Enabling Independence Through Sound Recognition

Sumit Autade
Smt. Indira Gandhi College of Engineering,
Navi Mumbai, Maharashtra
email: sumitautade100@gmail.com

Saurav Yadav
Smt. Indira Gandhi College of Engineering,
Navi Mumbai, Maharashtra
email:yadavsaurav11982@gmail.com

Chirag Rathod
Smt. Indira Gandhi College of
Engineering, Navi Mumbai, Maharashtra
email:rathodchirag15803@gmail.com

Deepti Vijay Chandran
Smt. Indira Gandhi College of
Engineering, Navi Mumbai, Maharashtra
email:dipti.chandran@sigce.edu.in

***Abstract*— "This research introduces a groundbreaking project, "Enabling Independence through Sound Classification," leveraging Artificial Neural Networks (ANNs), Convolutional Neural Networks (CNNs), Mel-frequency cepstral coefficients (MFCCs), and the Librosa library to offer real-time auditory feedback to individuals who are hearing impaired. The project's core objective is to enhance the independence and safety of this community by translating environmental sounds into meaningful alerts and descriptions. Beyond the technical aspects of sound classification, the study emphasizes the profound social impact of promoting inclusivity, self-reliance, and equity for those with auditory challenges. Through a comprehensive exploration of CNN and RNN architectures, along with comparisons to TensorFlow and PyTorch models on a prototype dataset, the proposed approach, incorporating envelope functions, normalization, segmentation, regularization, and dropout layers, demonstrates superior accuracy and reduced loss percentages. This research signifies a pivotal step towards a more accessible and inclusive society, harmonizing technology and empathy for the benefit of individuals with sensory challenges."**

Keywords—ANN, MFCC, DL, ML, RNN, CNN, API, ReLu, CPU, GPU.

I. INTRODUCTION

In a world that is constantly evolving, technology has been a driving force behind making everyday tasks more accessible and convenient for people with various abilities. One such advancement is sound recognition technology, which has the potential to revolutionize the way we interact with our surroundings and enhance the lives of individuals with diverse needs. Sound recognition technology is a burgeoning field that uses artificial intelligence and machine learning to identify and interpret sounds, thereby providing valuable information and support to users. From assisting individuals with hearing impairments to creating a more inclusive and efficient environment, sound recognition technology has far-reaching implications across various sectors, including healthcare, home automation, transportation, and more.

For those who are visually impaired or have limited mobility, sound serves as a crucial conduit to understand and interact with the world. This report delves into the burgeoning field of sound recognition, with a specific focus on how ANNs are revolutionizing the accuracy and scope of audio classification. The goal is to provide individuals with disabilities a pathway to greater independence and accessibility by harnessing the power of artificial intelligence. Individuals with disabilities encounter unique challenges in their daily lives, navigating a world that is often designed without their specific needs in mind. Whether it's recognizing everyday environmental sounds, understanding spoken communication, or responding to critical auditory cues such as alarms, sound recognition plays an indispensable role.

However, conventional sound recognition systems have historically fallen short in providing the nuanced and context-aware information needed for an independent and fulfilling life. This is where the integration of Artificial Neural Networks emerges as a game-changer. ANNs, a subset of artificial intelligence, have demonstrated the ability to learn and interpret audio data with remarkable accuracy. By training ANNs on extensive and diverse datasets of sounds, these neural networks can capture intricate patterns and features in the acoustic realm, enabling precise audio classification.

A. AIM AND OBJECTIVES:

- Assess Sound Recognition Technology Impact: Evaluate its significance, applications, and real-world impact on independence and inclusivity for diverse abilities.
- Showcase Positive Cases: Highlight real-world examples demonstrating the positive effects of sound recognition technology.
- Explore Trends and Future Tech: Identify current trends and emerging technologies in sound recognition, providing insights into potential future developments.
- Compare Neural Network Models: Conduct a comparative analysis of CNN and RNN models for audio data classification, pinpointing performance variables and assessing the impact of optimization techniques.

II. PROBLEM STATEMENT

Despite advances in assistive technologies, many individuals with disabilities continue to face barriers to independence, particularly in recognizing and responding to auditory cues and environmental sounds. The challenge is to leverage sound recognition technology to empower these individuals, making their lives more inclusive and self-reliant.

III. RELATED WORKS

Many significant contributions have been made in the same or related fields. For instance, traditional Automatic-Speech Recognition (ASR) systems have already been developed which works on the principles of the Gaussian mixture model [2]. However, instead of using Gaussian models, the Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) were used to classify audio data, which are models for processing datasets based on deep learning. Research has also been conducted to distinguish between environmental sounds by [16], using the Gaussian Mixture Model, Deep Neural Network, Recurrent Neural Network, Convolutional Neural Network, and i-vector. It was found that DNN was the best performing model while an RNN and CNN fusion provided the best average accuracy albeit, slower. Another comparative analysis was done by using different Machine learning algorithms to classify different audio signals in [6] which described how this procedure can be achieved by treating it as a pattern recognition problem in two stages, extraction, and classification. The authors in [6] have also described several purposes of these techniques such as speech recognition, and automatic bandwidth allocation, which allows for a telephone network to decide how much bandwidth should be allocated for the transmission, that is, music requiring higher while no bandwidth for no background noise, allowing for improved efficiency and also audio database indexing, which is currently done manually and is a very tedious task. [6] have used two k-nearest Neighbor and Support Vector Machine algorithms, concluding that SVM is more accurate while taking less time. Although primarily used in visual recognition contexts, convolutional architectures have also been applied in speech and music analysis [8], even though it takes a while to train the program, it shows that convolutional neural networks can be effectively applied in environmental sound classification tasks even with limited datasets and simple data augmentation. A recent venture called Audio AI by [27] aims to isolate vocals from an audio source such as a song by using Short-Time Fourier Transform (STFT) to expose the structure of human speech and then using Convolutional Neural Network (CNN) to identify and distinguish between voiced and unvoiced sections in a song, estimate the frequency of the fundamental over time and apply a mask to capture the harmonic content. In his algorithm, the author has treated audio signals as images to expose spatial patterns, as two-dimensional frequency against time graphs. It is observed that using CNN's Audio AI successfully separates the vocals from the instrumental section from the audio source. Several related works have also worked on the datasets. In [28], Urban8k Dataset was used to prove that an important piece for resolving (music) audio problems is the architecture alone using deep neural networks. An Extreme Learning Machine (ELM) model has been used and an accuracy of 89.82% was achieved using temporal models. In [26], a convolutional neural network and tensor deep stacking network were used to classify the environmental sounds of the ESC-50 dataset based on the generated spectrograms of these sounds. In [14], the authors tried to prove with the ESC-50 dataset that deep convolutional neural networks, which are designed specifically for object recognition in images, can be successfully trained to classify spectral images of environmental sounds. Using Convolutional Recurrent Neural Network (CRNN) 60.0% accuracy was achieved with Spectrogram, MFCC, and CRP combined and 60.3% with just Spectrogram.

IV. PROPOSED SYSTEM

A. Audio Pre-Processing

Before starting the experiments, the datasets were preprocessed. The pre-processing was done using normalization, Fast Fourier Transformation, Short-Time Fourier Transformation, Mel Filterbank, and Mel Frequency Cepstral Coefficient.

Librosa, a prominent Python library tailored for audio and music signal processing, played a pivotal role in our sound classification model training. Leveraging its robust functionality, Librosa facilitated the extraction of essential audio features, such as Mel-frequency cepstral coefficients (MFCCs), spectrograms, and other relevant representations. These features served as the foundational input for our model, enabling it to discern nuanced patterns within the audio data. Librosa's user-friendly interface and diverse capabilities expedited the preprocessing steps, allowing for seamless integration into the TensorFlow and Keras framework. Its comprehensive set of tools empowered us to navigate the complexities of audio data, contributing significantly to the model's accuracy in classifying diverse sound categories.

1) Signal Normalization: Audio files have different shapes of waves in a file. Microphones usually have a bit depth of 16 which can generate 2 to 16 integers in a time domain to generate waves. To normalize the signals apply a pre-emphasis filter. There are several reasons behind normalizing a signal. In a signal higher frequencies have less magnitude than the lower frequencies. Also to balance the audio noise ratio and numerical complexities in further calculation use pre-emphasis filters on the signal data. In a signal x, apply the following equation*:*

$$y(t) = x(t) - \alpha x(t-1) \tag{1}$$

Here α is the filter coefficient having a value of 0.95 or 0.97 [11].

2) Fast Fourier Transformation (FFT):

Fast Fourier Transformation (FFT) to convert complex wave signals into a coherent Frequency-Magnitude graph, commonly referred to as a Periodogram. The primary aim here is to represent audio data in a more decipherable form. Specifically, the FFT is employed to generate the periodogram, where the maximum frequency is plotted at 22 kHz over Magnitude. This choice aligns with the recording capacity of the microphone, set at 44.1 kHz. The Nyquist frequency, defined as half of the sampling frequency, is strategically set at 22 kHz, ensuring that any frequency above this threshold for our microphone will not be recorded. The outcome of this process is a spectrogram, produced by stacking periodograms over time. Notably, we generate four spectrograms for each 1/10th of a second in the audio file, providing a detailed and informative visual representation of the audio data.

3) Short-Time Fourier Transformation (STFT):

The application of Short-Time Fourier Transformation (STFT) is not merely about stacking periodograms to create a spectrogram; rather, it involves a nuanced approach to overlapping them. This methodology is rooted in the understanding that the frequency of an audio signal undergoes continuous changes over time. In the STFT process, the initial step involves segmenting the sample signal into small frames, advancing the frame window by 1 second. These frames, each spanning a duration of 20-40 ms, employ a 25 ms length in our specific experiment, with a 10 ms forward shift. The result is a new frame that shares 15 ms with the preceding frame, amounting to approximately 60% overlap. This deliberate overlap serves to generate contours, facilitating the assumption that the audio data is relatively static. Following frame segmentation, a window function is applied, introducing bell curves through overlaps, known as a Hamming Window. This specific window function, commonly used in FFT, tapers the wave shape of frames with two overlaps, mitigating spectral leakage and countering assumptions of infinite FFT. The Hamming window equation, defined as

$$W(\mathrm{n})=0.54-0.46\cos\,(2\pi\mathrm{n}\,/N) \qquad (2)$$

where $0 \leqslant \mathrm{n} \leqslant N$ and N represent the window length. While this process yields 400 points in FFT, FFT conventionally operates based on powers of 2, typically employing 512. Thus, after the initial 400 points, 112 zeros are added, culminating in the generation of periodograms for subsequent filtration processes.

4) Mel Filter Bank: The Mel filter bank concept employs triangular filters within the power spectrum, strategically designed based on the Mel Scale. Typically, 25-40 filters are utilized in the Mel filter bank, leveraging a logarithmic equation that emulates the human auditory system's ability to discern noise. This system is particularly adept at distinguishing low-frequency sounds, mirroring the human cochlea's capability to identify differences between, for instance, 70 Hz and 170 Hz. However, as frequencies increase, the human cochlea's discrimination diminishes, illustrated by its inability to differentiate between 17070 Hz and 17170 Hz. The Mel filter bank precisely capitalizes on this principle, focusing on fine distinctions in low frequencies and expanding its range for higher frequencies. In this experiment, a total of 26 Mel filters were employed, though it's common to extend this number up to 40 for specific purposes.

$$\mathrm{M(f)}=2595\log_{10}(1+f/700) \qquad (3)$$

Conversely, to convert a Mel value m back into frequency f (in Hz), the formula is:

$$\mathrm{F}(m) = 700(m^{10/2595}-1) \qquad (4)$$

This Mel filter bank equations generate a matrix of size 26x100, providing 26 vertically stacked values for each second of data and 100 horizontally aligned values. Despite the effectiveness of this approach, the generated images at this stage, as depicted in Figure 1, remain challenging to interpret in terms of associating specific sounds with their visual representations.

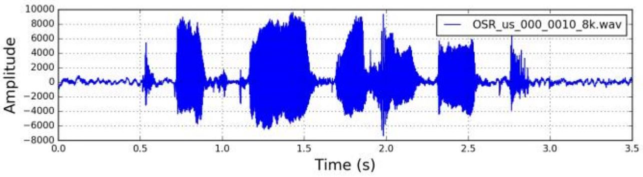


Fig 1: Filter Bank Coefficient of classes

5) Mel Frequency Cepstral Coefficient (MFCC): At the culmination of the signal data preprocessing, the Mel Frequency Cepstral Coefficients (MFCC) represent a refined and compact version of the previously generated image using the Mel filter bank. Employing the Discrete Cosine Transform (DCT), MFCC serves to decorate numerous energies from the preceding energy bands. DCT, a widely employed technique in image and audio compression, excels at discarding specific high-coefficient bits during compression, akin to applying a low-pass analog filter. This strategic compression mitigates data redundancy and soothes edges, embodying the core principle of DCT for audio and image compression. In our context, MFCC adeptly condenses the energy bank image from a 26x100 matrix to a more efficient 13x100 matrix. This compression facilitates the transformation of higher frequencies into lower frequencies, ensuring distinct sounds do not share identical representations. The comprehensive data preprocessing phase concludes with the completion of the MFCC stage.

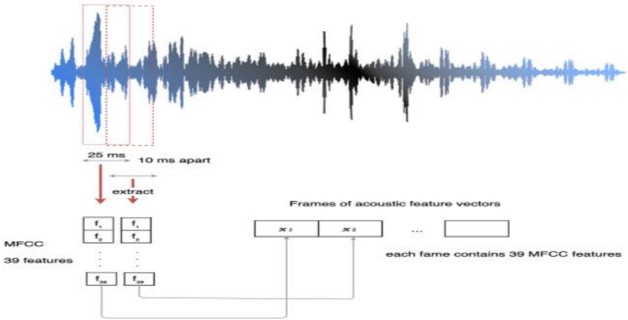

25 ms
10 ms apart
extract
Frames of acoustic feature vectors
MFCC
39 features
f1
f2
f39
x1
x2
...
each fame contains 39 MFCC features

Fig 2: MFCC Feature Extraction

B. Model Architecture

1) Artificial Neural Network (ANN):

The architecture of the Artificial Neural Network (ANN) designed for sound classification on the UrbanSound8k dataset is structured to efficiently process and classify audio features. The model comprises one input layer, two hidden layers, and one output layer. In the input layer, audio features extracted from the UrbanSound8k dataset are fed into the network. These features could include parameters derived from the audio signal, such as Mel-frequency cepstral coefficients (MFCCs) or spectrogram data.

The two hidden layers, essential for capturing intricate patterns within the audio data, employ Rectified Linear Unit (ReLU) activation functions. ReLU is chosen for its ability to introduce non-linearity into the model, aiding in the extraction of complex representations from the input features. The ReLU activation function replaces negative values with zero, mitigating the vanishing gradient problem and allowing the model to effectively learn and adapt during training. The output layer is designed with the number of nodes corresponding to the different sound categories within the UrbanSound8k dataset. The output layer generates predictions based on the learned features from the hidden layers, assigning probabilities to each sound class. During training, the model adjusts its weights and biases to minimize the difference between predicted and actual labels using a suitable loss function.

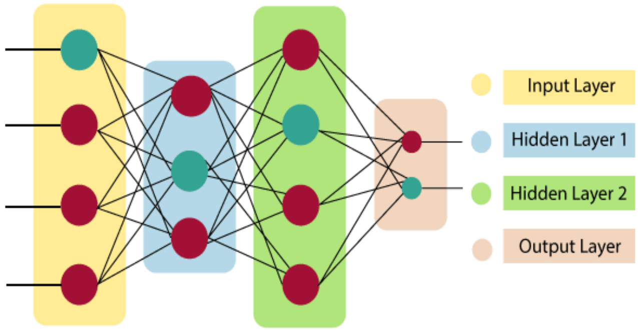


Fig 3 ANN Architecture

This ANN architecture aims to optimize sound classification performance by learning hierarchical features from the input data through the hidden layers and providing accurate predictions through the output layer. The choice of ReLU activation functions contributes to the model's capacity to capture complex patterns, enhancing its ability to discern various sound classes in the UrbanSound8k dataset.

To address the task of sound classification, the categorical cross-entropy loss function is utilized, complemented by ReLu activation in the output layer [17]. This combination aids in minimizing the losses associated with the ANN, particularly in the context of categorizing various sound classes. The ReLu activation function normalizes the output values, converting them into probabilities, and facilitating the efficient assignment of the model's predictions to different sound categories.

This architecture, coupled with the Adam optimizer, learning rate, categorical cross-entropy loss, and SoftMax and ReLu activation, collectively contributes to the effective training of the ANN model on sound data, enabling it to make accurate predictions and classifications.

2) Convolution Neural Network:

In the training process of our Convolutional Neural Network (CNN) model for sound classification using a prototype dataset featuring 10 labeled classes, we leverage the efficiency of the Adam optimization algorithm [15]. The architectural design incorporates five 3 x 3 convolutional layers, each seamlessly paired with a 2 x 2 pooling layer, facilitating the learning of distinctive features. To transform the original sequential audio data into sequential spectrograms, which are treated as images to optimize the CNN's effectiveness, we extract Mel-frequency cepstral coefficients (MFCCs) as key features.

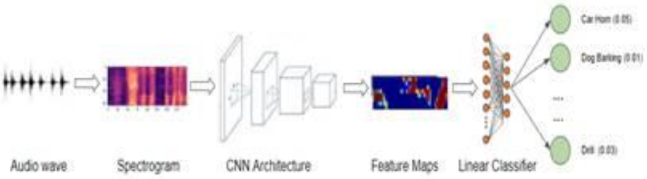


Fig: 4 CNN

To mitigate overfitting risks, a dropout technique with a probability of 0.2 is implemented during the training cycles. Rectified Linear Units (ReLU) are chosen for their effectiveness in modeling non-linear outputs from the layers [8]. The specific learning rate coefficient utilized in this context is set to 0.0001, a value carefully chosen to balance the trade-off between precision and training speed. The CNN's loss function is formulated using categorical cross-entropy in conjunction with softmax activation [17], well-suited for multi-class classification tasks. This approach quantifies the dissimilarity between the predicted probability distribution and the actual distribution of sound classes, providing a robust measure of model performance.

The softmax activation function is applied to the output layer, transforming the network's raw predictions into probability distributions across multiple classes.

To visualize the architecture of our CNN model. The diagram illustrates the arrangement of various layers, including convolutional layers, pooling layers, and possibly other components such as dropout layers. These layers collaboratively capture and hierarchically learn features from the sequential spectrograms, allowing the CNN to discern patterns and make accurate predictions for different sound categories.

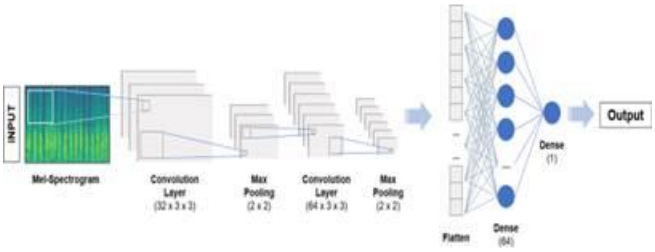


Fig 5: Model Architecture

3) Recurrent Neural Architecture RNN:

"In the training process of our Recurrent Neural Network (RNN) model for sound classification, specifically designed for a prototype dataset featuring 10 labeled classes and incorporating the temporal aspects of audio data, Long Short-Term Memory (LSTM) units play a crucial role. LSTMs enable the model to store and retrieve information, making them particularly effective in handling sequential data, essential for sound classification tasks.

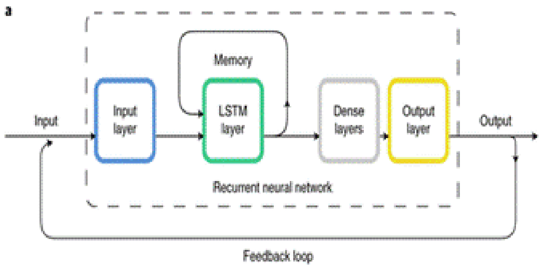


Fig 6: LSTM

To bolster the robustness of the RNN architecture during the training cycles, a dropout operator with a coefficient of 0.5 is employed. Dropout, a regularization technique, plays a vital role in preventing overfitting by randomly deactivating a fraction of neurons, encouraging the network to learn more robust and generalized representations.

The proposed RNN system is comprised of two RNN layers followed by four Rectified Linear Units (ReLU) [12]. ReLU activation functions introduce non-linearity into the model, assisting the network in learning complex patterns in the data. This architectural design is meticulously crafted to mitigate overfitting risks, striking a balanced trade-off between model complexity and generalization.

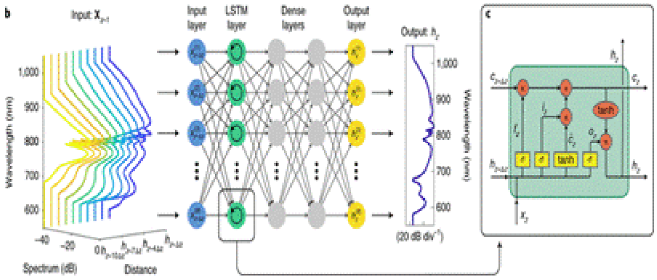


Fig 7: RNN Model Architecture

Similar to the CNN system, during the training of the datasets, categorical cross-entropy is utilized in conjunction with softmax activation. Categorical cross-entropy proves effective for classification tasks, offering a clear measure of dissimilarity between predicted and actual class distributions. The softmax activation function normalizes the output, ensuring that the model's predictions represent probability distributions across different sound classes.

For a visual representation of the RNN architecture and its application to the sound classification task, refer to Figure 4. The diagram provides insights into the organization of RNN layers, ReLU units, and their interconnections, showcasing how the network processes sequential data, particularly the MFCC features, for effective sound classification."

V. EXPERIMENTAL ANALYSIS

A. *Dataset*

"Prototype Dataset comprises a curated selection of sound excerpts, totaling 8,732 samples, with each lasting approximately 4 seconds. These samples have been meticulously categorized into 10 distinct classes: air conditioner, car horn, children playing, dog bark, drilling, engine idling, gunshot, jackhammer, siren, and street music. The diversity of sound categories in the Prototype Dataset allows for a comprehensive evaluation of the model's ability to discern and classify various real-world sounds.

Each sound recording in the Prototype Dataset was captured at a 44.1 kHz sampling rate, ensuring high-quality representations of the audio content. It is important to note that while this study utilized the Prototype Dataset for initial training cycles, the intention is to later transition to a customized dataset tailored to specific requirements. The model will be further trained on an extended range of sounds to enhance its adaptability to a broader spectrum of auditory stimuli. This shift aims to create a more specialized and effective model for the targeted application."

B. *Model Approach*

In crafting our sound classification model with the dynamic duo of TensorFlow and Keras, we embraced a holistic strategy that amalgamated the robust capabilities of these cutting-edge deep learning frameworks. TensorFlow, serving as the engine behind the scenes, delivered the foundational support needed for scalable and efficient neural network development. Meanwhile, Keras, a high-level API seamlessly integrated with TensorFlow, provided an expressive and user-friendly interface for the design and implementation of our intricate sound classification architecture.

The architectural blueprint of our model was meticulously tailored to accommodate the inherent sequential nature of audio data. We strategically integrated Convolutional Neural Network (CNN) layers to adeptly capture spatial features from sequential spectrograms. Concurrently, Recurrent Neural Network (RNN) layers, particularly Long Short-Term Memory (LSTM) units, were enlisted to capture temporal dependencies within the audio sequences. This thoughtful fusion of CNN and RNN layers endowed our model with a versatile capacity to discern both spatial and temporal intricacies, elevating its proficiency in extracting complex patterns from sound data.

For the training regimen, we curated a diverse dataset trifecta, encompassing UrbanSound8K, ESC-50, and FSDKaggle2018. These datasets encapsulated a rich spectrum of sound categories, ensuring the model's adaptability and generalization across an array of real-world auditory scenarios. The training process entailed meticulous parameter optimization, facilitated by the Adam optimizer with finely tuned learning rates. The categorical cross-entropy loss function coupled with SoftMax activation was employed for effective multi-class classification.

Throughout implementation, the streamlined expressiveness of Keras expedited prototyping and experimentation, while TensorFlow's robust backend seamlessly handled efficient computations and hardware acceleration integration. The resulting sound classification model, an orchestration of TensorFlow and Keras, not only showcased accuracy in categorizing diverse sound classes but also underscored the collaborative prowess of these two frameworks in pushing the boundaries of deep learning applications.

C. System FlowChart:

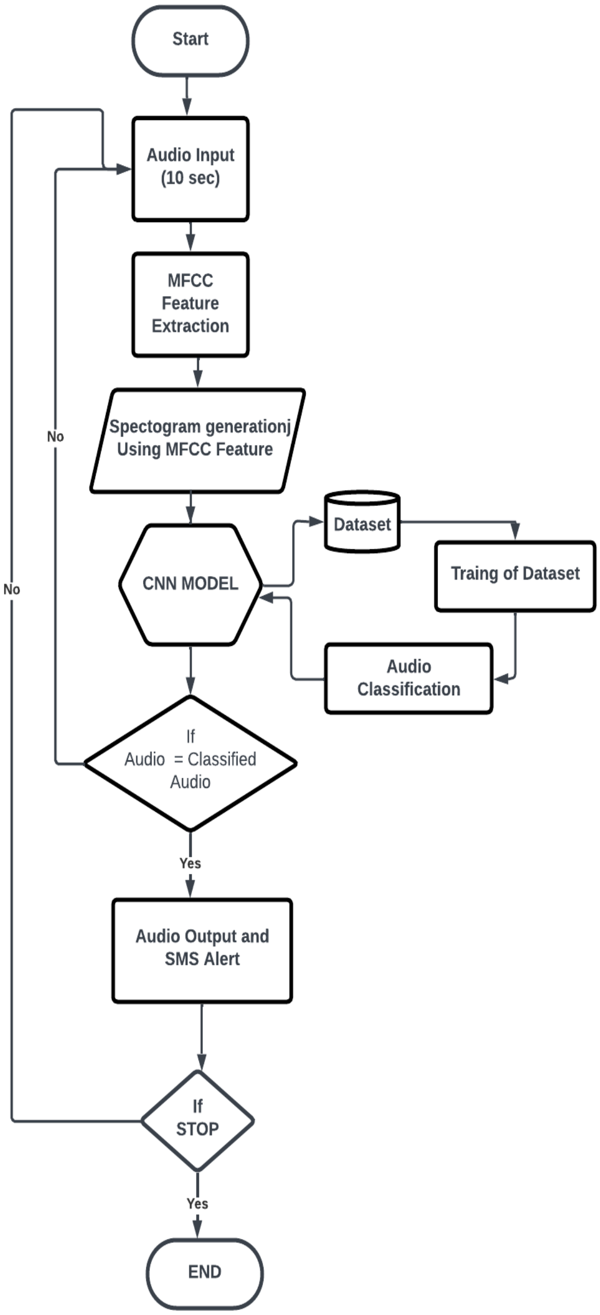


Fig 8: Flow Chart

D. Application Interface:

To seamlessly deploy our application, we leverage the power of Streamlit—an open-source Python library specifically crafted for the effortless creation of web applications. With a focus on simplicity, Streamlit facilitates the transformation of our sound classification system into a user-friendly app that can be run on various machines. This library excels at turning data scripts into shareable web applications, offering an intuitive interface that allows us to build interactive dashboards without the need for extensive web development expertise. Integrating our sound classification model with Streamlit ensures a smooth deployment process, enabling others to effortlessly interact with our project through a web browser. Streamlit's streamlined syntax and robust deployment capabilities make it the optimal choice for converting our sound classification model into a deployable and accessible application, showcasing its prowess in democratizing web development for data-driven projects.

E. Intelligent Classification and Notification System:

The system outputs highly accurate audio class classifications, serving as the foundation for subsequent actions. Leveraging a robust Text-to-Speech module using the gTTS (Google Text-to-Speech) library in Python simplifies the conversion of text to speech. Developed by Pierre-Nick Durette, it utilizes Google Translate's API to generate spoken audio files in various languages. With a user-friendly interface, gTTS is a concise and effective tool for integrating text-to-speech capabilities into Python applications. It converts the identified audio class into spoken words for real-time vocalization. Simultaneously, an SMS Alert module interfaces with a messaging service using Twilio API. Twilio provides a set of APIs (Application Programming Interfaces) that allows developers to easily add communication features to web and mobile applications without the need for complex telecommunications infrastructure. Twilio's services include SMS (Short Message Service) and MMS (Multimedia Messaging Service) messaging, voice calls, video calls, and more. Developers can use Twilio to build communication features, such as two-factor authentication, notifications, customer support, and interactive voice response systems. Twilio's platform is widely used across various industries for creating seamless and personalized communication experiences within applications. promptly notifying users about the detected audio class. This seamless integration ensures users receive timely and accessible information about their environment.

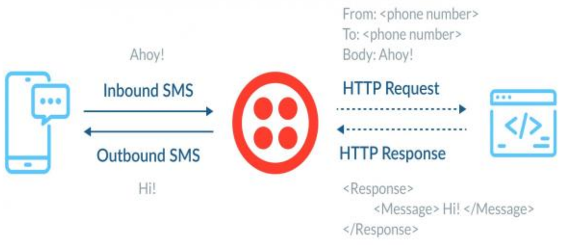
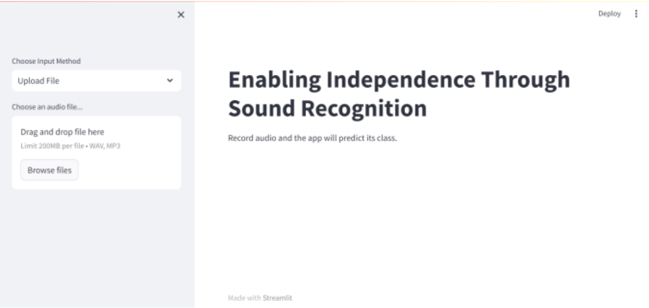


Fig 9: Twilio API

VI. Result

A. Result



Choose Input Method

Upload File

Choose an audio file...

Drag and drop file here

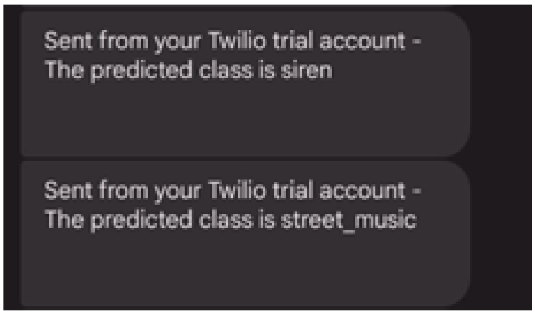
Limit 200MB per file • WAV, MP3

Enabling Independence Through Sound Recognition

Record audio and the app will predict its class.

Made with Streamlit

Fig 10: Application page



Sent from your Twilio trial account - The predicted class is siren

Sent from your Twilio trial account - The predicted class is street_music

Fig 11: SMS Alert

B. Comparative Result:

In the exclusive context of utilizing the UrbanSound8K dataset and implementing the model approach with TensorFlow and Keras, a consistent trend emerges, showcasing the Convolutional Neural Network (CNN) model's superiority over its Recurrent Neural Network (RNN) counterpart and Artificial Neural Network. This aligns seamlessly with CNNs' well-established prowess in excelling at classification tasks within the realm of computer vision [25]. The inherent challenges of optimizing RNNs, marked by issues like vanishing and exploding gradients, coupled with extended training times [30], underscore the pragmatic preference for CNNs in the nuanced landscape of sound classification.

Model	Dataset	Architecture	Accuracy (%)		Loss(%)	
			Train	Test	Train	Test
TensorFlow	Urban sound 8k	ANN	92.56	81.2	17.43	33.44
		CNN	96.44	76.5	22.34	22.33
		RNN	94.33	71.3	23.67	11.35
TensorFlow and Keras	Urban sound 8k	ANN	95.66	62.44	44.34	9.09
		CNN	98.77	55.67	23.33	11.65
		RNN	96.33	59.44	21.43	25.78
Pytroch	Urban sound 8k	ANN	93.44	61.22	34.65	23.32
		CNN	96.66	51.44	42.33	29.80
		RNN	92.44	55.43	44.21	22.55

Table 1: Tabulated Result

Notably, the CNN model consistently delivered higher accuracy results across all datasets. While occasional instances displayed elevated testing loss values, this nuanced behavior is inherent to the categorical cross-entropy nature of the loss function [21]. It is crucial to recognize that the output values denote probabilities for each labeled class, with the model's accuracy contingent on selecting the class with the highest probability for a given sample. This intricate dance between the loss function and accuracy highlights the model's acute sensitivity to any probability values linked with misclassified labels, leading to a consequential penalty.

VII. Conclusion

In conclusion, the project "Enabling Independence through Sound Classification" utilizes deep learning and Mel-frequency cepstral coefficients to empower individuals who are deaf and blind. With a focus on breaking down communication barriers, improving safety, and fostering community, the project demonstrates the potential of technology for positive social impact. The research identifies the CNN architecture, specifically the TensorFlow and Keras models, as highly effective, achieving a peak accuracy of 98.77% on the UrbanSound8k dataset. Future directions include exploring multitask learning and refining techniques for isolating specific audio elements, contributing to advancements in audio processing and classification. The study not only highlights performance disparities between CNN and RNN but also sets the stage for ongoing research on sound classification, detection, and noise mitigation using advanced neural network modifications, aligning to create a more inclusive and equitable future through innovative technology.

REFERENCE

[1] ——, ESC: Dataset for Environmental Sound Classification, version V2, 2015. DOI: 10.7910/DVN/YDEPUT. [Online]. Available: ht tps://doi.org/10.7910/DVN/ YDEPUT.

[2] H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," Jan. 2009, pp. 1096–1104.

[3] P. Mahana and G. Singh, "Comparative analysis of machine learning algorithms for audio signals classification," International Journal of Computer Science and Network Security (IJCSNS), vol. 15, no. 6, p. 49, 2015

[4] a. Salamon, J., Jacoby, C., & Bello, J. P. (2014). A dataset and taxonomy for urban sound research. In 22nd ACM international conference on Multimedia (pp. 1041-1044).

[5] K. J. Piczak, "Environmental sound classification with convolutional neural networks," in 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), IEEE, 2015, pp. 1–6.

[6] b. Salamon, J., & Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. IEEE Signal Processing Letters, 24(3), 279-283

[7] a. Gemmeke, J. F., Ellis, D. P., et al. (2017). Audio set: An ontology and human-labeled dataset for audio events. In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 776-780).

[8] M. S. Imran, Safwatimran/audio-classification. [Online]. Available: ht tps://github.com/Safwat Imran/Audio-Classification.

[9] J. Lyons, Mel frequency cepstral coefficient (mfcc) tutorial, http://practicalcryptography.com/miscellaneous/ machine - learning/guide - mel - frequency - cepstral - coefficients-mfccs/, Published Date is not specified

[10] V. Boddapat I, A. Peter, J. Rasmusson, and L. Lundberg, "Classifying environmental sounds using image recognition networks," Procedi Computer Science, vol. 112, pp. 2048–2056, 2017, Knowledge-Based and Intelligent Information Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France, ISSN: 1877-0509.DOI: https://doi.org/10.1016/j.procs.2017.08.250. [Online]. Available: http://www.sciencedirect .com/science/article/pii/S1877050917316599.

[11] A. Khamparia, D. Gupta, N. G. Nguyen, A. Khanna, B. Pandey, and P. Tiwari, "Sound classification using convolutional neural network and tensor deep stacking network," IEEE Access, vol. 7, pp. 7717–7727, 2019.

[12] A. Bhatnagar, R. Sharma, and R. Kumar, "Analysis of hamming window using advance peak windowing method," Jul. 2012.

[13] P. Mahana and G. Singh, "Comparative analysis of machine learning algorithms for audio signals classification," International Journal of Computer Science and Network Security (IJCSNS), vol. 15, no. 6, p. 49, 2015.

[14] A. H. Mansour, G. Z. A. Salh, and K. A. Mohammed, "Article: Voice recognition using dynamic time warping and melfrequency cepstral coefficients algorithms," International Journal of Computer Applications, vol. 116, no. 2, pp. 34–41, Apr. 2015, Full text available. https//ieeexplore.ieee.org/document/917701

[15] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://doi.org/10.1162/neco.1997.9.8.1735. [Online]. Available:

[16] a. Vaseghi, S. V. (2008). Advanced Signal Processing and Digital Noise Reduction. John Wiley & Sons.

[17] S. Adams, Audio-classification (paper version), https: //github.com/seth814/Audio-Classification, Apr. 2020.

[18] J. Bilmes, "Gaussian models in automatic speech recognition," in. Jan. 2009, pp. 521–555. DOI: 10.1007/9780-387-30441-0_29.

[19] H. Lee, P. Pham, Y. Largman, and A. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," Jan. 2009, pp. 1096–1104.

[20] P. Mahana and G. Singh, "Comparative analysis of machine learning algorithms for audio signals classification," International Journal of Computer Science and Network Security (IJCSNS), vol. 15, no. 6, p. 49, 2015.