Visualizing Similarities of Music from Different Instruments - A Novel Proposal to Project High-dimensional Music Features on a 2-dimensional Plane

Goutam Chakraborty *
Iwate Prefectural University, Information Science, IPU, Iwate, Japan
Email: goutam@iwate-pu.ac.jp

Cedric Bornand†
University of Applied Sciences HES-SO
Yverdon-les-Bains, Switzerland
Email: cedric.bornand@heig-vd.ch

Lokesh Ayyaswamy‡, Lakshman Patti§, Praveen Kumar Reddy Sangati¶, Subhash Molaka‖
Department of Computer Science Engineering and Artificial Intelligence‡§¶‖
Madanapalle Institute of Technology & Science, Madanapalle, India‡§¶‖
Email: lokeshreddy2680@gmail.com‡, lakshmanpatti99@gmail.com§,
prawinreddy1909@gmail.com¶, molakasubhash@gmail.com‖

***Abstract*—We perceive music from various perspectives: the melody, the rhythm, the emotions or passions they evoke, the richness of sound, and how it correlates with the time of day (like Morning Raga) or with seasons (like Vivaldi's Four Seasons). This is a multimodal classification challenge for which correct data annotation is a difficult issue. In this work, we propose a method for visualizing audio signals from various musical instruments to find their variances, quantify their similarities, and distances. To facilitate visualizing, we project the audio data on a 2-dimensional plane such that the distances (dis-similarities) of the audio signals are preserved as much as possible. The appropriate tools (algorithms) for this task were identified by experimental analysis. The work is conducted in two stages: the first is audio feature extraction and compression, and the second is the projection of high-dimensional audio features on a two-dimensional plane using various unsupervised visualization techniques. The aim is to determine which feature compression and visualization tools can produce clearly separated clusters of audio signals. The features of the Short Time Fourier Transform (STFT) spectrograms extracted using Convolutional Neural Networks (CNN) provided the best compressed representations, facilitating clear separation and meaningful projection of the audio signals using visualization tools t-Stochastic Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP). The scatterplots of the samples achieved silhouette scores of 84% and 81%, respectively, ensuring clear groups for sounds generated from different instruments. We also experimented with UNet to find numerical vector representations of the spectrogram images. UNet could achieve silhouette scores of around 75%.**

Keywords-MFCC; STFT; Spectrogram; CNN; U-Net; t-SNE; UMAP.

I. INTRODUCTION

This study analyzes audio signals from ten musical instruments, a combination of traditional Indian and Western ones. The features extracted from these signals are high-dimensional and projected onto a 2-dimensional plane to visualize similarities. The Indian instruments include flute, nadaswaram, and shehnai (wind-type); santoor and veena (string); and thavil and mridangam (percussion), while the Western ones are piano, guitar, and violin. Each instrument has distinct sound characteristics: wind instruments produce tones through vibration of air, string instruments through plucking or bowing, percussion by striking a tense diaphragm with a hand or a stick, and the piano by hammering strings via key presses.

This work builds upon our conference publication [1] expanding the number of instruments from six to ten and improving the visualization methodology.

The Fourier Transform (FT) [2] and the Fast FT (FFT) [2] were used to examine the audio signals. However, FFT cannot capture sequential information from the signal. Advances in speech processing introduced techniques such as STFT [3], Wavelet Transform (WT) [4], and Mel-frequency cepstral coefficients (MFCC) [5]. MFCC exploits the log scale of human audio perception and is widely used for audio signal analysis like speaker identification. As it uses small windows, the number of features increases linearly as the length of the audio signal increases. Using audio feature extraction methods and deep neural networks for compressing high-dimensional audio data enables us to achieve a relatively low-dimension representation of the audio signal, which is further used for visual representation as scatter plots on a 2-dimensional display. The overall plan is shown in Figure 1.

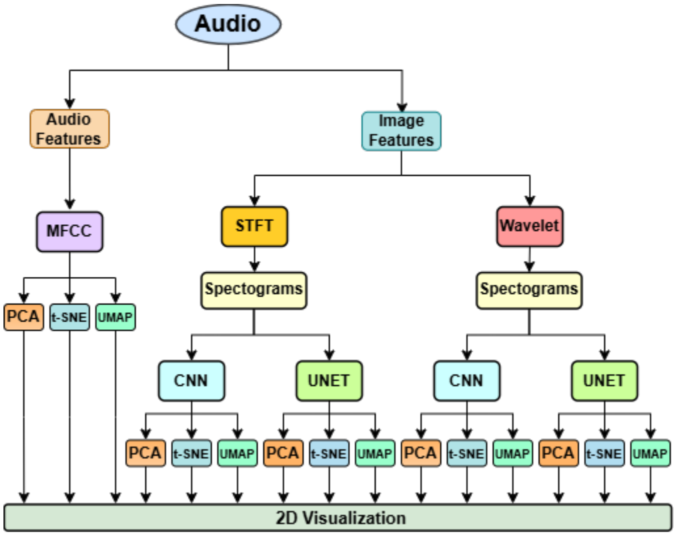


Figure 1. Overall plan for the Experiments.

The decorrelated MFCC features were extracted from the audio samples [5], using MFCC window lengths of 25 milliseconds. Even a few seconds of audio signal generate a high-dimensional feature vector. To be able to capture the characteristic features of the audio samples, we used signals of length 30 seconds. We had to find ways to represent them as a relatively low-dimensional vector. Here, we used spectrograms and neural networks to extract features from the spectrogram images.

For spectral analysis, we used the STFT. We converted the STFT features into spectrogram images [3]. These spectrograms serve as visual representations of the musical features. We also converted the audio to wavelet coefficients using Morlet wavelets, and that is being transformed to a spectrogram image representation.

To extract features from spectrograms, we trained a CNN model [6], and a U-Net model [7] on STFT and wavelet spectrograms. As our sample classes are known, the deep neural network was trained as a supervised classifier. Features were taken from the output of the convolutional layers, which are inputs to the dense classification layer. In the U-Net model, features were extracted from the bottleneck layer at the end of the encoder.

The effectiveness of the proposed methods was validated through several experiments by projecting the features onto a two-dimensional plane [8]. Section III details how CNN and U-Net architectures extract features from STFT and WT spectrogram images. To visualize music signals on a two-dimensional plane, we used Principal Component Analysis (PCA) (a linear method), t-SNE, and UMAP (nonlinear methods).

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the methodology, including data collection and preprocessing, feature extraction, and the proposed solution. Section IV presents the experimental results and their analysis. Finally, Section V concludes the paper and discusses future research directions.

II. Related work

The previous works on the visualization of audio sample characteristics are discussed below.

The authors used three different datasets in their work reported in 2024 [9]. Two datasets with 10 classes and an augmented version of one (using pitch shifting, time-stretching, and added random noise) were used. MFCC features were extracted, and CNN and Recurrent Neural Network (RNN)-(Long Short Term Memory) LSTM models were trained. CNN performed better on smaller datasets, while RNN-LSTM excelled on larger ones.

In the work reported in 2020 [8], the authors experimented with audio data of 10 classes, extracting MFCC features. They visualized these high-dimensional features using PCA, t-SNE, Iso-Map, and SOM. t-SNE produced well-separated clusters. SOM showed slight separation, while Iso-Map failed to capture any meaningful clustering. The conclusion was that Iso-Map failed to work with this high-dimensional data.

In another work on the audio classifier, reported in 2020 [6], the authors used a public dataset and converted the audio signals into Mel power spectrograms. They applied two approaches to capture features: a CNN model trained from scratch and a pre-trained VGG19 model using transfer learning. Both models performed well. The CNN model trained from scratch slightly outperformed the VGG19 model.

III. Proposed Methods

This Section presents the workflow of our experiments, covering the collection of audio samples of ten musical instruments, preprocessing, feature extraction, dimensionality reduction, and finally 2-dimension projection.

A. Data Collection and Pre-processing

Audio samples were sourced from public platforms like YouTube and recorded media, ensuring that each sample captures the unique tonal and spectral qualities of the instrument without background noise or interference from other instruments.

We gathered 300 audio samples, 30 samples per instrument, using YTMP3 and converted them to MP3. We processed them with Clideo. Clips were segmented into 30–45 seconds, then converted to WAV, ensuring standard audio quality for audio signal analysis.

B. Feature Extraction

1) MFCC Feature Extraction: MFCC features are widely used in audio analysis for music, speech recognition, and speaker identification. Pre-processing involves standardizing samples to 30 seconds through padding or trimming, then sampling at 44,100 Hz to preserve the high-quality of audio samples.

The MFCC extraction process starts with splitting the 30-second audio into 25 ms non-overlapping frames (1,201 frames, each with 1,103 samples). The segmented audio will get distortions at the end points due to abrupt change. To smooth out the change, we apply Hanning window [10] to maintain smooth transition betweem frames. The Discrete Fourier Transform (DFT) transforms the signal to the frequency domain, capturing spectral characteristics. A Mel filter bank emulates human hearing by dividing the spectrum into 26 bands, reducing dimension while maintaining required information. A logarithmic transformation follows to constrict the dynamic range. The last step is Discrete Cosine Transform (DCT) to decorrelates Mel-spectral coefficients, retaining the first 13 MFCCs.

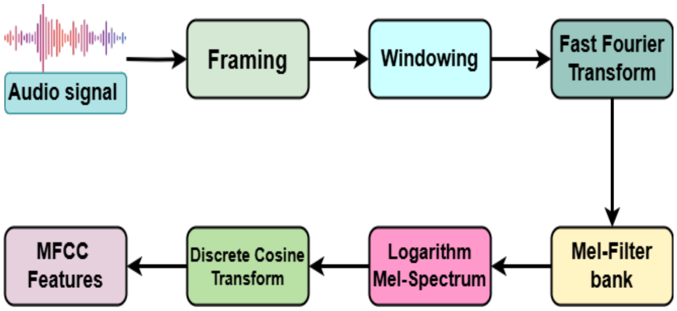


Figure 2. The process of MFCC Feature Extraction.

The MFCC extraction process is shown in Figure 2. Each 30-seconds sample is converted into 13 MFCCs × 1,201 frames and flattened into a 1-D vector of 15,613 elements. MFCCs capture important audio characteristics, preserving tonal, timbral, and rhythmic features for analysis of the audio signal.

Finally we have 300 samples, 30 samples each from ten musical instruments. Each sample is represented by a vector of 15,613 elements, 13 MFCC values from 1201 frames.

2) STFT Spectrogram Generation: The audio sample of 30 seconds is segmented into windows of size 25 milliseconds. Each segment of 25 millisecond contains 1,103 sample points, because the sampling rate is 44,120. FFT is applied to extract the frequency content (frequency band and corresponding intensity) of the signal within the segmented window. These frequency domain representations are then concatenated to form the spectrogram images. The spectrogram is an image of the frequency spectrum where the intensity of a frequency band is represented as brightness [11]. Figure 3 shows the STFT spectrograms of different musical instruments, capturing the tonal and spectral characteristics of the instruments as images.

3) Wavelet Spectrogram Generation: The 30-second audio is segmented into 2-second chunks. We use a minimum frequency of 64 Hz and a maximum frequency as the Nyquist frequency which is half of the sampling rate 44100/2. We used 32 scales. The sampling rate high so as to be able to capture the delicate abrupt changes that happen with musical instruments. For voice a much lower sampling rate suffices. We apply Continuous wavelet transform using the Morlet wavelet. The Morlet wavelet is shown in Figure 4. Morlet wavelets are a product of a sine wave and a Gaussian function. The standard deviation of the Gaussian determines the window size. Different sine wave frequencies capture different frequency components of the audio signal. As mentioned, 32 scales were used. The wavelets convolve over the signal and gets the wavelet coefficients. Amplitudes are converted into decibels. Wavelet coefficients from all the segments are concatenated to represent the entire audio signal.

The wavelet spectrogram displays the frequency spectrum where the intensity of a frequency is converted into brightness. Each spectrogram represents the unique characteristic of an audio signal [12]. The wavelet spectrograms of the sample for each instrument are shown in Figure 5.

C. Projection of higher dimension into 2D

We thus have five sets of features - one obtained from MFCC, two sets from STFT spectrogram features and then using CNN and U-Net; and two sets of wavelet spectrogram features and then using CNN and U-Net. To visualize these features in 2 Dimension, we employed three different visualization techniques. Firstly, we visualized the extracted features using PCA by taking the first two principal components and projecting the samples on two dimension where the two principal eigen vectors are the basis [13]. Its capability for proper projection is limited due to its linearity restriction. We used two nonlinear tools for projection t-SNE [14] and UMAP [3]. t-SNE takes the distribution in the high dimension and maps it to two dimension using a non-linear method, by which data that are closer in high dimension become closer, and data that are far are pushed further. This helps to represent the data in tight clusters in 2-dimension. UMAP [3] on the other hand, works on manifold space. Distances between samples are measured in terms of probabilities, as a random walk on a Markov chain. Thus, the distances between two data points are measured on the manifold space, not in the Euclidean space. The two prominant embedding space directions were used as the two axes of 2-dimensional visualization space. UMAP could give excellent results, even when high dimension MFCC features were used.

D. Proposed Method

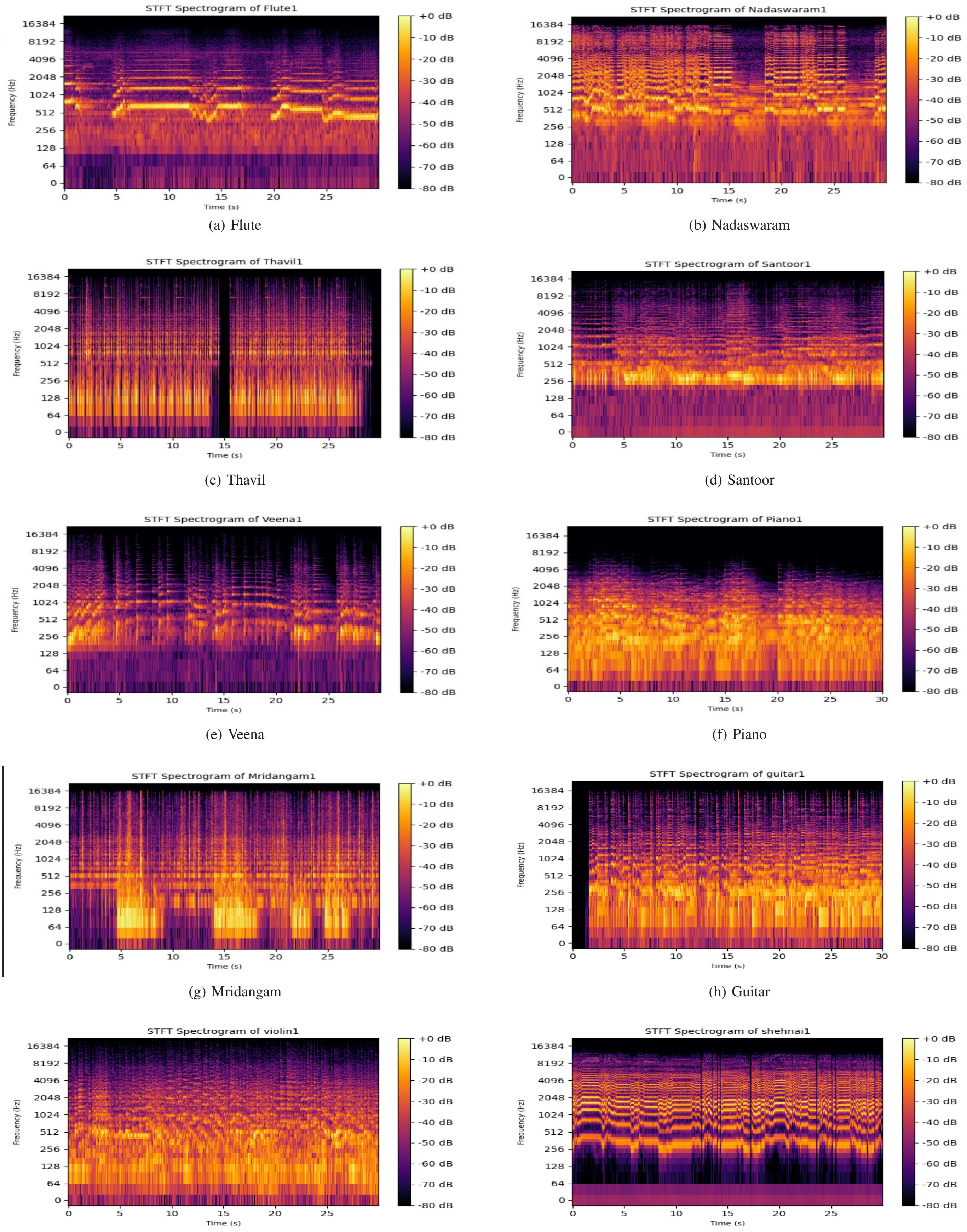
To map high-dimensional audio samples onto a two-dimensional plane, we employed three distinct visualization algorithms: PCA, t-SNE, and UMAP.

1) MFCC features onto 2D plane: MFCC features are extracted from audio signals, resulting in a high-dimensional data set with 15,613 dimensions for each 30-second music sample. In total, we have 300 samples from 10 different instruments. This dataset matrix of dimensions 300x15613 is the input to PCA, t-SNE, and UMAP to visualize the data on a 2-dimensional plane.

2) Feature Extraction using CNN: The STFT and WT spectrogram images of audio samples are used to train a Deep Neural Network (CNN) classifier model. The samples are labeled with the respective musical instruments. The output from the CNN layers are taken out as image features. The architecture of the CNN model consists of two convolutional layers, each succeeded by max - pooling layers, followed by a flatten layer and two dense layers for classification, one hidden and one output layer. The output from hidden layer is used as the compressed feature vector.

In Figure 6, the architecture of the CNN model utilized for training is shown. The model receives an STFT spectrogram images of size 400x600x3 as input. The first convolution layer comprises of 16 filters yielding an output of 400x600x16. Subsequently, a MaxPooling layer with a 2x2 kernel diminishes the size to 200x300x16. Following this, a second convolution layer with 32 filters is applied, and after executing 2x2 size max pooling, the output is further reduced to 100x150x32. The output is then flattened into a vector and processed through a dense layer classifier with a hidden layer of 64 nodes. The network is trained as a supervised classifier with 48,000 features as input, and 10 output nodes for 10 musical instruments. As mentioned, the hidden layer consists of 64 nodes. The extracted features are visualized using PCA, t-SNE, and UMAP algorithms.

For the wavelet spectrograms, which are of size 390×584×3, we use the same CNN architecture as for the STFT spectrograms. The only modification is that the first convolutional layer is configured with 20 filters. We then train a separate CNN model with this architecture and extract the features from the hidden dense layer for visualization.


STFT Spectrogram of Flute1

STFT Spectrogram of Nadaswaram1

STFT Spectrogram of Thavil1

STFT Spectrogram of Santoor1

STFT Spectrogram of Veena1

STFT Spectrogram of Piano1

STFT Spectrogram of Mridangam1

STFT Spectrogram of guitar1

STFT Spectrogram of violin1

STFT Spectrogram of shehnai1

(a) Flute

(b) Nadaswaram

(c) Thavil

(d) Santoor

(e) Veena

(f) Piano

(g) Mridangam

(h) Guitar

(i) Violin

(j) Shenai

Figure 3. STFT spectrogram samples for ten musical instruments.

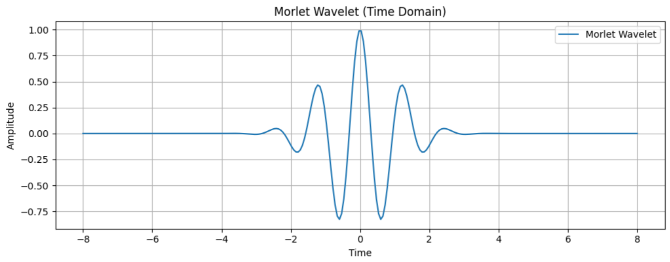


Figure 4. Morlet Wavelet In Wavelet Transform.

3) Feature Extraction using UNet: UNET, which was proposed for medical image segmentation, is used to compress the image features. The UNet architecture consists of encoder and decoder. The encoding part uses convolutional layers followed by max-pooling layers to extract features and reduce dimensions, while the decoding part employs upsampling layers to reconstruct the input. Essential compressed audio features are available in the bottom layer of the UNet.

In Figure 7, the architecture of UNET is shown. The STFT spectrograms of size 400×600×3 are input to the UNET model. Initially, two convolution layers with 32 filters are used, followed by max pooling with a pool size of 4×4. Next, two more Convolution layers with 64 filters are appended, followed by another MaxPooling operation. After that, two additional convolution layers are added, one with 128 filters and the next one with 64 filters, leading to the bottleneck layer, which captures the encoded representation of the input.

From the bottleneck layer, the features are upsampled using a transposed convolution operation with 64 filters of size 4×4. These up-sampled features are concatenated with corresponding layer from the encoding side. The process of convolution, upsampling, and concatenation is repeated for feature reconstruction. Finally, this process produces the reconstructed image. It is an unsupervised algorithm for encoding-decoding of images.

The elements from the bottleneck layer represent the input spectrogram image features. These values from the UNET bottom layer are the input to the three visualization software, namely PCA, t-SNE, and UMAP.

First, we train the UNet using data from all ten classes. The compressed features from the UNet bottleneck, from different classes, are superimposed in the feature space and fail to be presented as separate clusters on a 2-Dimensional plane. In the next experiments, we trained UNet separately with individual classes of samples. After this training, the features from the UNet bottleneck layer are used. The visualization algorithms projected them as isolated groups. Both STFT and wavelet spectrograms are used to train the UNET model and extract the features from the bottleneck layer, and the features are visualized.

IV. Experiments and Results

In this Section, we present the visualization results of MFCC features and the extracted features from STFT and WT spectrogram images using CNN and UNet.

A. Visualization of MFCC Features

To visualize high-dimensional MFCC features on a 2-dimensional plane, we used PCA, t-SNE, and UMAP. The resulting scatter plots of 300 music samples are shown in Figures 8, 9, and 10.

In Figure 8, the scatter plot shows that different instrument classes are mixed, with poor separation between samples from different musical instruments. All audio samples from different musical instruments are randomly placed on the 2-dimensional projection. This is because (1) the number of MFCC features is too large, and (2) PCA, being a linear method, cannot correctly project the data, which are on a curved manifold space in the MFCC feature space.

Both t-SNE and UMAP reveal distinct clustering patterns for different instruments based on their MFCC features. Piano consistently appears well-separated in both visualizations, reflecting its unique timbral profile. Thavil also shows clear separation in the t-SNE plot, whereas UMAP emphasizes tighter groupings among other instruments. Figure 10(UMAP) demonstrates a slightly better overall grouping of audio samples from different instruments than as Figure 9 (t-SNE). But neither of the two methods could achieve clear separation of all samples generated from 10 different instruments.

B. Visualization of Extracted features from CNN model

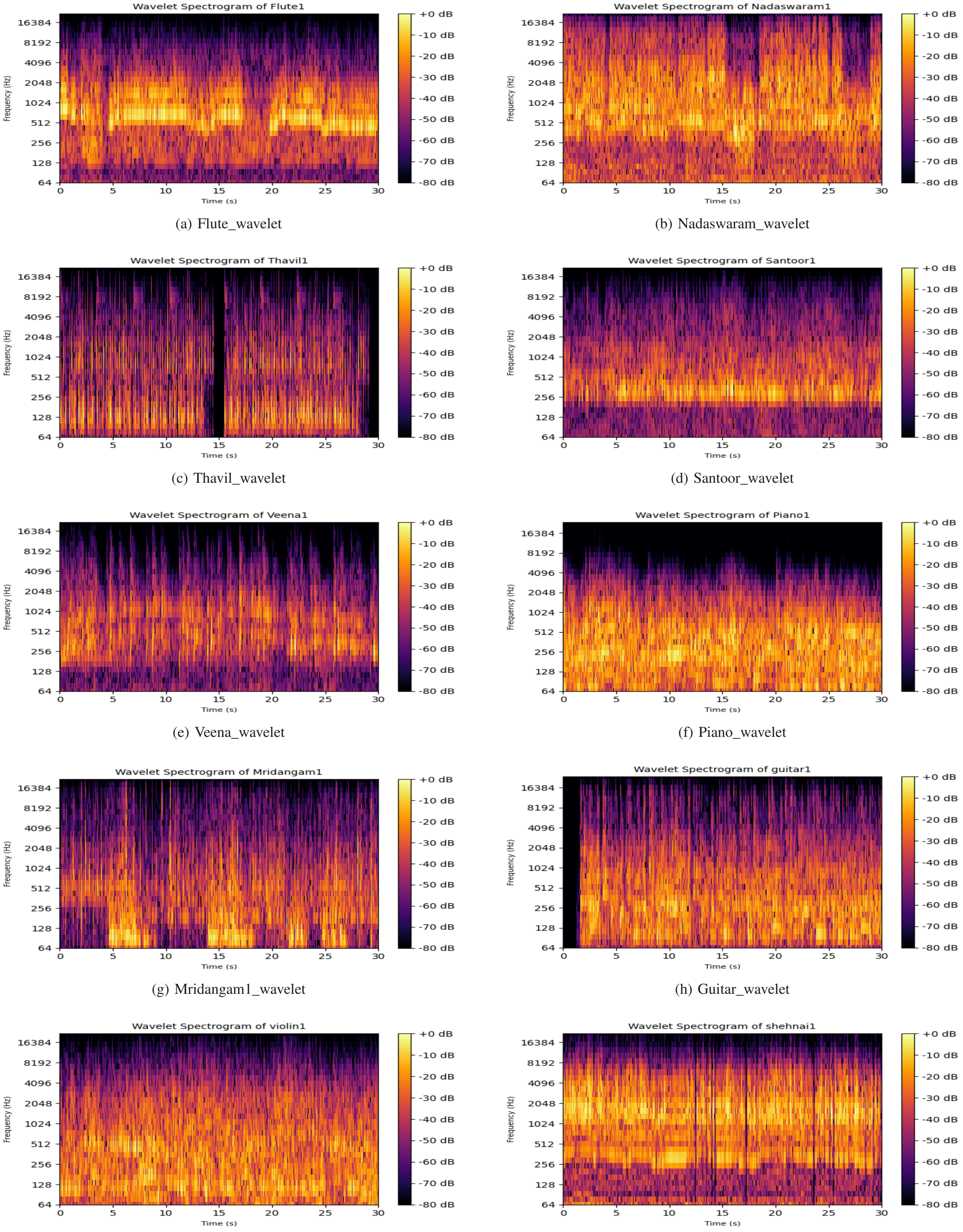
The STFT spectrogram and wavelet spectrogram are trained with CNN, and the features extracted from the hidden layer of the densenet classifier are now used as input to the visualization software. These features are much lower in number (64) than the MFCC features. These features extracted by CNN from both STFT and wavelet spectrograms are then input to PCA, t-SNE, and UMAP for visualization.

1) STFT spectrogram Features through CNN: The STFT spectrogram results are shown in Figures 11, 12 and 13.

In Figure 11, the PCA plot of CNN features displays moderately distinct groupings, but some overlaps persist. Clusters for classes like Santoor and Nadaswaram are visible, though not as tightly packed as in UMAP or t-SNE.

In Figure 12, the t-SNE representation of CNN features reveals distinct, compact clusters for all ten instruments. For instance, the shehnai and violin are far from the piano and mridhangam, suggesting meaningful class separation. By meaningful, we mean how we perceive the sounds from those 10 instruments - some perceived as more similar than others. This visualization confirms the CNN model's ability to learn feature representations that preserve inter-class differences effectively.

When UMAP was used for the 2-dimensional projection, as shown in Figure 13, highly compact clusters of samples from individual instruments were formed. The intra-cluster distances between violin, veena, thavil, guitar and mridhangam music



(a) Flute_wavelet

(b) Nadaswaram_wavelet

(c) Thavil_wavelet

(d) Santoor_wavelet

(e) Veena_wavelet

(f) Piano_wavelet

(g) Mridangam1_wavelet

(h) Guitar_wavelet

(i) Violin_wavelet

(j) Shenai_wavelet

Figure 5. Wavelet spectrograms of different musical instruments.

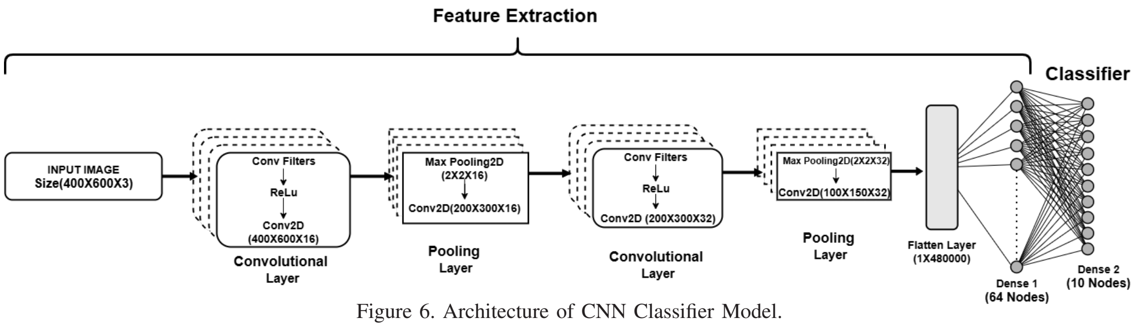


Figure 6. Architecture of CNN Classifier Model.

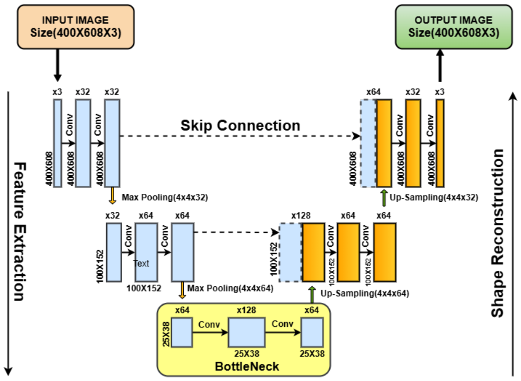


Figure 7. UNet Architecture.

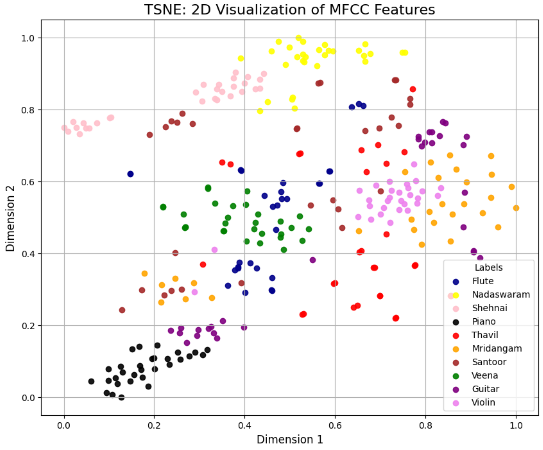


Figure 9. t-SNE visualization of MFCC features.

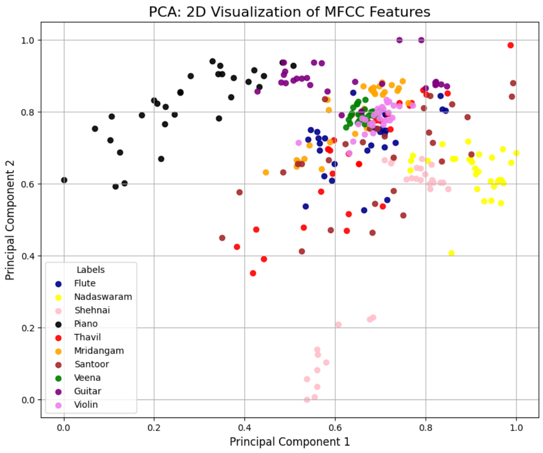


Figure 8. PCA visualization of MFCC features.

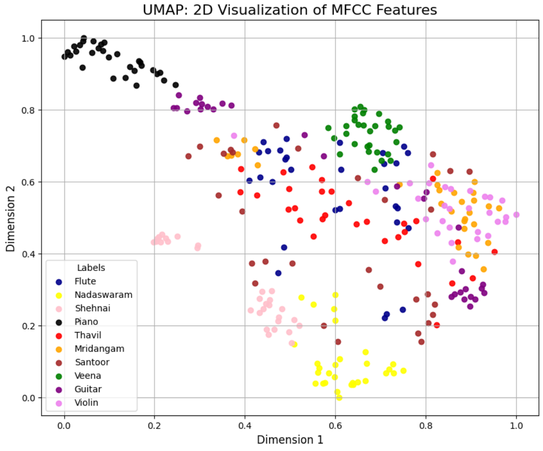


Figure 10. UMAP visualization of MFCC features.

samples are less. Compared to MFCC-UMAP, this visualization reflects a significant improvement in class separation, making UMAP effective and similar to projecting CNN features onto 2-dimensional space.

2) Wavelet spectrogram features using CNN: The results are shown in Figures 14, 15 and 16.

In Figure 14, PCA is used for 2-dimensional projection. As expected, the scatter plot of samples do not present clear clusters. Yet, we see samples from similar-sounding instruments are closely placed, and though there are no clear clusters, there is a vague separation visible, suggesting that the most significant variances in the features are somewhat aligned with class boundaries.

In Figure 15, t-SNE is used for visualization of samples from CNN features extracted from spectrograms. t-SNE is especially good in preserving the local structure of the data distribution in high dimension. The resulting visualization displays well-separated and compact clusters for each instrument class. Instruments such as the shehnai, thavil, and flute show clearly defined boundaries, indicating strong class discrimination.

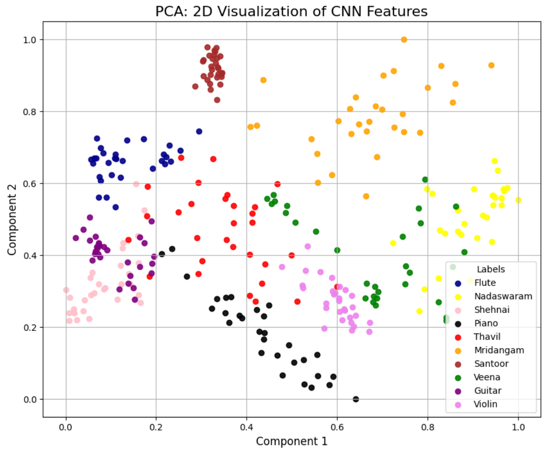


Figure 11. PCA visualization of STFT CNN features.

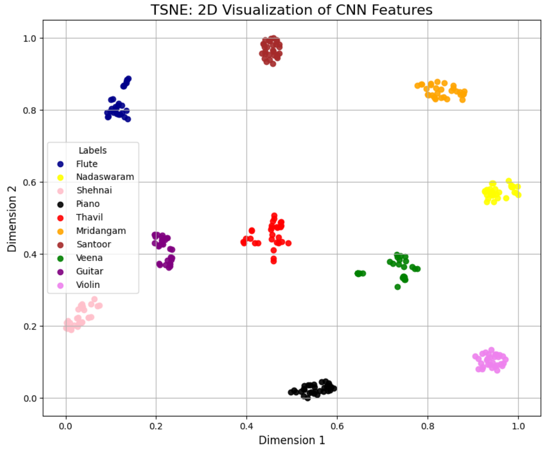


Figure 12. t-SNE visualization of STFT CNN features.

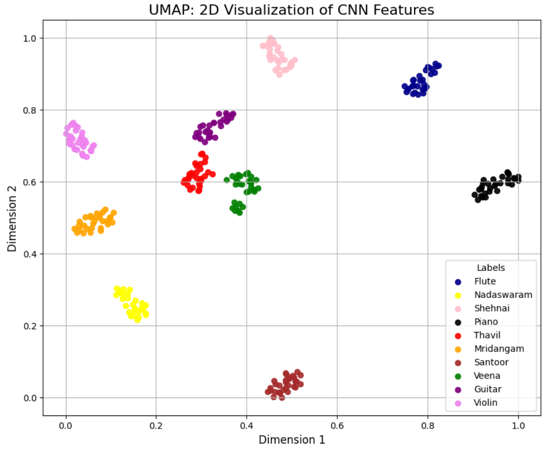


Figure 13. UMAP visualization of STFT CNN features.

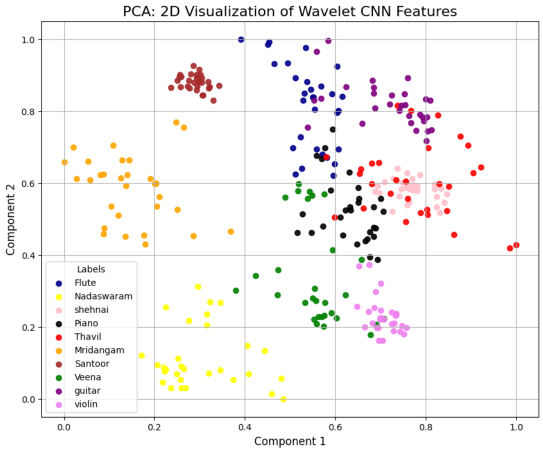


Figure 14. PCA visualization of wavelet CNN features.

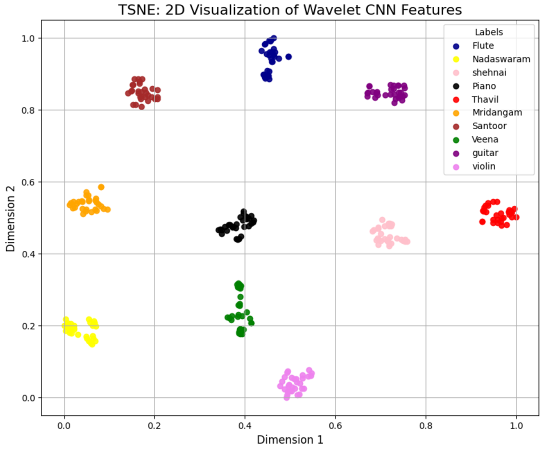


Figure 15. t-SNE visualization of wavelet CNN features.

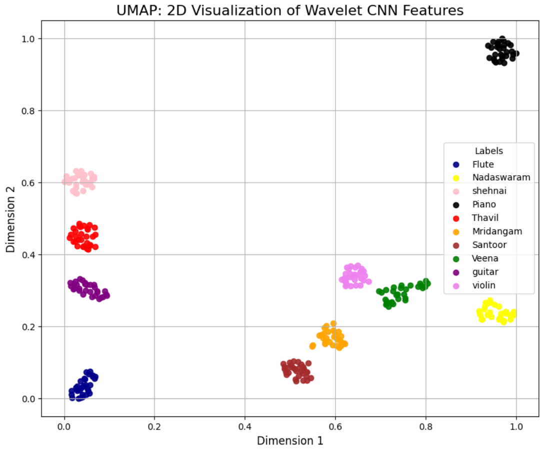


Figure 16. UMAP visualization of wavelet CNN features.

Unlike t-SNE, UMAP attempts to preserve both local and global structure. The UMAP projection, as shown in Figure 16, exhibits even better-separated clusters compared to t-SNE. Each instrument class occupies distinct regions in the scatter plot. This ensures that the Wavelet-CNN model has successfully captured both fine-grained and broad distinctions among instrument types.

C. *Visualization Results of UNet Features*

The STFT spectrograms and wavelet spectrograms were input into the UNet model, and features at the bottleneck layer were extracted. Thus, the original STFT features are compressed, and more abstraction is achieved at the UNet bottleneck. These compressed features are then used to visualize the data as a scatter plot on a 2-dimensional plane using PCA, t-SNE, and UMAP.

1) STFT spectrogram features through UNET: The STFT features extracted from the bottleneck are visualized, and results are shown in Figures 17, 18 and 19.

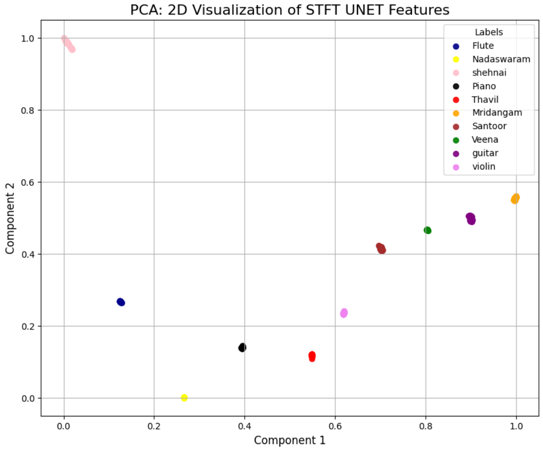


Figure 17. PCA visualization of UNET features.

In Figure 17, PCA projected samples forming compact clusters but widely separated. There are overlappings of samples from different instruments as well as splitting of samples from the same instrument. Flute forms a well-separated cluster. The other distinctive characteristics of Figure 17 is that samples appear to be compressed along the first principal component. As a linear dimensionality reduction method, PCA is less effective in separating the complex, non-linear relationships among UNET features.

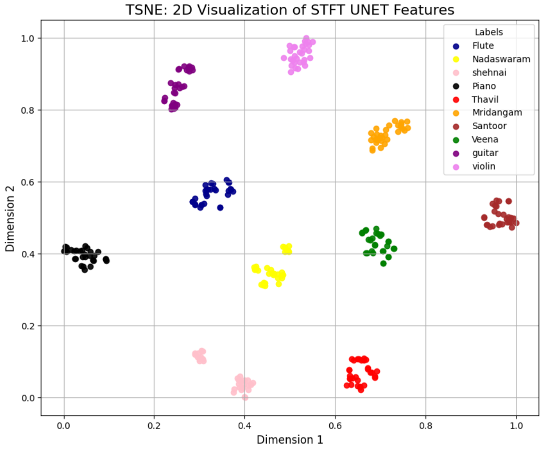


Figure 18. t-SNE visualization of UNET features.

In Figures. 18 and 19, the UMAP and t-SNE results on UNET features show clear, well-separated clusters for almost all instruments. UMAP reveals compact groupings, especially for flute, piano, nadaswaram, and violin, with low intra-cluster distances for thavil, piano, and shehnai. t-SNE also highlights distinct clusters. While minor dispersion exists in t-SNE, overall class separability is clear.

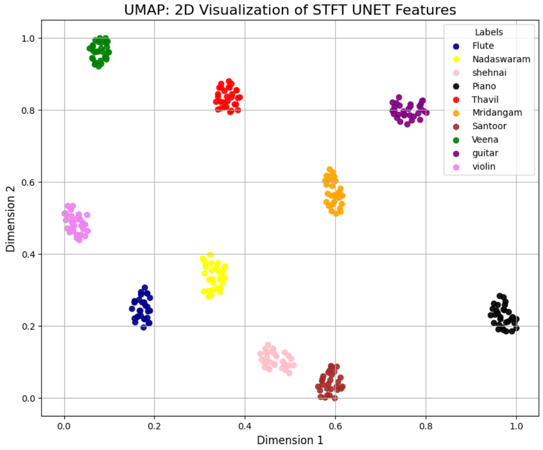


Figure 19. UMAP visualization of STFT UNet features.

2) Wavelet spectrogram features through UNET: The STFT features extracted from the bottleneck layer of UNET are used as input to the visualization tools. The results are shown in Figures 20, 21 and 22.

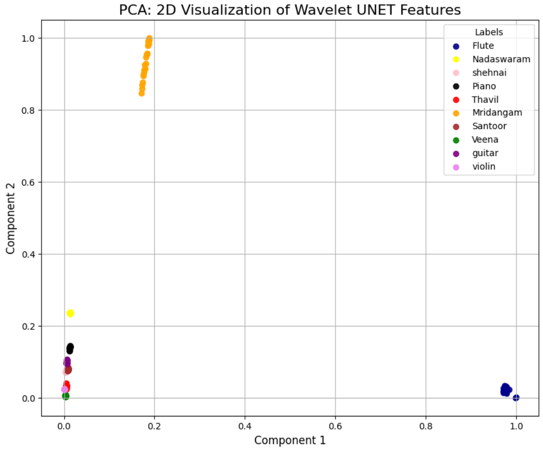


Figure 20. PCA visualization of Wavelet UNET features.

Figure 20 presents PCA results. Some instrument classes show partial separation, while a few samples from different classes overlapped. Despite limitations of the PCA being a linear projection algorithm, classes like Mridhangam and Piano form relatively distinct groups. But the shehnai and guitar music samples are overlapping.

In Figure 21, results using t-SNE are presented. We can see tightly clustered and well-separated groups. The strong local separation indicates that the model effectively distinguishes between instrument classes. This assures the improved ability of UNET to capture reliable feature representation.

Figure 22 shows the result when UMAP is used as the visualization tool. The figure shows distinct and compact clusters for each instrument class. The clear separation suggests that the combination of the Wavelet-UNET model effectively captures both local and global patterns in the data. Instruments like the flute, nadaswaram, and violin exhibit strong class distinction, which is what is perceived by human.

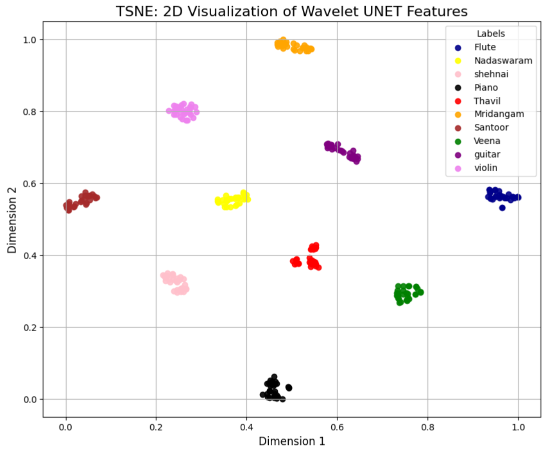


Figure 21. t-SNE visualization of Wavelet UNET features.

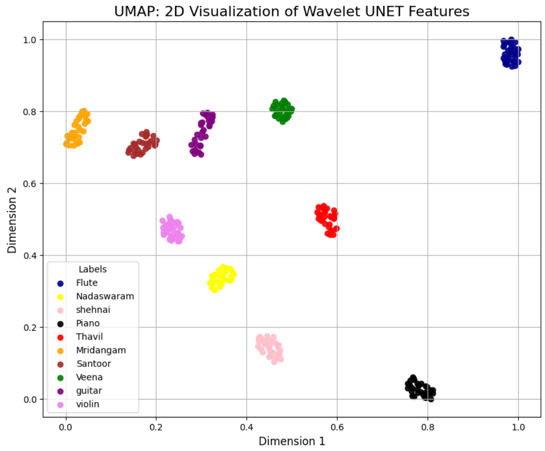


Figure 22. UMAP visualization of Wavelet UNet features.

TABLE I. COMPARISON OF VISUALIZATION TECHNIQUES BASED ON SILHOUETTE SCORES

Features Technique	MFCC	STFT (CNN)[a]	STFT (UNet)[b]	Wavelet (CNN)[a]	Wavelet (UNet)[b]
PCA	7.30	30.32	73.74	31.99	69.52
t-SNE	14.22	**84.72**	**78.09**	**84.35**	**78.83**
UMAP	**19.18**	78.66	75.05	79.20	74.47

The Silhouette scores, which are the ratio of interclass and intraclass distances, are displayed in Table I. High silhouette score means better separation between more compact clusters.

PCA demonstrates moderate performance for MFCC features and STFT spectrogram features using CNN but performs significantly better for the STFT features using UNET. t-SNE and UMAP outperform PCA in all experiments, with t-SNE achieving the highest silhouette scores. Both t-SNE and UMAP show similar performance for the STFT features using UNET, indicating their suitability for high-dimensional feature visualization. t-SNE gives a better compression closer samples are more tightly packed, and far-away samples are pushed farther away.

V. CONCLUSION AND FUTURE WORK

This study aims to find the correct tools to successfully visualize complex audio signals from musical instruments using machine learning and deep learning techniques. MFCC features, STFT, and wavelet spectrogram features were extracted to visualize the music samples in a two-dimensional plane. Three visualizations tools namely PCA, t-SNE, and UMAP are used. STFT features and wavelet coefficients were converted to spectrograms. Deep learning models and UNet were used to obtain a compressed version of the spectrogram image features. t-SNE and UMAP gave the best results, showing well-separated clusters. Though t-SNE gives better silhouette scores, UMAP projections are more akin to human perception. This is because t-SNE works locally whereas UMAP works on the manifold space which preservea the global information about sample distances.

MFCC features are proven to be the best for applications like speaker identification. But, for our application it is not. Instead of using MFCC features, concatenated spectrograms from sequential segments form an image for the whole musical sample. Analyzing those images are a better way to get distinct, separated clusters. The CNN model is sensitive to hyper parameters. The UNet model is robust and works better at all times. We train and extract the features, and it gives distinct separation between clusters. It is difficult to quantify the correctness of the results as far as human perception is concerned. For further investigation, we will

- Find the first few eigenvalues to check how fast the eigenvalues are diminishing and how that is reflected when the data is projected on the plane of the first two eigenvectors.
- Compare the interclass distances resulting from three different visualization algorithms and whether the relative distances from different methods are similar or not.
- Implement SOM as a tool for 2D visualization.

We will also extend this work for music generation, combining music generated by different instruments.

REFERENCES

[1] Goutam Chakraborty, Cedric Bornand, Lokesh Ayyaswamy, Subhash Molaka, Praveen Reddy, and Lakshman Patti, “Visualizing proximity of audio signals from different musical instruments: A two step approach”, in *DBKDA 2025: The Seventeenth International Conference on Advances in Databases, Knowledge, and Data Applications*, Accessed: March 9, 2025, Porto, Portugal: IARIA, Mar. 2025, pp. 25–31, ISBN: 978-1-68558-244-9.

[2] Meinard Müller, “The Fourier Transform in a Nutshell”, in *Fundamentals of Music Processing*. Aug. 2015, pp. 39–57, ISBN: 978-3-319-21944-8.

[3] Leland McInnes, John Healy, and James Melville, “Umap: Uniform manifold approximation and projection for dimension reduction”, *arXiv preprint arXiv:1802.03426*, 2018.

[4] Anca Popescu, Inge Gavat, and Mihai Datcu, “Wavelet analysis for audio signals with music classification applications”, in *2009 Proceedings of the 5-th Conference on Speech Technology and Human-Computer Dialogue*, 2009, pp. 1–6. DOI: 10.1109/SPED.2009.5156187

[5] S. Memon M. A. Hossan and M. A. Gregory, "A novel approach for MFCC feature extraction", *2010 4th International Conference on Signal Processing and Communication Systems*, pp. 1–5, 2010. DOI: 10.1109/ICSPCS.2010.5709752

[6] Boyang Zhang, Jared Leitner, and Samuel Thornton, "Audio recognition using mel spectrograms and convolution neural networks", *Noiselab University of California: San Diego, CA, USA*, 2019.

[7] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation", in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, Springer, 2015, pp. 234–241.

[8] Tamás Pál and Dániel T Várkonyi, "Comparison of Dimensionality Reduction Techniques on Audio Signals", in *ITAT*, 2020, pp. 161–168.

[9] Karim Mohammed Rezaul, Md Jewel, Md Shabiul Islam, KNEA Siddiquee, N Barua, MA Rahman, et al., "Enhancing Audio Classification Through MFCC Feature Extraction and Data Augmentation with CNN and RNN Models", *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 7, pp. 37–53, 2024.

[10] Hann function, *Hann function — Wikipedia, the free encyclopedia*, Online availability verified on 2 December, 2025, 2025. [Online]. Available: https://en.wikipedia.org/wiki/Hann_function

[11] Eva Wesfreid, "STFT Time-Frequency Visualization: Application to Sound Signals", *Image Processing On Line*, 2013, Preprint, September 18, 2013. Online availability verified on 2 December 2025, ISSN: 2105-1232. [Online]. Available: https://dev.ipol.im/~eva/STFT.pdf

[12] Dongyu Wang, Canghong Shi, Xiaojie Li, Kai Peng, Xianhua Niu, and Sani M. Abdullahi, "Audio Manipulation Detection Based on Wavelet Spectrogram and Multidimensional Feature Fusion with Dual-Channel CNN", *SSRN Electronic Journal*, 2024. DOI: 10.2139/ssrn.5106730

[13] Samuele Battaglino and Erdem Koyuncu, "A generalization of principal component analysis", in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 3607–3611.

[14] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-SNE", *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.