Power System Technology
ISSN:1000-3673

Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

Transformer-Based Audio Classification: Adapting Sequence-Classification Techniques from Nlp

C. S. Sonali[1], Suma K V[2,*], Chinmayi B S[3], Ahana Balasubramanian[4]

[1,2,3,4]*Department of Electronics and Communication*
Ramaiah Institute of Technology, India
**Corresponding author: sumakv@msrit.edu*

Abstract:- Audio classification is very useful in fields such as music and speech recognition. An important step in this direction is feature extraction from the signal and popularly used features are MFCCs and Mel-Spectrograms. These features are converted into spectrograms for the purpose of classification. Researchers have employed techniques such as machine learning and deep learning, for classifying spectrograms, however they can have high computational cost. An effort in reducing the computational cost is exploring a more straightforward approach inspired by sequence classification in NLP. This paper proposes a Transformer-based model for audio classification utilizing MFCCs as features. The proposed model is benchmarked against the Speech Commands v0.02, UrbanSound8k and ESC-50 datasets and has presented strong performance, with the highest accuracy of 95.2% attained upon training the model on the UrbanSound8k dataset. The model consists of a mere 127,544 total parameters, and hence is lightweight yet highly efficient at the task of audio classification. This work leads to an efficient and light on computation cost solution for audio classification which can be helpful in the field of Machine Learning and Data Science.

Keywords: *Transformer, MFCCs, Audio classification, Sequence classification.*

1. Introduction

Spoken audio is a temporal phenomenon, i.e., it consists of a sequence of acoustic events over time. Classification of audio has multiple applications in the field of Artificial Intelligence (AI) and Data Science such as Music Genre Classification, Anomaly detection in surveillance systems, Snore Detection, Speaker Identification, etc. Audio classification typically involves feature extraction as the first step after preprocessing. Feature extraction of the audio usually involves extraction of frequency domain representation as it offers greater insight into the signal. In the past, Convolutional Neural Networks (CNNs) have been heavily used for classifying audio using these extracted features. Recently, there has been a surge in exploration of new architectures whose performance is better in the audio classification task.

Recurrent Neural Networks (RNNs) and Long-Short Term Memory (LSTM) networks have been used to implement encoder-decoder models for Natural Language Processing (NLP) applications. In the encoder, the model creates a context vector representing the significant information in the input sequence. In the decoder, the model generates output sequences using

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

the context vector. These have been used in applications such as machine translation. Despite their strengths, these networks may encounter difficulties with the bottleneck problem, which involves representing all input information as a single context vector. The attention mechanism was developed as a solution to the bottleneck problem [1]. Attention can hence be a supportive tool in classifying audio, which is inherently a temporal phenomenon. Transformers [2], which have been used in NLP tasks, can be modified to classify audio.

Mel Frequency Cepstral Coefficients (MFCCs) embodies parts of human vocalization and experience. MFCC represents the logarithmic perception of intensity and tone. Actions for evaluating MFCC features are to be performed in order. Initially, for each set up an estimate of periodogram of the flexibility band or range is created. Subsequently, assessment of the DCT of the filter bank strengths is obtained. DCT coefficients starting from 1 to 30 are taken into further consideration. These 30 features represent the vital information of audio signal [3]. The MFCCs reflect the nonlinear relationship between the human ear and the frequency of the sound heard [4]. The log-spectrum already takes into account perceptual sensitivity on the magnitude axis, by expressing magnitudes on the logarithmic-axis. The other dimension is then the frequency axis. The beneficial properties of the MFCCs include: Quantification of the gross-shape of the spectrum (the spectral envelope), which is important in, for example, identification of vowels. It also it removes fine spectral structure (micro-level structure), which is often less important. It thus focuses on that portion of the signal which is typically most informative. Straightforward and computationally reasonably efficient calculation. This representation of audio signal is well-tested and understood.

The following sections are organized as follows: Section 2 provides a background, with discussion on attention, self-attention, multi-head attention and the transformer encoder, which is necessary to understand the proposed architecture; Section 3 provides a literature survey on the current transformer-based models used in audio classification; Section 4 discusses the proposed methodology with insight into the choice of feature extraction method, the inspiration for the architecture and the architecture itself; Section 5 provides the results achieved for multiple datasets, a comparison of performance with the work mentioned in literature and Section 6 concludes the paper while discussing possible future work using the proposed technology.

2. Background

This section aims to introduce and explain the fundamental concepts necessary to grasp the proposed work. It begins by discussing the concept of Attention and its significance, followed by an explanation of Self-Attention and its improvement through Multi-Head Attention. Finally, the Transformer Encoder layer is explained.

a. Attention Mechanism

RNNs consist of a recurrent link within their architecture that connects the present architecture with values from a preceding point in time. Researchers have stacked RNNs and also combined them to form bidirectional networks. These types of architectures have been heavily used in NLP tasks, such as Sentiment Analysis, Spam Detection, Part-of-Speech (POS) Tagging, Named Entity Recognition (NER), etc. Despite their ability to handle sequential data, these networks often face challenges when attempting to utilize crucial information that is

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

distant in time due to the vanishing gradient problem, where gradients can exponentially decay or explode during backpropagation through time.

LSTM Networks were introduced to overcome this shortcoming. LSTMs incorporate a specific context component into their architecture and employ specialized neural units that manage the flow of information using gates. Despite their ability to overcome the vanishing gradient problem, LSTM networks can still struggle with capturing long-term dependencies in sequences. If the information needed for prediction or classification is located far back in the sequence, it can be challenging for LSTMs to retain and propagate that information accurately.

Attention mechanism [1] was introduced to address the restrictions of traditional sequential models, such as RNNs, LSTMs, in capturing long-range dependencies effectively. In tasks such as POS tagging and NER, it is vital to consider both local and global context to make accurate predictions. Attention mechanisms provide a solution by allowing models to focus on relevant portions of the input sequence and dynamically weigh their importance during processing. The core of the transformer architecture is the attention mechanism. The attention mechanism provides a flexible and interpretable way to capture relationships between words or tokens in the input sequence. The idea of attention is to create the single fixed-length vector c by taking a weighted sum of all the encoder hidden state's. The weights focus on ('attend to') a particular part of the source text that is relevant for the token the decoder is currently producing. Attention thus replaces the static context vector with one that is dynamically derived from the encoder hidden states, different for each token in decoding.

b. *Self-Attention*

Self-attention is a mechanism used in transformers for capturing the relationships between different works or tokens in an input sequence. The input embedding, which are dense vectors representing the tokens, can play three roles during attention process, i.e., query (the current focus of attention), key (the preceding input being compared to the current focus) and value (used to compute the output for the current focus of attention). For capturing these, transformers use weight matrices W^Q, W^K and W^V. These weights will be utilized to project each input vector x_i into its role as a query, key or value. Given the computed projections, the score between the current focus of attention, x_i, and a preceding element x_j contains a dot product between the query projection q_i and the key projection k_j. The value of this dot product can range from $-\infty$ to ∞. The larger the value, the more similar the vectors being compared. These score computed for all $j \leq i$ can be made more effective by normalizing them using a softmax to create a vector of weights α_{ij}, that shows the proportional significance of each preceding input to the current focus. Given the proportional scores in α, an output vector y_i is now founded on a weighted sum over the value vectors, as shown in eq.1.

$$y_i = \sum_{j \leq i} \alpha_{ij} v_j \quad (1)$$

The dot product can result in an arbitrarily large value. Exponentiation of such values can result in numerical issues as well as loss of gradient during training. A scaled dot-product approach divides the outcome of the dot product by a factor linked to the size of the embeddings prior to passing them through the softmax. The typical approach consists of dividing the dot-

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

product with the square root of the dimensionality of the query and key vectors, as shown in eq. 2.

$$score(x_i, x_j) = \frac{q_i \cdot k_j}{\sqrt{d_k}} \quad (2)$$

This mechanism can now be parallelized by packing the input embeddings of the tokens of the input sequence into a single matrix X. This matrix can now be multiplied with the query 'Q' (eq. 3.), key 'K' (eq. 4.) and value 'V' (eq. 5.) weights to obtain projections.

$$Q = XW^Q \quad (3)$$

$$K = XW^K \quad (4)$$

$$V = XW^V \quad (5)$$

With these matrices, the query-key computations can be done simultaneously by multiplying Q and K^T in a single matrix multiplication. After scaling this product, applying softmax and multiplying the result with V, a vector embedding for each token in the input is obtained, as shown in eq. 6.

$$SelfAttention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (6)$$

c. *Multi-head Attention*

To enhance the expressiveness and robustness of the self-attention mechanism, transformers typically employ multiple attention heads. Each attention head learns different relationships and captures different aspects of the input sequence. The outputs of multiple attention heads are concatenated and linearly transformed to generate the final self-attention output. The attention heads operate in parallel, attending to different parts of the sequence and learning different aspects of the data.

For implementation of this notion, each head, i, in a self-attention layer is given its own set of key, query and value matrices: ${W_i}^K$, ${W_i}^Q$ and ${W_i}^V$. These are employed to project the inputs into separate key, value, and query embeddings separately for each head, with the remaining of the self-attention computation being unaltered. In multi-head attention, instead of using the model dimension d that's used for the input and output from the model, the key and query embeddings have dimensionality d_k, and the value embeddings are of dimensionality d_v (in [2] $d_k = d_v = 64$). Thus for each head i, there are weight layers ${W_i}^Q$, ${W_i}^K$, and ${W_i}^V$, and these get multiplied by the inputs packed into X to produce Q, K and V. To use these vectors in processing further, they have to be combined followed by reducing down to the original input dimension d. This is acieved by concatenating the outputs from each head followed by using yet another linear projection, to diminish it to the original output dimension for each token.

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

d. *Transformer-Encoder Layer*

The transformer encoder (fig. 1.) is a key component of the transformer architecture. It is a popular neural network architecture used for various NLP tasks, such as machine translation, language modeling and text classification. It is responsible for processing the input sequences and generating encoded representations that capture the contextual information of each element in the sequence.

Input Embeddings are continuous vector representations of the input sequence which capture the semantic meaning of the words or tokens in the sequence. These representations are the input to the encoder. Since the Transformer model doesn't have any sequential information, positional encodings are added to the word embeddings to capture the order or position of the words in the sequence. The transformer encoder layer, as introduced in [2], contains two sub-layers. The first is a multi-head self-attention mechanism and the second is a fully-connected feed-forward network.

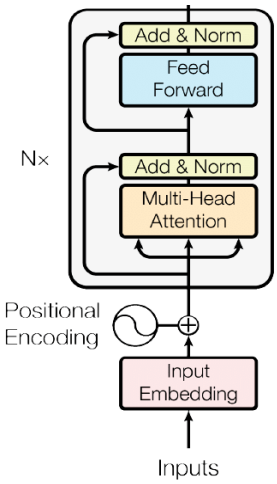

Fig. 1. Transformer encoder layer

The attention mechanism allows each word in the sequence to attend to other words in the same sequence, capturing the dependencies and relationships between different words. It calculates attention weights for each word by taking into account the relationships with all other words. Residual connection is employed around each of the two sub-layers, followed by layer normalization. The residual connections are added around each sub-layer to help with the flow of gradients during training. Layer Normalization is applied to normalize the outputs of each sub-layer.

3. Literature Survey

In recent times, an emergence of transformer based audio classification is observed. Table I. summarizes some of the literature surrounding application of transformers for audio classification. All of them have used Mel-Spectrograms for extracting the features from the

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

audio data, but the treatment of the spectrograms varies in different works. Apart from the surveyed papers [31, 32, 33], other works also employ similar mechanisms with some variations. As observed, all of the given works explore classification of the audios using the spectrograms extracted from the audio signals, in turn, converting the audio classification problem into an image classification problem. Taking inspiration from natural language processing, it is possible to introduce a more computationally efficient approach to audio classification.

TABLE I. LITERATURE SURVEY OF TRANSFORMER-BASED AUDIO CLASSIFICATION

Sl. No.	Feature extraction method	Model	Architecture Description	Reference
1	Mel-spectrogram	Audio Spectrogram Transformer (AST)	Spectrograms split into 16×16 patches with overlap and projected onto 1D patch embeddings. Positional embeddings are added to these patch embeddings. The embeddings are then fed through a Transformer and the classification output is obtained.	[5]
2	Mel-spectrogram	The Patchout faSt Spectrogram Transformer (PaSST)	Spectrograms undergo patch extraction and linear projection. Time and Frequency positional encodings are added to the projections. Dropping parts of the transformer input i.e. Patchout and adding of classification tokens is employed. The sequence is then flattened and passed through layers of self-attention.	[6]
3	Mel-spectrogram	Hierarchical Token-Semantic Audio Transformer (HTS-AT)	Mel Spectrograms split into patch windows and then split inside each window to capture the same time frame relation among frequency bins. Patch tokens sent into transformer-encoder blocks. Patch-Merge Layer implemented at the end of each block to reduce sequence size. Token-Semantic module combined to map final outputs into class feature maps.	[7]
4	Multi-Resolution Multi-Feature (MRMF)	Causal Audio	Multi-resolution Multi-filter feature patches with 3D positional embeddings are extracted. Feature	[8]

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

Sl. No.	Feature extraction method	Model	Architecture Description	Reference
		Transformer (CAT)	patches are sent as input to an acoustic attention network. Causal module proposed to reduce overfitting and improve interpretability.	
5	Mel-Spectrogram	Self-Supervised Audio Spectrogram Transformer (SSAST)	Similar architecture to AST. During self-supervised pre-training, random portions of the spectrogram patches are masked and the model is asked to find the correct patch at each masked position and to reconstruct the masked patch, which forces the AST to learn both temporal and frequency structure of the audio data.	[9]
6	Log-Mel Spectrogram	Multi-Scale Audio Spectrogram Transformer (MAST)	A Patch-Embed CNN is utilized to divide the Spectrogram into Spectrogram tokens, which are then processed using the MAST backbone for feature extraction. The class of the speech Spectrogram is projected by the MAST prediction head.	[10]
7	Mel-Spectrogram	Masked Autoencoding Audio Spectrogram Transformer (MAE-AST)	2D spectrograms split into 16×16 patches. Mask applied to some tokens for pretraining. Unmasked patch unrolled by the channel dimension, followed by the time dimension and then embedded into a vector via linear projection. These are then added to 1D sinusoidal positional embeddings and fed through the encoder. Mask tokens combined with encoder outputs and positional embeddings are added to all tokens before sending through the decoder during pretraining. For fine-tuning, encoder outputs are mean pooled to create a representation vector and the decoder is left untouched.	[11]

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

Sl. No.	Feature extraction method	Model	Architecture Description	Reference
8	Mel-Spectrogram	Separable Transformer (SepTr)	The spectrograms undergo tokenization and projection, followed by processing through vertical and horizontal sequential transformers. The Separal Transformer block is repeated L times, resulting in a SepTr model with a depth of L blocks. Finally, an MLP head takes the last class token as input to determine the final output.	[12]
9	Mel-Spectrogram	CNN+Transformer Cross Model Knowledge Distillation (CMKD)	The spectrogram is fed through CNN and AST. When either model is used as a teacher to train the other model via knowledge distillation, the student model's performance improves significantly and outperforms the teacher.	[13]
10	Mel-Spectrogram	Vision Transformer (ViT) with Patch-level Feature Fusion	The spectrogram is fed into a ViT to generate a feature map to exploit the advantage of using a pre-trained network. The feature map is reconstructed into a number of patches and a transformer is then applied to learn the patch-level information.	[14]
11	Mel-Spectrogram	Audio Spectrogram vIsion Transformer (ASiT)	From a 10-second audio spectrogram, two random augmented views of 6-seconds each are generated and fed to Group Masked Model Learning (GMML)-based manipulation block to obtain corrupted spectrograms. The clean and corrupted spectrograms are fed to the teacher and student networks, respectively. The network is capable of recovering transformed information from the non-transformed class-token and data-token, indicating learning of local and global representation of the given audio along with useful	[15]

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

Sl. No.	Feature extraction method	Model	Architecture Description	Reference
			inductive bias by learning local statistical correlation in the spectrogram.	

4. Proposed Methodology

The proposed methodology follows a step-by-step approach. Initially, feature extraction from the audio samples is done, which play a central role in creating a representation of the audio suitable for effective classification. These features act as input and are fed to the proposed architecture. To improve the model's ability to generalize, the methodology incorporates techniques such as Early Stopping and Stratified K-Fold Cross Validation. These contribute to better model performance and ensure robustness in the classification process.

1.1. Feature Extraction

Frequency domain features such as STFT, Mel-Spectrograms, MFCCs are often found to comprehend the frequency domain behavior of a signal. STFT [16] is a technique used to visualize the frequency content of a signal over time. In comparison to Fourier transform, which offers a frequency-domain representation of an entire signal, STFT provides a time varying frequency representation of a signal by calculating the Fourier Transform of small, overlapping time windows of the signal.

$$STFT\{x[n]\}(m,\omega) \equiv X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]\, w[n-m] e^{-i\omega n} \quad (7)$$

where $w[n]$ is the window function, commonly a Gaussian window or Hann window centered around zero, and $x[n]$ is the signal to be transformed and the frequency ω .

Mel Spectrogram [17] is a modification upon STFT that uses a filter bank to transform the frequency domain of a signal from the linear scale to Mel Scale, which is an approximate scale that represents the way humans perceive sound. It effectively combines perceptually similar frequencies, to give a smaller number of frequency bands. The relation between the Mel-frequency (m) and the frequency (f) is defined as shown in eq. 8.

$$m = 2595\, log(1 + \frac{f}{700}) \quad (8)$$

MFCCs [18] [19] are a further modification upon Mel Spectrograms. They are a cluster of features that are obtained from the Mel Spectrogram, effectively being a more compact depiction of the spectral information of the signal. Hence, MFCCs are used in this work, as they are powerful and efficient at capturing the spectral characteristics of audio signals. The initial steps for obtaining MFCC include pre-emphasis, framing, windowing and Discrete

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

Fourier transform (DFT) computation. DFT is applied to extract information in the frequency domain as shown in eq. 9.

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{\frac{-j2\pi nk}{N}};\ 0 \le k \le N \qquad (9)$$

Fourier transformed signal is passed through a set of band-pass filters known as Mel-filter banks to compute Mel Spectrum. Filter banks are generally implemented in the frequency domain, to compute MFCC. The most frequently used filter shape is triangular. Mel filter bank outputs a power spectrum. But, humans' perception of sound is logarithmic in nature. Therefore, the log of the output of the power spectrum is taken. Discrete Cosine Transform (DCT) is applied to the transformed Mel frequency coefficients which produces a set of cepstral coefficients (eq. 10.).

$$c(n) = \sum_{m=0}^{M-1}\big(s(m)\big)\, cos\ \left(\frac{\pi n(m-0.5)}{M}\right) \qquad (10)$$

The plots of MFCC obtained for samples from each dataset considered in this work can be observed in the fig. 2. The first row displays the MFCCs of five random audio samples from the ESC-50 dataset. The second row displays the MFCCs of five random audio samples from the Speech Commands v0.2. Finally, the third row displays the MFCCs of five random audio samples from the UrbanSound8k dataset. In all cases, 30 MFCCs were considered from each audio sample to capture relevant acoustic features.

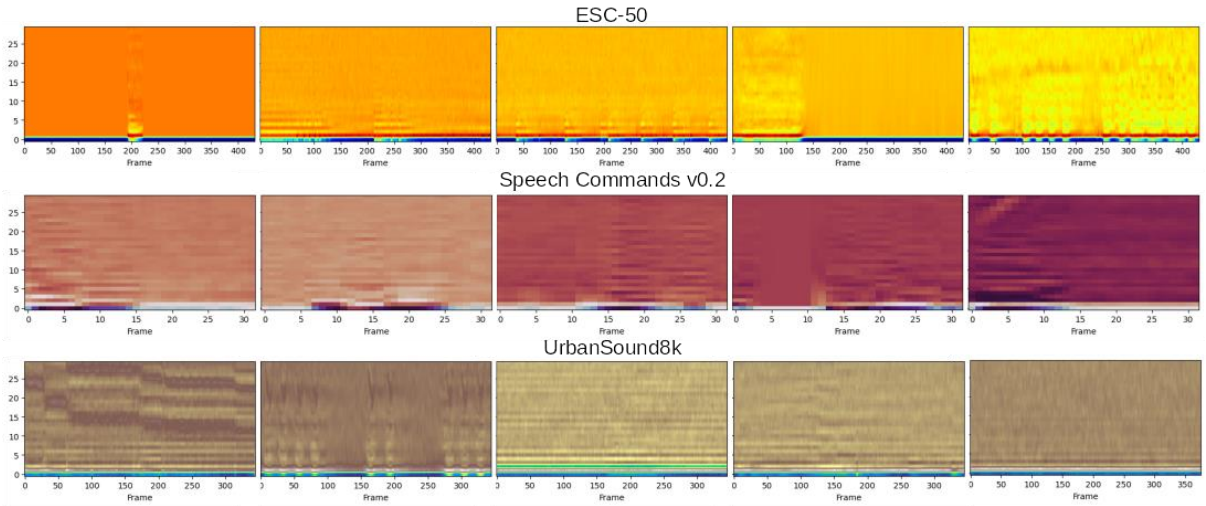

Fig. 2. MFCCs for audio samples from ESC-50, Speech Commands v0.2 and UrbanSound8k

1.2. Natural Language Processing: Sequence Classification

NLP is a subfield of artificial intelligence that intends to develop technology which can imitate human understanding, interpretation, and generation of language. Tasks in NLP can be

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

sequence-to-sequence, e.g. summarization, machine translation; sequence labeling, e.g. POS tagging, NER; sequence classification, e.g. sentiment analysis, language detection, topic labeling.

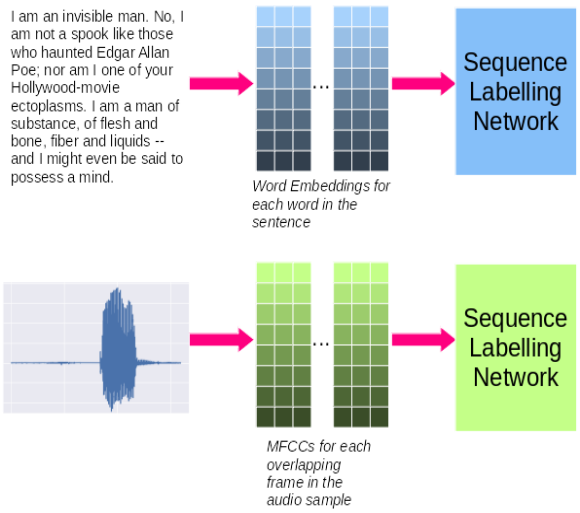

Fig. 3. Parallels between sequence labeling in NLP and Signal Processing

Sentiment analysis employs sequence classification task in NLP that seeks to decide the sentiment expressed in a piece of text. The procedure can be split into a number of stages. First, information is gathered from a variety of sources, including social media and customer reviews. Next, the data is pre-processed, which involves removing irrelevant parts of the corpus, such as punctuations, special characters, and numbers. The pre-processed data is then tokenized into words, which are used to create word embeddings, i.e. dense vectors used to denote words in the corpus. Using rule-based methods, lexicon-based methods, machine learning methods, and deep learning methods, the sentiment of the text is identified [20]. HuggingFace Transformers [21] provides APIs and tools to use state-of-the-art models for this task. Models such as DistilBERT [22], roBERTa [23], FinBERT [24], DeBERTa [25], etc. have been trained on datasets such as imdb [26], SST-2 [27], etc., for the sentiment analysis task.

A similar procedural architecture that has been modified to be more suitable for signal classification tasks can be employed to conduct audio classification. MFCCs for audio signals bear resemblance to the word embeddings for text corpora (fig. 3.). Hence, they can be utilized to classify audio signals.

1.3. Proposed Architecture

A Transformer-encoder [2] based model that uses MFCCs to classify audio is proposed as in fig. 4, taking inspiration from the methodology used for sequence classification tasks.

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

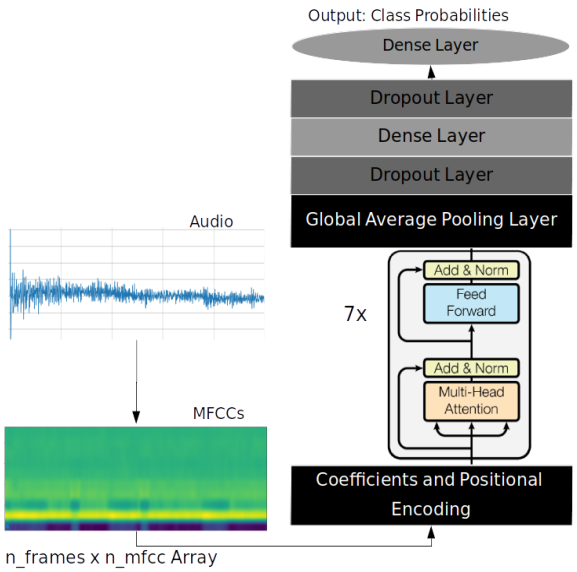

Fig. 4. Proposed Architecture

First, audio clips are used to produce the MFCCs. The Coefficients and Positional Embedding block is used to add positional/temporal information to the MFCC vectors. The sequences are passed through a series of 7 Transformer-Encoder blocks and the output from these blocks is now a sequence of encoded representations, one for each element in the input sequence. The encoders apply Multi-Head Attention and fully connected FeedForward layers for capturing complex relationships between different parts of the sequence.

At each layer, the output of the previous layer is fed as input to the next layer, resulting in a sequence of encoded representations that are progressively refined with each layer. Each encoded representation captures both local and global dependencies between the input elements and is generated by applying a series of operations to the input sequence. Transformers leverage attention in their encoder architecture to improve sequence labeling tasks like POS tagging and NER. The self-attention mechanism in transformers allows the model to dynamically weigh the importance of different elements in the input sequence, resulting in better contextual understanding and more accurate predictions. The encoded representation is then fed through a Global Average Pooling layer. This is followed by two dense layers, each separated by dropout layers. The output obtained is a vector with the class probabilities of the various classes.

4.4. Datasets

Datasets used in this work are ESC-50, Speech Commands v2 and UrbanSound8k. Inclusion of ESC-50 allows exploration of the performance of the methodology proposed, on a broad range of environmental sound categories. With incorporation of Speech Commands v2 in this

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

work, the research aims to address the challenges associated with real-world speech recognition scenarios. UrbanSound8k is used to investigate the effectiveness of the proposed methods in classifying urban sounds and understanding their implications in real-world applications. Details of the three datasets used in this study are elaborated below.

4.4.1. ESC-50

The ESC-50 dataset [28] is a collection of 2000 labeled environmental recordings. The dataset consists of 5-second recordings organized into 50 classes. The categories include animal sounds (such as birds, dogs, and cats), natural sounds (such as rain, thunder, and wind), and man-made sounds (such as vehicles, tools, and domestic sounds). The recordings of the audio were composed from various sources and have varying lengths, with a total duration of approximately five hours. Each audio clip is in WAV format and is sampled at 44.1 kHz with a bit depth of 16 bits.

4.4.2. Speech Commands v0.2

The Speech Commands v0.2 dataset [29] is a publicly available collection of audio recordings and associated transcriptions, designed for use in training and evaluating automatic speech recognition (ASR) systems. The dataset was released by Google in 2018 as an updated version of their original Speech Commands dataset. The Speech Commands v0.2 dataset contains over 100,000 one-second audio clips of spoken words, recorded by thousands of different people. The words include common commands like "yes", "no", "stop", "go", and "hello", names of people and places. The audio clips are in WAV format and have a sampling rate of 16 kHz and a bit depth of 16 bits.

4.4.3. UrbanSound8k

The UrbanSound8k dataset [30] consists of 8,732 annotated sound excerpts of length less than or equal to 4 seconds. It contains 10 low-level classes, i.e. car horn, air conditioner, gun shot, dog bark, children playing, engine idling, drilling, street music, siren, and jackhammer. The recordings of audio were obtained from various locations in urban areas, including parks, streets, and residential areas. Each audio clip is in WAV format and has a duration of 4 seconds, with a sampling rate of 44.1 kHz and a bit depth of 16 bits.

4.5. Model Training

The obtained MFCCs have a dimension of number of frames × number of MFCCs. A 4-head multihead attention block is utilized by the Transformer block. Each time, training of the model was carried out for 128 epochs with a sample size of 128. The Sparse Categorical Cross Entropy loss function and the Adam algorithm were both used. Stratified 10-Fold cross-validation was utilized to counteract the consequence of imbalanced datasets. Early Stopping is utilized to prevent overfitting during training by monitoring the training loss and stopping training when

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

the loss stops improving. The use of early stopping with Stratified K-Fold Cross Validation helps prevent overfitting and improves the generalization performance of the model. The model's performance is evaluated using precision, recall, f1-score and accuracy.

5. Results and Discussion

Experimentation and comparison was carried out on the three datasets mentioned earlier. The metrics considered for performance are precision, recall, F1 score and accuracy. Precision (eq. 11.) is the ratio of the correct positive predictions to the total number of positive predictions. Recall (eq. 12.) calculates the ratio of predicted positives to the total number of positive labels. F1 score (eq. 13.) depends on both the Recall and Precision, it is the harmonic mean of both the values. Accuracy (eq. 14.) is the ratio of correct predictions to the total number of predictions.

$$Precision = \frac{True\ Positives}{Total\ Number\ of\ Positive\ Predictions} \quad (11)$$

$$Recall\ = \frac{True\ Positives}{Total\ Number\ of\ Positive\ Labels} \quad (12)$$

$$F1\ Score = 2 * \frac{Recall * Precision}{Recall+Precision} \quad (13)$$

$$Accuracy = \frac{Correct\ Prediction}{Correct\ Prediction + Incorrect\ Prediction} \quad (14)$$

5.1. ESC-50

Each sample fed through the model had 216 frames and 30 extracted MFCCs across these frames, resulting in a total of 132,874 trainable and total parameters. Since the dataset contains a mere 2000 recordings, 5 folds were utilized making the training dataset size 1600 and testing dataset size 400. Table II. tabulates the precision, recall, F1-score and accuracy across the different folds in the Stratified K-Fold Cross-Validation while training the model on the ESC-50 dataset. As the classes are balanced, they yield similar weighted and macro statistics and are hence tabulated in a single column reporting the statistics. The model achieved an average accuracy, precision, recall, and F1-score of 75.2%, 77.2%, 75.2%, and 74.8%, respectively. Fig. 5 (a) helps visualize the metrics across the folds.

5.2. Speech Commands v0.2

Each sample fed through the model had 32 frames and 30 extracted MFCCs across these frames, giving a total of 125,940 trainable and total parameters. 10 folds were utilized for cross-validation. Table III. tabulates the precision, recall, F1-score and accuracy across the different folds in the Stratified K-Fold Cross-Validation while training the model on the Speech Commands v0.02 dataset. Due to the imbalance of the classes, the model yields different macro-statistics and weighted-statistics, which are reported as such. The average Macro-Precision, Macro-Recall, Macro-F1-Score obtained are 92.3%, 92.1% and 92.7%, respectively. The average Weighted-Precision, Weighted-Recall, Weighted F1-Score obtained

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

are all 92.6%. The average accuracy is 92.6%. Fig. 5 (b) helps visualize the metrics across the folds.

5.3. UrbanSound8k

Each sample fed through the model had 173 frames and 30 extracted MFCCs across these frames, giving a total of 127,544 trainable and total parameters. 10 folds were utilized for cross-validation. Table IV. tabulates the precision, recall, F1-score and accuracy across the different folds in the Stratified K-Fold Cross-Validation while training the model on the Speech Commands v0.02 dataset. Because of imbalance of the classes, the model yields different macro-statistics and weighted-statistics, which is reported as such. The average Macro-Precision, Macro-Recall, Macro-F1-Score obtained are 95.7%, 95.8%, 95.7%, respectively. The average Weighted-Precision, Weighted-Recall, Weighted F1-Score obtained are 95.3%, 95.2%, 95.2%, respectively. The average accuracy is 95.2%.

Fig. 5 presents the performance statistics of the trained models on three different datasets, i.e., ESC-50, Speech Commands v0.2 and UrbanSound8k. The figure consists of three subplots (a), (b), (c) corresponding to each dataset. In each subplot, the performance metrics are displayed. The x-axis represents the fold number indicating the different folds used during cross-validation or model evaluation. The y-axis represents the metric values, which vary depending on the specific statistic being measured. All the subplots show an improvement in the metrics with each subsequent fold.

Table V shows a comparison between the proposed methodology and the architectures surveyed for the literature. Since the proposed network is trained with any pre-training, an attempt was made to draw no-pre training results from the papers describing these architectures. The proposed model, which is not pre-trained and has a mere 128k parameters, demonstrates competitive performance across various datasets when compared to other models.

One notable aspect of the proposed model is its relatively small number of parameters, which stands at 128k. Despite its compact size, the model achieves remarkable results on the ESC-50, Speech Commands V2 and UrbanSound8k datasets. This highlights the efficiency and effectiveness of the model in terms of parameter utilization and computational resources needed for training and inference. Overall, the proposed model showcases several positives, including its compact size, competitive performance across datasets, and strong accuracy scores. These findings suggest that the model holds promise for multiple audio classification tasks, making it a valuable addition to the existing literature on audio-based deep learning models.

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

TABLE II. ESC-50: STATISTICS PER FOLD

Fold	Average Statistics			Accuracy
	Precision	*Recall*	*F1-score*	
1	.50	.47	.46	.47
2	.69	.67	.66	.67
3	.85	.83	.83	.83
4	.91	.89	.89	.89
5	.91	.90	.90	.90
Mean	.772	.752	.748	.752
Std. Dev.	.177	.182	.188	.182

TABLE III. SPEECH COMMANDS V0.2: STATISTICS PER FOLD

Fold	Macro-Average Statistics			Weighted-Average Statistics			Accuracy
	Precision	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*	
1	.90	.90	.90	.91	.90	.90	.90
2	.91	.90	.91	.91	.91	.91	.91
3	.92	.92	.92	.93	.93	.93	.93
4	.92	.91	.91	.92	.92	.92	.92
5	.93	.94	.93	.94	.94	.94	.94
6	.93	.93	.93	.93	.93	.93	.93
7	.93	.92	.92	.93	.93	.93	.93
8	.93	.93	.93	.93	.93	.93	.93
9	.93	.93	.93	.94	.94	.94	.94
10	.93	.93	.93	.93	.93	.93	.93
Mean	.923	.921	.921	.927	.926	.926	.926
Std. Dev.	.010	.014	.011	.010	.013	.013	.013

TABLE IV. URBANSOUND8K: STATISTICS PER FOLD

Fold	Macro-Average Statistics			Weighted-Average Statistics			Accuracy
	Precision	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*	
1	.90	.90	.90	.89	.89	.89	.89
2	.93	.93	.93	.92	.92	.92	.92
3	.97	.96	.96	.96	.96	.96	.96

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

Fold	Macro-Average Statistics			Weighted-Average Statistics			Accuracy
	Precision	*Recall*	*F1-score*	*Precision*	*Recall*	*F1-score*	
4	.97	.97	.97	.97	.97	.97	.97
5	.95	.95	.95	.94	.94	.94	.94
6	.95	.96	.95	.95	.95	.95	.95
7	.98	.98	.98	.98	.97	.97	.97
8	.97	.97	.97	.97	.97	.97	.97
9	.97	.98	.98	.97	.97	.97	.97
10	.98	.98	.98	.98	.98	.98	.98
Mean	.957	.958	.957	.953	.952	.952	.952
Std. Dev.	.025	.026	.026	.029	.028	.028	.028

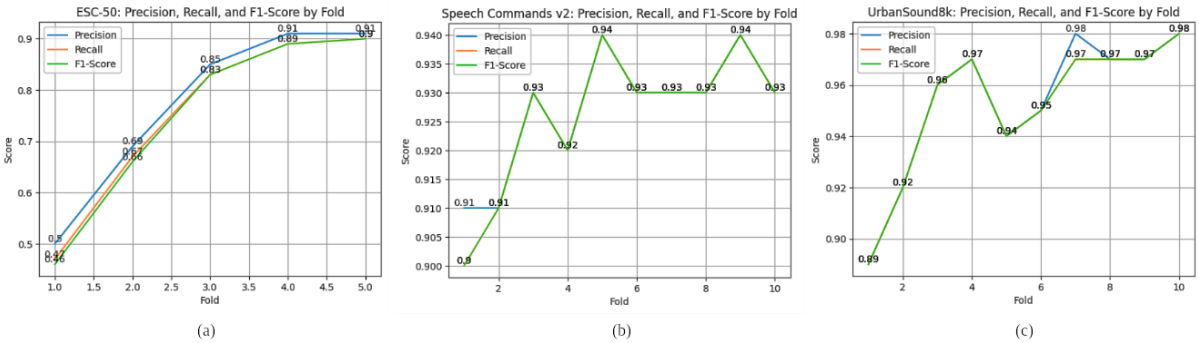

Fig. 5. Plots of performance metrics for the three datasets. (a) shows the metrics for the ESC-50 dataset, (b) shows the metrics for the Speech Commands v2 dataset and (c) shows the metrics for the UrbanSound8k dataset

TABLE V. COMPARISON WITH OTHER MODELS ON THE SAME DATASETS

Model	Pretrained (P) or not (N)	Number of Parameters	Datasets		
			ESC-50	*Speech Commands V2*	*UrbanSound8k*
AST [5]	N	86M	.887	.9811	-
HTS-AT [7]	P	31M	.97	.98	-
CAT [8]	N	Not mentioned	.894	-	.961
SSAST [9]	P	~86M	.888	.98	-

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

Model	Pretrained (P) or not (N)	Number of Parameters	Datasets		
			ESC-50	***Speech Commands V2***	***UrbanSound8k***
MAST [10]	P	Not mentioned	-	.931	.644
MAE-AST [11]	P	>86M	.863	.973	-
SepTr [12]	N	8.8M	.911	.985	-
ASiT [15] with ViT backbone	P	Not mentioned	.92	.988	-
Proposed work	N	128k	.772	.923	.957

6. Conclusion and Future Scope

This study has demonstrated the implementation of a new methodology for audio classification using the transformer encoder to create an encoded representation of the MFCCs obtained from the audio samples which is further used for classification, mimicking the sequence classification task in NLP. The network shows strong performance in audio classification tasks on various datasets, with particularly impressive results on the Speech Commands v0.02 (Accuracy- 92.6%) and UrbanSound8k (Accuracy- 95.2%) datasets, which have 125,940 and 127,544 trainable and total parameters respectively. The ESC-50 dataset, however, yielded lower statistics as in average precision, recall F1-score and accuracy in comparison to the other datasets. The low statistics on the ESC-50 dataset might be credited to the relatively low sample size, i.e. 2000 samples, in the dataset as compared to the other datasets. Hence the model may not be having sufficient exposure to the wide range of environmental sounds in the dataset. Despite the lower performance on the ESC-50 dataset, the overall results demonstrate the potential of the Transformer-encoder based network for audio classification tasks.

There is scope for future work so as to advance the proposed methodology and extend its capabilities. Firstly, a good step towards future research is to evaluate the proposed model on additional datasets. This would validate its effectiveness further and assess its generalizability across a wider range of audio classification tasks. Secondly, exploring the option of pretraining the model could lead to improvements in performance. Pretraining on a large scale audio dataset, such as AudioSet or a combination of multiple datasets, can enable the model to learn high-level representations and capture more intricate audio features. Furthermore, optimizing the proposed model's architecture is one more fruitful opportunity for future work. One potential modification could involve increasing the depth of the network by adding more layers.

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

Deeper architectures may enable the model to absorb more abstract and complex representations, potentially enhancing its discriminative power. But, it is important to carefully balance depth and model complexity to avoid overfitting or diminishing returns. Lastly, researchers can explore alternative methods for feature extraction to advance the quality of the input data.

References

[1] Daniel Jurafsky and James H. Martin. 2000. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition (1st. ed.). Prentice Hall PTR, USA.

[2] A. Vaswani et al., “Attention Is All You Need.” arXiv, 2017. doi: 10.48550/ARXIV.1706.03762.

[3] B. Vimal, M. Surya, Darshan, V. S. Sridhar and A. Ashok, "MFCC Based Audio Classification Using Machine Learning," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-4, doi: 10.1109/ICCCNT51525.2021.9579881.

[4] Muqing Deng, Tingting Meng, Jiuwen Cao, Shimin Wang, Jing Zhang, Huijie Fan, Heart sound classification based on improved MFCC features and convolutional recurrent neural networks, Neural Networks, Volume 130, 2020, Pages 22-32.

[5] Y. Gong, Y.-A. Chung, and J. Glass, “AST: Audio Spectrogram Transformer.” arXiv, 2021. doi: 10.48550/ARXIV.2104.01778.

[6] K. Koutini, J. Schlüter, H. Eghbal-zadeh, and G. Widmer, “Efficient Training of Audio Transformers with Patchout,” arXiv, 2021, doi: 10.48550/ARXIV.2110.05069.

[7] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, “HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection.” arXiv, 2022. doi: 10.48550/ARXIV.2202.00874.

[8] X. Liu, H. Lu, J. Yuan, and X. Li, “CAT: Causal Audio Transformer for Audio Classification.” arXiv, 2023. doi: 10.48550/ARXIV.2303.07626.

[9] Y. Gong, C.-I. J. Lai, Y.-A. Chung, and J. Glass, “SSAST: Self-Supervised Audio Spectrogram Transformer.” arXiv, 2021. doi: 10.48550/ARXIV.2110.09784.

[10] Liu, F.; Fang, J. Multi-Scale Audio Spectrogram Transformer for Classroom Teaching Interaction Recognition. Future Internet 2023, 15, 65. https://doi.org/10.3390/fi15020065.

[11] A. Baade, P. Peng, and D. Harwath, “MAE-AST: Masked Autoencoding Audio Spectrogram Transformer.” arXiv, 2022. doi: 10.48550/ARXIV.2203.16691.

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

[12] N.-C. Ristea, R. T. Ionescu, and F. S. Khan, “SepTr: Separable Transformer for Audio Spectrogram Processing.” arXiv, 2022. doi: 10.48550/ARXIV.2203.09581.

[13] Y. Gong, S. Khurana, A. Rouditchenko, and J. Glass, “CMKD: CNN/Transformer-Based Cross-Model Knowledge Distillation for Audio Classification.” arXiv, 2022. doi: 10.48550/ARXIV.2203.06760

[14] J. Luo, J. Yang, E. S. Chng, and X. Zhong, “Vision Transformer based Audio Classification using Patch-level Feature Fusion,” 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE, Nov. 07, 2022. doi: 10.23919/apsipaasc55919.2022.9980194.

[15] S. Atito, M. Awais, W. Wang, M. D. Plumbley, and J. Kittler, “ASiT: Audio Spectrogram vIsion Transformer for General Audio Representation.” arXiv, 2022. doi: 10.48550/ARXIV.2211.13189.

[16] M. Woelfel and J. McDonough, Distant Speech Recognition. Nashville, TN: John Wiley & Sons, 2009.

[17] D. O’Shaughnessy, Speech communication: Human and Machine. Universities press, 1987.

[18] K. S. Rao and M. K E, Speech Recognition Using Articulatory and Excitation Source Features. Springer International Publishing, 2017. doi: 10.1007/978-3-319-49220-9.

[19] J.R. Deller, J.H. Hansen, J.G. Proakis, Discrete Time Processing of Speech Signals (Prentice Hall, NJ, 1993).

[20] E. Rudkowsky, M. Haselmayer, M. Wastian, M. Jenny, Š. Emrich, and M. Sedlmair, “More than Bags of Words: Sentiment Analysis with Word Embeddings,” Communication Methods and Measures, vol. 12, no. 2–3. Informa UK Limited, pp. 140–157, Apr. 03, 2018. doi: 10.1080/19312458.2018.1455817.

[21] "Hugging Face Transformers Documentation," Hugging Face, 2021. [Online]. Available: https://huggingface.co/docs/transformers/index. [Accessed: June 10, 2023]

[22] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, “DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.” arXiv, 2019. doi: 10.48550/ARXIV.1910.01108.

[23] F. Barbieri, J. Camacho-Collados, L. Espinosa Anke, and L. Neves, “TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classification,” Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.findings-emnlp.148

Power System Technology
ISSN:1000-3673
Received: 16-10-2025 Revised: 05-11-2025 Accepted: 06-12-2025

[24] Huang, Allen H., Hui Wang, and Yi Yang. "FinBERT: A Large Language Model for Extracting Information from Financial Text." Contemporary Accounting Research (2022)

[25] P. He, X. Liu, J. Gao, and W. Chen, "DeBERTa: Decoding-Enhanced BERT with Disentangled Attention" in International Conference on Learning Representations, 2021. [Online]. Available: https://openreview.net/forum?id=XPZIaotutsD

[26] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning Word Vectors for Sentiment Analysis. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

[27] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

[28] K. J. Piczak, “ESC: Dataset for Environmental Sound Classification” Proceedings of the 23rd ACM international conference on Multimedia. ACM, Oct. 13, 2015. doi: 10.1145/2733373.2806390.

[29] P. Warden, “Speech Commands: A Dataset for Limited-Vocabulary Speech Recognition.” arXiv, 2018. doi: 10.48550/ARXIV.1804.03209.

[30] J. Salamon, C. Jacoby, and J. P. Bello, “A Dataset and Taxonomy for Urban Sound Research,” Proceedings of the 22nd ACM international conference on Multimedia. ACM, Nov. 03, 2014. doi: 10.1145/2647868.2655045.

[31] P. Li, J. Wu, Y. Wang, Q. Lan, and W. Xiao, “STM: Spectrogram Transformer Model for Underwater Acoustic Target Recognition,” Journal of Marine Science and Engineering, vol. 10, no. 10. MDPI AG, p. 1428, Oct. 04, 2022. doi: 10.3390/jmse10101428.

[32] S. Park, Y. Jeong, T. Lee, “Many-to-Many Audio Spectrogram Transformer: Transformer for Sound Event Localization and Detection” 6th Workshop on Detection and Classification of Acoustic Scenes and Events (DCASE 2021), Nov. 2021

[33] Y. Khasgiwala and J. Tailor, “Vision Transformer for Music Genre Classification using Mel-frequency Cepstrum Coefficient,” 2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON). IEEE, Sep. 24, 2021. doi: 10.1109/gucon50781.2021.9573.