SONG MOOD CLASSIFIER: A HYBRID DUAL-BRANCH MODEL

Combining Audio Signal Processing and Natural Language Processing for Automated Music Emotion Recognition

Saurabh Gupta, Prof. Nitin Goyal
Student, Professor
Department of Computer Science and Engineering,
R.D. Engineering College, Ghaziabad, India

Abstract: Music streaming platforms now serve hundreds of millions of listeners worldwide, yet the underlying systems that tag and recommend songs still lean heavily on manual labeling or metadata-driven heuristics, an approach that is both subjective and difficult to scale. This paper presents a Song Mood Classifier built on a Hybrid Dual-Branch architecture that combines acoustic signal processing and natural language processing. The system runs two parallel analysis pipelines: an audio branch powered by a Convolutional Neural Network (CNN) trained on spectrograms and Librosa-extracted features, alongside a lyric branch employing a transformer-based NLP model (BERT) to analyze the emotional content of song text. Outputs from both branches are combined through a dense fusion layer to produce a unified mood prediction across four categories: Happy, Sad, Calm, and Energetic. We frame mood misclassification as a data fusion engineering problem and demonstrate that the hybrid model improves accuracy by rethinking how audio and text signals are processed and merged.

Index Terms - Music Mood Classification, CNN, BERT, NLP, Hybrid Model, Audio Processing, Sentiment Analysis, Deep Learning.

I. INTRODUCTION

Music streaming platforms now serve hundreds of millions of listeners worldwide, yet the underlying systems that tag and recommend songs still lean heavily on manual labeling or metadata-driven heuristics, an approach that is both subjective and difficult to scale.
In this paper, we examine a recurring inefficiency in modern music platforms that we term mood classification latency: the mismatch between what a song actually conveys emotionally and the label an automated system assigns to it. This mismatch, we argue, is not the product of any single technical flaw. Rather, it stems from the mono-modal design philosophy that dominates existing classifiers, these systems examine either acoustic features or lyric sentiment, but never both at the same time.

At a technical level, this limitation manifests in two distinct ways. The first is feature incompleteness: systems that extract audio characteristics such as tempo and MFCCs while completely disregarding the lyrical and textual layer of a song. The second is what we call semantic blindness, systems that parse lyric sentiment but remain entirely unaware of how the music is actually performed, ignoring rhythm, pitch, and energy. When a song's audio and lyrical signals pull in different directions, these classifiers routinely assign the wrong mood.

To address this, we propose a Song Mood Classifier built on a Hybrid Dual-Branch architecture. The system runs two parallel analysis pipelines: an audio branch powered by a Convolutional Neural Network (CNN) trained on spectrograms and Librosa-extracted features, alongside a lyric branch that employs a transformer-based NLP model (BERT) to analyze the emotional content of song text. Outputs from both branches are then combined through a dense fusion layer to produce a unified mood prediction.

We target four mood categories: Happy, Sad, Calm, and Energetic. Rather than processing audio and lyrics through separate, disconnected pipelines, our system aligns acoustic signal features with semantic lyric content at a shared prediction stage. From an engineering standpoint, this demands parallel handling of two fundamentally different data types i.e. audio spectrograms and tokenized text before late-stage fusion.

That said, the approach comes with real constraints. Classification quality is closely tied to the size and labeling consistency of the training data we have access to. The current model is bounded by four to five discrete mood categories and cannot yet capture more nuanced emotional states like bittersweet or nostalgic. Real-time audio inference, while a natural next step, falls outside the current implementation scope.

We frame mood misclassification as a data fusion engineering problem rather than a purely statistical one. The hybrid model improves accuracy by rethinking how audio and text signals are processed and merged while being transparent about the data limitations and architectural trade-offs that affect real-world deployment.

II. LITERATURE REVIEW: EVOLUTION OF MUSIC MOOD CLASSIFICATION

Automated music mood classification did not arrive at its current limitations all at once. It evolved in stages, with each generation solving one problem while quietly creating another. The result is a field where powerful individual tools exist, but genuine multimodal understanding remains largely absent.

A. Generation 1.0: Audio-Only Classification

Early automated music analysis was built entirely around acoustic signals. Researchers in this period extracted hand-crafted audio features like Mel-Frequency Cepstral Coefficients (MFCCs), tempo, spectral contrast, and chroma, and passed them into classical machine learning classifiers like Support Vector Machines (SVMs) and Gaussian Mixture Models (GMMs). These methods worked reasonably well for genre classification, but fell short for mood, where emotional nuance is difficult to capture through acoustic features alone. Without semantic context, acoustic analysis cannot fully account for how humans emotionally experience music.

B. Generation 2.0: Lyric-Only NLP Models

The second wave introduced lyric analysis. NLP models including Recurrent Neural Networks (RNNs) and transformer architectures like BERT were applied to song text to extract sentiment and emotional cues. Systems built around lyric analysis performed well on songs with explicitly emotional language. However, they were fundamentally blind to musical delivery. A melancholic poem set against a driving, upbeat melody would consistently be labeled sad, simply because the system never heard the music it was classifying.

C. Commercial Platforms: Metadata Without Intrinsic Analysis

Major commercial platforms, including Spotify and Apple Music, took a different path. Rather than analyzing audio or lyrics directly at inference time, they relied on Collaborative Filtering — learning from large volumes of user listening behavior — supplemented by metadata such as genre tags, artist information, and release year. At scale, this worked surprisingly well. Recommendation engines produced results that users found broadly satisfying, and the approach was commercially defensible. But it has a fundamental flaw: these systems learn what users have already listened to, not what a song actually contains. A track released last week, with no listening history, gets classified poorly or not at all. A song that an artist labeled a particular genre may carry emotional content that diverges entirely from genre expectations. This cold-start problem is the Achilles' heel of behavior-driven classification. Without prior engagement data, the system has no real basis for categorization. It's commercially workable but intrinsically incomplete. However, these systems learn what users have already listened to, not what a song actually contains. A track released last week, with no listening history, gets classified poorly or not at all. This cold-start problem is the Achilles' heel of behavior-driven classification.

D. Critical Gaps Across Existing Models

Stepping back across all three generations, certain structural weaknesses appear consistently. The most pervasive is mono-modal analysis. A related problem is weak hybrid fusion — the small number of systems that do attempt multimodal classification typically perform early fusion, concatenating raw features before training begins. Additionally, almost all classifiers predict a single mood label from a narrow predefined set, failing to capture the emotional complexity of music.

E. Positioning the Song Mood Classifier Within This Context

The Song Mood Classifier we propose here represents a structural rethinking of the problem. Rather than treating audio analysis, lyric analysis, and mood prediction as sequential steps in a pipeline, the hybrid model processes them through a unified dual-branch architecture. Each input song generates both a spectrogram and a tokenized lyric representation; these are processed independently through a CNN and an NLP branch before their probability vectors are concatenated and passed to a shared fusion layer. This late fusion approach gives each branch the space to build a complete representation of its own modality.

III. PROPOSED SYSTEM ARCHITECTURE: THE HYBRID DUAL-BRANCH MODEL

The Song Mood Classifier is designed to address classification failure at the architectural level. Rather than adding features onto an existing approach, we restructured how two fundamentally different data types — audio signals and natural language — are processed and combined into a single mood prediction. The model follows a Hybrid Dual-Branch design with a shared fusion layer at the output stage.

A. Architectural Overview

The system is organized into three components that work in parallel:

- Audio Branch (Branch 1) — Processes the audio file
- Lyric Branch (Branch 2) — Processes the lyric text
- Fusion Layer — Combines outputs and produces final mood classification

This separation is deliberate. Each branch is allowed to specialize i.e. to learn the patterns most relevant to its modality without interference from the other. The two branches communicate only through probability vector concatenation at the fusion stage, ensuring that the final classification has access to the full predictive output of both modalities before a decision is made.

B. Audio Branch (Branch 1): Feature Extraction and CNN

Feature extraction is handled by the Librosa library. For each audio file, we extract four types of features:

- MFCCs (Mel-Frequency Cepstral Coefficients) — Capture timbral texture
- Chroma Features — Represent harmonic and tonal content
- Spectral Contrast — Measures the difference between peaks and valleys in the spectrum
- Tempo — Captures rhythmic pace in beats per minute

The audio file is also converted into a Mel spectrogram, a time-frequency representation that captures how audio energy is distributed across frequencies over time. This spectrogram is treated as an image and fed into a Convolutional Neural Network, which learns to recognize spatial patterns in the audio signal. The CNN architecture includes convolutional layers for feature map extraction, followed by pooling layers to reduce spatial dimensionality, and dense layers that output a probability distribution across the target mood classes.

C. Lyric Branch (Branch 2): NLP and Sentiment Analysis

The lyric branch processes the song text through a four-step pipeline:

- Preprocessing — Tokenization, stop-word removal, and normalization
- Embedding — Text is passed into a pre-trained BERT transformer model
- Sentiment and Emotion Extraction — The model outputs contextual representations of the lyric's emotional content
- Probability Output — A probability distribution over mood classes is generated

BERT's bidirectional attention mechanism is particularly well-suited to lyric analysis. It captures not just the sentiment of individual words, but the emotional meaning that emerges from the sequence as a whole — which matters enormously for music, where irony, metaphor, and non-literal language are commonplace. For deployments where computational resources are constrained, a lighter Tf-Idf vectorization approach via Scikit-learn can substitute for BERT, trading some accuracy for faster inference.

D. Fusion Layer: Combining the Branches

The probability vectors produced by the CNN (audio branch) and the NLP model (lyric branch) are concatenated into a single combined feature vector, which is then passed through a set of dense fully-connected layers. These layers learn, through training, how to weight the contributions

of each modality and produce a final mood label. This late fusion strategy preserves each branch's independently developed representation. The output is a single mood label drawn from our target set: Happy, Sad, Calm, or Energetic.

IV. THE CLASSIFICATION ALGORITHM: MOOD SCORING LOGIC

Not all songs fall neatly into one emotional category. We address classification ambiguity through a two-stage process: independent branch scoring followed by learned fusion.

A. Problem Definition

Single-modality classification does not generalize well. Audio-only classification misses lyric meaning; lyric-only classification ignores musical performance. A genuine solution must evaluate both signals independently and then combine them in a principled way.

B. Input Parameters

Four types of parameters feed into the prediction:
- Audio Features (MFCCs, Chroma, Spectral Contrast, Tempo) — Captured via Librosa and processed through the CNN branch
- Spectrogram Image — Visual representation of frequency patterns over time
- Lyric Text — Preprocessed and embedded through the BERT or Tf-Idf lyric branch
- Branch Confidence Scores — Each branch outputs a probability distribution representing classification confidence per mood class

C. Algorithm Design

The classification algorithm operates as a dual-branch probability fusion model. Each branch independently produces a probability vector across the four mood categories. A simplified representation of the fusion logic is:

Final Mood Score = Dense(Concat(P_audio, P_lyric))

In this formulation, P_audio is the probability vector from the CNN audio branch and P_lyric is the probability vector from the NLP lyric branch. Crucially, the fusion weights are not manually set, they are learned during training through backpropagation, which means the model discovers the optimal balance between audio and lyric signals from the data itself.
The full inference process proceeds as follows:
- Receive audio file and corresponding lyric text
- Extract audio features using Librosa; generate spectrogram
- Preprocess lyric text; tokenize and embed using BERT
- Pass audio features through CNN branch; obtain P_audio
- Pass lyric embeddings through NLP branch; obtain P_lyric
- Concatenate P_audio and P_lyric; pass through fusion dense layers
- Output final mood classification label

V. TOOLS, TECHNOLOGIES, AND DATASET

A. Dataset

We train the system on two publicly available, labeled music datasets:
- DEAM (Database for Emotional Analysis of Music) — Contains audio tracks with valence and arousal labels that can be mapped to discrete mood categories
- Million Song Dataset — A large-scale dataset providing audio features and metadata for over one million songs
- Corresponding lyric data — Retrieved via lyric APIs and aligned with the audio dataset by track ID

Preprocessing involves cleaning and aligning audio files with their corresponding lyrics and mood labels. Where datasets provide continuous valence/arousal scores, we map these to our four discrete categories: Happy, Sad, Calm, and Energetic.

B. Backend and Core Logic

Python serves as the primary programming language, chosen for its mature machine learning ecosystem and broad library support. Data handling relies on Pandas for tabular operations and NumPy for numerical computation.

C. Audio Processing

Librosa is the backbone of the audio processing pipeline. It handles MFCCs, chroma features, spectral contrast, tempo extraction, and Mel spectrogram generation. Audio files are loaded, normalized, and converted into fixed-length feature vectors suitable for model input.

D. Machine Learning and Model Building

The two branches draw on different model architectures suited to their respective data types:
- CNN (Convolutional Neural Network) — Built using TensorFlow and Keras; processes spectrogram images and extracted audio features to learn acoustic mood patterns
- BERT / RNN — Implemented using the HuggingFace Transformers library (for BERT) or Scikit-learn with Tf-Idf vectorization (for the lighter RNN variant); processes lyric text for sentiment and emotion classification

The fusion layer is implemented as a multi-layer dense network in Keras, combining the probability vectors from both branches into a final classification.

E. Frontend Prototype

For the user-facing prototype, we built a Streamlit web application. Users can upload an audio file, enter or paste corresponding lyrics, and receive a mood classification in real time.

F. Database

Dataset storage relies primarily on CSV and JSON files. A lightweight SQLite database is used where needed to track model evaluation results across experimental runs.

VI. SOCIO-TECHNICAL CONSIDERATIONS: WHY MUSIC CLASSIFIERS MISS

Most mood classifiers do not fail because of poorly written code. They fail because the emotional categories used to label music are culturally situated, contextually dependent, and vary meaningfully across individual listeners.

A. Subjectivity of Mood Labels

The mood labels in training datasets are assigned by human annotators whose judgments are shaped by their cultural backgrounds, personal histories, and the specific annotation guidelines they were given. Disagreements between annotators are common, particularly for songs with complex or ambiguous emotional content. This introduces noise directly at the data level — a model trained on ambiguous labels will learn ambiguous category boundaries.

B. The Lyric-Audio Divergence Problem

A number of songs deliberately exploit a mismatch between their audio and lyric emotional content. Upbeat, high-energy instrumentation layered over melancholic or ironic lyrics is a well-established artistic device. Mono-modal classifiers cannot handle this — they commit to one channel or the other, producing a label that captures only half of what the artist intended. Our hybrid model partially addresses this through the fusion layer, which learns to navigate conflicting signals.

C. Language and Cultural Scope

BERT's lyric branch performs best on English-language text. For music catalogs that span multiple languages, extending this approach would require fine-tuning on language-specific data or adopting a multilingual transformer architecture like mBERT or XLM-RoBERTa — both of which add computational cost and pipeline complexity.

VII. IMPLEMENTATION ROADMAP: FROM DESIGN TO DEPLOYMENT

Building the Song Mood Classifier requires a staged development process — methodical, incremental, and honest about what each phase is meant to validate. We organize the roadmap into four phases, each targeting a specific layer of risk: data integrity, model correctness, integration quality, and empirical validation.

A. Phase 1: Data Collection and Preprocessing

Phase 1 focuses on assembling and aligning the training dataset:

- Collect audio tracks from DEAM and Million Song Dataset
- Retrieve corresponding lyrics and align with audio files by track ID
- Map continuous valence/arousal labels to discrete mood categories: Happy, Sad, Calm, Energetic
- Normalize audio files to consistent sample rate and duration
- Apply NLP preprocessing to lyric text: tokenization, stop-word removal, and case normalization

This phase is data-intensive and unglamorous. It also determines the ceiling on everything that follows.

B. Phase 2: Model Development (Branch Training)

Once the data is in order, we train each branch independently before attempting fusion. The audio CNN is trained on spectrogram images and Librosa-extracted features. The lyric BERT or RNN model is fine-tuned on the labeled lyric dataset. We evaluate each branch independently using accuracy and F1 score to confirm that both achieve meaningful performance above baseline before proceeding. Specifically, this phase involves:

- Audio CNN is tested for accuracy on held-out audio samples
- Lyric NLP model is evaluated on held-out lyric samples
- Initial baseline metrics are recorded for each individual branch

C. Phase 3: Fusion and Integration

Once both branches have been validated individually, we integrate them into the full hybrid model. The fusion dense layer is trained on the concatenated probability vectors from both branches. The key objectives for this phase are:

- Train fusion layer on combined branch outputs
- Evaluate hybrid model accuracy against individual branch baselines
- Identify cases where fusion improves or degrades classification
- Test Streamlit UI with end-to-end song upload and mood prediction

D. Phase 4: Evaluation and Reporting

After successful integration, we evaluate the full system comprehensively. Evaluation involves:

- Computing accuracy, precision, recall, and F1 score for each mood class
- Generating a confusion matrix to identify systematic misclassifications (e.g., Calm confused with Sad)
- Comparing hybrid model performance against audio-only and lyric-only baselines
- Documenting cases of lyric-audio divergence and model behavior on ambiguous inputs

The central hypothesis to validate: does the hybrid model achieve measurably higher classification accuracy than either individual branch alone?

VIII. RESULTS (EXPECTED)

Once the prototype is complete, this section will present full performance metrics and qualitative observations from end-to-end testing.

A. Model Performance Comparison

We expect results across three model configurations as shown in Table 1 below.

Table 1. Model Performance Comparison

Model	Input Modality	Expected Accuracy
Audio-Only (Baseline)	Audio (CNN on Spectrogram)	[To be populated]
Lyrics-Only (Baseline)	Lyric Text (BERT/RNN)	[To be populated]
Hybrid Model (Proposed)	**Audio + Lyrics (CNN + BERT)**	**Expected Highest**

B. Final Evaluation Metrics

We will report the following metrics upon completion:

- Confusion Matrix — To visualize per-class accuracy and identify systematic misclassifications between mood categories
- Accuracy — Overall percentage of correctly classified samples across all mood categories
- F1 Score — Harmonic mean of precision and recall per class, providing a balanced performance measure particularly useful given potential class imbalance in the dataset

IX. CONCLUSION AND FUTURE SCOPE

A. Conclusion

This paper presents a hybrid AI model for automated song mood classification that targets the core limitation of mono-modal analysis prevalent in existing systems. By combining acoustic signal processing with natural language understanding of lyrics, the proposed classifier develops a more complete picture of a song's emotional content, one that more closely approximates how human listeners actually experience music.

The dual-branch architecture, with a CNN handling audio spectrograms and a BERT-based NLP model handling lyric text, allows the system to analyze both what is being performed and what is being communicated. Late-stage fusion then allows the model to reconcile any tension between these two signals and arrive at a mood classification that neither branch could reliably produce on its own. The system has practical applications in personalized music recommendation, automatic playlist generation, and richer music discovery features for streaming platforms.

B. Limitations

Several constraints limit the current implementation:

- Model performance is strongly dependent on the quality, size, and annotation consistency of the labeled training dataset. Noisy or ambiguous mood labels in the training data will degrade classification boundaries.
- Classification is limited to four to five discrete mood categories (Happy, Sad, Calm, Energetic). Complex or nuanced emotional states bittersweet, nostalgic, anxious are outside the current scope.
- The lyric branch performs optimally on English-language text. Multilingual music classification requires additional model fine-tuning or multilingual transformer architectures.

C. Future Scope

We identify the following as high-priority directions for future work:

- Multi-label Classification — Enhance the model to predict multiple simultaneous moods for a single song, reflecting the emotional complexity of music more accurately
- Real-time Analysis — Adapt the model architecture for real-time mood detection from microphone input, enabling live music mood identification
- Personalization — Integrate the model with individual user profiles to learn personal mood-music associations, producing mood classifications calibrated to listener preference
- Multilingual Support — Extend the lyric branch to support non-English languages using multilingual transformer models

X. REFERENCES

[1] 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.

[2] Y. Hu and L. Chen, "Lyric-based song sentiment classification with sentiment vector space model," in Proc. 12th Int. Society for Music Information Retrieval (ISMIR) Conf., 2011.

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv:1810.04805, 2018.

[4] B. McFee, C. Raffel, D. Liang, et al., "librosa: Audio and Music Signal Analysis in Python," in Proc. 14th Python in Science Conf., 2015.

[5] F. Chollet, Deep Learning with Python, Manning Publications, 2018.

[6] M. Soleymani, M. N. Caro, E. M. Schmidt, C. Sha, and Y. H. Yang, "1000 Songs for Emotional Analysis of Music," in Proc. ACM SIGMM Workshop on Music Information Retrieval with User-Centered and Multimodal Strategies, 2013.

[7] T. Li and M. Ogihara, "Detecting emotion in music," in Proc. 4th Int. Society for Music Information Retrieval (ISMIR) Conf., 2003.
[8] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd ed., O'Reilly Media, 2019.
[9] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.
[10] TensorFlow and Keras Documentation, https://www.tensorflow.org, Accessed 2024.
[11] Librosa Documentation (Music and Audio Analysis Library), https://librosa.org, Accessed 2024.