Visualizing Proximity of Audio Signals from Different Musical Instruments - A Two Step Approach

Goutam Chakraborty *
Madanapalle Institute Of Technology & Science, Madanapalle, India
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

A. Lokesh‡, Subhash Molaka§, Praveen Kumar Reddy Sangati¶, Lakshman Patti‖
Department of Computer Science Engineering and Artificial Intelligence‡§¶‖
Madanapalle Institute Of Technology & Science, Madanapalle, India‡§¶‖
Email: lokeshreddy2680@gmail.com‡, molakasubhash@gmail.com§,
prawinreddy1909@gmail.com¶, lakshmanpatti99@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 the 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 identify their variances and quantify their similarities and distances. 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 STFT spectrogram extracted using CNN provide the best compressed representations, which are better visualized using t-SNE and UMAP techniques, achieving silhouette scores of 84% and 81%, respectively. The STFT spectrogram features are compressed more effectively using UNet, resulting in improved cluster visualization with t-SNE, UMAP, and even with PCA, with silhouette scores of around 75%.**

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

I. Introduction

Each music sample has unique context-dependent audio characteristics, making audio classification a challenging multi-modal task. Additionally, training classifiers requires annotated signals, which are difficult to obtain.

In this project, our aim is to analyse audio signals from different musical instruments. The extracted features are high-dimensional. We project them onto a two-dimensional plane to visualize their similarities and dissimilarities. For our experimental study, six musical instruments were selected, five of which are traditional Indian Instruments: Flute, Nadaswaram, Thavil, Santoor, and Veena. We also included the Western classical instrument piano and compared the audio characteristics of the music produced by the instruments. These traditional instruments possess unique tonal features. Flute and Nadaswaram are wind instruments that produce sound through air vibration, the flute being side-blown and made of bamboo, while Nadaswaram is a long wooden pipe with a conical end. Santoor (struck) and Veena (plucked) are string instruments, with Veena's dual resonators, add uniqueness to the music. Thavil, a South Indian percussion drum, has a hollow wooden shell with stretched leather membranes, and is played by hand or stick. The piano produces sound by hammering strings when keys are pressed.

Traditionally, Fourier Transform (FT) [1] and Fast Fourier Transform (FFT) [1] was used for signal analysis. But this approach cannot capture sequential contextual audio information. Advances in speech processing introduced techniques like Short-Time FT (STFT) [2] , Wavelet Transform (WT), and Mel-Frequency Cepstral Coefficients (MFCC) [3]. After using such audio feature extraction tools, machine learning techniques including deep neural networks that compress high-dimensional audio data. This effectively facilitated viewing music pieces, originiated from different instruments, as compact scatterplots on a plane. The overall plan is shown in Figure 1.

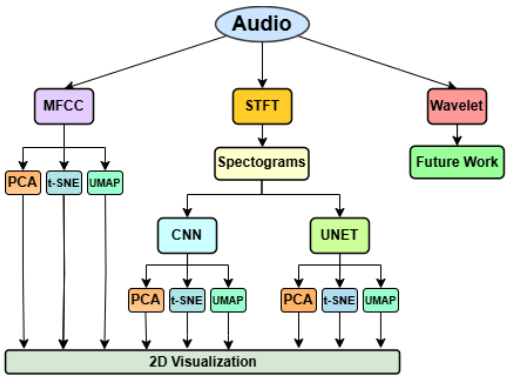


Figure 1. Overall plan for the Experiments.

MFCC features were extracted from the musical samples [3]. Using standard MFCC window lengths (25 milliseconds), even a few seconds of audio signal generate a high-dimensional feature vector. In this high dimension, the distribution of distances between samples exhibits low variance, making two-dimensional visualization ineffective.

We used a second step of feature compression using deep neural network. The effectiveness of the proposed methods was validated through several experiments by projecting the data onto two dimensions [4]. For spectral analysis, we used the

Short-Time Fourier Transform. We converted the STFT features into spectrogram images [2]. These spectrograms serve as visual representations of the music features. To extract features from spectrogram images, we used a CNN model [5], and a U-Net model [6]. CNN and UNet were trained on STFT spectrograms. Features were extracted from the output of filter layers of the CNN where they are input to the dense classification layer. Similarly, features were extracted from the bottleneck layer in UNet. Section III details how CNN and U-Net architectures extract features from images through their distinct approaches.

To visualize music signals on a two-dimensional plane, we used PCA (a linear method), t-SNE, and UMAP. MFCC features are directly fed into the above three visualization tools.

The remainder of this paper is organized as follows. Section II reviews related work. Section III describes the methodology, including data collection and pre-processing, 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 [7]. Two datasets with 10 classes and an augmented version of one (using pitch shifting, time-stretching, and noise) were used. MFCC features were extracted, CNN and RNN-LSTM models were trained. CNN performed better on smaller datasets, while RNN-LSTM excelled on larger ones.

In the work reported in 2020 [4], 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 meaningful structure. The conclusion was that Iso-Map failed to work with this high-dimensional data.

In another work on the audio classifier, reported in 2020 [5], 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 and Experiments

This section outlines the paper's workflow, including data collection and preprocessing, feature extraction, projection of higher dimension into 2D, and the proposed method, as detailed below.

A. Data Collection and Pre-processing

Audio samples are collected from open public platforms like YouTube and recorded media, ensuring each sample captures the instrument's unique tonal and spectral characteristics without background noise or audio from other instruments.

We collected 180 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 high-quality data for analysis.

B. Feature Extraction

1) MFCC Feature Extraction: MFCC feature is a standard for audio analysis in music, speech recognition, and speaker identification. Pre-processing standardizes samples to 30 seconds by padding or trimming, followed by sampling at 44,100 Hz for high-quality signal preservation.

The MFCC extraction process begins with pre-emphasis to enhance high-frequency components. The 30-seconds audio is segmented into 25ms non-overlapping frames (1,201 frames, each with 1,103 samples). A Hanning window is applied to smooth edges and reduce distortions. The Discrete Fourier Transform (DFT) converts the signal to the frequency domain, capturing spectral characteristics. A Mel filter bank mimics human hearing by dividing the spectrum into 26 bands, reducing dimensionality while retaining essential information. A logarithmic transformation follows to compress the dynamic range. Finally, the Discrete Cosine Transform (DCT) decorrelates Mel-spectral coefficients, retaining the first 13 MFCCs used for classification.

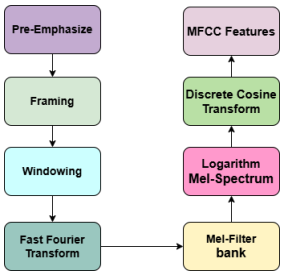
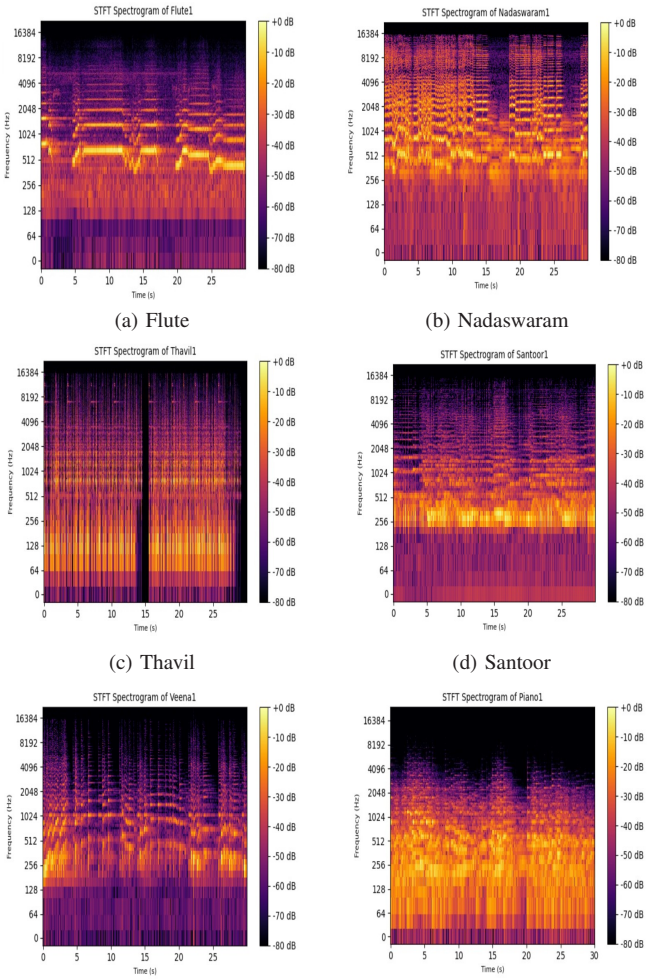


Figure 2. The process of MFCC Feature Extraction.

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 essential audio characteristics, preserving tonal, timbral, and rhythmic features for classification and visualization.

The dataset for each musical instrument consists of 30 samples, each with 15,613 values, resulting in a data matrix of 30x15,613 for each instrument.

2) STFT Spectrogram Generation: The audio waveform of 30 seconds is divided into equal parts with a window of size 25 milliseconds. Each segment contains 1,102 samples. DFT of each segment is computed using the Fast Fourier Transform (FFT). The result of the FFT for each segment represents the frequency content of the audio within that window. These frequency domain representations are then concatenated to form the spectrogram image. The spectrogram displays the frequency spectrum where the intensity of a frequency is converted into brightness. The images corresponding to sequential windows provide a view of the audio spectral characteristics over the duration of the signal [8]. Figure 3 shows the STFT spectrograms of each musical instrument capturing the tonal and spectral characteristics of the instruments as images.



(a) Flute (b) Nadaswaram

(c) Thavil (d) Santoor

(e) Veena (f) Piano

Figure 3. Sample Spectrograms for each musical instrument.

C. Projection of higher dimension into 2D

1) Principal Component Analysis (PCA): PCA is a lightweight linear dimensionality reduction algorithm that identifies variance directions (eigenvectors) from the covariance matrix, which is symmetric with orthogonal eigenvectors. Projecting data onto the first two eigenvectors in 2-D highlights key data distributions [9]. In our project, PCA's effectiveness depends on the compression algorithm applied to MFCC or STFT data. We noted that UNet-compressed STFT spectrograms form well-clustered visuals even with PCA.

2) t-Distributed Stochastic Neighbor Embedding (t-SNE): t-SNE [10] is a non-linear dimensionality reduction technique, preserves the local structure of high-dimensional data by converting distances into probabilities and placing similar points close in lower dimensions. When applied to high dimensional feature space, t-SNE effectively captures complex patterns, revealing distinct clusters of audio data from musical instruments and uncovering structures.

3) Uniform Manifold Approximation and projection (UMAP): UMAP [2] is a non-linear dimensionality reduction technique that preserves both local and global structures, making it more effective for visualizing high-dimensional data in 2-D. By modeling data relationships as a graph and maintaining these connections in lower dimensions, UMAP faithfully presents complex distributions in low dimension.

D. Proposed Method

To visualize the audio features on a two-dimensional plane, we employed three visualization algorithms.

1) Visualization of MFCC Features: The MFCC features are extracted from the audio signals, resulting in a dataset with dimension 15,613 from every music piece of 30 seconds duration. In total, we have 180 samples for 6 instruments. Then we directly visualized them using PCA, t-SNE and UMAP.

2) Feature Extraction using CNN: The spectrograms of audio samples are used to train a CNN model with labels of the musical instruments. The CNN model architecture includes two convolutional layers, each followed by max-pooling layers, a flatten layer, and a couple of dense layers.

In Figure 4, the architecture of the CNN model used for training is illustrated. The input to the model is a spectrogram of size 400x600x3. The first convolution layer is with 16 filters to extract features, resulting in an output dimension 400x600x16. A MaxPooling layer with a 2x2 kernel reduces the spatial dimensions to 200x300x16. This is followed by a second Convolution layer with 32 filters and after applying 2X2 size max pooling, the resulting output is reduced to 100x150x32. The output is flattened into a vector and fed into a dense layer classifier. Since the musical instruments are known, the network is trained as a supervised classifier with feature vectors as input.

The extracted features are visualized using the visualization techniques: PCA, t-SNE, and UMAP. Feature vector scatter plots are projected on a 2-D plane to visualize the similarities and distances of the audio signals.

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 structure: 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 5, the spectrogram of size 400×600×3 is input to the UNET model. Initially, two Convolution layers with 32 filters are used, followed by MaxPooling with a pool size of 4×4. Next, two more Conv2D layers with 64 filters are applied, followed by another MaxPooling operation. After that, two additional Convolution layers are applied one with 128 filters and the other with 64 filters leading to the bottleneck layer, which captures the encoded representation of the input.

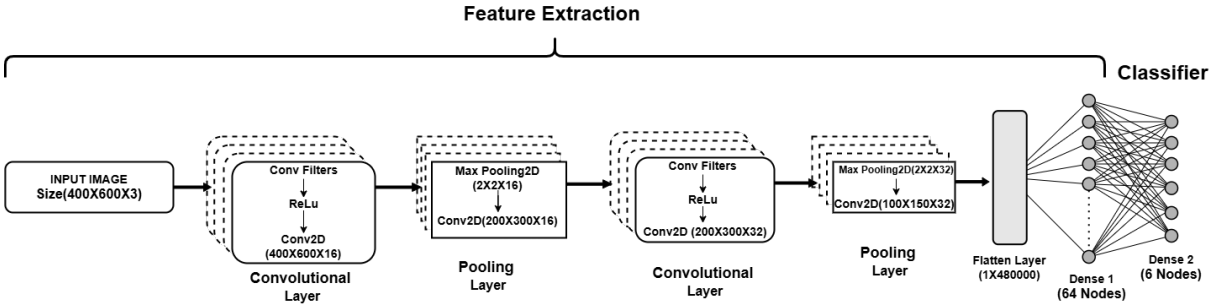


Figure 4. Architecture of CNN Classifier Model.

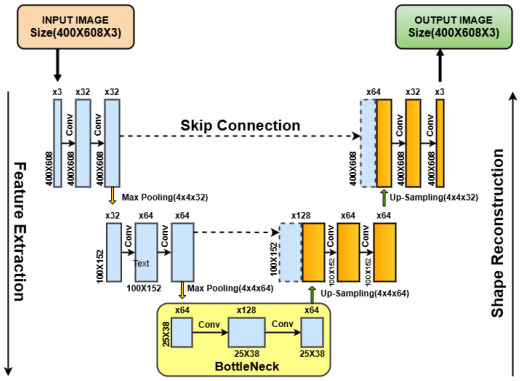


Figure 5. UNet Architecture.

From the bottleneck layer, the features are upsampled using a transposed convolution operation with 64 filters of size 4×4. These upsampled features are concatenated with the previous layer from the bottleneck. The process of Convolution, Upsampling, and concatenation is repeated for feature reconstruction. Finally, this process produces the reconstructed image. It is an unsupervised method.

The features from the bottleneck layer are extracted and used to visualize the data in a 2-D scatter plot using PCA, t-SNE and UMAP.

First, we train the UNet using data from all six classes. The compressed features from the UNet bottleneck, from different classes, are superimposed in the feature space and cannot be projected as separate clusters on 2-Dimensional plane. In the next experiments, we trained UNet separately with individual classes of features. After this training, the features from UNet bottleneck layer are used. The visualization algorithms projected them into clear isolated groups.

IV. Experiment and Results

In this section, we present the visualization results of MFCC features, and the extracted features from 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 plot of 30×6 datapoints are shown in Figures 6, 7, and 8.

In Figure 6, the scatter plot for different instruments are completely mixed up. In Figure 7, we observe that Piano and Thavil samples are far apart and compact. It is as expected, because they produce very different types of sound. Veena samples are compact but mixed with Flute, Santoor, and Nadaswaram, which are overlapping and spread out.

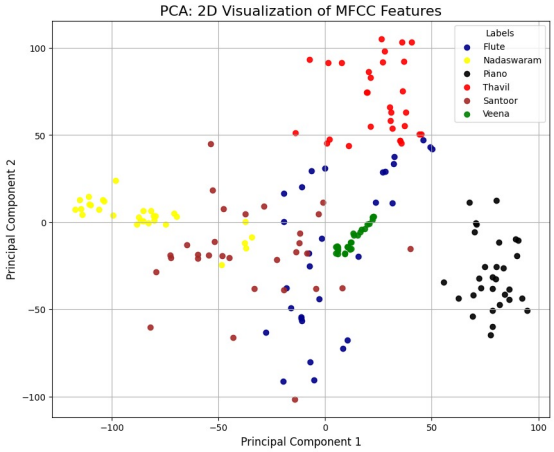


Figure 6. PCA visualization of MFCC features.

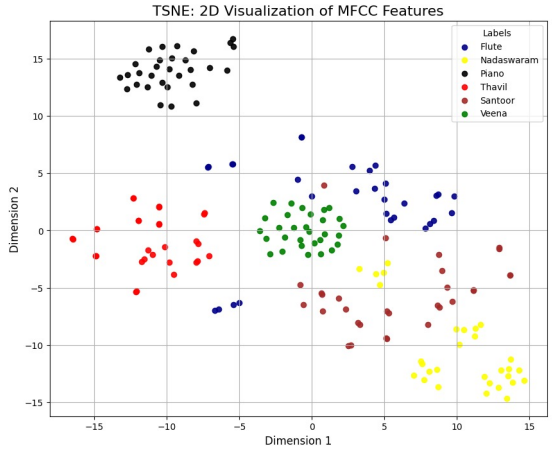


Figure 7. t-SNE visualization of MFCC features.

In Figure 8, the Piano samples are far from the others, while the rest form compact clusters. But these clusters are close to each other due to small interclass distances. The overall conclusion is that the MFCC feature, used directly as input to the visualization software, does not result in clear clusters.

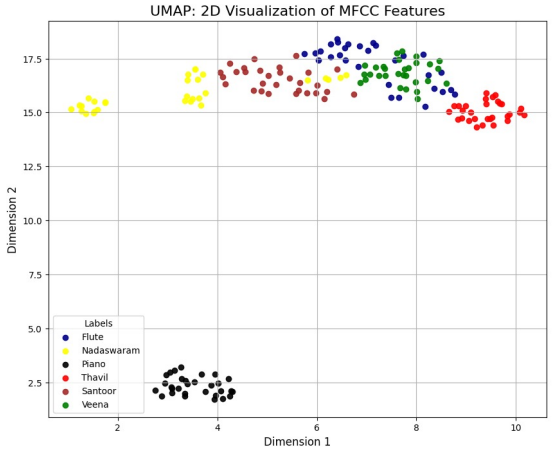


Figure 8. UMAP visualization of MFCC features.

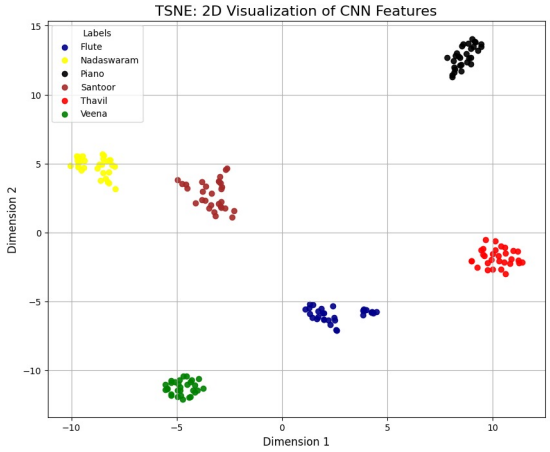


Figure 10. t-SNE visualization of CNN features.

B. *Visualization of Extracted features from CNN model*

Features extracted from the output of CNN, i.e., input to the dense layer for classification, are much lower in number than the MFCC features. These extracted features are fed into PCA, t-SNE, and UMAP for visualization. The results are shown in Figures 9, 10 and 11.

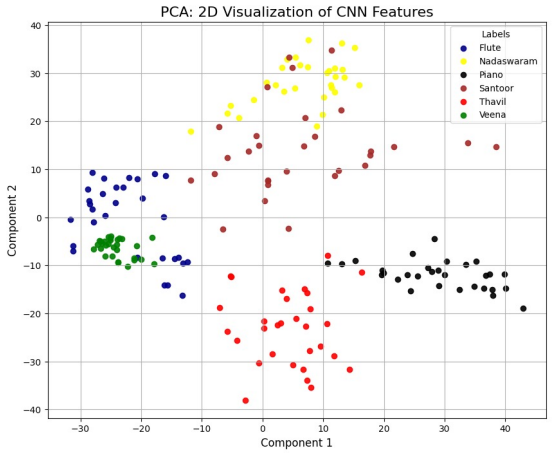


Figure 9. PCA visualization of CNN features.

In Figure 9, the thavil, nadaswaram, and piano samples form well-separated clusters though the sample points are scattered over a wide area. The santoor samples overlap with the nadaswaram cluster, suggesting some similarity in their features. The flute and veena clusters are positioned very close, indicating the related characteristics of the two instruments.

In Figure 10, all the music samples are clearly separated with large inter-cluster distances. The clusters are fairly compact, i.e., intra-cluster distances are not large. The piano and veena samples form compact clusters.

When UMAP was used for the 2D projection, as shown in Figure 11, we got compact non-overlapping clusters with clear large inter-cluster distances. The santoor and nadaswaram clusters are close to each other.

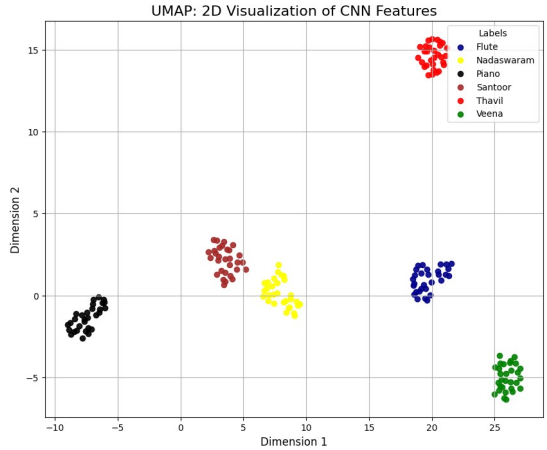


Figure 11. UMAP visualization of CNN features.

C. *Visualization Results of UNet Features*

The STFT spectrograms features were input into the UNet model and features at 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 scatter plot on a 2D plane using PCA, t-SNE and UMAP. The scatter plot results are shown in Figures 12, 13 and 14.

In Figure 12, we got clusters in which samples of every instrument are very compact. Veena and santoor clusters are very close to each other, which is quite contrary to what we got using UMAP and t-SNE. Thavil and piano cluster distances also close. Finally, nadaswaram and flute clusters are very far from the remaining clusters. Two things are to observe here, (1) the first eigenvalues are large and the second eigenvalues are around one-third of the first eigenvalues, (2) the interclass distances are very different from t-SNE or UMAP results, whereas the t-SNE and UMAP results are similar. This is due to linear projection with PCA.

In Figures 13 and 14, we got very well separated clusters where the intra-class distances are small resulting in compact

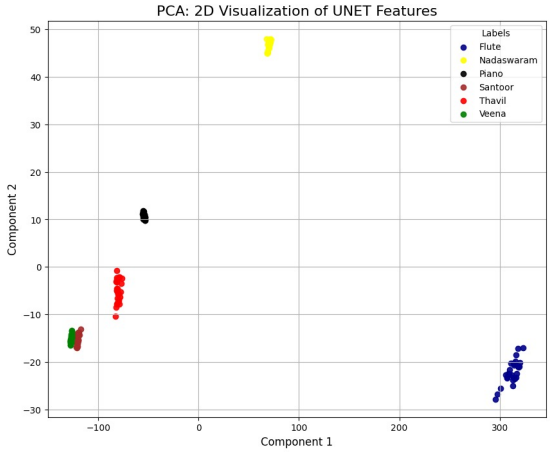


Figure 12. PCA visualization of UNET features.

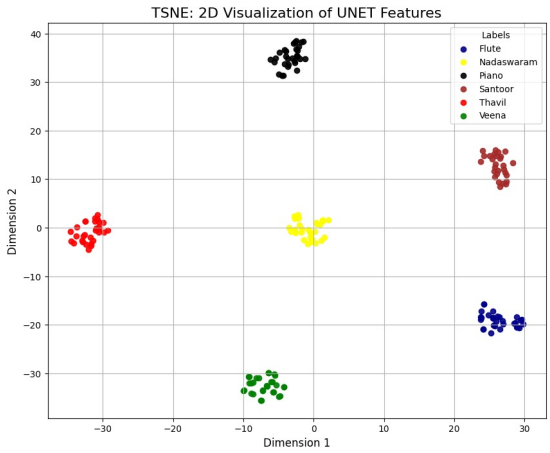


Figure 13. t-SNE visualization of UNET features.

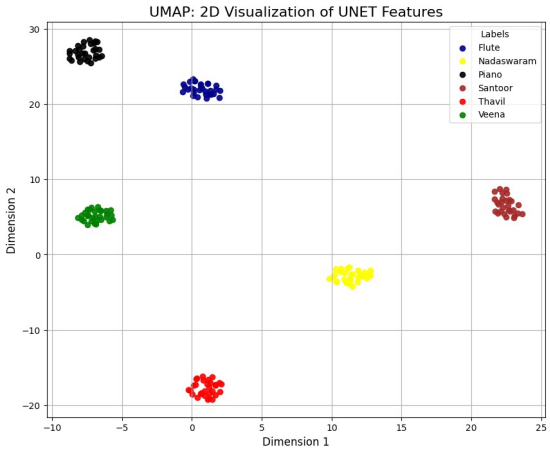


Figure 14. UMAP visualization of UNet features.

clusters. The inter-class distances are as expected and they are similar for the two visualization algorithms.

TABLE I
COMPARISON OF VISUALIZATION TECHNIQUES BASED ON SILHOUETTE SCORES

Visualization Technique \ Feature Selection	MFCC Features	STFT Spectrogram Features (CNN)	STFT Spectrogram Features (UNet)
PCA	32.85	35.21	76.50
t-SNE	37.37	83.99	74.60
UMAP	40.78	81.02	74.64

The Silhoutte score, which is the ratio of interclass and intraclass distances, are displayed in Table I.
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 for non-linear feature distributions, with t-SNE achieving the highest silhouette score for STFT Features using CNN and UMAP performing best for MFCC. Both t-SNE and UMAP show similar performance for the STFT features using UNET, indicating their suitability for high-dimensional feature visualization.

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 and STFT features were extracted and used to visualize their scatterplots on a two-dimensional plane by PCA, t-SNE, and UMAP. STFT features were converted to spectrograms, and Deep learning models CNN and UNet, were used to obtain a compressed version of the spectrogram image features. To visualize them in 2D, t-SNE and UMAP gave the best results, showing well-separated clusters.

It is difficult to quantify the correctness of the results. 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 wavelet transform to extract music features, and then use wavelet spectrogram like, STFT spectrogram, and compare the results.
- 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] M. Müller, “The fourier transform in a nutshell”, in Aug. 2015, pp. 39–57, ISBN: 978-3-319-21944-8.

[2] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction”, *arXiv preprint arXiv:1802.03426*, 2018.

[3] S. M. 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.

[4] T. Pál and D. T. Várkonyi, "Comparison of dimensionality reduction techniques on audio signals.", in *ITAT*, 2020, pp. 161–168.

[5] B. Zhang, J. Leitner, and S. Thornton, "Audio recognition using mel spectrograms and convolution neural networks", *Noiselab University of California: San Diego, CA, USA*, 2019.

[6] O. Ronneberger, P. Fischer, and T. 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.

[7] K. M. Rezaul *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.

[8] E. Wesfreid, "Preprint september 18, 2013 stft time-frequency visualization application to sound signals",

[9] S. Battaglino and E. 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.

[10] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.", *Journal of machine learning research*, vol. 9, no. 11, 2008.