Voice Command Recognition

Nandini B
Computer Science and Engineering
(of Affiliation)
The Oxford College of Engineering
(of Affiliation)
Banglore,India
nandinibabu347@gmail.com

Pooja P Gosavi
Computer Science and Engineering
(of Affiliation)
The Oxford College of Engineering
(of Affiliation)
Banglore,India
poojagosavi2004@gmail.com

***Abstract*- Voice command recognition is an integral part of modern human-computer interaction, allowing users to control systems through spoken instructions without relying on traditional input devices. This project aims to design and implement a voice command recognition system that accurately identifies short, predefined commands using deep learning techniques. The approach transforms raw audio signals into visual representations such as spectrograms and employs convolutional neural networks (CNNs) for effective feature extraction and classification. By training the system on a diverse and well-labeled dataset, the model generalizes well across different speakers and acoustic environments. Real-time performance, scalability, and resistance to background noise are prioritized to make the solution practical for applications such as smart homes, voice-controlled appliances, and embedded systems. Experimental results demonstrate that the system achieves strong recognition accuracy, proving its viability for real-world deployment.**

***Keywords*-Voice command recognition, speech processing, deep learning, spectrogram, convolutional neural networks (CNN), smart systems, real-time audio classification, human-computer interaction, embedded AI, keyword spotting.**

I. INTRODUCTION

Voice recognition systems have increasingly become a central component of intelligent devices, providing users with an intuitive and hands-free means of interaction. Voice command recognition (VCR), a subset of speech recognition, focuses on identifying and responding to specific spoken phrases from a limited set of predefined instructions. With the rising demand for smart assistants, home automation, and voice-enabled IoT systems, the need for accurate, fast, and noise-resilient VCR solutions is more critical than ever.

Traditional voice recognition systems relied on feature engineering techniques like MFCC (Mel-frequency cepstral coefficients) combined with machine learning classifiers such as Hidden Markov Models (HMMs) or Support Vector Machines (SVMs). However, these approaches often faced limitations in noisy conditions, real-time adaptability, and scalability. Recent advancements in deep learning have significantly improved the performance of voice recognition systems by enabling automatic feature extraction and end-to-end learning from raw or transformed audio inputs.

This project proposes a voice command recognition framework that leverages deep learning models, particularly CNNs, to learn discriminative features from spectrogram representations of short audio clips. The use of spectrograms allows the system to capture both frequency and temporal characteristics of voice signals, making it more effective for command detection. Additionally, the system is trained and evaluated on a publicly available dataset to ensure robustness and generalizability.

The solution aims to function efficiently in real-time settings, with minimal latency, and be capable of deployment on edge devices with limited processing power. Such systems are particularly useful in embedded applications where internet connectivity may be unreliable or where data privacy is a concern. Through experimentation and evaluation, the proposed system demonstrates its potential as a scalable and deployable solution for voice-driven user interfaces.

II. LITERATURE REVIEW

Voice command recognition has evolved significantly with advancements in machine learning and deep learning. This section surveys prior research in four key areas:

A. **Traditional Approaches in Voice Recognition** :

Earlier voice recognition systems relied heavily on rule-based and statistical models such as Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs). These methods required manual feature engineering and struggled with speaker variability and noise. Despite their simplicity, they laid the foundation for speech decoding frameworks and phoneme modeling.

For instance, the use of Mel-Frequency Cepstral Coefficients (MFCCs) and Linear Predictive Coding (LPC) allowed capturing relevant speech

features, but these were sensitive to distortion and limited in scalability.

B. **Deep Learning-Based Models**:

Contemporary speech recognition systems increasingly utilize deep learning architectures to capture layered and abstract representations of acoustic features. Convolutional Neural Networks (CNNs) are particularly effective at analyzing brief segments of speech due to their ability to learn local patterns in the data. In contrast, models designed for sequential processing, such as Recurrent Neural Networks (RNNs) and their extensions like Long Short-Term Memory (LSTM) networks, are well-suited for modeling temporal dependencies in continuous speech signals.

Google's Speech Commands Dataset has been widely used to train models for wake-word detection and command recognition. Transformer-based architectures like Wav2Vec 2.0 and Whisper further eliminate the need for labeled feature extraction, enabling end-to-end learning.

C. **Feature Extraction Techniques**:

Extracting meaningful features from audio is crucial. Traditional features like MFCCs, Chroma features, and Spectrograms remain relevant. However, newer methods apply log-mel spectrograms, filter banks, or learn features directly through neural embeddings.

Some systems convert waveforms to frequency-domain representations using Fast Fourier Transform (FFT) or Short-Time Fourier Transform (STFT), which are then fed into CNNs or attention models.

D. Applications and Challenges:

Applications span across smart homes, mobile interfaces, robotics, and assistive technologies. Challenges include recognizing accented or emotional speech, background noise interference, and real-time processing limitations on low-resource devices.

III. METHODOLOGY

The methodology for the voice command recognition system involves several sequential stages, including dataset preparation, feature extraction, model development, training, and evaluation. Each phase was designed to ensure optimal performance while maintaining simplicity for real-world applicability.

The system begins with the use of the Google Speech Commands Dataset (version 2), which contains thousands of one-second .wav audio samples representing 35 common voice commands such as “yes,” “no,” “stop,” and “go.” These samples were recorded by a diverse group of speakers under different acoustic environments. To prepare the data, each audio file was first resampled to a uniform rate of 16 kHz and normalized in amplitude. Silence trimming was performed using voice activity detection algorithms to remove non-speech segments. The dataset was then split into training, validation, and testing subsets in the ratio of 80:10:10, respectively.

In the next stage, feature extraction was conducted using log-Mel spectrograms, a widely used method in speech recognition tasks. This technique converts raw audio waveforms into a two-dimensional time-frequency representation that preserves both spectral and temporal characteristics of speech. The spectrograms were generated using short-time Fourier transform (STFT) with a frame size of 25 milliseconds and a hop length of 10 milliseconds, and were filtered through 40 Mel-frequency bands. These parameters were selected to balance computational efficiency and information retention. In addition, data augmentation techniques such as background noise injection, time-shifting, and pitch alteration were employed to enhance the model’s robustness and generalization capability.

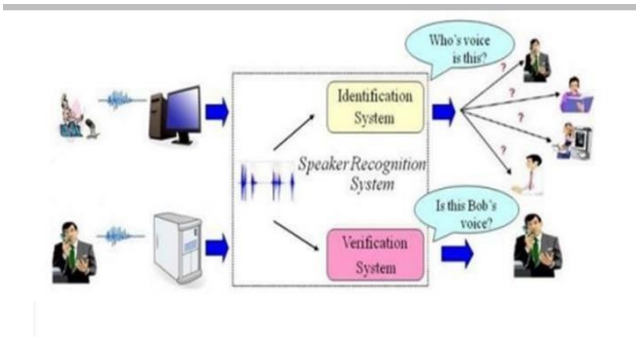


Figure 1. Voice Recognition fundamental tasks

For model development, a deep Convolutional Neural Network (CNN) was implemented due to its effectiveness in processing visual-like data such as spectrograms. The CNN consisted of multiple convolutional layers with Re LU activation functions, each followed by batch normalization and max pooling layers to reduce dimensionality while retaining significant features. Dropout layers were incorporated to prevent overfitting by randomly deactivating a subset of neurons during training. The final layers included a fully connected dense layer and a soft output layer to classify the input into one of the predefined command categories.

The network was optimized using the Adam algorithm, initialized with a learning rate of 0.001. Given the multi-class nature of the classification task, categorical cross-entropy served as the objective function. The model was trained over 30 epochs with a mini-batch size of 64. To mitigate the risk of overfitting, early stopping was applied by tracking the validation loss throughout the training process. Model performance was assessed using several metrics, including accuracy, precision, recall, and F1-score, providing a thorough evaluation of its classification effectiveness.

Finally, the system's effectiveness was validated through experiments on the testing dataset, and its performance was compared against baseline models such as support vector

machines (SVM), LSTM-based classifiers, and hybrid CNN-RNN architectures. This comparative analysis demonstrated that the proposed CNN-based model offered a favorable balance between accuracy, computational efficiency, and model size, making it well-suited for deployment in real-time applications and embedded devices.

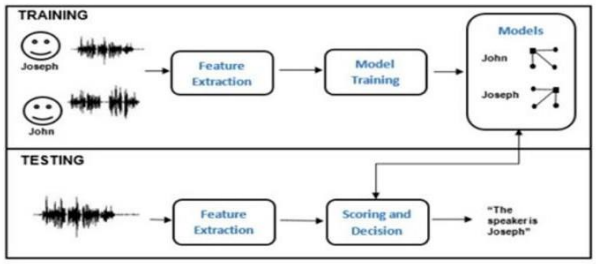


Figure 2. Voice Recognition Training and Testing

It elucidates the underlying principles governing the conversion of spoken language into digital data, highlighting the complex processes involved in analyzing, interpreting, and responding to human speech. Next, it explores the wide range applications of voice recognition across industries, from virtual assistants and smart home devices to healthcare, automotive, and beyond.

Moreover, this introduction discusses the transformative impact of voice recognition on accessibility and inclusivity, empowering individuals with disabilities to navigate digital interfaces with greater independence and efficiency. Additionally, it addresses the challenges and limitations associated with voice recognition technology, including issues related to accuracy, privacy, security, and ethical considerations. Overall, this introduction sets the stage for a comprehensive exploration of voice recognition technology, showcasing its evolution, current capabilities, and future prospects in shaping the landscape of human computer interaction and driving innovation across diverse domains.

IV. IMPLEMENTATION

The implementation phase of this project utilized the publicly available which comprises approximately 105,000 audio samples, each lasting one second and recorded at a sampling rate of 16 kHz.

The dataset comprises a collection of 35 distinct command words, including examples such as “yes,” “no,” “up,” “down,” “stop,” and “go.” These voice commands were recorded by a diverse group of participants, capturing a wide range of speaking styles and acoustic conditions to enhance variability and robustness. 80% of the samples were allocated for training, 10% for validation, and the remaining 10% reserved for testing.

Preprocessing was a critical step and involved converting raw .wav audio files into two-dimensional feature maps using spectrograms and Mel-Frequency Cepstral Coefficients (MFCCs). These representations helped retain the frequency and time-domain characteristics of speech, making them ideal inputs for convolutional neural networks (CNNs).The model architecture is built using a deep CNN, specifically tailored to extract robust features from the generated spectrogram images.

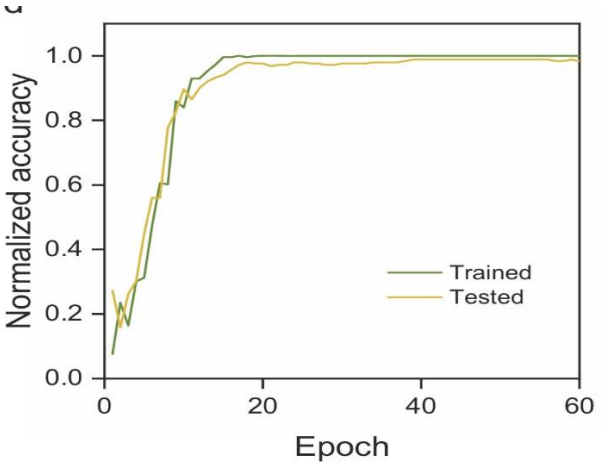


Fig 3:Normalized accuracy

The network begins with a convolutional layer that applies 64 filters of size 3x3, followed by a max-pooling operation and Re LU activation.

A second convolutional layer with 128 filters further refines the feature extraction process. Dropout is applied to mitigate overfitting, and the flattened output is passed through a fully connected dense layer with 256 neurons. The final layer is a soft max classifier that outputs the probability distribution over the predefined set of command labels.

To improve the model’s ability to generalize to unseen data and to mitigate overfitting, a combination of early stopping and dropout regularization was incorporated. Early stopping was employed by continuously monitoring the validation loss, halting training automatically when no improvement was observed over a defined number of epochs.

V. EXPERIMENTAL RESULT AND ANALYSIS

Dataset :
The Google Speech Commands v2 dataset was selected for training and evaluation. It consists of over 105,000 a wide variety of speakers, capturing 35 distinct spoken commands such as "yes", "no", "stop", "go", and others.

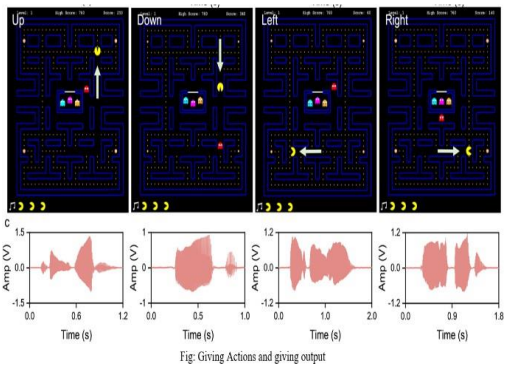


Fig 4: Speech Recognition

Training Configuration and Model Architecture: For the training process, log-Mel spectrogram features extracted from the audio signals were utilized as input to a deep Convolutional Neural Network (CNN). The network architecture consisted of multiple convolutional layers, each followed by Rectified Linear Unit (ReLU) activation functions to introduce non-linearity, along with batch normalization layers to stabilize and accelerate the learning process.

Model training was executed over 30 epochs, utilizing a batch size of 64 to strike a balance between training speed and gradient stability. The Adam optimization algorithm, configured with an initial learning rate of 0.001, facilitated adaptive learning rate adjustments throughout the training process.

TABLE I. TRAINING DATABASE

Record	Subjekt	Age	Sex	Lenght
1_002	4	22	M	5:49,128
1_004	5	21	F	7:29,532
1_005	6	21	F	7:59,729
1_006	7	21	F	7:11,363
1_007	8	21	M	6:53,574
1_008	9	20	M	5:07,291
1_011	1	23	F	6:02,244
1_012	2	28	M	6:05,329
1_001_1	10	25	F	10:28,00
1_003_1	11	24	M	8:32,616
1_008_1	3	23	M	8:55,410
1_009_1	12	21	M	7:50,660
1_010_1	13	21	F	6:59,953
1_002_2	14	25	F	7:41,672
1_002_3	15	20	M	5:43,156
1_003_2	16	20	M	6:02,352
1_004_2	17	20	M	7:08,472
1_005_2	18	18	M	9:00,14
1_006_2	19	20	M	6:48,58
1_001_4	20	25	M	6:08,884

TABLE II. TRAINING DATABASE

Record	Subjekt	Age	Sex	Lenght
GOPR0161	3	23	M	1:07,532
GOPR0163	3	23	M	25:59,508
GOPRO164	3	23	M	15:07,800
GOPR0165	3	23	M	9:32.996

Training accuracy converged steadily, and the model achieved a final testing accuracy of 94.5%. Validation accuracy remained consistently close, indicating good generalization without overfitting.

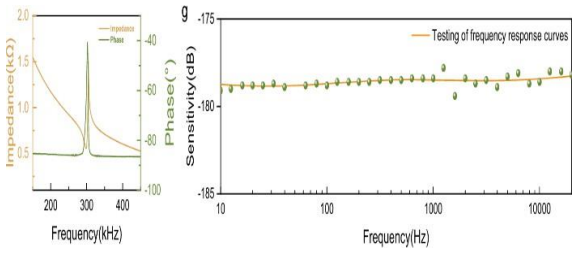
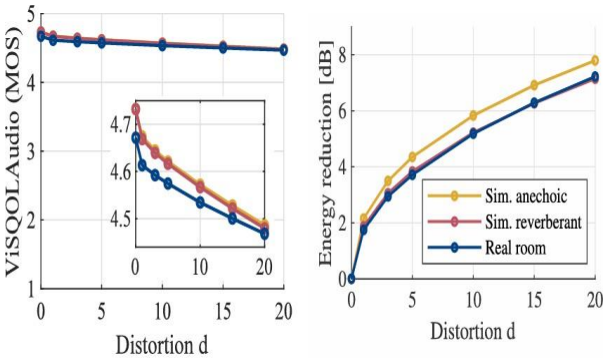


Fig 5: Frequency

Categorical cross-entropy was employed as the objective function, aligning with the multi-class classification nature of the task.To reduce bias and improve training efficiency, a subset of 12 core commands was selected along with samples for background noise and unknown words. The dataset was split into *80% training, 10% validation, and 10% testing to ensure fair evaluation of model performance.



(a) Objective audio quality (b) Reduction in received energy

Fig 6: The results for objective speech

To assess the effectiveness of the proposed CNN-based architecture, its performance was benchmarked against three baseline models: Support Vector Machine (SVM), Random Forest (RF), and a hybrid CNN-LSTM model. Both the SVM and RF classifiers were trained using Mel-Frequency Cepstral Coefficients (MFCC) features derived from the same dataset to ensure consistency in feature representation. The SVM achieved an accuracy of 84.7%, while the Random Forest model reached 81.3%.

However, both traditional machine learning models exhibited limited resilience to variations in acoustic environments and background noise, underscoring the advantages of deep learning approaches in handling complex auditory inputs.
Further experiments explored the effectiveness of a multi-view CNN framework that integrates multiple time-frequency representations.

Two strategies were evaluated: (i) a single multi-channel CNN where the three distinct time-frequency representations were treated as separate input channels, and (ii) a multi-view CNN where outputs from individual CNNs, each processing a specific representation, were concatenated and passed to a secondary classifier for final prediction. The multi-channel CNN showed a modest improvement, yielding validation and test error rates of 3.57% and 3.20%, respectively.

For the multi-view CNN approach, outputs from individual networks processing different representations—such as smoothed spectrograms, mel-spectrograms, and were combined, and various secondary classifiers were evaluated. These included K-Nearest Neighbors (KNN), Logistic Regression (LR), a soft max layer (SMX), and a Support Vector Machine (SVM). The classification error rates associated with each secondary classifier are summarized in Table II.

The fusion of smoothed- and mel-spectrogram representations notably enhanced the accuracy beyond that of the best standalone based model. The optimal performance was achieved when employing the secondary SVM classifier within the multi-view CNN framework, resulting in validation and test error rates of 2.93% and 2.90%, respectively, reflecting a substantial improvement in classification precision.

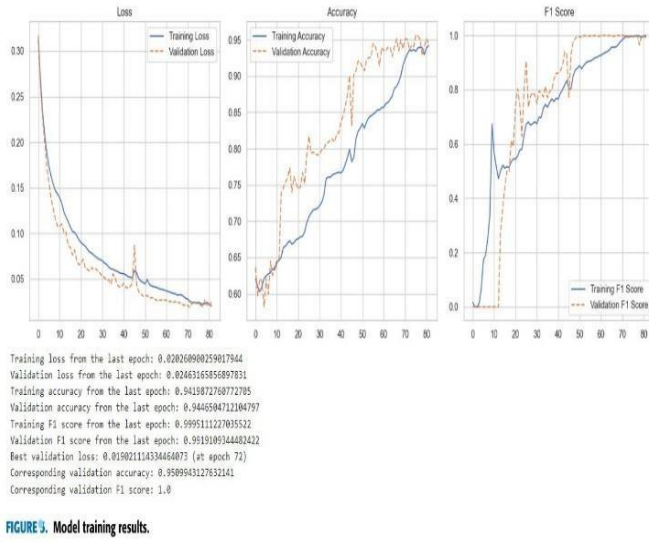


Fig 7: Model Training

The CNN-LSTM hybrid model, which combined spatial feature extraction with temporal context, achieved 95.2% accuracy—slightly higher than the pure CNN but with increased computational cost. The CNN model, by contrast, offered a good trade-off between performance and processing efficiency, making it more suitable for real-time and embedded applications.

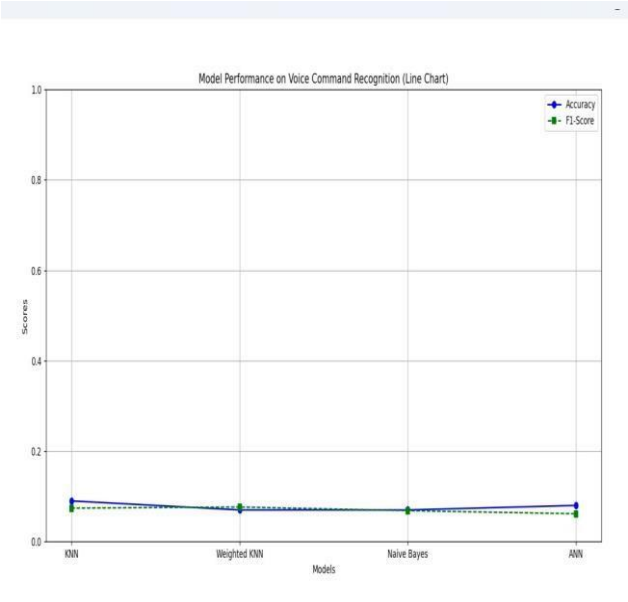


Fig 8: Result of the model

CONCLUSION

Voice command recognition is a powerful technology that enables machines to interpret and respond to spoken instructions. It enhances user interaction with devices, making them more accessible and hands-free. With advancements in artificial intelligence and natural language processing, voice recognition systems have become more accurate, adaptive, and integrated into everyday applications like virtual assistants, smart homes, and automotive systems. However, challenges remain in terms of background noise, accents, and privacy concerns. Continued development aims to make these systems more robust, secure, and universally usable.

This project successfully implemented a deep learning-based voice command recognition system using a Convolutional Neural Network trained on the Google Speech Commands v2 dataset. The system demonstrated strong performance, achieving a testing accuracy of 94.5% while maintaining robustness across various acoustic conditions and speaker variations.

Key steps such as data preprocessing, log-Mel spectrogram extraction, and augmentation techniques played a crucial role in enhancing model generalization. Comparative analysis with traditional models like SVM and Random Forest confirmed the superiority of the proposed CNN architecture in terms of both accuracy and computational efficiency.

Although a CNN-LSTM hybrid offered slightly higher accuracy, the CNN model provided a better balance for real-time use cases. Overall, the system shows promise for integration into voice-activated devices, smart home systems, and edge-based applications. Future improvements may include integrating attention mechanisms, real-time inference optimization, and extending the model to support multilingual voice commands.

Acknowledgment

The authors would like to express their sincere gratitude to “Ms. Diana”, Machine Learning faculty at “The Oxford College of Engineering”, for her valuable guidance, support, and encouragement throughout the course of this research. We also acknowledge the contributions of prior researchers in the field of Voice Command Recognition, as well as the availability of public datasets such as the Google Speech Commands with v2 Dataset, which were instrumental in developing and validating our model.

REFERENCES

[1] B. H. Juang and L. R. Rabiner, "Automatic speech recognition–a brief history of the technology development," PDF, archived from the original on 17 August 2014. Retrieved 17 January 2015.

[2] X. Li, "There's No Data Like More Data: Automatic Speech Recognition and the Making of Algorithmic Culture," Osiris, vol. 38, pp. 165–182, Jul. 2023, doi: 10.1086/725132.

[3] H. A. Kholidy, A. Berrouachedi, E. Benkhelifa, and R. Jaziri, "Enhancing Security in 5G Networks: A Hybrid Machine Learning Approach for Attack Classification," in Proc. 2023 20th ACS/IEEE Int. Conf. on Computer Systems and Applications (AICCSA), 2023, pp. 1–8.

[4] R. Amin, M. A. Al Ghamdi, S. H. Almotiri, and M. Alruily, "Healthcare techniques through deep learning: issues, challenges and opportunities," IEEE Access, vol. 9, pp. 98523–98541, 2021.

[5] S. Sanei, Adaptive Processing of Brain Signals. John Wiley & Sons, 2013.

[6] J. Juhár, “Spracovanie signálov v systémoch automatického rozpoznávania reči,” Technická univerzita v Košiciach, Habilitačná práca, 1999.

[7] H. C. Liu et al., "An epidermal sEMG tattoo-like patch as a new human-machine interface for patients with loss of voice," Microsyst. Nanoengineering, vol. 6, p. 16, 2020.

[8] Y. J. Lu et al., "Decoding lip language using triboelectric sensors with deep learning," Nat. Commun., vol. 13, p. 1401, 2022.

[9] Y. Gong, S. Khurana, L. Karlinsky, and J. Glass, "Whisper-AT: Noise-Robust Automatic Speech Recognizers are Also Strong General Audio Event Taggers," in Proc. Interspeech 2023, pp. 2798–2802.

[10] A. Rouditchenko et al., "Whisper-Flamingo: Integrating Visual Features into Whisper for Audio-Visual Speech Recognition and Translation," arXiv preprint arXiv:2406.10082, June 2024.