

## Research

# Scanning dial: the instantaneous audio classification transformer

Huawei Jiang<sup>1</sup> · Husna Mutahira<sup>2</sup> · Unsang Park<sup>2</sup> · Mannan Saeed Muhammad<sup>3,4</sup>

Received: 14 October 2023 / Accepted: 12 February 2024

Published online: 27 February 2024

© The Author(s) 2024 [OPEN](#)

## Abstract

A number of remarkable accomplishments have been achieved in the field of audio classification using algorithms based on Transformers in recent years. As addressed in the literature, sound classification commonly involves the analysis of audio recordings that are usually five seconds or longer in duration. This raises a secondary question: Can Transformers effectively classify extremely short audio samples? The main objective of this study is to utilize the Transformer model for sound classification, focusing on extremely brief audio clips, with an average sound duration of  $1.24 \times 10^{-2}$  seconds, which is too short for human recognition. In addition, a new filter is developed to obtain an instantaneous audio dataset. This filter is applied individually to the ESC-50, UrbanSound8K, AESDD, ReaLISED and RAVDESS datasets to obtain corresponding instantaneous datasets. Moreover, a new data augmentation technique is introduced with the objective of increasing classification accuracy. A comparative analysis between the proposed scheme and the mainstream data augmentation methods is performed on the instantaneous audio datasets, resulting in accuracy rates of 94.16%, 96.40%, 70.98%, 89.28%, and 53.51%, respectively. This study has the main advantage of being able to classify sounds efficiently for extremely short audio duration.

## Article highlights

- The proposed method filters out the muted part from the audio files, resulting in a new dataset called as instantaneous audio dataset.
- A novel data augmentation technique and an adapted Transformer model are proposed for the effective classification of short audio.
- This algorithm can effectively classify sounds within extremely short audio durations, which makes it useful for situations requiring swift decision-making.

**Keywords** Data augmentation · Instantaneous audio classification · Instantaneous audio dataset · Transformer

---

✉ Unsang Park, [unsangpark@sogang.ac.kr](mailto:unsangpark@sogang.ac.kr); ✉ Mannan Saeed Muhammad, [mannan@skku.edu](mailto:mannan@skku.edu) | <sup>1</sup>Department of Computer Science and Engineering, College of Computing and Informatics, Sungkyunkwan University, Natural Sciences Campus, Suwon 16419, South Korea. <sup>2</sup>Department of Computer Science and Engineering, Sogang University, Seoul 04107, South Korea. <sup>3</sup>Department of Electrical and Computer Engineering, College of Information and Communication Engineering, Sungkyunkwan University, Natural Science Campus, Suwon 16419, South Korea. <sup>4</sup>Department of AI and Robotics, Sejong University, Seoul 05006, South Korea.



## 1 Introduction

In the era of vacuum tubes, radios were equipped with luminous dials for tuning frequencies and locating stations, complemented by a scanning dial designed for exploring different frequency options. Today's digital tuning retains the term's meaning of seeking content across stations. For radio listeners, the speedy identification of their preferred content closely resembles the quick channel selection that humans effortlessly perform within milliseconds through auditory perception [10]. The manuscript is motivated by the question of whether deep learning algorithms can similarly classify sounds accurately in durations shorter than human recognition.

Over the past years, convolutional neural networks (CNN) have achieved a significant impact on several image and audio processing tasks [23, 27, 30]. In recent times, there has been a growing focus on the Transformer model as a viable alternative to CNNs [11, 14, 21]. The Transformer is a type of neural network based on a self-attention mechanism [14]. Initially developed for natural language processing tasks, the Transformer has seen broader applications in recent years, including image and audio processing [13, 35]. There are two main approaches to using the Transformer for sound classification i.e., the pure Transformer approach [11, 18] or, the combination of Transformer and CNN [7]. Transformer-based algorithms outperform CNN and achieve state-of-the-art (SOTA) performance in the field of audio classification [4, 11, 12]. Additionally, pre-training models are employed to facilitate transfer learning in this context [11, 19].

Classifying very short segments of audio data holds significance across a wide range of applications due to the prevalence of brief sounds in our surroundings, often lasting only milliseconds. These applications include gunshot detection for public safety [3], rapid identification of bird calls in ornithology [31], swift and effective monitoring of coughs for health purposes [1], quick identification of faults in industrial machinery, and the detection of keywords in audio for security surveillance [24]. In essence, it is crucial to preserve as much feature information as possible in the acquired audio data.

For sound classification using the Transformer, an audio spectrogram is generated through Short-Time-Fourier-Transformation from an audio file. The spectrogram is split into a sequence of patches, which are then sequentially fed into a linear projection layer. Each patch-embedding is augmented with a position-embedding, and a class token is introduced for the classification task. Then, the sum of each patch-embedding and position-embedding is passed through the Transformer encoder, and a classifier is applied to the first vector of output, which serves as the final result.

While the above method provides a systematic approach to sound classification, the effective training of the model, especially with extremely short audio files, poses distinct challenges. Data augmentation is a crucial step in the efficient classification of extremely short audio files, especially when the available data is limited. In such scenarios, the augmentation process takes place prior to segmenting the spectrogram into patches. The primary objective is to enrich data diversity, given that shorter audio files often contain limited feature information. However, working with short audio files presents unique challenges due to their constrained feature content, necessitating a cautious approach to data augmentation. Traditional large-scale data augmentation techniques are unsuitable in this context because they can significantly alter the original audio file characteristics, resulting in the loss of essential information crucial for short audio files. To address this issue, a data augmentation method named Squeezing-Toothpaste-Time-Shift (STTS) is proposed. The STTS technique optimizes the process of data augmentation while preserving the integrity of the original features.

This study is focused on the audio classification at extremely short intervals of time using a Transformer referred to as the instantaneous Audio Classification Transformer, *i*-ACT. Additionally, a new dataset is generated with a very short segmentation time measured in decibels relative to full scale (dBFS). It is important to note that there are no voiceless (silent) segments within any of the audio files in this new dataset. The experiments conducted have shown that audio files can be classified swiftly and with a high degree of accuracy within exceptionally brief time intervals. The proposed scheme excels in localizing sound sources at the millisecond scale, distinguishing it from earlier methods, by Senocak et al. [33, 34] and Arandjelovic et al. [2], that primarily focused on classifying audio files at the second scale.

The structure of this manuscript is as follows: Sect. 1 gives the motivation of this manuscript and an overview of how the transformer is applied to audio classification, data augmentation methods, and the significance of this study. With the limited existing research on classifying short audio files, addressing audio classification at the millisecond scale presents even bigger challenges. As a result, Sect. 2 exclusively presents the related work of data augmentation. Section 3 provides details about datasets used in this study and how to obtain the instantaneous audio datasets. Section 4 displays the attributes of newly generated instantaneous datasets. Section 5 discusses the proposed data augmentation method and the adapted Transformer. A description of the experimental setup is described in Sect. 6, while a discussion of the results can be found in Sect. 7. Finally, Sect. 8 presents the conclusion and future research.

## 2 Related work

In the field of audio processing and analysis, two common ways to visually represent audio signals are through waveforms and spectrograms. These representations serve as the basis for various data augmentation techniques in tasks related to sound processing. While working with sound waveforms, techniques such as adding noise [43], applying time stretch [44], and pitch shifting [42] are frequently employed to diversify the training data. However, as this article predominantly focuses on the classification of spectrograms, our subsequent emphasis shifts to exploring data augmentation techniques specific to the spectrogram domain. A variety of methods have been developed for enhancing the diversity of audio data within spectrograms. One notable technique is known as SpecAugment [26], which directly augments the spectrogram data. SpecAugment comprises three primary operations applied to the spectrogram: time warping, frequency masking, and time masking. Another approach, named SpecMix [16], is designed specifically for manipulating time-frequency domain features. This augmentation method involves blending two distinct data samples using time-frequency masks that effectively preserve spectral correlation within each audio sample. A different strategy, MixUp [38], revolves around blending two sounds from different classes in a randomized ratio, resulting in a sound that lies between classes. Additionally, the VH-MixUp technique involves augmenting data through a nonlinear blending process applied to multiple images [36].

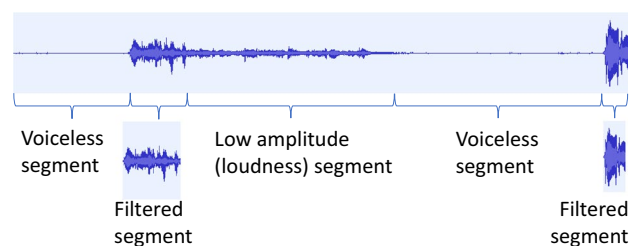
However, it is important to acknowledge that the aforementioned methods have the potential to partially diminish the feature information inherent in audio files. To address this concern and mitigate the loss of critical feature information, the STTS is introduced. This approach aims not only to enhance data diversity and reduce overfitting but also to strengthen the generalization capabilities of neural network-based algorithms. The STTS method achieves this by selectively shifting a segment of the audio spectrogram's feature information along the temporal dimension, thereby striking a balance between augmentation and preservation.

## 3 Instantaneous audio datasets

Most publicly available audio datasets consist of audios longer than 5sec, which may or may not contain silent parts [22, 25, 28, 32, 40, 41]. This research navigates through the details of audio classification within remarkably brief time intervals, utilizing a Transformer model- the Instantaneous Audio Classification Transformer (*i*-ACT). Unlike previous methodologies, that classify sounds within a minimum duration of 5sec or 10sec [9, 18], *i*-ACT specializes in performing instantaneous audio classification in extremely short intervals, to be precise, a minimum 0.025sec.

However, to effectively utilize *i*-ACT for instantaneous audio classification, thorough preprocessing of existing datasets is requisite. Simply segmenting existing datasets into instantaneous portions proves insufficient; it is important to excise silent (unvoiced) sections from the audio files. To create the updated dataset, the dBFS [29] is employed to filter out non-silent portions from the audio data used in the experimental datasets. The dBFS is a unit of measurement for amplitude levels in digital systems and provides a standard value [20]. It determines the absence of vocal activity in an audio segment by comparing its sample value with a baseline value. The silent segments within the dataset are identified by analyzing the amplitude of the audio files using the *Pydub* library of Python.

**Fig. 1** Waveform of an audio file. An audio segment is divided into voiceless segment, low amplitude segment and filtered segment, respectively



Given that the waveform of an audio file consists of multiple periodic signals, calculating the average value over a time period involves taking the Root Mean Square (RMS) of the sample values in the signal. This process identifies non-silent audio segments by comparing the average value of the periodic function with the sample value over a given time or amplitude, as illustrated in Fig. 1. As shown in the figures two filtered segments obtained by dBFS discard the unvoiced and lower amplitude segments. The RMS of the audio file is calculated as follows:

$$X_{RMS} = \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}, \tag{1}$$

where  $x_1, x_2, x_3, \dots, x_n$  are the sample values and  $x_n \in [-1, 1]$  and  $n$  represents the total number of samples of an audio file sampled at 44.1 kHz. After calculating the RMS, a threshold value  $T$  is defined as follows:

$$T = 20 \cdot \log_{10} \left( X_{RMS} \cdot \sqrt{2} \right), \tag{2}$$

such that  $T = 0$  for a full-scale sine wave. Since the maximum of  $X_{RMS}$  is  $\frac{1}{\sqrt{2}}$ , therefore multiplying  $X_{RMS}$  by  $\sqrt{2}$  ensures that the argument of  $\log_{10}$  in (2) returns 0 for the full-scale sine wave. To differentiate between silent and non-silent segments, the audio signal is divided into voiceless, low-amplitude, and high-amplitude segments. Therefore, the sample values are modified to get new values,  $l_i$ , by the following equation:

$$l_i = 20 \cdot \log_{10} (|x_i|), \tag{3}$$

where  $1 \leq i \leq n$ . The audio file segment is recognized as the part to be discarded when the stream of modified sample values  $l_i$  remains less than the threshold  $T$  for a minimum of 50msec. In this way, several audio segments of higher amplitude (loudness) can be obtained from one audio file.

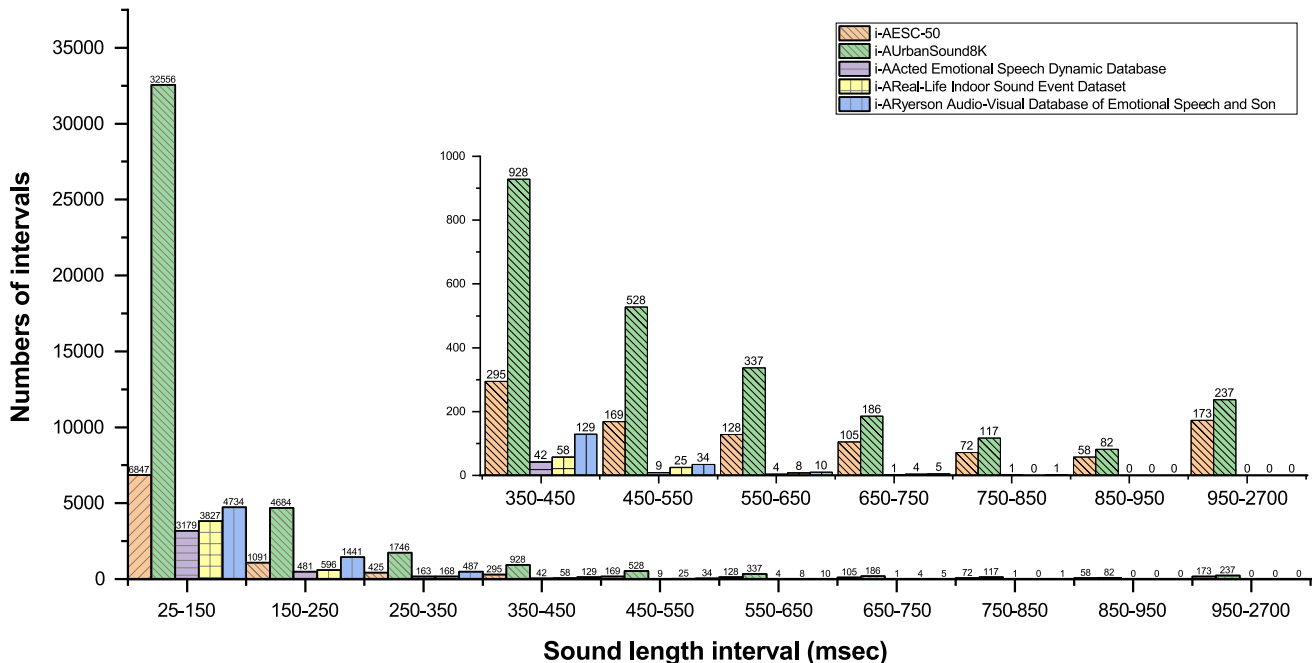


Fig. 2 *i-AESC-50, i-AUrbanSound8K, i-AAESDD, i-ARealISED and i-ARAVDESS* interval histogram of different audio lengths at the millisecond level. A smaller subfigure is the zoomed-in range 350 to 2700 of the larger subfigure

Contrary to configuration presented by Iqbal et al. in [15], which establishes the threshold in terms of a fixed value of  $-48$ , this study determines the threshold in terms of the RMS of each audio file in order to filter out silent parts. The exact silent parts of all audio files cannot, however, be detected. The distribution duration of the audio datasets used in this study is shown in Fig. 2. The horizontal axis is the audio length interval, and the vertical axis is the number of intervals. The length of each interval is marked on the histogram. Newly generated datasets through dBFS-based filter and original datasets are homogeneous.

## 4 Generated datasets for training and evaluation

The experiment utilized five datasets: ESC-50 [28], UrbanSound8K [32], Real-Life Indoor Sound Event Dataset (ReaLISED) [25], Ryerson Audio-Visual Database of Emotional Speech & Song (RAVDESS) [22], and Acted Emotional Speech Dynamic database (AESDD) [40, 41]. The proposed filtering method was applied to generate instantaneous audio datasets for each.

The values for category count, file count, training set size, testing set size, and average duration for each generated dataset are provided in Table 1. The average duration of all generated audio files combined ( $i$ -AESC-50  $\sim$   $i$ -ARAVDESS) is  $1.24 \times 10^{-2}$ seconds.

## 5 Data augmentation and transformer architecture

This section provides a comprehensive explanation of the proposed data augmentation method and the Transformer architecture in detail.

### 5.1 Proposed data augmentation method

Following the procurement of the instantaneous audio datasets, the ensuing procedural step encompasses the execution of data augmentation. This phase is necessary for enhancing the robustness and generalization capabilities of the model, especially in diverse and potentially challenging acoustic environments. This manuscript introduces an innovative and effective data augmentation approach known as Squeezing-Toothpaste-Time-Shift (STTS). The proposed technique aims to elevate the model's performance by expanding its ability to generalize across a spectrogram of acoustic scenarios for the instantaneous audio datasets.

The STTS method leverages the understanding that the frequency content of an audio signal can be visually represented as a color image through a spectrogram. A spectrogram serves as a multidimensional representation, essentially a matrix where rows  $r$  correspond to *time*, columns  $c$  correspond to *frequency*, and the intensity is depicted by numerical values. This graphical representation illustrates the relationship of sound elements over time and frequency, similar to a snapshot of the audio landscape.

**Table 1** The attributes of newly generated instantaneous datasets (for extremely short audio) through the proposed filter

Dataset Name	Category Count	File Count	Training Set Size	Testing Set Size	Average Duration
<i>i</i> -AESC-50	50	9363	7828	1535	158 msec
<i>i</i> -UrbanSound8K	10	41,401	31,538	9863	121 msec
<i>i</i> -AAESDD	5	3880	3232	648	98 msec
<i>i</i> -AReaLISED	18	4686	3909	777	99 msec
<i>i</i> -ARAVDESS	8	6281	5704	1137	128 msec

**Algorithm 1** STTS Method

---

**Algorithm 1** STTS Method

---

**Require:** Random value  $-10 \leq R \leq 10$   
**Ensure:**  $R \in \mathbb{Z}$   
**Input:** Spectrogram matrix  $V_{r,c}; r \in [1, 128]; c \in [1, 70]$   
**Output:** Updated Spectrogram matrix  $V_{r,c}^*$

```

while  $R \neq 0$  do
  if  $R > 0$  then
     $V_{r,c=71-k}^* \leftarrow V_{r,c=k}$ 
     $V_{r,c=d-R}^* \leftarrow V_{r,c=d}$ 
  else if  $R < 0$  then
     $V_{r,c=k}^* \leftarrow V_{r,c=71-k}$ 
     $V_{r,c=d}^* \leftarrow V_{r,c=d-R}$ 
  end if
end while
    
```

*/\*  $k \in [1, |R|]; d \in (|R|, 70]; k, d \in \mathbb{Z}^*$  \*/*

---

The STTS process is a dynamic operation involving the manipulation of the spectrogram columns. Specifically, a “squeezing” action takes place on a designated number of leftmost data columns, determined by the absolute value of a random integer value  $R$  within the range  $[-10, 10]$ . This value is assigned to the spectrogram matrix  $V_{r,c}$ , consequently generating a modified spectrogram matrix, denoted as  $V_{r,c}^*$ . Following this, the affected columns are transposed to the rightmost side of the spectrogram. This action necessitates a simultaneous shift of the existing data on the right side to accommodate the transposed columns. The direction of this shift depends on the sign of  $R$ : a positive  $R$  value indicates a squeeze and shift to the right, whereas a negative  $R$  implies a leftward operation. If  $R$  equals zero, the original spectrogram remains unaltered.

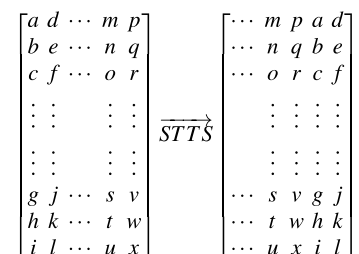
In the resultant spectrogram matrix  $V_{r,c}^*$ , the pixel values undergo updates using the variables  $k$  and  $d$  as indices. This updating process is designed to ensure that no data is lost or omitted during the transition. By maintaining the integrity of the original information while introducing variations through the squeezing and shifting operations, the STTS method successfully injects controlled variations into the spectrogram representation. This controlled augmentation strategy is fundamental to the robustness and adaptability of the model, allowing it to effectively learn and generalize across diverse acoustic scenarios without compromising the quality of the underlying audio data. Algorithm 1. gives the concrete step of the STTS method.

To facilitate a comprehensive understanding of the proposed STTS method, Fig. 3 illustrates its application to the base input matrix. The figure aids in explaining the dynamic transformations that occur during the STTS process. In Fig. 3, a random value of 2 is chosen as an example, influencing the behavior of the STTS method. This specific value results in the squeezing of the leftmost two columns of the base input matrix, which are then transposed to the right side of the spectrogram. Simultaneously, the data that was originally on the right side shifts to make space for these newly transposed columns, as depicted in Fig. 3.

It is important to note that the central portion of the matrix remains unaffected by the squeezing operation. The data in the center merely shifts to the left or right, ensuring that no information is lost during this process. This careful design ensures the accuracy of the audio data is maintained, keeping all the detailed features in the spectrogram intact.

For consistency and uniformity, the size of all audio spectrogram matrices in this context is maintained at  $128 \times 70$ . This standardization underscores the applicability and scalability of the STTS method across diverse instantaneous audio datasets, promoting its versatility as a data augmentation technique.

**Fig. 3** Squeezing-Toothpaste-Time-Shift (STTS) applied to the base input matrix



## 5.2 Transformer architecture

Figure 4 illustrates the *i*-ACT architecture with the Squeezing-Toothpaste-Time-Shift (STTS) method. Initially, the input audio waveform, with a duration of  $\Delta T$  seconds, is transformed into an audio spectrogram using a 15msec Hamming window computed at intervals of 6msec. Following the approach outlined in [11, 12, 18], the frequency dimension is consistently maintained at 128. To ensure uniformity before feeding the spectrograms into the Transformer, a fixed value for the time dimension of the spectrogram needs to be provided. Through a series of experiments, it was determined that the optimal Transformer performance is achieved when the time dimension is set to 70 for such brief audio files i.e., an average duration of  $1.24^{-2}$ second. Consequently, most spectrogram matrices will have a shorter time dimension. If the time dimension of the generated spectrogram is less than 70, the lower section of the spectrogram matrix is filled with zeros. Conversely, if it exceeds 70, the segment beyond 70 will be omitted.

The spectrogram undergoes a patch division process, as illustrated in Fig. 4, wherein it is segmented into smaller units, each measuring  $16 \times 16$ . In contrast to traditional CNN, which processes entire images simultaneously, ViT [6] operates by dividing images into smaller patches. This patch-based processing, inspired by the Transformer, enhances computational efficiency and facilitates parallelization.

Following Gong et al.'s insights [11], the overlapping patches is employed, characterized by a six-unit overlap in both the time and frequency dimensions. This strategy is employed to improve the accuracy of classification. Subsequently, these patches undergo sequential processing through a linear projection layer, leading to the generation of patch embeddings. In order to retain positional information, position embeddings are introduced. These embeddings signify the position of each patch embedding within the overall sequence and are incorporated into the patch embedding sequence using linear addition. This approach ensures the preservation of crucial positional details during the transformation process.

Both the patch embeddings and position embeddings have a size of 768 elements, with dimensions of  $16_{height} \times 16_{width} \times 3_{item}$ . The  $3_{items}$  refer to the three components required by the Transformer encoder: Query, Key, and Value [6]. These components are essential for the attention mechanism of the Transformer, as they enable it to analyze the relationships and relevance of each sequence element in the audio signal.

Class tokens, initialized tokens without inherent information, are added to the beginning of the patch embedding sequence, following the approach by Devlin et al. in [5]. While the class token itself lacks specific information, it aggregates information from other tokens in the sequence. Subsequently, the sum of each patch embedding and its corresponding position embedding is fed into the Transformer. The Transformer architecture comprises two parts: an encoder and a decoder. In this study, only the encoder is utilized for classification tasks, as it analyzes the relationships between input patches. The architecture of the Transformer encoder aligns with the design presented by Dosovitskiy et al. in [6]. The Transformer encoder's output is then directed to a *softmax* classifier, which assigns confidence scores to various sound events for classification purposes.

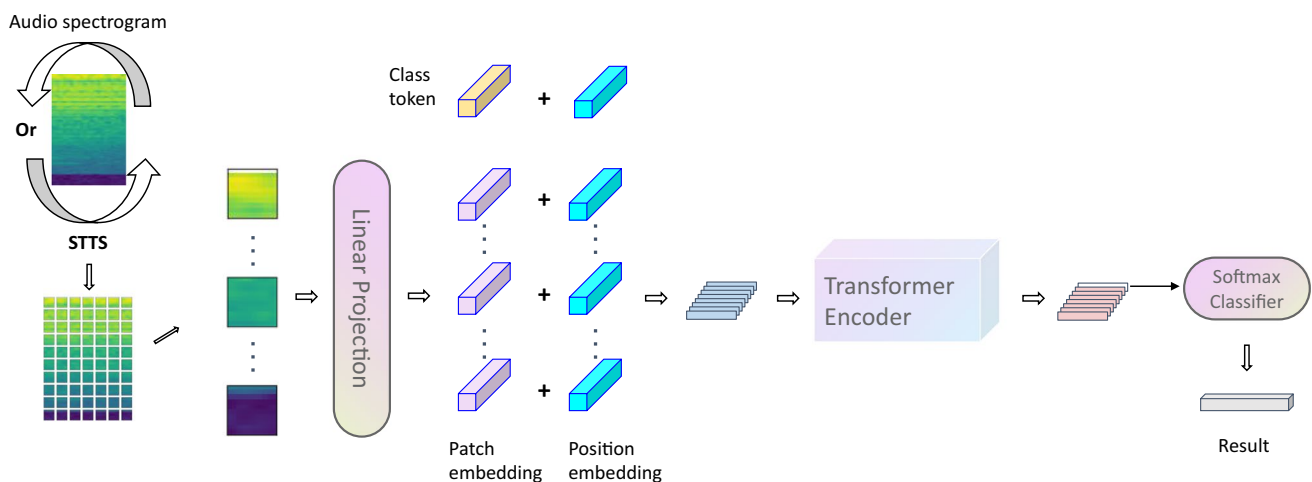


Fig. 4 The Instantaneous Audio Classification Transformer (*i*-ACT) architecture with the Squeezing-Toothpaste-Time-Shift (STTS) method

## 6 Experimental setup

For the experiments, PyTorch library is used and the data is trained on a GeForce GTX 3060 GPU with 12GB memory. In this study, the datasets for training and testing were divided in a 5:1 ratio using stratified sampling, based on individual sound event categories, such that 83.34% dataset is used for training, and 16.66% is used for testing. The *i*-UrbanSound8K dataset was divided in a 3:1 ratio to verify performance consistency across different ratios.

The scarcity of files in the training set was a notable constraint, leading us to optimize the scheduling of datasets for model training. Therefore, the validation set is not employed in this analysis [8, 11, 18]. The focus of this study was on including as many datasets as possible while ensuring that only the essential training and testing sets were allocated for the effective evaluation of model performance.

To promote convergence and minimize the impact of varying scales within the dataset, the input audio spectrogram was normalized, resulting in a dataset with 0 mean and a 0.5 standard deviation. Given the brief audio duration, the dataset is treated as a single-label dataset, with only one event considered within a unit interval. Performance assessment relies on the *accuracy* metric for single-label classification, while for multi-label classifications, mean average precision (mAP) is utilized.

The Adam optimizer [17] is used for training, with an initial learning rate of  $10^{-5}$ . The learning rate is subsequently decreased by a factor of 0.5 for every 10 epochs. To prevent underfitting, the model is trained for 100 epochs and each experiment is repeated at least five times. Accuracy results reported in all cases are the averages obtained at the 100<sup>th</sup> epoch.

## 7 Results and discussion

This section describes the results and performance comparison of STTS under varied settings. The performance of *i*-ACT, examined across multiple datasets, is comparatively analyzed in the context of the pre-training model and diverse data augmentation methods. Moreover, the *i*-ACT is compared with preceding methodologies, such as AST [11] and PSLA [12]. AST is purely a transformer-based algorithm, while PSLA is an ensemble of model training techniques that have been proven to significantly enhance model accuracy. The core architecture underpinning PSLA is a convolutional neural network called as EfficientNet [37]. The decision to contrast PSLA and AST is strategically made to underscore the advantages of *i*-ACT, which employs a transformer architecture with the objective of accurately classifying extremely short audio files. The standard deviation for all results obtained from the subsequent experiments remains less than 0.4.

### 7.1 Performance analysis of PSLA & AST using the ESC-50 & *i*-AESC-50 datasets

This section evaluates the performance of AST and PSLA and highlights the crucial functionality of *i*-ACT in the classification of extremely short audio clips within the *i*-AESC-50 dataset, leveraging optimal configurations initially devised for the ESC-50. According to preceding studies [11, 12], AST has demonstrated impressive outcomes, notably when utilizing pre-training models derived from ImageNet and AudioSet, and implementing masking techniques across both time and frequency domains. It is critical to acknowledge that, while AST achieves an accuracy of 90.92% when deployed on the ESC-50 dataset, its accuracy perceptibly recedes to 89.20% when applied to the more demanding *i*-AESC-50 dataset, as indicated in Table 2.

In contrast, PSLA used ImageNet as a pre-training model and registers an accuracy of 82.75% on the ESC-50 dataset but experiences a decrease to 78.89% when applied to the *i*-AESC-50 dataset, as shown in Table 2. This table demonstrates that both AST and PSLA methods encounter a dip in classification accuracy when interfacing with the *i*-AESC-50

**Table 2** Performance comparison of PSLA and AST from the ESC-50 to the *i*-AESC-50

Methods+Data augmentation	ESC-50	<i>i</i> -AESC-50
AST + ImageNet + AudioSet + Masking	90.92%	↓ 89.20%
PSLA + ImageNet + Masking	82.75%	↓ 78.89%



**Table 3** Classification accuracy of Squeezing-Toothpaste-Time-Shift method, frequency shift and non-random time shift

<i>i</i> -ACT	Range of Random Value	Accuracy
		75.01%
Time-Shift	<b>(− 10, 10)</b>	<b>↑ 84.34%</b>
Time-Shift	(− 5, 5)	↑ 83.10%
Time-Shift	(− 10, 15)	↑ 83.23%
Frequency-Shift	(− 10, 10)	↓ 71.58%
Frequency-Shift	(− 20, 20)	↓ 71.37%
Time-Shift	(− 10, 10) $\otimes$	↓↓ 25.27%

Legend: ↑ indicates that the accuracy has been improved by the data augmentation, and ↓ represents decline in the accuracy. A sharp decline is represented by ↓↓, and

$\otimes$

indicates that the squeezing value is set according to the rule

**Table 4** Classification accuracy of pretraining models with *i*-ACT

No pre-training model	ImageNet	ImageNet and AudioSet
75.01%	85.59%	85.63%

dataset, thereby highlighting the complex nature of classifying extremely short audio files within this dataset, relative to the ESC-50 dataset.

## 7.2 Performance analysis of STTS across various configurations

The Squeezing-Toothpaste-Time-Shift (STTS) method is investigated under diverse configurations, with accuracy results being evaluated across varied scenarios: different ranges of the *random value R*, *frequency-shift*, and *non-random time-shift*. Three specific ranges of *R* are considered for *R*: [−10, 10], [−5, 5], and [−10, 15]. In the case of *non-random time-shift*, the *R* is fixed and correspondingly allocated according to the sound event classes for instance, the *i*-AESC-50 comprises 50 types of sound events, each assigned a fixed value corresponding to its category. The experimental findings reveal that the highest accuracy is achieved when *R* resides within the range of [−10, 10], as detailed in Table 3. As mentioned previously, all audio spectrogram matrices maintain a size of  $128 \times 70$ . The optimal time-shift range is determined to be  $0.1429 \times$  the number of columns in the spectrogram matrix for achieving the highest classification accuracy.

Despite the fact that both frequency and time can be squeezed along their respective axes, the performance of STTS exhibits a declination when the frequency is manipulated. This could potentially be ascribed to the frequency diversity inherent *i*-AESC-50, suggesting that random frequency shifts may lead to overfitting. Conversely, *non-random time-shift* squeezing, based on sound event categories, is assumed to enhance model performance due to fixed labels. However, the results contradict this assumption, indicating that the model benefits from greater randomness to increase its robustness.

## 7.3 Performance of *i*-ACT under pre-training model

This section presents a comparative assessment of *i*-ACT, exploring its performance against previously proposed pre-training models. In the study by Gong et al. [11], the incorporation of AudioSet as a pre-training model, used in conjunction with ImageNet, augmented the system's performance by 7% compared to utilizing only ImageNet as a pre-training model with the original ESC-50 dataset. It is important to note that both ImageNet alone and in combination with AudioSet exhibit equivalent efficacy as pre-training models for the *i*-AESC-50, which was derived from the original ESC-50, as evidenced in Table 4.

## 7.4 Performance analysis of *i*-ACT with data augmentation methods on *i*-AESC-50

Table 5 presents the accuracy outcomes of *i*-ACT classification when combined with the ImageNet pretraining model and various data augmentation methods, including MixUp, Masking, and the proposed STTS. To conduct the ablation

study, *i*-ACT is compared with the original AST and PSLA, i.e., without the use of any pretraining model or data augmentation technique. A notable enhancement in performance on *i*-AESC-50 was observed when the pretrained ImageNet model was deployed [11, 39]. Consequently, the performance of *i*-ACT is compared to that of the ImageNet pretrained *i*-ACT. Furthermore, a comparative analysis of time/frequency masking [26] and MixUp [38] is carried out, as both have demonstrated improved performance in audio classification tasks [11, 18], in conjunction with STTS. The experimental findings highlighted that time/frequency masking of 25 and a mix-ratio of 0.4, respectively, resulted in optimal performance. Based on the results presented in Table 5, it is evident that STTS has higher accuracy compared to the other data augmentation methods.

The learning curve of STTS, when compared with MixUp and Masking on *i*-AESC-50, is illustrated in Fig. 6. This figure shows accuracy values on the horizontal axis and epochs on the vertical axis. Examination of the learning curve indicates that STTS achieves better generalization for the *i*-ACT classifier.

To elucidate the classification capability of the *i*-ACT with data augmentation methods within the various categories of the *i*-AESC-50, refer to the confusion matrix illustrated in Fig. 5.

Additionally, Fig. 7 illustrates the classification accuracy of the *i*-ACT with different data augmentation methods as well as their combinations based on various sound events. The outcomes of *i*-ACT are uniformly colored to distinguish the effects of the proposed method from those of ImageNet and other data augmentation methods. The proposed method shows improved results than the previous methods.

### 7.5 Performance analysis of STTS on the original ESC-50

STTS was subjected to the conventional ESC-50 dataset to evaluate its efficacy in handling standard audio data. Nonetheless, the results, displayed in Table 6, reveal that neither the self-attention-based transformers nor the CNN-based PSLA exhibited a substantial performance enhancement when STTS was implemented. Contrarily, the utilization of STTS somewhat diminished the performance of these models. The performance demonstrated a marginal enhancement when utilizing Masking. Complementing the insights provided in Table 5, the findings from the same table affirm that STTS is effective when managing extremely short audio segments. Conversely, it does not produce satisfactory results when applied to audio files of standard/normal length.

### 7.6 Performance analysis of *i*-ACT with data augmentation methods on *i*-AUrbanSound8K & *i*-AAESDD

To verify the scalability of the proposed scheme, *i*-ACT is implemented on *i*-AUrbanSound8K and *i*-AAESDD datasets. Table 7 demonstrates the classification accuracy of *i*-ACT in conjunction with various data augmentation methods as compared to prior algorithms on *i*-AUrbanSound8K. It should be noted that while *i*-AUrbanSound8K predominantly comprises urban sounds, it contains fewer sound categories than ESC-50 dataset. According to the results presented in Table 7, *i*-ACT with data augmentation methods outperforms previous methods.

Table 7 illustrates a noteworthy finding: while the latest SOTA for UrbanSound8K stands at 90% [8], *i*-ACT, when paired with data augmentations, achieved an accuracy of 96.4% on the *i*-AUrbanSound8K. This discrepancy is partially

**Table 5** Classification accuracy of *i*-ACT with data augmentation methods on *i*-AESC-50

Methods + Data Augmentation	Accuracy (%)
PSLA	72.37
AST	67.42
<i>i</i> -ACT	75.01
<i>i</i> -ACT + <b>STTS</b>	84.34
<i>i</i> -ACT + ImageNet	85.59
<i>i</i> -ACT + ImageNet + <b>STTS</b>	91.04
<i>i</i> -ACT + ImageNet + MixUp	88.80
<i>i</i> -ACT + ImageNet + MixUp + <b>STTS</b>	93.02
<i>i</i> -ACT + ImageNet + MixUp + Masking	92.24
PSLA + ImageNet + MixUp + Masking + <b>STTS</b>	84.58
AST + ImageNet + MixUp + Masking + <b>STTS</b>	89.89
<i>i</i> -ACT + ImageNet + MixUp + Masking + <b>STTS</b>	94.16

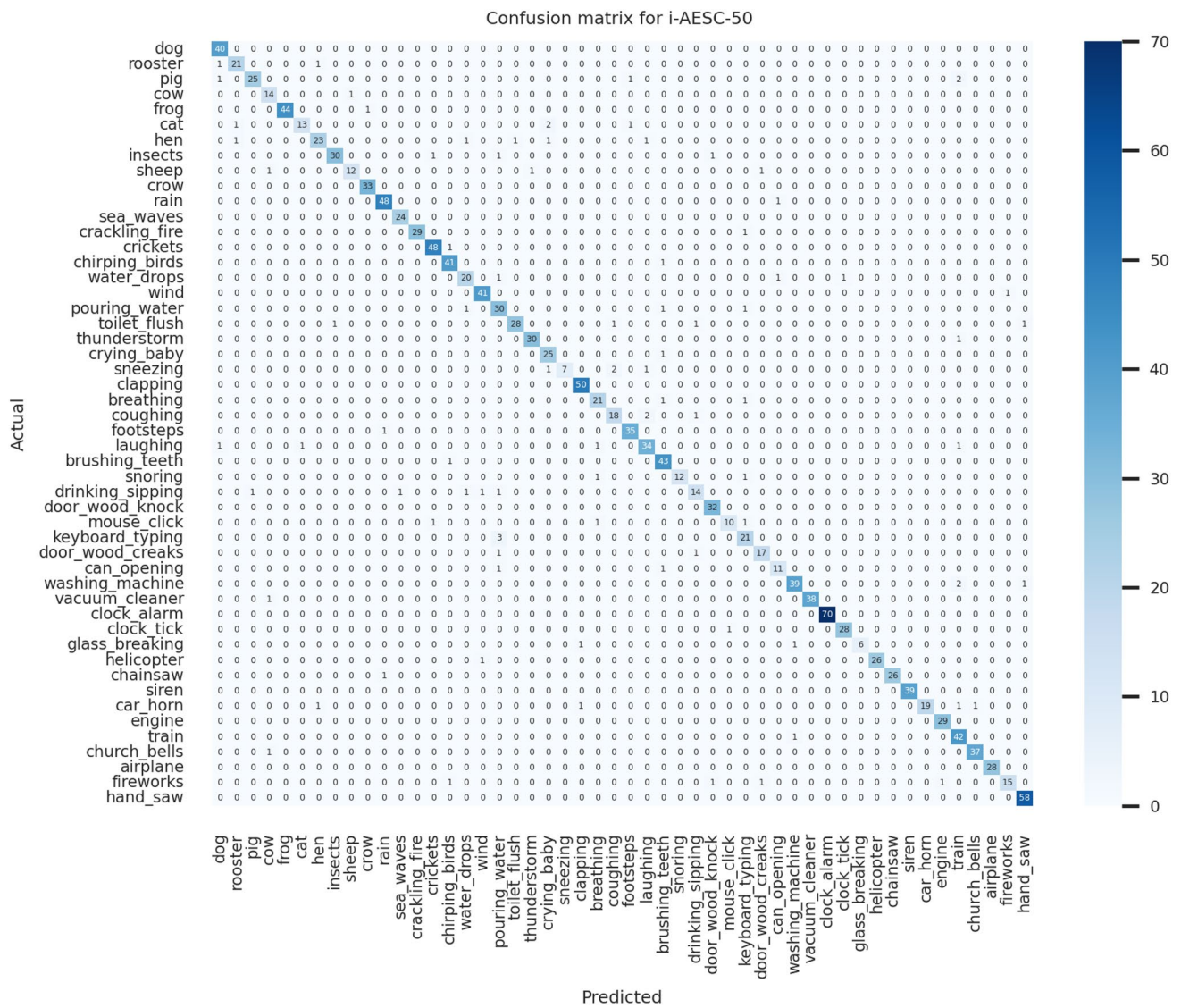
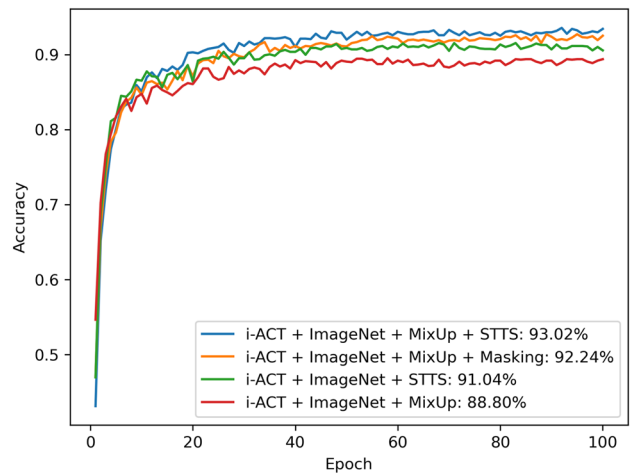


Fig. 5 The confusion matrix of *i*-ACT + ImageNet + MixUp + Masking + STTS on *i*-AESC-50

Fig. 6 Learning curve of STTS compared MixUp and Masking on *i*-AESC-50



**Table 6** Classification accuracy of AST and PSLA with and without STTS on the normal ESC-50

Methods+Data augmentation	Accuracy (%)	With STTS (%)
AST + ImageNet + AudioSet	89.31	89.28
AST + ImageNet + AudioSet + Masking	90.92	90.84
PSLA + ImageNet	82.20	81.90
PSLA + ImageNet + Masking	82.75	81.95

**Fig. 7** The classification accuracy of *i*-AESC-50 by *i*-ACT with different data augmentation methods and pretraining model



**Table 7** Classification accuracy of *i*-ACT with data augmentation methods based on *i*-AUrbanSound8K

Methods + Data Augmentation	Accuracy (%)
PSLA	89.52
AST	88.17
<i>i</i> -ACT	89.86
PSLA + ImageNet + MixUp + Masking + <b>STTS</b>	90.90
AST + ImageNet + MixUp + Masking + <b>STTS</b>	95.76
<i>i</i> -ACT + ImageNet + MixUp + Masking + <b>STTS</b>	96.40

attributed to the possibility that the two datasets, despite similarities, are not identical. Therefore, while they may not be directly comparable, this insight could lay the groundwork for future investigations, such as exploring amplitude-based local data augmentation. Nonetheless, *i*-ACT provided a substantial enhancement in the dataset of *i*-AUrbanSound8K.

Additionally, to demonstrate the superiority of our proposed strategy, it is compared with a SOTA method known as End-to-End Audio Transformer (EAT) [8], which is a combination of CNN and Transformer and has previously achieved the best results on the UrbanSound8K dataset. As shown in Table 8, while EAT achieved an accuracy of 86.32% on *i*-AUrbanSound8K, our method surpassed EAT by nearly 10%, achieving an accuracy of 96.40%.

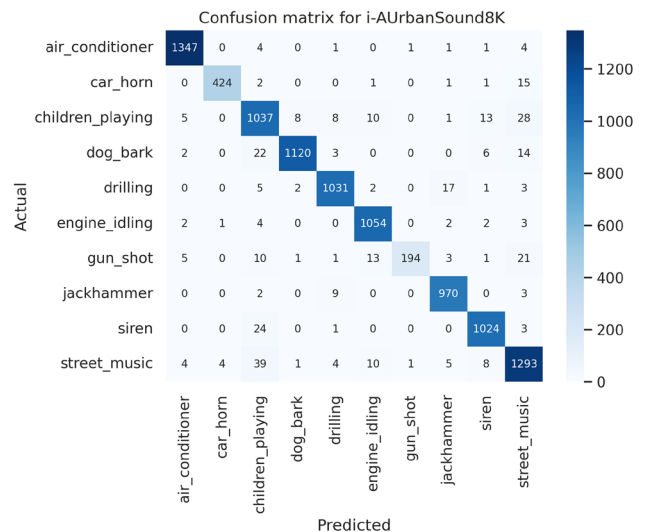
Table 9 reveals that *i*-ACT, when employed alongside data augmentation methods on the emotional speech dataset *i*-AAESDD, achieves an accuracy of 70.98%, surpassing both PSLA and AST.

To delineate the classification capability of the *i*-ACT with data augmentation methods across different categories in both *i*-AUrbanSound8K and *i*-AAESDD, refer to the confusion matrices shown in Figures 8 and 9.

### 7.7 Performance analysis of *i*-ACT with data augmentation methods on *i*-AReaLISED & *i*-ARAVDESS

In contrast to the *i*-AUrbanSound8K dataset, the *i*-AReaLISED dataset encompasses indoor sound events and features a broader array of sound types. The outcomes of implementing *i*-ACT on *i*-AReaLISED are depicted in Table 10. The original *i*-ACT presents an accuracy of 53.85%, whereas PSLA and AST exhibit accuracies of 50.30% and 53.24%, respectively. However, *i*-ACT, when harmonized with data augmentation strategies, transcends both PSLA and AST, achieving an accuracy of 70.98% as opposed to the 65.58% and 59.87% achieved by PSLA and AST (when utilizing identical data augmentation strategies), respectively, as evidenced in Table 10.

**Fig. 8** The confusion matrix of *i*-ACT + ImageNet + MixUp + Masking + **STTS** on *i*-AUrban-Sound8K



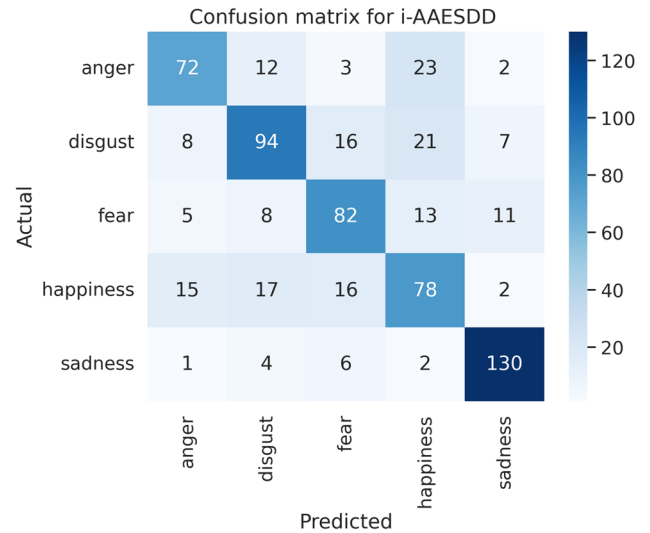
**Table 8** Classification accuracy of SOTA and proposed strategy based on *i*-AUrbanSound8K

SOTA method-EAT	Our method
86.32%	96.40%

**Table 9** Classification accuracy of *i*-ACT with data augmentation methods based on *i*-AAESDD

Methods + Data Augmentation	Accuracy (%)
PSLA	50.30
AST	53.24
<i>i</i> -ACT	53.85
PSLA + ImageNet + MixUp + Masking + <b>STTS</b>	59.87
AST + ImageNet + MixUp + Masking + <b>STTS</b>	65.58
<i>i</i> -ACT + ImageNet + MixUp + Masking + <b>STTS</b>	70.98

**Fig. 9** The confusion matrix of *i*-ACT + ImageNet + MixUp + Masking + **STTS** on *i*-AAESDD



**Table 10** Classification accuracy of *i*-ACT with data augmentation methods based on *i*-AReaLISED

Methods + Data Augmentation	Accuracy (%)
PSLA	80.31
AST	81.16
<i>i</i> -ACT	81.74
PSLA + ImageNet + MixUp + Masking + <b>STTS</b>	84.59
AST + ImageNet + MixUp + Masking + <b>STTS</b>	86.41
<i>i</i> -ACT + ImageNet + MixUp + Masking + <b>STTS</b>	89.28

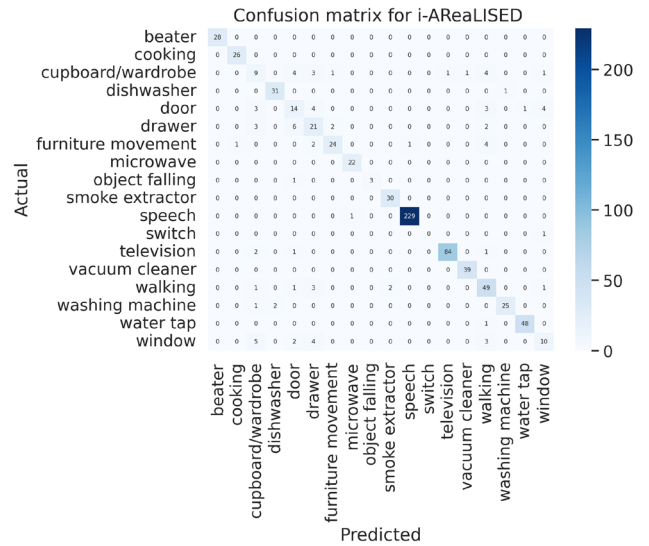
To outline the *i*-ACT with data augmentation methods' classification capability across diverse categories in both *i*-AReaLISED and *i*-ARAVDESS, consult the confusion matrices depicted in Figures 10 and 11.

### 7.8 Performance analysis of Speech emotion datasets *i*-AAESDD & *i*-ARAVDESS

The analysis of *i*-RAVDESS dataset, indicated an accuracy rate slightly above 50%. This relatively lower performance can be attributed to the complex nature of emotional expression in speech, which often requires analysis of the entire utterance. Short audio clips make it difficult to distinguish emotions, which are conveyed through tone, pitch, rhythm, and language in speech. The proposed model, designed primarily for short-duration audio, struggles to capture these intricate emotional cues effectively within such short snippets. Like with *i*-RAVDESS, the unique characteristics of emotional speech in *i*-AESDD pose challenges for our current model. Despite these challenges, the proposed method demonstrated approximately 22% greater efficiency for the *i*-ARAVDESS dataset compared to other methods, as shown in Table 11. A parallel improvement of 20% in efficiency was noted for the *i*-AAESDD dataset, as detailed in Table 9.

In contrast to *i*-AAESDD, *i*-ARAVDESS is created using English speech and has more varieties of sound events. It can be seen from Table 11, that classifying the emotional speech dataset in *i*-ARAVDESS is more challenging than in *i*-AAESDD.

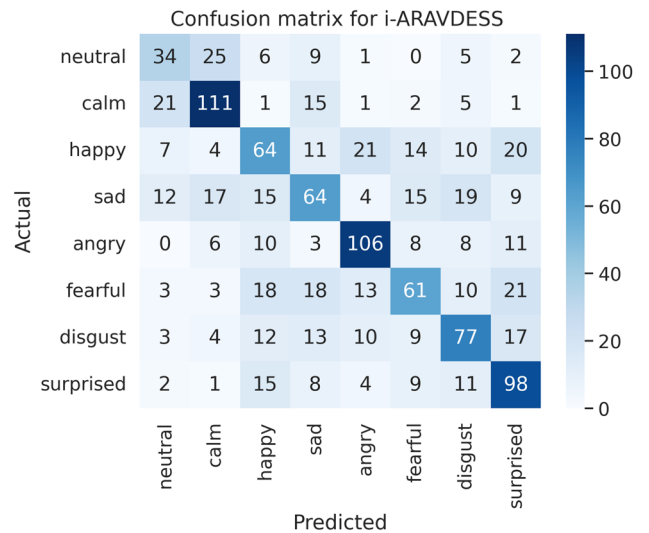
**Fig. 10** The confusion matrix of *i*-ACT + ImageNet + MixUp + Masking + **STTS** on *i*-ARaLISED



**Table 11** Classification accuracy of *i*-ACT with data augmentation methods based on *i*-ARAVDESS

Methods + Data Augmentation	Accuracy (%)
PSLA	31.72
AST	33.19
<i>i</i> -ACT	35.09
PSLA + ImageNet + MixUp + Masking + <b>STTS</b>	36.58
AST + ImageNet + MixUp + Masking + <b>STTS</b>	50.36
<i>i</i> -ACT + ImageNet + MixUp + Masking + <b>STTS</b>	53.51

**Fig. 11** The confusion matrix of *i*-ACT + ImageNet + MixUp + Masking + **STTS** on *i*-ARAVDESS



Nevertheless, both the original *i*-ACT and *i*-ACT equipped with augmentation approaches demonstrate superior performance compared to AST and PS LA.

## 8 Conclusion

This study introduces *i*-ACT, a Transformer-based algorithm for instantaneous audio classification. Initial research involved employing a dBFS-based filter to extract truncated audio recordings from publicly accessible datasets by removing silent segments. This filter is versatile for application to various audio datasets. A new data augmentation method, STTS, is also employed to further improve classification accuracy. The proposed methodology is evaluated against different sound events and proves effective in classifying very short-duration sounds. A comparison between the proposed data augmentation scheme and standard methods was conducted on instantaneous audio datasets, yielding accuracy rates of 94.16%, 96.40%, 70.98%, 89.28%, and 53.51%, respectively. This study establishes a foundation for the rapid and accurate interpretation of auditory signals.

The implementation of instantaneous audio classification holds substantial potential for a wide variety of applications and future studies across diverse domains. Encompassing real-time applications in emergency response and healthcare, enhancing user experiences in gaming and smart technologies, and facilitating advancements in academic, industrial, and environmental research.

**Acknowledgement** This work was partly supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-02068, Artificial Intelligence Innovation Hub) and (No. 2022-0-00621, Development of artificial intelligence technology that provides dialog-based multi-modal explainability). The authors would also like to express their gratitude to Ms. Zhao Rong for her assistance in creating graphs for this manuscript.

**Author contributions** HJ and MSM conceived the idea. Material preparation, data collection, and analysis were carried out by HJ, MSM and HM. The review and editing were conducted by MSM, HM and UP. The first draft of the manuscript was written by H. JIANG, and subsequent versions were written and edited by HM. All authors provided comments on previous versions of the manuscript. All authors have read and approved the final manuscript. The authors would also like to express their gratitude to Ms. Zhao Rong for her assistance in creating graphs for this manuscript.

**Funding** This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub) and (No.2022-0-00621, Development of artificial intelligence technology that provides dialog-based multi-modal explainability).

**Data availability** All processed audio datasets will be publicly available, after the manuscript is published.

## Declarations

**Ethics approval and consent to participate** This article does not contain any studies with human participants or animals performed by any of the authors.

**Competing interests** The authors have no competing interests to declare that are relevant to the content of this article.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Alqudaihi KS, Aslam N, Khan IU, Almuhaideb AM, Alsunaidi SJ, Ibrahim NMAR, Alhaidari FA, Shaikh FS, Alsenbel YM, Alalharith DM, et al. Cough sound detection and diagnosis using artificial intelligence techniques: challenges and opportunities. *IEEE Access*. 2021;9:102327–44.
2. Arandjelovic R, Zisserman A. Objects that sound. In: Proceedings of the European conference on computer vision (ECCV), 2018:435–451. [https://openaccess.thecvf.com/content\\_ECCV\\_2018/papers/Relja\\_Arandjelovic\\_Objects\\_that\\_Sound\\_ECCV\\_2018\\_paper.pdf](https://openaccess.thecvf.com/content_ECCV_2018/papers/Relja_Arandjelovic_Objects_that_Sound_ECCV_2018_paper.pdf)



3. Chacon-Rodriguez A, Julian P, Castro L, Alvarado P, Hernández N. Evaluation of gunshot detection algorithms. *IEEE TCAS-I*. 2010;58(2):363–73.
4. Chen K, Du X, Zhu B, Ma Z, Berg-Kirkpatrick T, Dubnov S. Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022; pp. 646–650. IEEE (2022). <https://ieeexplore.ieee.org/document/9746312>
5. Devlin J, Chang MW, Lee K, Toutanova K. BERT: Pre-training of deep bidirectional transformers for language understanding. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota, 2019; <https://doi.org/10.18653/v1/N19-1423>.
6. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S, Uszkoreit J, Houslyb N. An image is worth 16x16 words: Transformers for image recognition at scale. In: *International Conference on Learning Representations*, 2021; <https://openreview.net/forum?id=YicbFdNTTy>
7. Fang Z, Yin B, Du Z, Huang X. Fast environmental sound classification based on resource adaptive convolutional neural network. *Sci Rep*. 2022;12(1):1–18.
8. Gazneli A, Zimerman G, Ridnik T, Sharir G, Noy A. End-to-end audio strikes back: Boosting augmentations towards an efficient audio classification network. 2022, arXiv preprint [arXiv:2204.11479](https://arxiv.org/abs/2204.11479)
9. Giannakopoulos P, Pikrakis A, Cotronis Y. Improving post-processing of audio event detectors using reinforcement learning. *IEEE Access*. 2022;10:84398–404. <https://doi.org/10.1109/ACCESS.2022.3197907>.
10. Gjerdingen RO, Perrott D. Scanning the dial: the rapid recognition of music genres. *J New Music Res*. 2008;37(2):93–100. <https://doi.org/10.1080/09298210802479268>.
11. Gong Y, Chung YA, Glass J. Ast: Audio spectrogram transformer. *INTERSPEECH*; 2021. [https://www.isca-speech.org/archive/pdfs/inter\\_speech\\_2021/gong21b\\_interspeech.pdf](https://www.isca-speech.org/archive/pdfs/inter_speech_2021/gong21b_interspeech.pdf)
12. Gong Y, Chung YA, Glass J. Psla: Improving audio tagging with pretraining, sampling, labeling, and aggregation. *IEEE/ACM Trans Audio Speech Lang*. 2021;29:3292–306.
13. Gong Y, Lai C, Chung YA, Glass J. Ssast: Self-supervised audio spectrogram transformer. *Proc Innov Appl Artif Intell Conf*. 2022;36:10699–709.
14. Han K, Wang Y, Chen H, Chen X, Guo J, Liu Z, Tang Y, Xiao A, Xu C, Xu Y, et al. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 2022. <https://ieeexplore.ieee.org/abstract/document/9716741>
15. Iqbal T, Kong Q, Plumbley M, Wang W. Stacked convolutional neural networks for general-purpose audio tagging. *DCASE2018 Challenge*, 2018. [http://personal.ee.surrey.ac.uk/Personal/W.Wang/papers/IqbalKPW\\_DCASE2018\\_task2\\_technical\\_report.pdf](http://personal.ee.surrey.ac.uk/Personal/W.Wang/papers/IqbalKPW_DCASE2018_task2_technical_report.pdf)
16. Kim G, Han DK, Ko H. Specmix: A mixed sample data augmentation method for training withtime-frequency domain features. *INTERSPEECH*, 2021. [https://www.isca-speech.org/archive/pdfs/inter\\_speech\\_2021/kim21c\\_interspeech.pdf](https://www.isca-speech.org/archive/pdfs/inter_speech_2021/kim21c_interspeech.pdf)
17. Kingma DP, Ba J. Adam: A method for stochastic optimization. In: *ICLR (Poster)*, 2015. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
18. Koutini K, Schlüter J, Eghbal-zadeh H, Widmer G. Efficient training of audio transformers with patchout. *INTERSPEECH*, 2022. [https://www.isca-speech.org/archive/pdfs/inter\\_speech\\_2022/koutini22\\_interspeech.pdf](https://www.isca-speech.org/archive/pdfs/inter_speech_2022/koutini22_interspeech.pdf)
19. Kumar V, Choudhary A, Cho E. Data augmentation using pre-trained transformer models. In: *Proceedings of the Second Workshop on Life-long Learning for Spoken Language Systems*. Association for Computational Linguistics; 2020. <https://aclanthology.org/2020.lifelongnlp-1.3.pdf>
20. Lewis J. Understanding microphone sensitivity. *Analog Dialogue*. 2012;46(2):14–6.
21. Liu Z, Lv Q, Yang Z, Li Y, Lee CH, Shen L. Recent progress in transformer-based medical image analysis. *Comput Biol Med* 2023;107268. <https://www.sciencedirect.com/science/article/pii/S0010482523007333>
22. Livingstone SR, Russo FA. The Ryerson audio-visual database of emotional speech and song (RAVDESS): a dynamic, multimodal set of facial and vocal expressions in North American English. *PloS One*. 2018;13(5): e0196391.
23. Lu L, Yi Y, Huang F, Wang K, Wang Q. Integrating local CNN and global CNN for script identification in natural scene images. *IEEE Access*. 2019;7:52669–79. <https://doi.org/10.1109/ACCESS.2019.2911964>.
24. Michaely AH, Zhang X, Simko G, Parada C, Aleksic P. Keyword spotting for google assistant using contextual speech recognition. In: *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, IEEE, 2017; pp. 272–278. <https://ieeexplore.ieee.org/abstract/document/8268946>
25. Mohino-Herranz I, García-Gómez J, Aguilar-Ortega M, Utrilla-Manso M, Gil-Pita R, Rosa-Zurera M. Introducing the realised dataset for sound event classification. *Electronics*. 2022;11(12):1811.
26. Park DS, Chan W, Zhang Y, Chiu CC, Zoph B, Cubuk ED, Le QV. SpecAugment: a simple data augmentation method for automatic speech recognition. *INTERSPEECH*, 2019.
27. Peng Y, Liao M, Song Y, Liu Z, He H, Deng H, Wang Y. FB-CNN: Feature fusion-based bilinear CNN for classification of fruit fly image. *IEEE Access*. 2020;8:3987–95. <https://doi.org/10.1109/ACCESS.2019.2961767>.
28. Piczak KJ. Esc: Dataset for environmental sound classification. In: *Proceedings of the 23rd ACM international conference on Multimedia*, 2015:1015–1018. <https://github.com/karolpiczak/ESC-50>
29. Price J. Userstanding db. In: *Professional Audio 2007*.
30. Qamhan MA, Altaheri H, Meftah AH, Muhammad G, Alotaibi YA. Digital audio forensics: microphone and environment classification using deep learning. *IEEE Access*. 2021;9:62719–33. <https://doi.org/10.1109/ACCESS.2021.3073786>.
31. Rajan R, Johnson J, Abdul Kareem N. Bird call classification using DNN-based acoustic modelling. *Circuits, Systems, and Signal Processing*, 2022:1–12. <https://link.springer.com/article/10.1007/s00034-021-01896-2>
32. Salamon J, Jacoby C, Bello JP. A dataset and taxonomy for urban sound research. In: *Proceedings of the 22nd ACM international conference on Multimedia*, 2014:1041–1044. <https://urbansounddataset.weebly.com/urbansound8k.html>
33. Senocak A, Oh TH, Kim J, Yang MH, Kweon IS. Learning to localize sound source in visual scenes. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018:4358–4366. [https://openaccess.thecvf.com/content\\_cvpr\\_2018/html/Senocak\\_Learning\\_to\\_Localize\\_CVPR\\_2018\\_paper.html](https://openaccess.thecvf.com/content_cvpr_2018/html/Senocak_Learning_to_Localize_CVPR_2018_paper.html)

34. Senocak A, Oh TH, Kim J, Yang MH, Kweon IS. Learning to localize sound sources in visual scenes: analysis and applications. *IEEE PAMI*. 2019;43(5):1605–19.
35. Song Q, Sun B, Li S. Multimodal sparse transformer network for audio-visual speech recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 2022. <https://ieeexplore.ieee.org/abstract/document/9755926>
36. Summers C, Dinneen MJ. Improved mixed-example data augmentation. In: 2019 IEEE winter conference on applications of computer vision (WACV). IEEE 2019; pp. 1262–1270. <https://ieeexplore.ieee.org/abstract/document/8659168>
37. Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning, PMLR, 2019, pp. 6105–6114. <https://arxiv.org/pdf/1905.11946.pdf>
38. Tokozume Y, Ushiku Y, Harada T. Learning from between-class examples for deep sound recognition. In: International Conference on Learning Representations, 2018. <https://openreview.net/forum?id=B1Gi6LeRZ>
39. Touvron H, Cord M, Douze M, Massa F, Sablayrolles A, Jégou H. Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, PMLR, 2021; pp. 10347–10357. <https://proceedings.mlr.press/v139/touvron21a.html>
40. Vryzas N, Kotsakis R, Liatsou A, Dimoulas CA, Kalliris G. Speech emotion recognition for performance interaction. *J Audio Eng Soc*. 2018;66(6):457–67.
41. Vryzas N, Matsiola M, Kotsakis R, Dimoulas C, Kalliris G. Subjective evaluation of a speech emotion recognition interaction framework. In: Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion, 2018:1–7. <https://dl.acm.org/doi/abs/10.1145/3243274.3243294>
42. Yao S, Niu B, Liu J. Enhancing sampling and counting method for audio retrieval with time-stretch resistance. In: 2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM), IEEE, 2018; pp. 1–5. <https://ieeexplore.ieee.org/abstract/document/8499068>
43. Zhao W, Yin B. Environmental sound classification based on adding noise. In: 2021 IEEE 2nd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA), IEEE. 2021;2:887–892. <https://ieeexplore.ieee.org/abstract/document/9688248>
44. Zhao W, Yin B. Environmental sound classification based on pitch shifting. In: 2022 International Seminar on Computer Science and Engineering Technology (SCSET), 2022:275–280. IEEE. <https://ieeexplore.ieee.org/abstract/document/9700940>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.