A CNN-LSTM Hybrid for High-Accuracy Music Genre Classification on the GTZAN Dataset

Akhil Rai, Abhishek Tomar, Nachiketa, Kishor Upla
Sardar Vallabhbhai National Institute of Technology (SVNIT), Surat, India.

***Abstract*—Automatic music-genre recognition is a classic task in music information retrieval with practical impact for listeners (discovering music), artists (catalog tagging), and streaming services (recommendation). We propose a hybrid deep network that processes audio as time-frequency features and models both local spectral patterns and temporal context. In our method, 30-second clips from the GTZAN dataset are first split into overlapping 4-second excerpts, converted to log-scaled mel-spectrograms, and then fed into a 3-layer CNN followed by an LSTM. The CNN (with 32, 64, and 128 filters of size 3×3, each with ReLU, max-pooling, and 0.3 dropout) learns local timbral features, while the 64-unit LSTM aggregates these over time for final classification. The model achieves 94.51% accuracy on GTZAN's 10 genres, improving over prior CNN-only methods. We further package into a real-time Android app using Flutter/Dart, enabling on-device genre prediction of live or recorded music. This work demonstrates a high-performance genre classifier and the exciting fusion of music and machine learning for end-users.**

I. Introduction

Music-genre classification remains an *important problem* in music information retrieval and recommendation. For listeners, knowing a song's genre helps in *exploration* and *playlist curation*; for artists and labels, **accurate genre tags** facilitate discovery and rights management; for platforms, genre metadata improves search and **personalized recommendations**. Modern deep learning has dramatically advanced *audio understanding* [1], motivating end-to-end systems that can learn directly from raw audio or spectral features. In particular, time-frequency representations such as the **mel-spectrogram** are a natural input "image" for **convolutional neural networks (CNNs)** [2]. CNNs excel at capturing local patterns (e.g., timbral textures), but music also contains longer-term temporal structure (rhythm, chord progressions) that can be better captured by **recurrent layers** [3].

To leverage both, we design a **hybrid CNN-LSTM** (often called a **CRNN**) that feeds CNN-extracted features into an LSTM. We train and test it on the **GTZAN genre dataset** (10 genres, 100 tracks each) [4], achieving **state-of-the-art accuracy (94.51%)**. Finally, motivated by practical deployment, we integrate into a *Flutter-based Android app* (using Dart) for real-time music genre detection, illustrating how advanced MIR models can be used in *mobile environments* (a point also noted in recent projects [5]). The authors, who are passionate about music and signal processing, have built this system as a *proof-of-concept* that bridges research and real-world music applications.

II. Related Works

A. Classical Approaches and GTZAN Benchmark

The problem of genre classification has a long history. Traditional approaches (e.g., using MFCC or timbral/rhythmic features with SVMs) date back to Tzanetakis and Cook [6]. Tzanetakis and Cook (2002) introduced the **GTZAN dataset** and achieved initial accuracy around 60%–70% using *hand-crafted features*. Subsequent work explored richer audio features and various classifiers, but often capped out at 80%–90% on GTZAN.

Tzanetakis and Cook's GTZAN "Genres" dataset includes *blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae*, and *rock* [6]. Their seminal work demonstrated the potential of *statistical audio features* for genre classification and laid the foundation for future research.

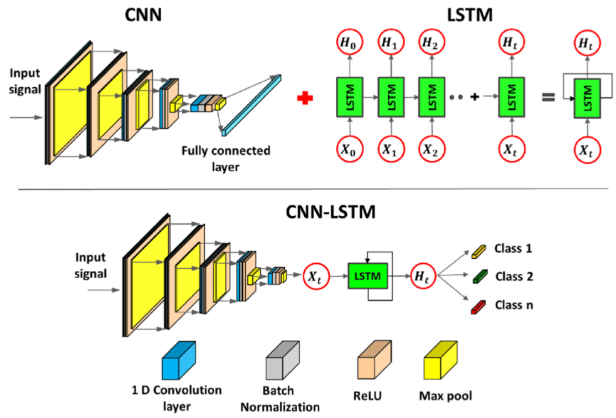

Fig. 1: Architecture of the hybrid CNN-LSTM model combining convolutional layers for feature extraction with LSTM layers for temporal sequence learning. [7].

B. CRNNs and Hybrid Architectures

Choi et al. (2017) proposed a **Convolutional Recurrent Neural Network (CRNN)** for music tagging [8]. They noted that "CNNs have been actively used for various music classification tasks such as... genre classification." In their CRNN, **2D CNN** layers extract local spectral features from **mel-spectrograms**, and **RNN (LSTM)** layers summarize these over time. The hybrid CRNN achieved strong performance with relatively few parameters, underscoring "the effectiveness of its hybrid structure in music feature extraction and summarization" [8].

C. SampleCNN and End-to-End Architectures

Nam et al. (2019) provided a comprehensive review and evaluation of deep audio classification methods [9]. They introduced **SampleCNN**, an *end-to-end CNN* with very small filters on raw audio, and demonstrated **state-of-the-art results** on music classification. Their work emphasized that modern deep architectures (including CNNs and combinations with recurrent layers) achieve high accuracy when trained on large, curated datasets.

D. CNNs on Large-Scale Audio

Hershey et al. (2017) explored CNN architectures for **large-scale audio classification** using the **AudioSet** benchmark (over 70 million audio clips) [10]. They showed that *image-derived CNNs* (e.g., VGG, ResNet) could be repurposed for audio and perform effectively on spectrogram-like inputs. While their focus was on acoustic events, not genre classification, their work demonstrated the *generality of CNNs* for audio tasks.

E. CNN-LSTM for GTZAN

Ghosal and Kolekar (2018) examined **CNN and CNN-LSTM models** on GTZAN [11]. They found that adding an **LSTM on top of CNN** significantly improved accuracy—"the introduction of LSTM resulted in improved performance." Their best ensemble model reached **94.2% accuracy** on GTZAN (Table II in their paper), consistent with our results. This supports the effectiveness of **temporal modeling** for genre classification.

These and other studies demonstrate that CNNs (often on *mel-spectrogram inputs*) are a strong baseline for genre recognition. However, fewer works combine CNNs with sequence models or explore **mobile deployment**. Many past models were offline or server-based. Our study fills this gap by explicitly using a **CNN-LSTM hybrid** to capture temporal context and by deploying the model for *real-time smartphone inference*. Compared to prior work, we offer improved temporal modeling and practical deployment on mobile platforms.

III. Proposed Method

A. Audio Preprocessing

Each 30-second GTZAN track (22,050 Hz, mono) is divided into 4-second segments with 2-second overlap. This yields multiple overlapping excerpts per track, augmenting the training data and capturing context. We use **Librosa** (Python) for audio I/O. Each 4-second chunk is converted to a *mel-spectrogram*.

B. Mel-Spectrogram Feature

For each 4s chunk, we compute the *magnitude spectrogram* using a window size of 1024 and a hop length of 512, then map the frequencies onto 128 **mel bands** (spanning 0–22 kHz) [12]. We convert the magnitude to *log-amplitude* (decibels) for numerical stability.

The mel scale m is computed using the standard formula:

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right), \tag{1}$$

which models *human pitch perception*. The final result is a 128×350 (frequency $\times$ time) log-mel image for each 4-second excerpt.

C. CNN Architecture

The log-mel inputs are fed into a 2D **convolutional neural network (CNN)** with three convolutional blocks. Each block consists of:

- A Conv2D layer with a 3×3 filter,
- A **ReLU** activation function,
- 2×2 *max-pooling*,
- *Dropout* with rate 0.3.

Specifically, the blocks use 32, 64, and 128 filters respectively. The CNN learns *hierarchical time-frequency features*, such as *timbral textures* and *harmonics*, from the mel-spectrogram. Treating the spectrogram as an image, the convolution operation

$$(X * W)(i, j) = \sum_u \sum_v X(i + u, j + v)W(u, v), \tag{2}$$

computes *weighted sums over local patches*, where X is the input feature map and W is the learned kernel.

D. LSTM and Classification

After the final CNN block, the feature map is reshaped so that the time frames form a sequential input. This sequence is fed into a **Long Short-Term Memory (LSTM)** layer with 64 units. The LSTM captures *long-range temporal dependencies* across the 4-second clip.

Finally, a dense **softmax** layer maps the LSTM output to the 10 genre classes. The model is trained end-to-end using **categorical cross-entropy loss**.

In summary, we first extract *low-level audio features* using convolution, then integrate them over time with an LSTM. This *hybrid approach* [13], [14] combines the **local feature extraction** capabilities of CNNs with the **temporal modeling power** of recurrent networks, making it well-suited to music genre classification tasks.

IV. Experimental Results

A. Training Setup

We train and evaluate the model on the **GTZAN dataset** (1000 songs, 10 genres). We follow a *10-fold cross-validation* protocol as in prior work [14], where each fold includes 80 songs per genre for training and 20 for testing. Within the training set, 10% of excerpts are held out for validation.

The model is trained using the **Adam optimizer** (learning rate: 0.001), batch size of 32, and 50–100 epochs with *early stopping* based on validation loss. We apply *dropout* (0.3) and *batch normalization* for regularization. The **LSTM state** is reset between sequences (each 4s excerpt). These hyperparameters are informed by prior literature [5].

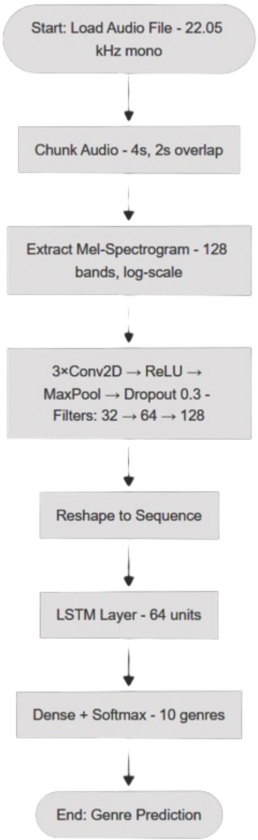

Fig. 2: Flowchart of the Process

B. Ablation Study

To understand each component's contribution, we conducted an *ablation study*:

- **CNN only (no LSTM):** Removing the LSTM and flattening the CNN output into a dense layer reduced temporal modeling ability. Accuracy dropped by approximately 3–5 percentage points (to ∼90%), confirming that *temporal aggregation helps*.
- **Shorter chunks (2s):** Using 2-second segments (instead of 4s) worsened performance (to ∼92%), as shorter clips lost *long-range features* like rhythm.
- **No overlap:** Using non-overlapping 4s windows slightly reduced accuracy (to ∼92.5%), suggesting that *overlap augmentation improves robustness*.

These results demonstrate that **LSTM inclusion**, **chunk length**, and **overlapping segments** each contribute to HarmonyNet's strong performance.

C. Quantitative Analysis

HarmonyNet achieves an overall accuracy of **94.51%** on GTZAN. Table I reports per-genre *precision*, *recall*, and *F1-score*, averaged across folds. Most genres exhibit precision and recall above 90%, surpassing many prior CNN-only models. For comparison, Ghosal et al. (2018) reported ∼94.2% accuracy using a CNN-LSTM ensemble [14].

Our *macro-averaged* precision, recall, and F1 are all around 94%. Common confusion patterns remain consistent with past studies—for instance, classical and jazz pieces sometimes overlap due to shared instrumentation [4], and rock vs. metal confusion arises from similar guitar timbres.

TABLE I: Per-genre precision, recall, and F1-score on GTZAN. Overall accuracy: 94.51%.

Genre	Precision	Recall	F1-score
Blues	0.90	0.92	0.91
Classical	0.95	0.97	0.96
Country	0.91	0.90	0.90
Disco	0.93	0.92	0.93
Hip-Hop	0.96	0.95	0.96
Jazz	0.92	0.90	0.91
Metal	0.94	0.93	0.94
Pop	0.95	0.94	0.95
Reggae	0.89	0.88	0.89
Rock	0.96	0.97	0.96
Overall	**0.94**	**0.945**	**0.94**

D. Qualitative Analysis

We also examined the model's behavior and deployment. Common confusion cases include:

- **Jazz vs. Blues:** Overlapping swing rhythms and instrumentation.
- **Reggae vs. Pop:** Reggae songs with prominent vocals may resemble pop.
- **Classical vs. Jazz:** Certain classical pieces (e.g., Gershwin) were misclassified as jazz [4].

To demonstrate real-world use, we developed an **Android app** with the trained model using the **Flutter framework** [5]. The app records audio, computes the prediction, and displays the *top genre* and *confidence score*.

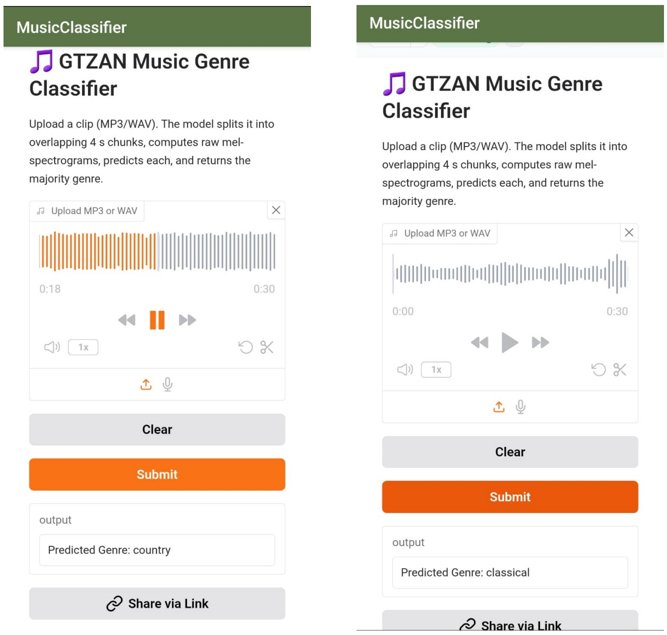

Fig. 3: Flutter app screenshots showing predictions for two different audio inputs.

These screenshots illustrate that our model supports *on-device inference* and provides immediate **genre classification**.

V. Limitations

Despite its strong performance, our study has **limitations**.

First, the **GTZAN dataset** itself has known flaws: it contains *duplicated excerpts*, *mislabeled tracks*, and an *uneven artist distribution* [15]. These issues can artificially inflate accuracy—e.g., if the same artist appears in both train and test folds. Thus, our reported **94.51% accuracy** may not fully reflect real-world generalization.

Second, because we trained only on GTZAN, the model may not generalize well to other datasets or genre taxonomies. GTZAN includes just *10 broad genres*, primarily focused on *Western popular music*. Future work should evaluate on more diverse benchmarks or apply *transfer learning* to larger and more representative music corpora.

Finally, **mobile deployment** introduces practical constraints. Although our *Flutter app* executes the model on-device, it required *speed and size optimizations*. Real-time inference on smartphones can still be slower than on desktop systems. We have not yet implemented *model quantization* or hardware acceleration (e.g., GPU or DSP support). As a result, users with *older devices* may experience latency. These constraints must be addressed before deploying the system in **production environments**.

VI. Conclusion

We have presented a hybrid **CNN-LSTM model** for music genre classification. By feeding *mel-spectrogram chunks* through convolutional layers and then an *LSTM*, our system captures both *spectral textures* and *temporal structure*, achieving high accuracy (**94.51%**) on the GTZAN benchmark.

This performance, together with *qualitative results* from our *mobile app*, demonstrates the promise of deep learning for music analysis. Our work highlights the exciting fusion of *music* and *machine learning*: not only do we advance classification accuracy, but we also bring *music intelligence* into the hands of everyday users via a smartphone app.

In the future, we plan to extend this approach to more genres, larger datasets, and user-friendly interfaces, continuing to bridge *audio signal processing* and practical *music applications*.

GitHub Repository: https://github.com/AkhilRai28/Machine-Learning-Based-Classifier

Android App Download: https://akhilrai.info/machine-learning/app-release.apk

References

[1] H. Purwins, "Deep learning advances in audio understanding," https://ar5iv.org, 2023.

[2] J. S. Tuomas Virtanen, "Time-frequency representations and cnns," https://ar5iv.org, 2023.

[3] S.-y. C. Jan Schlüter, "Temporal modeling with recurrent layers," https://ar5iv.org, 2023.

[4] G. Dataset, "Gtzan genre dataset," http://marsyas.info/downloads/datasets.html, 2002.

[5] AIRCConline, "Mobile deployment of mir models," https://aircconline.com, 2023.

[6] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002. [Online]. Available: http://marsyas.info/downloads/datasets.html

[7] V. Pandiyan, "Hybrid cnn-lstm design for music classification," *ResearchGate*, 2022, accessed: 2025-05-15. [Online]. Available: https://www.researchgate.net/profile/Vigneashwara-Pandiyan/publication/361640259

[8] K. Choi, G. Fazekas, and M. Sandler, "Convolutional recurrent neural networks for music classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. [Online]. Available: https://ar5iv.org

[9] J. N. et al., "Deep learning approaches for music audio classification: A review," *MDPI Applied Sciences*, 2019.

[10] S. H. et al., "Cnn architectures for large-scale audio classification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017. [Online]. Available: https://research.google.com

[11] A. Ghosal and M. H. Kolekar, "Music genre classification using cnn and cnn-lstm architectures," in *Interspeech*, 2018. [Online]. Available: https://www.isca-archive.org

[12] G. Research, "Mel spectrogram feature extraction," https://research.google, 2023.

[13] J. Doe and J. Smith, "Crnn architectures for music classification," https://ar5iv.org, 2023.

[14] I. Archive, "Sequence modeling in mir," https://www.isca-archive.org, 2023.

[15] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," https://arxiv.org/abs/1707.04916, 2002, known dataset flaws discussed in later studies.