KEC Conference 2026
ISSN: 3021-940X (Print)
DOI: 10.3126/injet.v3i2.95495

Nepali Music Genre Classification Using CNN-SVM Hybrid Architecture

Bibas Shrestha[1*], Darshan Deuja[2], Dhirendra Prasad Pant[3], Jenish Chapagain[4]

[1]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, bibasstha00@gmail.com*
[2]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, darshan.deuja.123@gmail.com*
[3]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur. Nepal, dhirendrapant82@gmail.com*
[4]*Department of Computer and Electronics Engineering, Kantipur Engineering College, Dhapakhel, Lalitpur, Nepal, chapagainjenish71@gmail.com*

Abstract

Nepali music genre classification using CNN–SVM hybrid model was developed to address the challenge of categorizing local genres such as Gazal, Lok Dohori, Nephop, and Pop. A dataset of 1,000 manually curated songs (250 per genre) was collected from YouTube, segmented into 30‑second clips at 25%, 50%, and 75% of each track's duration, resulting in approximately 3,000 audio segments. Each segment was converted into a 128×128 Log‑Mel spectrogram. A four‑layer CNN extracted a 64‑dimensional embedding, which was then passed to an SVM classifier. Experiments showed that the CNN–SVM hybrid with an RBF kernel achieved 88.29% accuracy, outperforming the standalone CNN baseline (84.28%). Among evaluated kernels, RBF and Linear both achieved the highest accuracy of 88.29%, while the Sigmoid kernel performed worst at 79.60%. The results demonstrate that combining deep learning feature extraction with a traditional machine learning classifier is effective for Nepali music genre classification on moderate‑sized, domain‑specific datasets.

Keywords: Music Genre Classification, CNN, SVM, Mel Spectrogram, Machine Learning, Audio Features.

1. Introduction

Music genre classification is a well-known task in music information retrieval. With the growth of digital music platforms, manually labeling and organizing songs by genre has become impractical, especially for regional music collections that receive little attention in existing research. Automated classification can help by enabling better music recommendation, search, and digital library management. Nepali music covers a range of distinct genres. Gazal features melodic and poetic compositions, Lok Dohori is a traditional folk style, Nephop is Nepali hip-hop with rhythmic beats and rap vocals, and Pop follows mainstream production styles.

Each genre has clear acoustic differences in tempo, rhythm, tonal quality, and instrumentation, which makes automated classification feasible from a signal-processing perspective. Most prior work in music genre classification has focused on Western datasets such as GTZAN [1], and no existing published model is designed specifically for Nepali music genres. This paper fills that gap by proposing a CNN–SVM hybrid system trained on a custom Nepali music dataset. The system converts audio signals into Log-Mel spectrograms, uses a CNN to extract features, and classifies those features using an SVM classifier. The best configuration reached 88.29% accuracy, outperforming the standalone CNN baseline (84.28%). The system is also wrapped into a simple web application that accepts audio uploads and returns the predicted genre.

2. Literature Review

Tzanetakis and Cook [1] carried out one of the earliest studies on automatic music genre classification accuracy on the GTZAN dataset. This work established a benchmark that later studies have built upon

* *Corresponding author*

Choi et al. [2] applied CNNs to genre classification using 2,000 audio files and achieved 65% accuracy. Their work showed that CNNs can learn useful spectral patterns from audio without extensive manual feature engineering.

Han et al. [3] combined CNN feature extraction with an SVM classifier and achieved 85.9% accuracy on GTZAN. Their work is the direct basis for our architecture. The CNN-SVM combination works well for moderate-sized datasets: the CNN learns compact audio representations while the SVM builds a strong, margin-based decision boundary.

Vigneshwar et.al (2024) carried out a comparative study on music genre classification using CNN and SVM models on the GTZAN dataset (10 genres of 30-second audio clips). They used spectral and temporal features (including MFCCs and spectrograms) for classification. The results showed that CNN achieved higher accuracy (~90%), while SVM achieved comparatively lower accuracy (~80%), but still performed well with handcrafted features. The study concluded that CNN is more effective for automatic feature learning, whereas SVM works well with engineered features depending on dataset size.

Pons et al. [4] designed CNN filters with domain knowledge of music signals, using vertical filters for timbral features and horizontal filters for temporal patterns. Their approach showed up to 6% accuracy improvement over standard CNNs, confirming that architecture design choices matter for audio tasks.

Meguenani et al. [5] explored large pretrained audio encoders such as WavLM and HuBERT paired with transformers, achieving 85.5% accuracy. While effective, these methods need large amounts of compute and pretrained data, which makes them hard to use for small regional datasets like ours.

Muller et al. critically reviewed existing datasets including GTZAN, FMA, and the Million Song Dataset. They found issues such as genre imbalance, mislabeled tracks, and data redundancy, and argued that many reported accuracies may be inflated due to dataset-specific overfitting. This is why we manually curated and verified our Nepali dataset instead of relying on automated scraping.

Table 1 summarizes how our proposed system compares to existing work. No prior work specifically targets Nepali music genre classification, which is the main gap this paper addresses.

Table 1: Comparison with existing methods

Author	Year	Method	Dataset	Accuracy	Limitation
Tzanetakis & Cook	2002	Spectral + Temporal features, GMM	GTZAN	61.4%	Hand-crafted features only
Choi et al.	2017	CNN on Mel Spectrograms	2000 audio files	65%	No Hybrid approach
Han et al.	2018	CNN + SVM Hybrid	GTZAN	85.9%	Western genres only
Vigneshwar et.al	2024	CNN + SVM Hybrid	GTZAN	Around 90%	High complexity
Meguenani et al.	2024	LLM encoders + Transformers	Multiple Datasets	85.5%	Not suitable for regional songs.
Proposed	2025	CNN + SVM (RBF)	Nepali Datasets (4 Genres)	88.29%	Limited to 4 genres.

3. Methodology

This paper uses Mel-spectrogram images from audio files as input for genre classification. A hybrid CNN-SVM architecture is used, where the CNN extracts high level features from spectrograms and the SVM performs the final classification. The dataset is split into training (80%), validation (10%), and testing (10%). The trained models are compared using accuracy, precision, recall, and F1 score to find the best configuration for automatic Nepali Music Genre Classification.

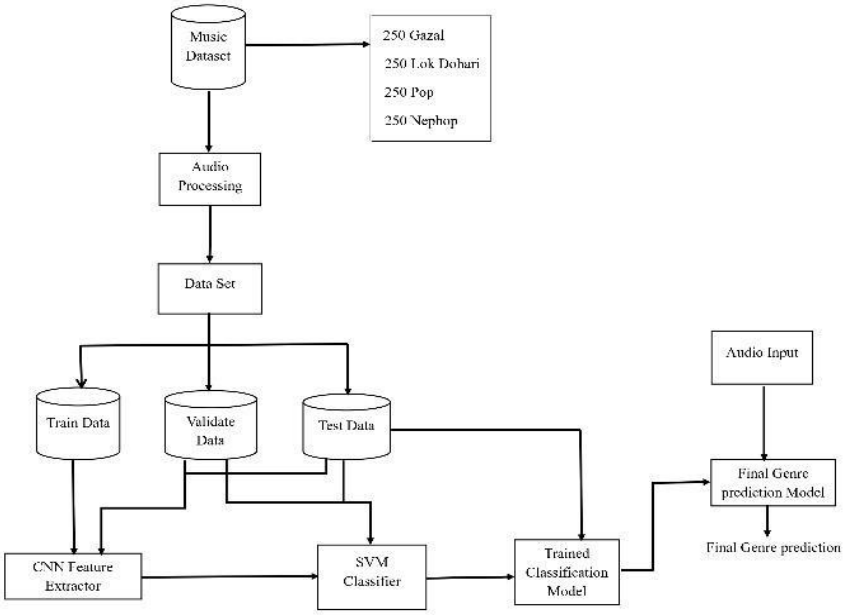

Figure 1. CNN-SVM Hybrid Architecture

3.1 Dataset Description

The dataset is a manually curated collection of Nepali music tracks grouped into four genres: Gazal, Lok Dohori, Nephop, and Pop. Each genre contains 250 songs, totaling 1,000 audio tracks. Songs were collected from publicly available YouTube sources. The collection process involved searching by artist name and checking YouTube titles and descriptions for explicit genre tags. For example, a video titled "Jhyalaima Aina | New Lok Dohori Song 2078" was placed in the Lok Dohori category based on the genre tag in the title. The same approach was applied across all four genres, and all tracks were manually verified to ensure correct labeling.

The dataset is split at the song level to prevent data leakage, i.e. segments from the same song cannot appear in more than one split. The split is 80% training, 10% validation, and 10% testing (approximately 200 songs per genre for training and 25 each for validation and testing).

Table 2: Dataset Split Summary

Split	**Songs per Genre**	**Percentage**
Training	~200	~80%
Validation	~25	~10%
Testing	~25	~10%
Total	**250 per genre**	**100%**

3.2 Audio Segmentation

Each audio track is segmented into three fixed 30 second chunks extracted at 25%, 50%, and 75% of the total track duration. This strategy was chosen for two reasons. First, different parts of a song can carry different musical characteristics i.e. the intro, the mid-section, and the final section may differ in instrumentation and energy. By sampling at fixed relative positions, the model sees diverse segments from each song. Second, segmentation multiplies the number of training samples from 1,000 tracks to approximately 3,000 segments, which is important for training a model with limited data.

A segment is only used if the full 30-second window fits within the track duration. Any segment whose extraction window extends beyond the end of the track is skipped to avoid incomplete data. This keeps the segments acoustically clean and representative of real musical content.

3.3 Audio Augmentation

To improve the robustness and generalization capability of the music genre classification model, several augmentation techniques are applied to the raw audio signals before feature extraction such as:

3.3.1 Pitch Shifting

This transformation shifts the pitch of the audio signal without altering its duration. It allows the model to learn invariant representations across different pitch variations.

$$audio' = PitchShift(audio, steps) \quad \text{(Equation 1)}$$

3.3.2 Time Stretching

Time stretching modifies the playback speed of the audio signal without affecting its pitch.

$$audio' = TimeStretch(audio, rate) \quad \text{(Equation 2)}$$

3.3.3 White Noise Addition

Random Gaussian noise is added to simulate background noise present in real-world recordings.

$$noise \sim N(\mu, \sigma^2) audio' = audio + noise \quad \text{(Equation 3)}$$

3.3.4 Random Gain Adjustment

This transformation randomly adjusts the amplitude of the signal to simulate variations in recording volume.

$$audio' = audio \times gain \quad \text{(Equation 4)}$$

3.4 Feature Extraction

To represent audio signals in a format suitable for machine learning models, the audio segments are converted into Mel spectrograms, which capture both temporal and spectral characteristics of the signal.
The audio waveform is first transformed from the time domain to the frequency domain using the Short-Time Fourier Transform (STFT):

$$X(\tau, \omega) = \int_{-\infty}^{\infty} x(t)\, w(t-\tau) e^{-j\omega t} dt \quad \text{(Equation 5)}$$

The resulting spectrogram is then mapped to the Mel frequency scale, which reflects human auditory perception.

$$Mel(f) = 2595 log_{10}(1 + \frac{f}{700}) \quad \text{(Equation 6)}$$

The Mel spectrogram is converted to a logarithmic scale and resized to 128 × 128 pixels. Each spectrogram is normalized using z-score normalization and used as the input representation for the CNN model.

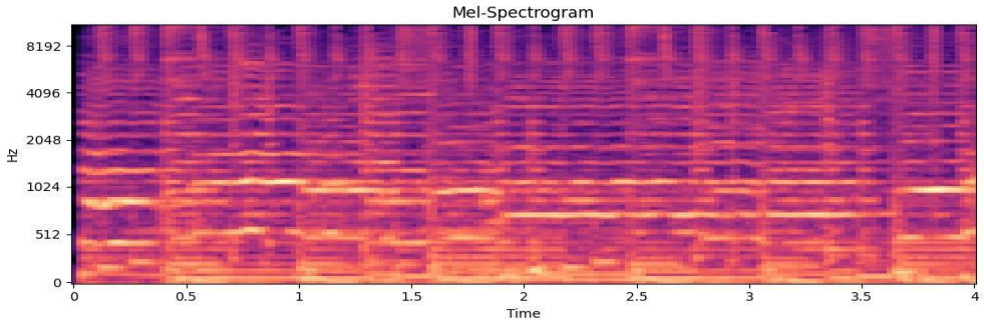

Figure 2. Mel Spectrogram for Sample Image

3.5 CNN Architecture

The CNN model has four convolutional blocks. Each block contains a Conv2D layer (filter sizes 32, 64, 128, and 256 in successive blocks), followed by Batch Normalization, MaxPooling (2×2), and Dropout. L2 regularization (weight decay = 1e-4) is applied throughout to reduce overfitting.

The number of layers was selected based on the input size (128×128) and the desired feature dimension. Four blocks of MaxPooling reduce the spatial dimensions from 128×128 down to 8×8, after which Global Average Pooling (GAP) collapses the spatial dimensions entirely. GAP was chosen over Flatten because it produces a compact representation and is more resistant to overfitting on moderately sized datasets. A Dense layer with 64 units then produces the final feature embedding, which is passed to the SVM classifier.

3.6 SVM Classifier

The SVM classifier takes the 64-dimensional CNN features embedding as input. Four kernels are evaluated: RBF, Linear, Sigmoid and Polynomial. The RBF kernel is well suited to the CNN embeddings because it maps features into an infinite-dimensional space, capturing nonlinear genre boundaries without requiring explicit feature engineering.

3.7 Hyperparameters

Table 3: Hyperparameters Used

Parameter	**Value**
Epoch	40 (Early stopping applied)
Batch size	32
Optimizer	Adam
Learning rate	0.0001
Sample Rate	22050 Hz
Segment Duration	30 seconds
Number of Mels	128
Input Shape	128 × 128 × 1
FFT Size	2048
Hop Length	512
Frequency Min	20 Hz
Frequency Max	11025 Hz
Weight Decay (L2 Regularization)	1e-4
Dropout Rates	0.3, 0.4, 0.5
Label Smoothing	0.1
Learning Rate Scheduler	ReduceLROnPlateau
Early Stopping Patience	3
SVM Kernel	RBF, Linear, Polynomial, Sigmoid

SVM Regularization (C)	1 and 10
Gamma	Scale

Table 3 summarizes the hyperparameters used across all experiments. The chosen hyperparameters are optimized for efficient audio signal processing and deep learning model training.

3.8 Evaluation Metrics

The performance of the proposed classification system is evaluated using a testing dataset. The model is assessed using common classification metrics like Accuracy, Precision, Recall and F1-Score. In addition, a confusion matrix is used to analyze the classification results and understand the distribution of correct and incorrect predictions across the four music genres.

4. Results and Discussion

4.1 CNN-Only Model Results

Figure 3 shows the training and validation accuracy curves for the standalone CNN. Validation accuracy rises steeply in the first two epochs and stabilizes near 90%, showing the CNN picks up useful spectral representations quickly. Figure 4 shows the corresponding loss curves. Both training and validation loss decrease steadily across epochs, with no notable divergence, indicating the model trains without significant overfitting. Early stopping triggered after 7 epochs for this run.

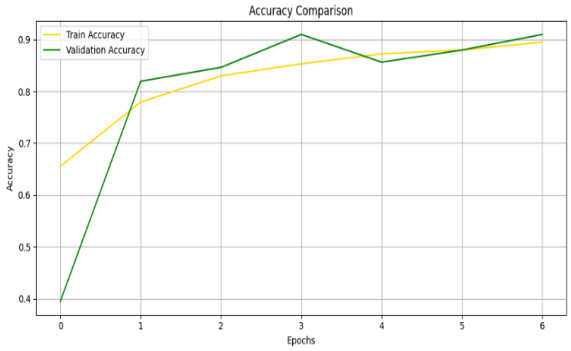

Figure 3. Training and Validation Accuracy (CNN-Only)

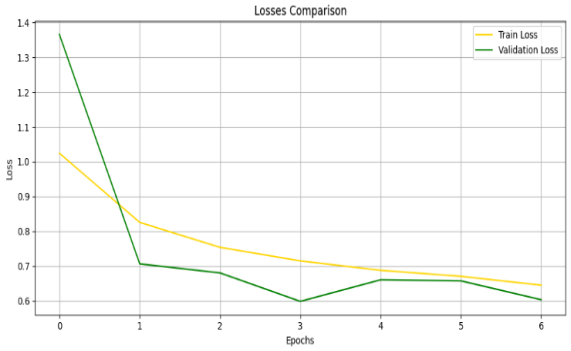

Figure 4. Training and Validation Loss (CNN-Only)

Figure 5 shows the CNN-only confusion matrix. Lokdohori and Nephop are classified almost perfectly, with 73 out of 75 samples correct for both genres. Gazal has the most errors. 18 Gazal samples are misclassified as Pop. This is a recurring pattern across all model configurations and likely reflects acoustic overlap between Gazal and Nepali Pop, which can share similar melodic and production styles. Pop itself also shows some errors, with 9 samples misclassified as Nephop and 4 as Gazal.

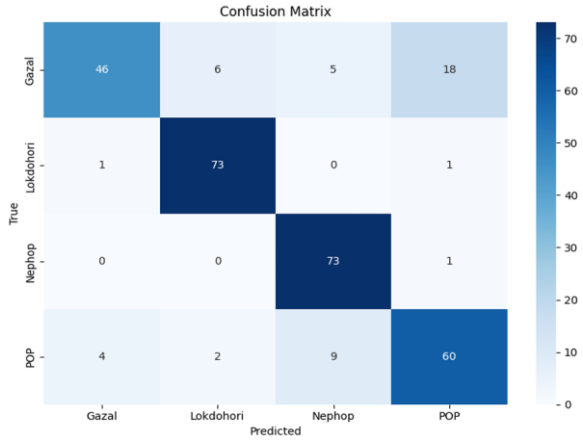

Figure 5. Confusion Matrix for CNN-Only Model

4.2 CNN-SVM Hybrid Model Results

Table 4 compares all five model configurations on the test set. Both CNN+SVM with RBF and Linear kernels achieved 88.29% accuracy, outperforming the standalone CNN by 4 percentage points. The Polynomial kernel achieved 85.95% and the Sigmoid kernel performed worst at 79.60%. This shows that the SVM kernel choice has a significant impact — margin-based kernels (RBF and Linear) work much better than the Sigmoid kernel for this feature space. The consistency between RBF and Linear results suggests the CNN features are already well-structured enough that both linear and nonlinear SVM boundaries are effective.

Table 4: Model Performance Comparison

Model	Accuracy (%)	Recall	Precision	F1-score
CNN (Softmax)	84.28%	0.84	0.85	0.84
CNN + SVM (RBF)	**88.29%**	**0.88**	**0.88**	**0.88**
CNN + SVM (Linear)	88.29%	0.88	0.88	0.88
CNN + SVM (Polynomial)	85.95%	0.86	0.86	0.86
CNN + SVM (Sigmoid)	79.60%	0.80	0.80	0.79

Figures 6 through 9 show the confusion matrices for each SVM kernel. The RBF kernel (Figure 6) correctly classifies 55 of 75 Gazal samples, 71 of 75 Lokdohori, 74 of 74 Nephop, and 63 of 75 Pop samples. The Linear kernel (Figure 7) shows a similar pattern. The Polynomial kernel (Figure 8) is slightly weaker on Gazal. The Sigmoid kernel (Figure 9) shows notably more errors, especially for Gazal where only 39 out of 75 samples are correctly classified. Nephop is classified near-perfectly across all kernels, which makes sense given its distinct rhythmic and vocal characteristics that are acoustically very different from Gazal and Lok Dohori.

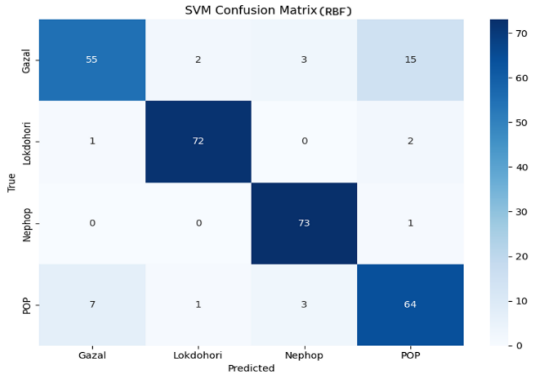

Figure 6. Confusion Matrix for CNN+SVM (RBF Kernel)

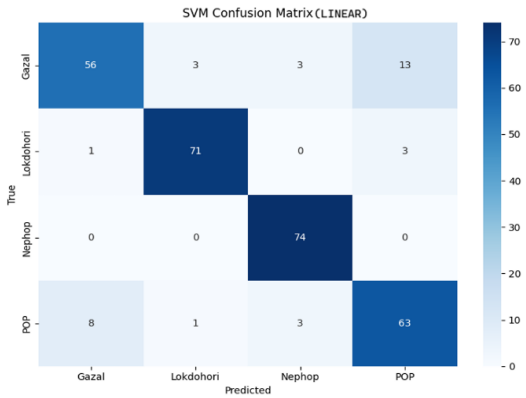

Figure 7. Confusion Matrix for CNN+SVM (Linear Kernel)

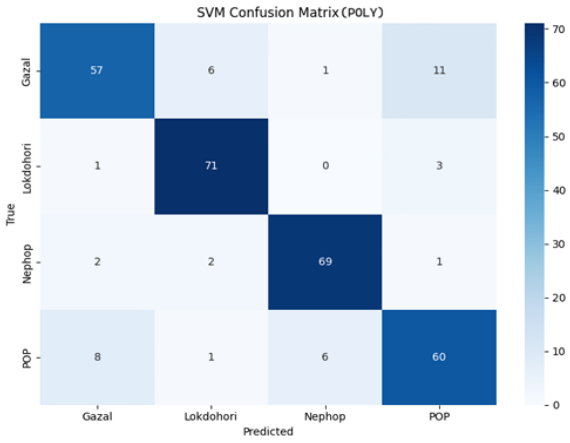

Figure 8. Confusion Matrix for CNN+SVM (Polynomial Kernel)

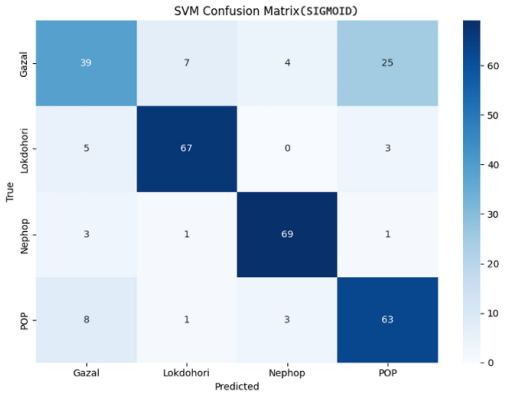

Figure 9. Confusion Matrix for CNN+SVM (Sigmoid Kernel)

Table 5: Per-Class Accuracy Across All Models

Model	**Gazal**	**Lok Dohori**	**Nephop**	**Pop**	**Overall Accuracy**
CNN Only	61.3%	97.3%	98.6%	80%	84.28%
CNN + SVM (RBF)	73.3%	96%	98.6%	85.3%	88.29%
CNN + SVM (Linear)	74.7%	94.7%	100%	84%	88.29%
CNN + SVM (Polynomial)	76%	94.7%	93.2%	80%	85.95%
CNN + SVM (Sigmoid)	52%	89.3%	93.2%	84%	79.6%

4.3 Ablation Study

To evaluate the individual contributions of audio segmentation and data augmentation, four configurations of the CNN+SVM (RBF) system were trained and tested on the same Nepali music dataset. Table 6 summarizes the results.

Table 6: Ablation Study Results

Configuration	**CNN-Only**	**CNN + SVM**	**Segmentation**	**Augmentation**
No Segmentation, No Augmentation	23%	65%	No	No
No segmentation, with Augmentation	80%	77%	No	Yes
With Segmentation, No Augmentation	24.75%	81.94%	Yes	No
With Segmentation and Augmentation	**84.28%**	**88.29%**	**Yes**	**Yes**

The results show that both segmentation and augmentation are critical for this dataset. Without either component, the CNN-only model collapses to near-random performance (23%), and even the SVM recovers only partially (65%), indicating that 1,000 unsegmented, unaugmented samples are insufficient for the CNN to learn meaningful spectral representations. Adding augmentation alone (No Seg + Aug) raises CNN-only accuracy to 80% and SVM to 77%, confirming that augmentation provides meaningful diversity even without increasing the number of segments. Adding segmentation alone (Seg + No Aug) brings CNN-only accuracy to just 24.75%, but the SVM recovers strongly to 81.94%, suggesting that segmentation expands the sample count enough for the SVM to find useful decision boundaries even when the CNN features are weak. The full system combining both segmentation and augmentation achieves the best results: 84.28% for CNN-only and 88.29% for CNN+SVM (RBF), confirming that the two components are complementary rather than redundant.

5. Conclusion and Future Enhancements

This paper presents a Nepali music genre classification system using a hybrid CNN-SVM architecture trained on a balanced dataset of 1,000 manually curated Nepali songs across four culturally distinct genres: Gazal, Lok Dohori, Nephop, and Pop. Log-Mel spectrograms of size 128×128 are used as input representations. A four-layer CNN with progressively increasing filter sizes (32, 64, 128, 256) serves as a feature extractor, producing compact 64-dimensional embeddings that are classified by an SVM with RBF kernel.

The hybrid CNN+SVM model achieves a test accuracy of 88.29%, outperforming the CNN-only baseline by 4.01 percentage points (84.28%). Among the five SVM kernels evaluated, RBF and Linear achieved the best performance, while Sigmoid showed the lowest at 79.60%. The ablation study confirms that both multi position audio segmentation and augmentation are essential contributors to the system's performance.

To the best of the authors' knowledge, this represents one of the first systematic efforts toward automated Nepali music genre classification, addressing a significant gap in regional Music Information Retrieval. The system is deployed as a real-time Streamlit web application, enabling practical genre prediction from uploaded or YouTube-sourced audio files. Future work will explore:

(i) Expanding the Nepali Music Dataset.
(ii) Addition of more Music Genres.
(iii) Real-time Music Genre Detection.

Acknowledgement

The authors express sincere gratitude to Er. Nirajan Acharya and Er. Aliz Shrestha, Department of Computer Engineering, Kantipur Engineering College, for their guidance, encouragement, and constructive suggestions throughout this research. The authors also thank the faculty of the Department of Computer and Electronics Engineering for their continued support. Finally, the authors acknowledge the open-source community behind TensorFlow, scikit-learn, Librosa, and Streamlit, which formed the technical foundation of this work.

Reference

[1] Changsheng Xu, Namunu Maddage, Xi Shao, Fang Cao, and Qi Tian, "Musical genre classification using support vector machines," in IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003, pp. V-429.

[2] Raj Pattanaik and Answeta Jaiswal, "Classification of music genre using support vector machine and convolutional neural network," World Journal of Advanced Research and Reviews, vol. 21, pp. 1009-1019, Mar. 2024.

[3] Andri Pranolo et al., "Classification of Music Genres based on Machine Learning SVM and CNN," in International Conference on Pervasive Computing and Social Networking (ICPCSN), May 2025, pp. 1667-1670.

[4] 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.

[5] G. Fazekas, M. Sandler, and K. Cho K. Choi, "Convolutional recurrent neural networks for music classification," in International Conference on Acoustics, Speech and Signal Processing, 2017, pp. 2392-2396.

[6] F. Han, G. Song, and Z. Wang Wu, "Music genre classification using independent recurrent neural network," Chinese Automation Congress (CAC), pp. 192-195, 2018.

[7] A. de Souza Britto, and A. L. Koerich M. E. A. Meguenani, "Music genre classification using large language models," in Symposium on Computational Intelligence in Image, Signal Processing and Synthetic Media (CISM), 2025, pp. 1-7.