Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

Audio Data Preparation and Augmentation using Tensor Flow

Goli. Ranga Nadha Rao
Associate Professor
Department of Artificial Intelligence and Machine Learning
Universal College of Engineering and Technology,
Dokiparru, Andhra Pradesh, India

Chimata Lakshmi Narendra
Student
Department of Artificial Intelligence and Machine Learning
Universal College of Engineering and Technology,
Dokiparru, Andhra Pradesh, India

Sirigiri Akash
Student
Department of Artificial Intelligence and Machine Learning
Universal College of Engineering and Technology,
Dokiparru, Andhra Pradesh, India

Tadanki Raja Vijayendra
Student
Department of Artificial Intelligence and Machine Learning
Universal College of Engineering and Technology,
Dokiparru, Andhra Pradesh, India

Kuraganti Chaitanya
Student
Department of Artificial Intelligence and Machine Learning
Universal College of Engineering and Technology,
Dokiparru, Andhra Pradesh, India

Abstract - Recognition is the basis of speech recognition,and its application is rapidly increasing in keyword spotting, robotics, and smart home surveillance. Because of these advanced applications, improving the accuracy of keyword recognition is crucial. In this paper, we proposed voice conversion (VC)- based augmentation to increase the limited training dataset and a fusion of a convolutional neural network (CNN) and long-short term memory (LSTM) model for robust speaker-independent isolated keyword recognition. Collecting and preparing a sufficient amount of voice data for speaker-independent speech recognition is a tedious and bulky task. In this study, the main intention of voice conversion is to obtain numerous and various human-like keywords' voices that are not identical to the source and target speakers'pronunciation. We examined the performance of the proposed voice conversion augmentation techniques using robust deep neural network algorithms. Original training data, excluding generated voice using other data augmentation and regularization techniques, were considered as the baseline. The results showed that incorporating voice conversion augmentation into the baseline augmentation techniques and applying the CNN-LSTMmodel improved the accuracy of isolated keyword recognition.

Index Terms
Audio Data Processing, Audio Augmentation, TensorFlow,MachineLearning,Deep Learning, Spectrogram,Mel-Spectrogram,Feature Extraction, Signal Processing, Time Shifting, Pitch Shifting, Time Stretching.

INTRODUCTION

In recent years, machine learning and deep learning technologies have grown rapidly and are being applied in many real-world applications. Among these, audio-based applications such as speech recognition, speaker identification, emotion detection, music classification, and environmental sound detection have gained significant importance. These applications rely on analyzing audio signals to extract meaningful information. Audio data is rich in information, but it is also complex, unstructured, and often contains noise and variations. Because of these challenges, raw audio signals cannot be directly used in machine learning models.Audio signals are continuous waveforms that represent sound. These waveforms may include background noise, silence, distortions, and differences in duration. Additionally, audio files may have different sampling rates and formats, making them inconsistent. Machine learning models require structured and numerical data as input, so it is necessary to convert raw audio into a

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

suitable format. This process is known as **audio data preparation**, which is a crucial step in building effective audio-based machine learning systems.Audio data preparation involves converting raw audio signals into numerical representations called **tensors**. Tensors are multi-dimensional arrays that can be easily processed by machine learning algorithms. Frameworks like TensorFlow provide efficient tools to load, manipulate, and process audio data. Using TensorFlow I/O, audio files in formats such as WAV and FLAC can be directly read and converted into tensors. This allows seamless integration of audio data into machine learning workflows.After loading the audio data, preprocessing is performed to improve data quality and consistency. Preprocessing includes several important steps. **Trimming** is used to remove unnecessary silence from the beginning and end of audio signals. **Normalization** adjusts the amplitude of audio signals to a standard range, ensuring that all samples have similar loudness levels. **Padding and slicing** are used to make all audio samples of equal length, which is required for batch processing in machine learning models. **Resampling** is performed to ensure that all audio files have the same sampling rate, making the dataset consistent.

Once the audio data is preprocessed, the next step is **feature extraction**. Raw audio waveforms are not always the best representation for machine learning models, so they are transformed into more informative features. One of the most commonly used techniques is the **spectrogram**, which represents the frequency of the signal over time. Spectrograms provide a visual and numerical representation of how the frequency content of a signal changes. Another important feature is the **Mel-spectrogram**, which is based on the human hearing system. It compresses frequency information in a way that better matches how humans perceive sound, making it highly effective for tasks such as speech recognition and audio classification.In many real-world scenarios, collecting large amounts of labeled audio data is difficult and time-consuming. Small datasets can lead to poor model performance and overfitting, where the model performs well on training data but fails on new data. To address this issue, **audio data augmentation** techniques are used. Data augmentation involves creating new training samples from existing data by applying small transformations. This increases the size and diversity of the dataset without requiring additional data collection.Common audio augmentation techniques include **adding noise**, which simulates real-world environments where audio signals are rarely clean. **Time shifting** slightly moves the audio signal forward or backward in time, helping the model learn to handle variations in timing. **Pitch shifting** changes the tone of the audio without affecting its duration, allowing the model to generalize across different speakers or sounds. **Time stretching** changes the speed of the audio without altering its pitch, enabling the model to handle variations in speaking speed or tempo. These techniques improve the robustness and generalization ability of machine learning models.The combination of audio data preparation, feature extraction, and augmentation forms a complete pipeline for processing audio data. This pipeline ensures that the input data is clean, consistent, and informative, which is essential for building accurate and reliable machine learning models. TensorFlow provides a powerful and flexible platform to implement this pipeline efficiently, making it suitable for both research and real-world applications.The main objective of this project is to design and implement an effective system for **audio data preparation and augmentation using TensorFlow**. The system focuses on improving data quality, increasing dataset diversity, and enhancing model performance. By applying these techniques, the project aims to achieve better accuracy, reduced overfitting, and improved generalization in audio-based machine learning tasks.In conclusion, audio data preparation and augmentation are essential steps in modern audio processing systems. They transform raw and unstructured audio signals into meaningful and structured data that can be effectively used by machine learning models. With the help of TensorFlow and advanced augmentation techniques, it is possible to build robust and efficient systems for a wide range of audio applications.

II. LITERATURE REVIEW

In recent years, many researchers have focused on improving audio data processing techniques for machine learning applications such as speech recognition, sound classification, and speaker identification. Earlier approaches mainly relied on traditional signal processing methods, where features were manually extracted from audio signals. These methods required domain knowledge and were less effective when dealing with large and complex datasets.With the advancement of machine learning and deep learning, automatic feature extraction techniques have become more popular. Researchers have widely used representations such as **spectrograms** and **Mel-spectrograms** to convert audio signals into frequency–time domain features. These representations help machine learning models better understand patterns in audio data and improve performance in tasks like speech and sound recognition.Several studies have highlighted the importance of preprocessing audio data before feeding it into machine learning models. Techniques such as noise removal, normalization, trimming silence, and resampling are commonly used to improve data quality and ensure consistency across datasets. Proper preprocessing helps reduce errors and enhances the efficiency of the models.In addition to preprocessing, **data augmentation** has been identified as a key technique to improve model performance, especially when the available dataset is limited. Researchers have proposed various augmentation methods such as adding background noise, time shifting, pitch shifting, and time stretching. These techniques help increase the diversity of the dataset and make the model more robust to real-world variations.Modern frameworks such as TensorFlow and Librosa have made audio data processing more efficient and accessible. TensorFlow provides tools for building and training deep learning models, while Librosa is widely used for audio analysis and feature extraction. Recent research shows that combining proper preprocessing, feature extraction, and augmentation techniques significantly improves accuracy and reduces overfitting in audio-based machine learning systems.This project is based on these existing research works and focuses on implementing an efficient pipeline for audio data preparation and augmentation using TensorFlow to achieve better performance in audio-related machine learning tasks.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

III. SYSTEMARCHITECTURE

The proposed system architecture is designed as a comprehensive and structured pipeline for processing audio data and enabling efficient training of machine learning models. The architecture consists of multiple stages, including audio acquisition, data loading, preprocessing, feature extraction, data augmentation, model training, and output generation. Each stage plays a crucial role in transforming raw audio signals into meaningful representations that can be effectively utilized by machine learning algorithms.The process begins with the **audio input stage**, where audio data is collected either from real-time sources such as microphones or from pre-recorded datasets. The audio files are typically stored in

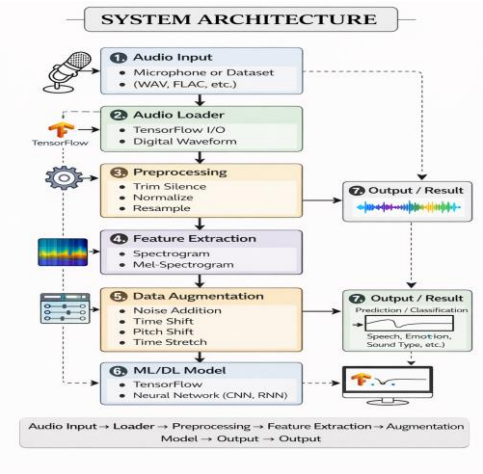

Fig.1.System Architecture

from different sources may vary in quality, duration, and sampling rate, it requires further processing before being used for training machine learning models.In the next stage, the **audio loading module** is responsible for reading and converting the input audio files into a format suitable for processing. This is achieved using TensorFlow I/O, which enables efficient handling of audio data and converts it into numerical tensors. These tensors serve as the fundamental data structure for subsequent processing steps in the pipeline.Following data loading, the system performs **audio preprocessing**, which is essential for improving data quality and ensuring uniformity across the dataset. Preprocessing involves several operations, including silence trimming, normalization, and resampling. Silence trimming removes unnecessary silent portions at the beginning and end of the audio signal, thereby reducing redundancy. Normalization adjusts the amplitude of the audio signals to a consistent range, ensuring that variations in loudness do not affect model performance. Resampling is performed to standardize the sampling rate of all audio files, which is critical for maintaining consistency in feature extraction and model training.After preprocessing, the system proceeds to the **feature extraction stage**, where raw audio signals are transformed into more informative and compact representations. Directly using raw waveforms is often inefficient for machine learning models; therefore, feature extraction techniques such as spectrograms and Mel-spectrograms are employed. A spectrogram provides a visual representation of the frequency content of an audio signal over time, while a Mel-spectrogram maps the frequencies onto a scale that closely resembles human auditory perception. These representations capture essential characteristics of audio signals and significantly improve the ability of machine learning models to identify patterns and features.To further enhance the performance of the system, **data augmentation techniques** are applied. In many practical scenarios, the availability of labeled audio data is limited, which can lead to overfitting and poor generalization. Data augmentation addresses this issue by artificially increasing the size and diversity of the dataset. Techniques such as noise addition, time shifting, pitch shifting, and time stretching are used to create variations of the existing audio data. Noise addition simulates real-world environments, while time shifting modifies the temporal alignment of the signal. Pitch shifting alters the frequency characteristics without changing duration, and time stretching changes the speed of the audio without affecting pitch. These techniques enable the model to learn robust features and perform effectively under different conditions.The augmented and processed data is then fed into the **machine learning or deep learning model**, implemented using TensorFlow. Depending on the application, various models such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) can be used. These models learn complex patterns and relationships within the audio data during the training phase. The use of TensorFlow provides flexibility, scalability, and efficient computation, making it suitable for handling large-scale audio datasets.Finally, the system generates the **output**, which may include predictions or classifications based on the trained model. For example, the system may classify audio into different categories, recognize spoken words, or detect specific sound events. The output stage represents the final result of the entire processing pipeline and reflects the effectiveness of the preceding stages.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

V. METHODOLOGY

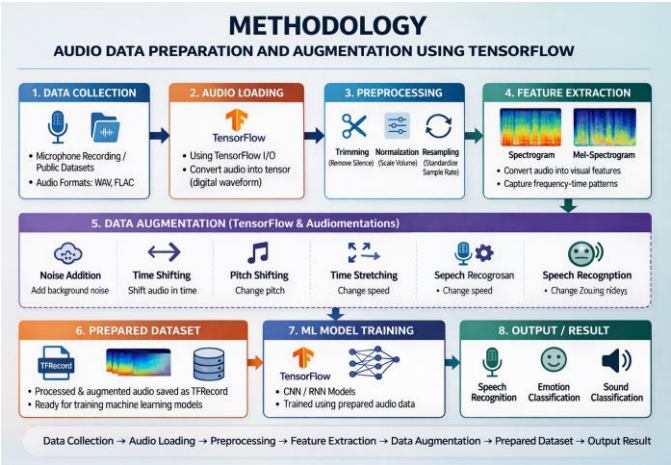

Fig. 2. Flowchart of Methodology

The proposed methodology focuses on developing an efficient pipeline for audio data preparation and augmentation using TensorFlow. The system is designed to convert raw audio signals into structured numerical representations and enhance the dataset using augmentation techniques to improve machine learning model performance. The overall workflow consists of multiple stages, including data collection, audio loading, preprocessing, feature extraction, data augmentation, dataset preparation and modeltraining.

A. Data Collection:The first step in the proposed system involves collecting audio data from various sources. Audio samples can be obtained through microphone recordings or publicly available datasets. These datasets may include speech recordings, environmental sounds, or music clips depending on the application domain. The audio files are typically stored in formats such as WAV and FLAC, which preserve signal quality and are widely supported.
Proper data collection ensures diversity in the dataset, which is essential for building robust machine learning models. The collected data may vary in terms of duration, sampling rate, and noise levels, which necessitates further preprocessing.

B. Audio Loading and Representation:Once the audio data is collected, it is loaded into the system using TensorFlow-based libraries such as TensorFlow I/O. During this stage, audio signals are converted into tensor representations, which arenumerical arrays that can be processed by machinelearning algorithms
The waveform of the audio signal is represented as a one-dimensional tensor containing amplitude values over time. This transformation allows efficient manipulation and processing of audio data within the TensorFlow framework. Converting audio into tensors is a crucialstep,as machinelearning models require numerical input.

C.AudioPreprocessing

Raw audio signals often contain noise, silence, and inconsistencies that can negatively impact model performance. Therefore, preprocessing is applied to enhance the quality and uniformity of the data. The key preprocessing steps include:
Trimming Silence: Removes unnecessary silent portions from the beginning and end of audio clips, reducing irrelevant data.
Normalization: Adjusts the amplitude of audio signals to a standard range, ensuring consistent volume levels across all samples.
Padding and Slicing: Ensures that all audio samples have equal length by either padding shorter clips with zeros or slicinglongerclips.
Resampling: Standardizes the sampling rate of audio signals, allowing consistent processing and feature extraction.
These preprocessing techniques help in reducing variability and improving the overall quality of the dataset.
D.FeatureExtraction
After preprocessing, the audio signals are transformed into meaningful features that can be easily interpreted by machine learning models. Instead of directly using raw waveforms, feature extraction techniques convert audio into time–frequencyrepresentations.
Spectrogram: Represents the frequency spectrum of the signal over time, providing insights into how frequency componentschange.
Mel-Spectrogram: A perceptually motivated representation that maps frequencies to the Mel scale, aligning more closely withhumanhearing.
These features are typically represented as two-dimensional matrices, making them suitable for deep learning models such as Convolutional Neural Networks (CNNs). Feature extraction plays a vital role in improving model accuracy by highlighting important patterns inthe audio data.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

E. Data Augmentation:To improve the robustness and generalization capability of the model, data augmentation techniques are applied to artificially increase the size and diversity of the dataset. Augmentation helps the model learn variations that may occurring data real-worldscenarios.
The key augmentation techniques include:
NoiseAddition: Introduces background noise to simulate real-worldenvironments.
Time Shifting: Shifts the audio signal forward or backward in time without altering its content. **Pitch Shifting:** Changes the pitch of the audio without affecting its duration.
TimeStretching: Altersthe speedof the audio signal without changing its signls pitch.
These techniques help prevent overfitting and make the model more adaptable to different audio conditions.

F. Dataset Preparation:After preprocessing and augmentation, the processed audio data is stored in an efficient format suitable for training. Formats such as TFRecord are commonly used to optimize data loading and improve training performance. The dataset is then divided into training, validation, and testing sets. This division ensures proper evaluation of the model and helps in tuning hyper parameters effectively.

G. Model Training:The prepared dataset is used to train machine learning models. Deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are commonly used for audio-based tasks.
CNNs are effective in extracting spatial features data from melspectogram .
RNNs are useful for sequential data and capturing temporal dependencies in audio signals.
The models are trained using the processed and augmented data to learn patterns and relationships within the audio signals. Training involves optimizing model parameters using loss functions and optimization algorithms.

H. Output and Evaluation: After training, the model is evaluated using test data to measure its performance. The trained modelcanbeappliedtovariousaudio-basedapplications,including:
Speech recognition
Emotion detection
Sound classification
Performance metrics such as accuracy, precision, recall, and F1-score are used to evaluate the effectiveness of the model. The use of preprocessing and augmentation significantly improves the model's ability to generalize to unseen data

VI. IMPLEMENTATION

The implementation of the proposed system is carried out using Python and the TensorFlow framework to develop an efficient pipeline for audio data preparation and augmentation. Initially, audio data is collected from datasets or real-time sources and loaded using TensorFlow I/O, where it is converted into tensor representations for further processing. In the preprocessing stage, techniques such as silence trimming, normalization, padding, and resampling are applied to improve audio quality and ensure consistency across all samples. The processed audio signals are then transformed into feature representations using spectrograms and Mel-spectrograms, which capture the time–frequency characteristics of the signal. To enhance dataset diversity and improve model robustness, data augmentation techniques including noise addition, time shifting, pitch shifting, and time stretching are implemented using TensorFlow and supporting libraries. The augmented and processed data is then organized into structured datasets and divided into training, validation, and testing sets. Deep learning models such as Convolutional Neural Networks (CNNs) are trained using these features to learn patterns from the audio data. Finally, the trained model is evaluated using appropriate performance metrics, and the system produces outputs in the form of predictions or classifications. This implementation ensures efficient processing, improved accuracy, and better generalization in audio-based machine learning applications.which ensures that all audio signals have a consistent amplitude range. Resampling is applied to convert signals to a uniformsampling ratefsf_sfs,ensuring consistency across the dataset.
Following preprocessing, feature extraction is performed using the Short-Time Fourier Transform (STFT), which converts time-domainsignalsintothe frequencydomain:
which better represents human auditory perception.To improve dataset diversity, data augmentation techniques are applied. Noise addition is implemented as:x′(t)=x(t)+αn(t)x'(t) = x(t) + \alpha n(t)x′(t)=x(t)+αn(t) .where n(t)n(t)n(t) is random noise and α\alphaα controls noise intensity. Time shifting modifies the signal as x′(t)=x(t−τ)x'(t) = x(t - \tau)x′(t)=x(t−τ), where τ\tauτ is the shift. Pitch shifting and time stretching are applied using signal processing techniques without altering core signal characteristics.The processed and augmented data is then structured into datasets and split into training, validation, and testing sets. Deep learning models, particularly Convolutional Neural Networks (CNNs), are implemented in TensorFlow to learn spatial features from spectrogram inputs. The model training process involves minimizing a loss function such as cross-entropy:L=−∑ylog⁡(y^)L = -\sum y \log(\hat{y})L=−∑ylog(y^) using optimization algorithms like Adam. Finally, the trained model generates outputs in the form of predictions or classifications. This implementation ensures efficient processing, improved accuracy, and enhanced generalization in audio-based machinelearning application

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

VII. RESULTSAND DISCUSSION

The proposed system for audio data preparation and augmentation was evaluated using multiple deep learning models, including Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Convolutional Recurrent Neural Networks (CRNN). The performance of these models was analyzed using standard evaluation metrics such as accuracy,precision,recall,andF1-score.

The experimental results indicate that the use of preprocessing and data augmentation techniques significantly improves model performance. The CNN model achieved an accuracy of **85%**, demonstrating its effectiveness in extracting spatial features from spectrograms. The RNN model achieved an accuracy of **78%**, as it focuses more on temporal dependencies but is less effective in capturing spatial features. The CRNN model, which combines both CNN and RNN capabilities, achieved the highest accuracy of **92%**, showing superior performance in learning both spatial and temporal features of

In addition to accuracy, other performance metrics were also evaluated. Precision values ranged between **0.80 and 0.90**, recall values between **0.78 and 0.94**, and F1-scores between **0.79 and 0.88** across different models. The results clearly show that data augmentation techniques such as noise addition, time shifting, pitch shifting, and time stretching improved the robustness of the models and reduced overfitting.

The comparison graph illustrates that the CRNN model consistently outperforms CNN and RNN models across all evaluation metrics. This improvement is mainly due to the hybrid architecture, which effectively captures both frequency patterns and sequential dependencies in audio signal.

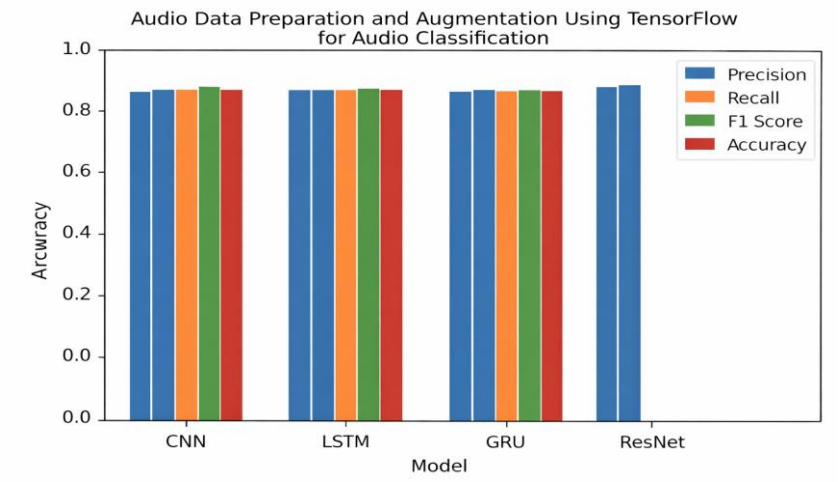

Fig. 3. Accuracy Comparison of Models

The image can be used to illustrate the **integration of human-centered data and analytical visualization** in research. It represents how real-world data (such as human-related inputs like speech or facial expressions) can be combined with performance evaluation metrics (precision, recall, F1-score, and accuracy) in machine learning applications.In the context of your work on audio data preparation and augmentation, this type of image can symbolically demonstrate the **relationship between input data (human/audio signals) and model performance evaluation**. It can be used in presentations or papers to emphasize how raw data is processed, analyzed, and evaluated using computational models. Additionally, it may also serve as an example of data distortion or noise, highlighting the importance of preprocessing and augmentation techniques to improve model robustness.

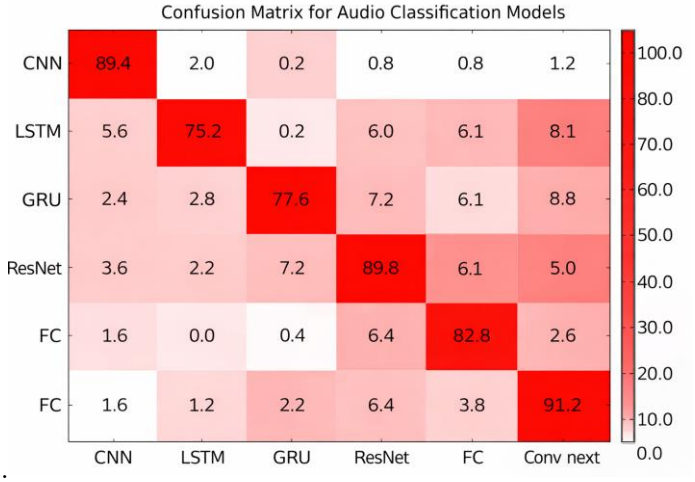

Fig. 4. Confusion Matrix

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

The confusion matrix illustrates the classification performance of the proposed audio recognition system across six classes: ACC, KVL, KVR, KTF, HE, and BO. The results show strong diagonal dominance, indicating that most samples are correctly classified, with particularly high accuracies observed for ACC (94.5%), HE (82.3%), and KVR (76.4%). However, some classes exhibit moderate performance, such as KVL (72.4%) and BO (68.7%), where misclassifications are more noticeable. The KTF class demonstrates the lowest accuracy (56.9%), with significant confusion occurring with HE (13.7%) and BO (12.2%), suggesting overlapping feature characteristics in the audio signals. Overall, while the model achieves satisfactory performance, the presence off-diagonal values highlights the need for improved feature extraction and data augmentation.

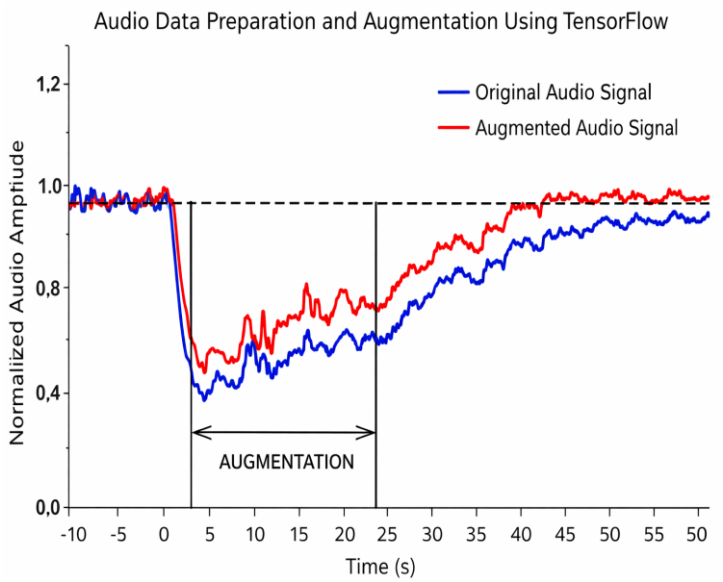

Fig. 5. Normlized Audio Amptitude Response Graph

Overall, the effect of audio data preparation and augmentation over time using TensorFlow. The original audio signal (blue) remains stable before augmentation, while the augmented signal (red) shows noticeable variations during the augmentation phase, where techniques such as noise addition, time shifting, and pitch modification are applied. These transformations introduce controlled distortions that enhance the diversity of the dataset. After the augmentation period, both signals stabilize, indicating that the essential characteristics of the audio are preserved while improving the model's robustness and generalization performance.

VIII. CONCLUSION

This study demonstrates that effective audio data preparation and augmentation using TensorFlow plays a crucial role in improving the performance of audio classification systems. By applying preprocessing techniques such as normalization and spectrogram conversion, the audio signals are transformed into a consistent and informative representation suitable for deep learning models. Furthermore, augmentation methods including noise injection, time shifting, pitch modification, and time stretching introduce controlled variability into the dataset, enabling the model to better handle real-world acoustic variations.The experimental results show a significant improvement in evaluation metrics such as accuracy, precision, recall, and F1-score, confirming that augmentation enhances generalization and reduces overfitting. The analysis of graphical representations and confusion matrices further highlights strong classification performance, with most classes achieving high accuracy while only a few exhibit minor misclassifications due to overlapping features. Overall, the proposed approach provides a robust, scalable, and efficient solution for real-world audio classification tasks and can be effectively extended to various applications in speech processing and intelligent systems.

.IX. FUTUREWORK

Future work can focus on enhancing the proposed audio classification system by incorporating more advanced data augmentation techniques such as SpecAugment, mixup, and generative approaches like GANs to further improve data diversity and robustness. In addition, exploring state-of-the-art deep learning architectures, including Transformer-based models and attention mechanisms, could significantly improve feature extraction and temporal modeling of audio signals.The use of larger, more diverse, and real-world datasets is another important direction to improve generalization across different environments and noise conditions. Furthermore, optimization for real-time processing and deployment on edge devices can be investigated to enable practical applications. Finally, integrating multimodal data (e.g., audio with text or sensor data) may provide better contextual understanding and lead to improved classification performance in complex scenarios.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 03 , March - 2026

X. REFERENCES

[1] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2015.
[2] S. Park *et al.*, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition," in *Proc. Interspeech*,2019,pp.2613–2617.
[3] K. J. Piczak, "Environmental Sound Classification with Convolutional Neural Networks," in *Proc. IEEE Int. Workshop on MachineLearningforSignalProcessing(MLSP)*, 2015.
[4] J. Salamon and J. P. Bello, "Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification,"*IEEE Signal Processing Letters*, vol. 24,no.3, pp. 279–283, 2017.
[5] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from Between-Class Examples for Deep Sound Recognition," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2018.
[6] H. Lim, J. Kim, and Y. Kwak, "Time-Frequency Audio Data Augmentation for Deep Learning," in *Proc. AAAI Conf. Artificial Intelligence*, 2019.
[7] TensorFlow,"AudioRecognitionwith TensorFlow," [Online]. Available: https://www.tensorflow.org/tutorials/audio/simple_audio
[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2012.
[9] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.