on textural and structural patterns in the spectrograms, which are crucial for identifying musical genres. To quantify the performance of the model and guide its learning, the loss function employed was sparse categorical crossentropy. This function is particularly suited for classification tasks with multiple classes where each class label is provided as an integer; it measures the disparity between the predicted probability distribution and the true distribution, with a focus on penalizing incorrect categorical predictions.

The optimization of the model's weights was performed using the Adam optimizer [36], a widely-used variant of stochastic gradient descent that adjusts the learning rate dynamically, enhancing the efficiency and effectiveness of training deep neural networks.

Furthermore, **two (2) accuracy metrics** were used to evaluate the model's performance: top-1 accuracy and *SparseTopKCategoricalAccuracy* with *k=2*. Top-1 accuracy measures the proportion of instances where the top prediction corresponds to the true label, providing a straightforward metric of classification success. Meanwhile, *SparseTopKCategoricalAccuracy*, with *k* set to 2, was specifically chosen because many songs can straddle two different genres.

The *SparseTopKCategoricalAccuracy* for $k = 2$ is a metric that assesses how frequently the true label appears within the top k predictions of the model. For $k = 2$, the mathematical formulation is as follows:

Let y_i denote the true label for the i^{th} sample, and $P(i, j)$ represent the predicted probability for the j^{th} class of the same sample, where j spans all possible classes. The probabilities for each sample are ranked from highest to lowest. The function $f(y_i, P)$ is defined as:

$$f(y_i, P) = \begin{cases} 1 & \text{if } y_i \in \{\text{indices of the top 2 max } P(i,j)\} \\ 0 & \text{otherwise} \end{cases}$$

The overall *SparseTopKCategoricalAccuracy* for the dataset is then computed as the average of $f(y_i, P)$ across all samples i:

$$\text{SparseTopKCategoricalAccuracy} = \frac{1}{N} \sum_{i=1}^{N} f(y_i, P)$$

This metric is particularly beneficial for tasks where classifications are not mutually exclusive or might be ambiguous, as it evaluates the model's accuracy in placing the true label within its top two predictions, offering a more lenient yet realistic assessment of the model's performance.

The detailed results of this training process will be discussed in the subsequent Experimental results section of this paper.

3.5. Harnessing Pre-trained Networks

Along with our model, this study employs transfer learning and fine-tuning with **VGG16** [37] and **MobileNet** [38], models originally designed for image classification, to enhance music genre classification from spectrograms. These models were utilized directly as they are built into the Keras library [39], facilitating their adaptation. Given their success in visual pattern recognition, adapting these architectures to interpret audio data as visual spectrograms offers a promising approach to improve classification accuracy and system robustness in the auditory domain.

Transfer Learning: The application of transfer learning techniques involved utilizing pre-trained neural networks, specifically VGG16 and MobileNet, which are typically designed for image classification tasks. These models were adapted for the task of music genre classification from spectrograms, which are essentially grayscale images representing sound frequency and time. To accommodate our single-channel (grayscale) spectrogram data, modifications were made to the input layer of these models to transform the 1-channel input into a 3-channel format, mimicking RGB images. This adaptation allowed the use of architectures initially intended for color images without substantial architectural changes.

In the case of VGG16 during the transfer learning phase, the model had a total of 16,812,554 parameters, out of which 2,097,866 were trainable. For MobileNet, the model comprised 7,423,882 total parameters with 4,195,018 being trainable. Upon configuring the input layers, all layers of the base models were frozen to retain the learned features relevant to image processing, which are similarly beneficial for analyzing textures and patterns in spectrograms. The networks were extended by adding a flatten layer followed by two dense layers. The first dense layer comprised 64 units with ReLU activation function, which introduces non-linearity to the learning process, enhancing the model's ability to learn complex patterns. The second dense layer, equipped with 10 units and a softmax activation, served the purpose of classifying the inputs into one of ten music genres. The training was conducted over 12 epochs with a batch size of 32, employing the same metrics as the initial training: top-1 accuracy and *SparseTopKCategoricalAccuracy* with $k = 2$, to comprehensively assess the model's classification prowess. The default learning rate of 0.001 was maintained during this phase to leverage the pre-trained weights without causing drastic updates that might disrupt the learned features.

Fine-tuning: Following the transfer learning phase, fine-tuning was employed to tailor the pre-trained networks more closely to the specific task of genre classification from spectrograms. In this phase, the entire network was made trainable, allowing updates to all weights across the model. For VGG16, all 16,812,554 parameters were finetuned, while for MobileNet, all 7,423,882 parameters underwent fine-tuning. This approach aims to refine the high-level features in the deeper layers of the network, making them more relevant to the task at hand. The fine-tuning was also conducted over 12 epochs with a batch size of 32, but crucially, at a reduced learning rate of 0.0001. Lowering the learning rate for fine-tuning is a strategic choice to prevent overwriting the nuanced features already learned in the transfer learning phase [40]. A smaller learning rate ensures that the updates to the model weights are incremental, thus preserving the integrity of the pre-existing features while gradually adapting them to better suit the classification of music genres [41].

The experiments were conducted using the TensorFlow framework on a Linux operating system, leveraging an RTX 3090Ti NVIDIA GPU.

4. EXPERIMENTAL RESULTS

The convolutional neural network (CNN) architecture developed in this study, along with the VGG16 and MobileNet models adapted for music genre classification using mel spectrograms, have all demonstrated exceptional performance on the GTZAN dataset. Each model achieved an outstanding 100% accuracy for both top-1 and top-2 genre classifications during validation. This perfect score across all three models underscores the effectiveness of the specifically designed CNN and the adapted pre-existing architectures when validated in a controlled environment. This result demonstrates their robust ability to accurately classify music genres under standardized testing conditions provided by the validation dataset.

In contrast, the real-world application of this model, tested using the Spotify dataset, presented more challenging conditions, leading to reduced accuracies. Here, the model achieved a top-1 accuracy of 53% and a top-2 accuracy of 72%. These results, while lower, are significant as they illustrate the model's robustness and generalization capabilities across a more diverse and uncontrolled music collection. The discrepancy in performance between the GTZAN and Spotify datasets can largely be attributed to the inherent differences in audio quality, genre distribution, and possibly more nuanced genre overlaps in the Spotify dataset.

Further insights were gained through the application of fine-tuning techniques on the pre-trained MobileNet architecture, which showed an enhanced top-1 accuracy of 62.50% and a top-2 accuracy of 80.13% on the Spotify dataset. This improvement indicates that fine-tuning the pre-trained networks to the specific task of music genre classification from spectrograms can significantly enhance the model's ability to discern subtle features relevant to genre classification in a more complex dataset.

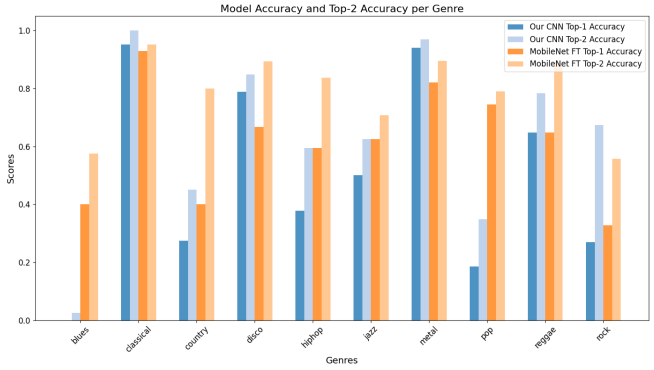

Fig. 9. Accuracy and top-2 accuracy per genre of custom CNN model and fine-tuned MobileNet

Table 4 summarizes the performance of our custom CNN, VGG16, and MobileNet through transfer learning and fine-tuning on the Spotify dataset, highlighting the effectiveness of fine-tuning pre-trained models compared to custom and standard architectures.

Model	Top-1 Acc. (%)	Top-2 Acc. (%)
Our CNN	52.90	71.88
VGG16 TL	56.25	72.76
VGG16 FT	58.92	73.21
MobileNet TL	58.03	73.88
MobileNet FT	62.50	80.13

Table 4. Accuracy of CNN Models on Spotify Dataset. TL - Transfer Learning, FT - Fine-tuning

Figure 9 shows the accuracy and top-2 accuracy per genre for the custom CNN model and the fine-tuned MobileNet. From the results, it can be observed that our models perform exceptionally well on genres such as metal, classical, and disco, achieving nearly 100% top-2 genre prediction accuracy. Genres such as hip-hop, jazz, country, pop, and reggae achieve around 80% top-2 genre prediction accuracy, while rock and blues have the lowest accuracies. This lower performance for rock and blues is likely due to the wide spectrum of songs that can be classified under these genres, leading to greater variability and complexity in their classification.

This comparison clearly demonstrates the enhanced accuracy achieved by MobileNet through fine-tuning, as well as illustrating the challenges of applying these models to real-world data compared to more controlled environments.

Building on the insights from the CNN performance metrics, another crucial factor emerged regarding the use of image formats within our neural network processing. Initially, our models, including the custom CNN, VGG16, and Mo-

bileNet, were trained using mel spectrograms derived from music tracks. A notable discovery was that models trained on grayscale images - or, in our case, single-channel spectrograms yielded better results compared to those trained with RGB (multi-channel) spectrograms. This likely reflects the greater simplicity and reduced computational complexity of processing single-channel data, which allows the models to focus more effectively on texture and shape patterns pertinent to music genres.

Furthermore, an additional aspect of our experimental setup involved the audio file formats. Our models were originally trained on uncompressed WAV files from the GTZAN dataset, which generally contain more detailed audio information due to the lack of compression. This contrasts sharply with the real-world testing environment using the Spotify dataset, where the audio was in compressed MP3 format. To investigate the impact of this discrepancy, we conducted an experiment where we converted the GTZAN dataset from WAV to MP3 and trained the models on this transformed data. The results were markedly inferior, implying that the loss of data fidelity due to MP3 compression negatively affects the performance of the models.

This was tested during early experimentation with a smaller dataset and simpler CNN, comprising 1000 instances with a 70% training and 30% test split, and no data augmentation techniques used. The results, as seen in Table 5 below, show that models trained on MP3 files generally perform worse than those trained on WAV files. Similarly, models trained with grayscale spectrogram images consistently outperform those trained with RGB images.

Data Description	**Top-1 Acc. (%)**	**Top-2 Acc. (%)**
WAV & RGB	17.44	29.13
WAV & Grayscale	23.84	43.92
MP3 & RGB	17.21	26.26
MP3 & Grayscale	22.73	40.61

Table 5. Performance Comparison of CNN Models Based on Audio and Image Formats

This observation underscores a key challenge in applying deep learning models to real-world scenarios: the quality of the input data can significantly influence the accuracy and effectiveness of the models. It appears that despite the lower fidelity of the MP3 data used in the Spotify test, our WAV-trained models still performed well, suggesting a certain robustness but also pointing to the potential for improvement if high-fidelity data were available in real-world applications. Moreover, our model trained on a 1000 WAV and grayscale dataset performed even better with additional data (total of 6537) generated with augmentation techniques discussed earlier, reaching 59.40% for top-1 accuracy and 71.88% for top-2 musical genre accuracy prediction.

5. CONCLUSION

This study successfully demonstrates the application of convolutional neural networks (CNNs) and Mel spectrograms for music genre classification, leveraging the GTZAN dataset and a novel Spotify dataset to evaluate performance across different music collections. Our custom CNN model, along with adaptations of VGG16 and MobileNet using transfer learning and fine-tuning, achieved high accuracy in controlled test environments, with perfect scores on the GTZAN dataset validation. The real-world application using the Spotify dataset provided a rigorous test of the models' generalization capabilities, where they exhibited robust performance albeit with a reduction in accuracy, which was expected due to the dataset's complexity and diversity.

Key findings include the superior performance of models utilizing grayscale spectrograms over RGB formats, which suggests that simpler input data formats can effectively enhance CNN training by focusing on textural and structural details essential for genre differentiation. Furthermore, the fine-tuning of pre-trained models on genre-specific data proved significantly beneficial, particularly with the MobileNet architecture, which outperformed all other models on the Spotify dataset.

The application of data augmentation techniques, such as time stretching and pitch shifting for raw audio, as well as frequency and time masking for spectrograms, substantially increased the diversity and size of our training dataset, which was crucial in enhancing the model's ability to handle real-world data variabilities. These strategies, combined with the empirical insights gained from comparing performance across different audio formats (WAV vs. MP3), underline the importance of high-quality, well-preprocessed data for training effective deep learning models in music genre classification.

6. FUTURE WORK

Future work will focus on several key areas to enhance the current models and explore new dimensions of music genre classification. Firstly, expanding the diversity and size of the training datasets beyond the GTZAN and Spotify collections will be crucial. Investigating more varied and extensive datasets could further improve the robustness and accuracy of the models, especially in real-world applications.

Secondly, exploring additional deep learning architectures and more advanced data augmentation techniques could yield better performance and insights. For example, integrating recurrent neural network (RNN) layers could enhance the models' ability to capture temporal dynamics in music, which are crucial for recognizing and differentiating genres with similar spectral characteristics but different rhythmic patterns.

Moreover, a comparative analysis of different loss functions and their impact on multi-class classification tasks like music genre classification could provide deeper insights into

model optimization and performance enhancement.

Finally, developing a real-time classification system that can be deployed as a practical application in music streaming services would be an exciting direction. This would not only test the models in dynamic scenarios but also potentially contribute to enhanced user experience through more accurate and responsive music recommendation systems based on genre classification.

7. REFERENCES

[1] A.V. Oppenheim, A.S. Willsky, and I.T. Young, *Signals and Systems*, Prentice-Hall signal processing series. Prentice-Hall, 1983.

[2] M. Puckette, *The Theory and Technique of Electronic Music*, World Scientific Publishing Company, 2007.

[3] Julius Orion Smith, *Mathematics of the discrete Fourier transform (DFT): With audio applications*, Julius Smith, 2008.

[4] Jayesh Kumpawat and Shubhajit Dey, "Acoustic scene classification using auditory datasets," 2022.

[5] Chiyu "Max" Jiang, Jingwei Huang, Karthik Kashinath, Prabhat, Philip Marcus, and Matthias Niessner, "Spherical cnns on unstructured grids," *arXiv preprint arXiv:1901.02039*, 2019.

[6] P. F. Antonietti and E. Manuzzi, "Refinement of polygonal grids using convolutional neural networks with applications to polygonal discontinuous galerkin and virtual element methods," *arXiv preprint arXiv:2102.05738*, 2021.

[7] Tsung-Wei Ke, Michael Maire, and Stella X. Yu, "Multigrid neural architectures," *arXiv preprint arXiv:1611.07661*, 2016.

[8] Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr, "An evaluation of convolutional neural networks for music classification using spectrograms," *Applied soft computing*, vol. 52, pp. 28–38, 2017.

[9] Jason Goldthwaite, "The value of digital signal processing," Sensear Blog, 2023, Accessed: 10/09/2023.

[10] Gavin Wright, "What is the Nyquist theorem? — techtarget.com," https://www.techtarget.com/whatis/definition/Nyquist-Theorem: :text=The [Accessed 11-09-2023].

[11] D O'Shaughnessy, *Speech Communication*, Addison-Wesley series in electrical engineering. Digital signal processing. Addison Wesley Longman Publishing, New York, NY, Aug. 1987.

[12] Ervin Sejdić, Igor Djurović, and Jin Jiang, "Time–frequency feature representation using energy concentration: An overview of recent advances," *Digital Signal Processing*, vol. 19, no. 1, pp. 153–183, 2009.

[13] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho, "A comparison of audio signal preprocessing methods for deep neural networks on music tagging," in *2018 26th European Signal Processing Conference (EUSIPCO)*, 2018, pp. 1870–1874.

[14] BL Roberts, "The decibel scale," *Department of Physics Boston University*, pp. 2–5, 1984.

[15] Jash Mehta, Deep Gandhi, Govind Thakur, and Pratik Kanani, "Music genre classification using transfer learning on log-based mel spectrogram," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, 2021, pp. 1101–1107.

[16] OpenAI, "Gpt-4: Generative pre-trained transformer 4," OpenAI Blog, 2021, Prompt: [Can you plot a diagram representing relationship between frequency, mel scale and amplitude. Put frequency on X axis, mel scale on y on the left side, and amplitude on the right side.].

[17] T. O'Brien, "Learning to understand music from shazam," https://medium.com/@tsob/learning-to-understand-music-from-shazam-56a60788b62f, 2023, Accessed: 2023-05-05.

[18] Ndiatenda Ndou, Ritesh Ajoodha, and Ashwini Jadhav, "Music genre classification: A review of deep-learning and traditional machine-learning approaches," in *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2021, pp. 1–6.

[19] Hareesh Bahuleyan, "Music genre classification using machine learning techniques," 2018.

[20] Tao Feng, "Deep learning for music genre classification," *private document*, 2014.

[21] Sunil Kumar Prabhakar and Seong-Whan Lee, "Holistic approaches to music genre classification using efficient transfer and deep learning techniques," *Expert Systems with Applications*, vol. 211, pp. 118636, 2023.

[22] Safaa Allamy and Alessandro Lameiras Koerich, "1d cnn architectures for music genre classification," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, pp. 01–07.

[23] R. K. Pattanaik and A. Jaiswal, "Classification of music genre using support vector machine and convolutional neural network," *World Journal of Advanced Research and Reviews*, vol. 21, no. 3, pp. 1009–1019, 2024.

[24] Colin Raffel Dawen Liang Daniel PW Ellis Matt McVicar Eric Battenberg McFee, Brian and Oriol Nieto, "librosa/librosa: 0.9.2," 2022.

[25] Spotify, "Spotify web api," 2024.

[26] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*. Sept. 2019, ISCA.

[27] Rafael L. Aguiar, Yandre M.G. Costa, and Carlos N. Silla, "Exploring data augmentation to improve music genre classification with convnets," in *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018, pp. 1–8.

[28] Rémi Mignot and Geoffroy Peeters, "An analysis of the effect of data augmentation methods: Experiments for a musical genre classification task," *Transactions of the International Society for Music Information Retrieval*, Dec 2019.

[29] Wenhao Bian, Jie Wang, Bojin Zhuang, Jiankui Yang, Shaojun Wang, and Jing Xiao, "Audio-based music classification with densenet and data augmentation," in *PRICAI 2019: Trends in Artificial Intelligence*, Abhaya C. Nayak and Alok Sharma, Eds., Cham, 2019, pp. 56–65, Springer International Publishing.

[30] Nirmal M R and Shajee Mohan B S, "Music genre classification using spectrograms," in *2020 International Conference on Power, Instrumentation, Control and Computing (PICC)*, 2020, pp. 1–5.

[31] Abien Fred Agarap, "Deep learning using rectified linear units (relu)," 2019.

[32] Vincent Christlein, Lukas Spranger, Mathias Seuret, Anguelos Nicolaou, Pavel Kral, and Andreas Maier, "Deep generalized max pooling," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. Sept. 2019, IEEE.

[33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, jan 2014.

[34] Jonghoon Jin, Aysegul Dundar, and Eugenio Culurciello, "Flattened convolutional neural networks for feedforward acceleration," 2015.

[35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[36] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2017.

[37] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[38] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[39] François Chollet et al., "Keras," https://github.com/fchollet/keras, 2015.

[40] Youngmin Ro and J. Choi, "Autolr: Layer-wise pruning and auto-tuning of learning rates in fine-tuning of deep networks," pp. 2486–2494, 2021.

[41] Aziz Fajar, R. Sarno, C. Fatichah, R. Susilo, and Gusti Pangestu, "Cyclical learning rate optimization on deep learning model for brain tumor segmentation," *IEEE Access*, vol. 11, pp. 119802–119810, 2023.