© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715778

AI-Driven Synesthetic Music Visualizer: Real-Time Cross-Modal Audio-to-Visual Translation Using Machine Learning

MOHAMMED YUSOOF S[1], HARINI C N[2], HARINI S[1], SARANYA R[3], DR. LAKSHMIDEVI[4]
[1, 2, 3,4]*III B.Sc. Artificial Intelligence and Machine Learning, Sri Krishna Adithya College of Arts and Science, Coimbatore, Tamil Nadu, India*
[5]*Guide, B.Sc. Artificial Intelligence and Machine Learning, Sri Krishna Adithya College of Arts and Science, Coimbatore, Tamil Nadu, India*

Abstract- ***This paper presents the design, implementation, and evaluation of an AI-Driven Synesthetic Music Visualizer — a real-time system that computationally emulates the neurological phenomenon of synesthesia by translating auditory signals into semantically congruent, dynamic visual art. Audio tracks in MP3 or WAV format are ingested and decomposed into perceptual acoustic features — including Root Mean Square (RMS) energy, spectral centroid, chroma vector, tempo, and spectral rolloff — through Short-Time Fourier Transform (STFT)-based signal processing using the Librosa library. Six machine learning architectures are trained and benchmarked on an annotated audio-visual mapping corpus: Multi-Layer Perceptron (MLP), Long Short-Term Memory (LSTM), Random Forest, Support Vector Machine (SVM), K-Nearest Neighbours (KNN), and Gradient Boosting. The Gradient Boosting model achieves the highest classification performance with an F1-score of 88.8 % and an average inference latency of 29 ms — well within the perceptual synchronisation budget. Predicted visual parameters (colour palette, shape morphology, animation velocity) are forwarded to a GPU-accelerated OpenGL rendering engine sustaining 62 frames per second on commodity hardware. The complete pipeline is deployed as a browser-accessible Gradio application. Results demonstrate that intelligent cross-modal synthesis is achievable in genuine real time, opening avenues for generative art, live performance, and assistive technology for hearing-impaired users.***

Indexed Terms- ***Synesthesia, Music Visualisation, Deep Learning, LSTM, GAN, Audio Feature Extraction, Real-Time Processing, Generative AI, Cross-Modal Synthesis, Gradient Boosting, Gradio***

I. INTRODUCTION

The convergence of deep learning and digital media has created fertile ground for intelligent multimedia systems that go beyond passive consumption and enter the domain of automated creative synthesis. Among the most intriguing problems in this space is audio-to-visual cross-modal translation — the automatic generation of meaningful visual content driven by acoustic stimuli. Nature offers a compelling template: synesthesia, a well-documented neurological condition in which activation of one sensory pathway involuntarily triggers a response in another. Individuals with chromesthesia, the most studied subtype, reliably perceive specific colours, spatial forms, and textures when they hear particular musical tones, chords, or timbres. Computational synesthesia seeks to replicate these learned cross-sensory mappings through data-driven models.

Conventional music visualisers — as found in legacy media players such as Winamp — operate on deterministic, rule-based logic anchored to raw amplitude or bass-frequency thresholds. They produce oscilloscopes, spectrum bars, and pulsating circles, but lack any semantic understanding of the music itself. A system driven by hard-coded rules cannot distinguish between the delicate timbre of a solo cello and the kinetic aggression of heavy metal; its visual output is therefore generic, repetitive, and emotionally incongruent with the audio. This gap motivates the design of a learning-based approach.

The system described in this paper addresses these limitations by learning the latent statistical relationships between rich acoustic feature representations and annotated aesthetic visual parameters. A pipeline of audio preprocessing, deep feature extraction, multi-model benchmarking, GPU-

accelerated rendering, and web-based deployment is engineered with strict latency constraints to guarantee perceptual audio-visual synchronisation during live operation.

A. Research Objectives

1) Construct a low-latency audio preprocessing pipeline that decomposes musical signals into perceptual features encompassing frequency, tempo, amplitude, spectral centroid, and chroma content.
2) Train and rigorously benchmark six machine learning architectures — MLP, LSTM, Random Forest, SVM, KNN, and Gradient Boosting — for the audio-to-visual parameter mapping task.
3) Design a hardware-accelerated rendering engine that converts predicted visual parameters into fluid animations at a sustained frame rate of 60+ FPS.
4) Deploy the integrated system as a Gradio web application accessible to non-technical end-users without requiring local software installation.
5) Quantitatively evaluate system performance using accuracy, F1-score, synchronisation latency, and frame-rate consistency as primary metrics.

II. RELATED WORK

Briot, Hadjeres, and Pachet (2017) surveyed deep learning methods applied to musical content, establishing that recurrent and generative architectures can learn high-level structural representations of audio well beyond simple pattern matching. Their framework provided the conceptual basis for adopting LSTM and GAN models in the present work.

McFee et al. (2015) introduced the Librosa Python library, which consolidates efficient implementations of spectral and temporal audio analysis algorithms including MFCCs, chroma features, beat tracking, and spectral rolloff. Librosa has since become the de-facto standard for audio feature engineering in the machine learning research community and serves as the primary feature-extraction backbone in this study.

The introduction of Generative Adversarial Networks (Goodfellow et al., 2014) enabled a paradigm shift in generative modelling, allowing a discriminator-guided generator to synthesise high-fidelity content conditioned on arbitrary input representations. Subsequent conditional GAN variants demonstrated feasibility of audio-conditioned image synthesis, directly informing the generative mapping architecture explored here.

Ward (2013) documented consistent cross-modal associations among synesthetic individuals — for example, the pitch class C major reliably evoking bright red hues across a statistically significant population. These empirical psychoacoustic correspondences were used as ground-truth labels when constructing the audio-visual mapping dataset.

Paszke et al. (2019) described PyTorch's flexible dynamic computation graph and native CUDA integration. This framework was adopted for implementing the LSTM and GAN components owing to its support for custom recurrent cell definitions and GPU-accelerated backpropagation.

III. SYSTEM ARCHITECTURE

The proposed system is organised as a five-stage modular pipeline, each stage communicating through well-defined data interfaces to permit independent optimisation and future extensibility. The end-to-end flow moves from audio ingestion, through feature extraction and AI-driven mapping, to real-time visual rendering and user-facing deployment.

A. Audio Capture and Preprocessing

Audio input is accepted as an uploaded MP3/WAV file or as a live microphone stream captured via PyAudio at a uniform 22,050 Hz sampling rate. The continuous stream is segmented into overlapping short-time frames of 512 samples (approximately 23 ms each at the chosen sample rate). Each frame is subjected to a Short-Time Fourier Transform (STFT) to obtain a time-frequency magnitude spectrogram, which is then converted to decibel scale. Z-score standardisation is applied per-feature to ensure that numerically disparate quantities (e.g., Hz versus dB) contribute proportionally during model inference.

B. Acoustic Feature Extraction

From each processed frame the following six perceptual features are extracted using Librosa: (1) RMS Energy — a measure of signal loudness; (2) Spectral Centroid — the weighted centre-of-mass of the frequency spectrum, indicating perceived

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715778

brightness; (3) Chroma Vector — a 12-element distribution over musical pitch classes; (4) Tempo — beats per minute estimated via autocorrelation of the onset-strength envelope; (5) Spectral Rolloff — the frequency below which 85 % of spectral energy is concentrated; and (6) Dominant Frequency — the peak-energy bin of the magnitude spectrum. These features collectively provide a compact but expressive acoustic representation of each audio frame.

Table I. Audio-to-Visual Mapping Dataset — Representative Samples

Dominant Freq. (Hz)	Tempo (BPM)	Amplitude (dB)	Assigned Colour (Hex)	Generated Shape	Animation Speed
440	120	−12	#FF5733 Red	Pulsing Sphere	Fast
220	90	−20	#33A1FF Blue	Flowing Wave	Slow
880	140	−6	#F1C40F Yellow	Expanding Star	Very Fast
110	60	−24	#8E44AD Purple	Slow Ripple	Very Slow
660	128	−10	#2ECC71 Green	Geometric Grid	Fast

C. Audio-Visual Mapping Dataset

A supervised audio-visual mapping corpus was constructed by processing a curated library of music tracks spanning six genres: Classical, Jazz, Electronic, Rock, Lo-Fi, and Ambient. Each audio frame was annotated with target visual parameters — assigned colour hex code, geometric shape type, and animation speed category — derived from psychoacoustic synesthetic association literature (Ward, 2013) and supplemented by a structured annotation protocol. Table I presents representative samples from the dataset. The complete corpus comprises approximately 14,000 labelled frame instances across 120 audio tracks.

Table II. Acoustic Feature Schema with Visual Mapping Correlations

Feature Name	Data Type	Description	Visual Mapping Correlation
Timestamp	Float	Time marker (seconds) within the audio track.	Progress bar / timeline synchronisation
RMS Energy	Float	Root-mean-square loudness of the audio frame.	Scale and brightness of rendered shapes
Spectral Centroid	Float	Centre-of-mass of the frequency spectrum.	Edge sharpness: jagged vs. smooth geometry
Chroma Vector	Array[12]	12-element semitone distribution vector.	Colour hue and palette selection
Tempo (BPM)	Integer	Detected beats per minute via autocorrelation.	Pulse frequency and transition speed
Spectral Rolloff	Float	Frequency below which 85 % of energy resides.	Particle density and glow intensity

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715778

D. Machine Learning Models

Six candidate architectures were trained and evaluated on an 80/20 stratified train–test split of the annotated corpus. Five-fold cross-validation was applied during training to mitigate variance in performance estimates. The architectures evaluated were:

1) Multi-Layer Perceptron (MLP): A feed-forward network with three hidden layers (256 → 128 → 64 neurons) and ReLU activations, used as the baseline.
2) Long Short-Term Memory (LSTM): A two-layer recurrent network with hidden size 128, exploiting temporal dependencies across consecutive audio frames for smoother visual transitions.
3) Random Forest: An ensemble of 200 decision trees with Gini impurity splitting, providing strong generalisation with minimal hyperparameter sensitivity.
4) Support Vector Machine (SVM): An RBF-kernel classifier effective in high-dimensional spectral feature spaces.
5) Gradient Boosting: A sequential additive ensemble that iteratively minimises a differentiable loss, selected as the primary model based on benchmark results.
6) K-Nearest Neighbours (KNN, k=5): A distance-based classifier used as a non-parametric computational baseline.

E. Visual Rendering Engine

The rendering subsystem receives the predicted visual parameter vector from the inference engine and translates it into real-time graphical output using an OpenGL-based pipeline (PyOpenGL or Vispy). Outputs include dynamic geometric primitives (spheres, grids, ripple fields, expanding stars), GPU particle systems whose density is governed by spectral rolloff, and colour palettes derived from the predicted chroma-based hex codes. Double-buffered rendering and GLSL shader programs maintain a sustained throughput of 62 FPS on the reference hardware platform, ensuring perceptual synchrony with the audio stream.

F. Metadata Schema

A lightweight relational metadata store tracks acoustic parameters extracted at each processing step, enabling replay, dataset augmentation, and debugging. Table III summarises the schema.

Table III. Metadata Database Schema

Field Name	Data Type	Description	Sample Value
Dominant Frequency	INTEGER	Primary pitch of the audio chunk in Hz; drives base colour hue.	440
Tempo (BPM)	INTEGER	Beats per minute; governs the speed of visual state transitions.	120
Amplitude	DECIMAL	Peak loudness in dB; controls rendered shape scale.	−12.5
Spectral Centroid	INTEGER	Brightness/timbre measure; adjusts texture complexity and particle spread.	1500

IV. IMPLEMENTATION

A. Development Environment and Tool Stack

The complete technology stack is detailed in Table IV. Python 3.9 was selected as the primary development language for its breadth of scientific and machine learning library support. Model training was conducted in Google Colab with GPU acceleration; inference and rendering were tested locally on the reference hardware configuration (Intel Core i5, 16 GB RAM, NVIDIA GTX 1660 with CUDA 11.8).

Table IV. Software Tool Stack and Roles

Library / Tool	Role in the System
Python 3.9+	Core language; bridges audio analysis, ML models, and UI logic.
Librosa	Spectral feature extraction: MFCCs, chroma, tempo, spectral centroid, RMS.
PyAudio	Captures live microphone / system-audio streams for real-time mode.
NumPy	High-throughput array mathematics on audio frame buffers.
PyTorch / TensorFlow	Trains and evaluates LSTM and GAN architectures.
PyOpenGL / Vispy	GPU-accelerated rendering of 2-D/3-D animations at ≥ 60 FPS.
scikit-learn	Trains classical ML models (RF, SVM, KNN, Gradient Boosting, MLP).
Gradio	Browser-based deployment interface for end-user interaction.

B. Data Preparation

Audio tracks are loaded with Librosa's core loading routine, resampling all signals to 22,050 Hz. Frame-level feature matrices are assembled and standardised using scikit-learn's StandardScaler, fitted exclusively on the training partition and applied without refitting to the test partition to prevent data leakage. SelectKBest mutual-information selection identified Tempo, Spectral Centroid, and RMS Energy as the three most predictive features; however, all six features were retained to preserve model robustness across diverse musical genres.

C. Model Training and Selection

Each architecture was trained on 80 % of the labelled corpus. Hyperparameters for Gradient Boosting (n_estimators = 300, learning_rate = 0.05, max_depth = 5) were determined via a randomised grid search with five-fold cross-validation. The best-performing model was serialised using Joblib's compressed dump facility, producing a portable artefact (visualizer_engine.pkl, ≈ 4.2 MB) suitable for low-overhead runtime loading.

D. Real-Time Inference Pipeline

During live operation, incoming 512-sample audio frames are preprocessed and standardised using the frozen training-partition scaler. The loaded Gradient Boosting model processes each six-dimensional feature vector and produces a visual complexity score in the range [0, 1]. Scores exceeding the empirically determined threshold of 0.70 activate the high-intensity kinetic visual preset (vibrant colours, fast geometric transitions); lower scores engage the ambient melodic preset (soft gradients, slow flow animations). The predicted visual directive is forwarded to the OpenGL rendering engine, which generates and presents the corresponding frame within the 29 ms mean inference budget.

E. Deployment Interface

The complete pipeline is packaged as a Gradio web application. The browser-based interface accepts numerical acoustic feature inputs directly or triggers the full preprocessing–inference–rendering cycle when an audio file is uploaded. A public shareable tunnel is generated via Gradio's integrated proxying service, permitting remote access without any local software installation on the client machine.

V. RESULTS AND EVALUATION

All six models were evaluated on the held-out test partition (20 % of the corpus, ≈ 2,800 frame instances) using four classification metrics — accuracy, precision, recall, and F1-score — together with average per-frame inference latency. Table V presents the complete comparison.

© MAR 2026 | IRE Journals | Volume 9 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV9I9-1715778

Table V. Model Performance Comparison on the Audio-Visual Mapping Task

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Latency (ms)
MLP Neural Network	81.4	79.3	80.1	79.7	38
LSTM Network	86.7	85.2	87.4	86.3	52
Random Forest	83.5	82.1	81.9	82.0	22
SVM (RBF Kernel)	79.8	78.5	77.6	78.0	18
Gradient Boosting	89.2	88.6	89.0	88.8	29
KNN (k = 5)	74.3	72.8	73.5	73.1	15

Gradient Boosting achieves the highest classification accuracy (89.2 %) and F1-score (88.8 %), reflecting a well-balanced trade-off between precision and recall. Its mean inference latency of 29 ms falls comfortably within the 50 ms threshold required to maintain the perceptual illusion of audio-visual synchrony at standard display refresh rates. The LSTM network records the second-highest F1-score (86.3 %) but incurs a 52 ms average latency owing to its sequential hidden-state propagation, making it marginally unsuitable for strict real-time constraints on lower-end hardware. Random Forest offers an attractive accuracy–latency trade-off (83.5 %, 22 ms) and is recommended as a fallback model in resource-constrained deployment scenarios. SVM and KNN, while computationally frugal, exhibit notably weaker classification performance, indicating that the audio-to-visual mapping task involves non-linear feature interactions that kernel boundaries and distance metrics handle less effectively.

At the system level, the integrated pipeline sustains a mean end-to-end latency of 31 ms from audio frame capture to displayed visual frame. The rendering engine maintains 62 FPS on the reference platform throughout stress tests involving genre-switching and dynamic amplitude transitions. Visual coherence — rated on a five-point Likert scale by a ten-participant user panel assessing the perceptual alignment between audio characteristics and generated visuals — achieved a mean score of 4.1 / 5.0, confirming qualitative congruence between the AI-predicted and psychoacoustically expected visual states.

VI. DISCUSSION

The experimental results affirm that ensemble tree-based methods, specifically Gradient Boosting, outperform both shallow distance-based classifiers and deep sequential models for this particular cross-modal mapping task when strict latency constraints are imposed. The advantage of Gradient Boosting stems from its native capacity to capture non-monotonic interaction effects among acoustic features — for instance, the compound influence of spectral centroid and tempo on animation fluidity — without requiring architectural modifications or expensive sequential computation.

A key challenge encountered during development was maintaining temporal coherence across consecutive audio frames. Purely frame-level prediction, without any contextual memory, can produce abrupt visual discontinuities at musical transitions. The LSTM model partially resolves this by encoding inter-frame temporal dependencies; however, its latency overhead limits applicability on commodity hardware in strict real-time scenarios. A productive avenue for future research is a lightweight hybrid architecture in which Gradient Boosting performs fast visual state classification and a compact LSTM post-processing head applies temporal smoothing to the predicted parameter sequence, combining the latency efficiency of the former with the temporal continuity of the latter.

The Gradio deployment interface proved highly accessible for user testing. However, streaming full-resolution visual animations over web connections introduces bandwidth-dependent variable latency absent in locally executed versions. Future iterations may benefit from a WebGL-based client-side rendering layer, where only the compact visual parameter predictions are transmitted over the network and all rendering computation is offloaded to the browser's GPU — a strategy that would dramatically reduce bandwidth requirements while preserving visual fidelity.

VII. CONCLUSION

This paper presented an AI-Driven Synesthetic Music Visualizer that bridges computational audio signal processing and generative visual art through a rigorously benchmarked machine learning pipeline. By extracting six perceptual acoustic features from musical signals and learning their mapping to aesthetic visual parameters via a Gradient Boosting ensemble, the system achieves real-time, emotionally congruent audio-visual synchronisation that substantially surpasses the capabilities of conventional rule-based visualisers.

The modular five-stage pipeline — encompassing DSP-based feature extraction, multi-model benchmarking, OpenGL-accelerated rendering, and Gradio-based web deployment — provides a reproducible and extensible foundation for future research in intelligent cross-modal synthesis. The system records an F1-score of 88.8 %, an end-to-end latency of 31 ms, and a sustained rendering frame rate of 62 FPS on commodity hardware, demonstrating practical viability for real-world deployment scenarios ranging from live performance to therapeutic applications for the hearing-impaired community.

Future work will explore: (1) conditional GAN-based texture generation for richer visual outputs; (2) a genre-aware contextual embedding layer that adapts visual mappings to detected musical genre in real time; (3) integration of the rendering engine with Virtual Reality head-mounted displays for fully immersive audio-visual experiences; and (4) a deeper personalisation mechanism enabling users to guide the aesthetic profile of generated visuals through preference feedback. This work contributes evidence that artificial intelligence can function not merely as an analytical instrument but as a creative collaborator, meaningfully expanding the boundaries of human sensory experience.

REFERENCES

[1] J. P. Briot, G. Hadjeres, and F. Pachet, "Deep Learning Techniques for Music Generation — A Survey," Springer Nature, 2017.

[2] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," in Proc. 14th Python in Science Conf., pp. 18–24, 2015.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," in Advances in Neural Information Processing Systems (NIPS), vol. 27, 2014.

[4] A. Paszke, S. Gross, F. Massa, A. Lerer, et al., "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in Advances in Neural Information Processing Systems, vol. 32, pp. 8024–8035, 2019.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[6] D. Shreiner, G. Sellers, J. Kessenich, and B. Licea-Kane, OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3, 8th ed. Addison-Wesley Professional, 2013.

[7] J. Ward, The Frog Who Croaked Blue: Synesthesia and the Mixing of the Senses. Routledge, 2013.