SpeakSense – Final Project Report

Machine Learning CSCI 6364 – Prof. Sardar Hamidian (sardar@gwu.edu), Prof. Armin Mehrabian (armin@gwu.edu)
Abde Manaaf Ghadiali – G29583342, Gehna Ahuja – G35741419, Venkatesh Shanmugam – G27887303
Submission Date – 3rd May 2024, ***Project Link*** *– SpeakSense – AI Language Detection Tool (GitHub.com)*

Abstract

Language is vital for communication, bridging individuals, communities, and cultures. Accurately identifying diverse languages is crucial in our interconnected world. **Artificial intelligence (AI)** offers powerful solutions for language detection and understanding. This paper introduces "**SpeakSense**," an AI tool for identifying languages from **audio data**. It discusses the importance and challenges of language detection in audio content, detailing SpeakSense's architecture and methods. Through experimentation, it demonstrates SpeakSense's effectiveness in identifying languages across contexts. It explores practical applications such as transcription and speech analytics, addressing ethical considerations. Speak Sense signifies a significant advancement in AI-driven language detection, facilitating effective communication in our multilingual world.

Introduction

Language detection from audio recordings stands as a critical task in enabling effective communication and comprehension across linguistic barriers. This project endeavors to develop a robust and precise system tailored for this purpose, leveraging cutting-edge **machine learning** and **deep learning** techniques alongside **signal processing** methodologies. By accurately identifying spoken languages from diverse audio inputs, including varying accents, dialects, and environmental conditions, the system aims to facilitate practical applications in speech recognition, transcription, translation, and beyond.

In pursuit of this objective, the project draws upon diverse and representative datasets sourced from **Kaggle**, encompassing a broad spectrum of languages, accents, dialects, and speaking styles. These datasets include the "**Audio Dataset with 10 Indian Languages**," featuring Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Punjabi, Tamil, Telugu, and Urdu, as well as the "**Spoken Language Identification**" dataset, comprising English, Spanish, and German recordings. Additionally, a test (holdout) set comprising audio samples collected from friends, family, and faculty members serves to demonstrate live prediction capabilities.

To manage computational constraints effectively, the project plans to utilize 120,000 audio files for **deep neural network (DNN)** and **classical machine learning models**, while restricting the dataset to 12,000 audio files for **convolutional neural network (CNN)** models. Through the integration of these diverse datasets and computational strategies, the project endeavors to develop a robust and scalable language detection system capable of meeting the demands of real-world applications across linguistic boundaries.

Previous Work

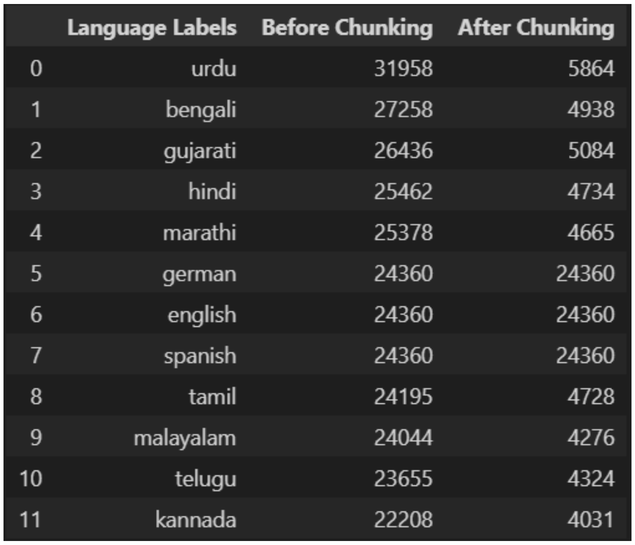
Several research endeavors have contributed significantly to the field of language detection from audio recordings, advancing methodologies, techniques, and applications. **[Juang & Rabiner, 1990][1]** presents a comprehensive exploration of various techniques for language identification in speech signals. The authors compare different approaches, including template matching, hidden Markov models (HMMs), and Gaussian mixture models (GMMs), evaluating their performance across multiple languages and conditions. **[Reynolds, Rose, & Lyu, 1995][2]** propose a neural network-based approach for language identification from speech signals. They investigate the efficacy of different neural network architectures and training strategies, demonstrating the potential of neural networks in accurately discerning languages from audio inputs. **[Waibel & Lee, 1990][3]** provides a comprehensive overview of the state-of-the-art techniques in multilingual speech recognition. The authors discuss various methodologies, including language modeling, acoustic modeling, and feature extraction, highlighting their applications and challenges in achieving accurate language detection and recognition. **[Bahdanau et al., 2016][4]** explore the application of deep learning models, specifically recurrent neural networks (RNNs) and convolutional neural networks (CNNs), for language recognition tasks. They investigate different network architectures and training strategies, demonstrating the effectiveness of deep learning in automatically extracting language features from audio signals. **[Cavnar & Trenkle, 1994][5]** while not focused exclusively on audio data, is relevant for its exploration of language identification from short textual inputs. The authors investigate various statistical and machine learning approaches for automatically identifying the language of short texts, laying the groundwork for similar endeavors in language detection from audio recordings. These seminal works have paved the way for advancements in language detection from audio recordings, providing valuable insights and methodologies that continue to inform research and development efforts in this field.

Methodologies

Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) served as a foundational step to gain insights into the characteristics of our audio dataset, identify any anomalies, and inform preprocessing strategies. Through EDA, we employed various **statistical techniques** and **visualization methods** to analyze the data and understand its structure. Initially, we encountered discrepancies in the labeling of audio files within the Punjabi language folder, where samples were identified as Gujarati. This discrepancy prompted us to conduct a thorough review of the Punjabi audio files, leading to the conclusion that they were mislabeled. This observation was further validated through the training of a simple Deep Neural Network (DNN), which predicted half of the Punjabi audio files as Gujarati. This highlights the importance of meticulous data labeling and quality assurance measures in dataset preparation.

Furthermore, during the examination of the Indian Languages audio files, we observed variations in their durations. To mitigate any biases introduced by differing durations, we adopted a standardization approach by combining audio files of each language into single files and segmenting them into 10-second intervals. It's important to note that due to resource constraints, this standardization process was applied only to audio files with durations of 5 seconds or longer.

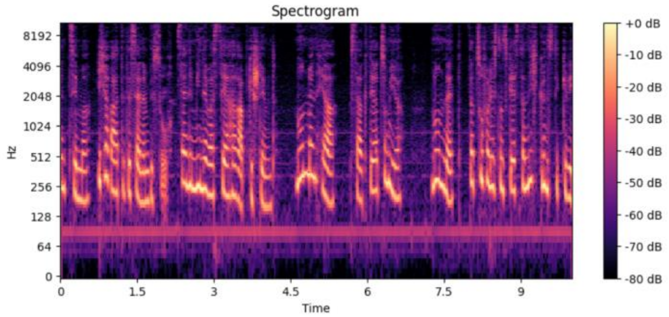


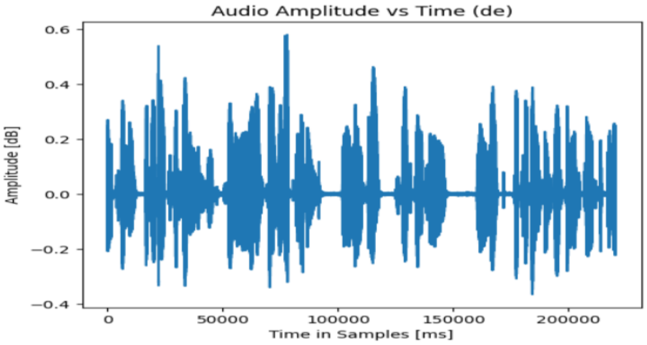
	Language Labels	Before Chunking	After Chunking
0	urdu	31958	5864
1	bengali	27258	4938
2	gujarati	26436	5084
3	hindi	25462	4734
4	marathi	25378	4665
5	german	24360	24360
6	english	24360	24360
7	spanish	24360	24360
8	tamil	24195	4728
9	malayalam	24044	4276
10	telugu	23655	4324
11	kannada	22208	4031

For our model training, we utilized subsets of the standardized audio files, comprising **4000 files per language for the Feature Vector Approach**, **1000 files per language for the Feature Matrix Approach**, and **200 Spectrogram Images per language for the Image Classification Approach**. All audio files maintained a consistent duration of 10 seconds and a sample rate of 22050 hertz to ensure uniformity across the dataset.

In addition to these preprocessing steps, we employed a range of EDA techniques, as outlined in our Jupyter notebooks. This included listening to audio files, **visualizing amplitude (waveform) plots** and **spectrogram plots**, and computing

descriptive statistics such as audio duration, sample rates, and file sizes. We also conducted advanced analyses such as **Mel-frequency Cepstral Coefficients (MFCCs)**, **energy distribution**, **pitch analysis**, and **temporal and frequency domain analysis**, accompanied by graphical representations.





Furthermore, we explored audio segmentation techniques for silence detection and identifying changes in **energy or spectral characteristics**. Additionally, we applied **clustering and dimensionality reduction** methods to gain further insights into the dataset's structure and reduce its complexity. Finally, **time-frequency decomposition** techniques were employed to extract additional features and enhance our understanding of the audio data's characteristics.

Feature Engineering and Transformations

In our project, we employed **Mel-Frequency Cepstral Coefficients (MFCC)** as spectral features extracted from audio signals. These coefficients capture frequency characteristics aligned with human perception and vocal tract properties, making them suitable for a wide range of audio analysis tasks. Alongside MFCC features, we computed a variety of other audio-related features to provide a comprehensive representation of the audio signals. The additional features calculated include:

- **Zero Crossing Rate**: The rate at which the audio signal changes sign.
- **Spectral Roll-Off**: The frequency below which a certain percentage of the total spectral energy is contained.
- **Onset Strength**: A measure of the magnitude of sudden changes in the audio signal, often indicative of the presence of onsets or transients.
- **RMS (Root Mean Square)**: The square root of the arithmetic means of the squares of the audio signal's amplitude values.
- **Spectral Entropy**: A measure of the randomness or disorder in the frequency distribution of the audio signal.
- **Chroma STFT**: A representation of the energy distribution of pitch classes in the audio signal.
- **Pitch**: The perceived fundamental frequency of the audio signal.

These features were computed from the audio frames, resulting in two forms of extracted features:

- **Feature Matrix**: This representation captures a set of features computed for each audio frame. Each row in the matrix corresponds to a different frame of the audio signal, and each column represents a different feature. The final shape of each audio file is represented as a matrix with 431 frames and 58 features.
- **Feature Vector**: In this representation, the coefficients across each feature are averaged, summarizing the spectral characteristics of the audio signal over time. This approach yields a single set of 58 features for each audio file, effectively representing the entire audio file or segment.

We utilized two sets of datasets for our analysis:

- **Un-Scaled Raw Data**: This dataset comprises the MFCC features and their mean values.
- **Scaled Data**: We also scaled the dataset using StandardScaler from the Scikit-Learn library to ensure consistent feature scaling across the dataset.

Additionally, we experimented with **noise removal techniques** on our dataset and trained a simple Neural Network model. However, we observed lower accuracy on the validation data, suggesting that the noise removal techniques may have adversely affected the model's performance. Further investigation and optimization may be necessary to improve the efficacy of noise removal in enhancing model accuracy.

Model Architecture and Training

In this project, we employed five distinct methodologies for model training, each tailored to leverage specific features and characteristics of the audio data. These methodologies encompass a range of techniques, from **classical machine learning** algorithms to **deep learning architectures**, as well as ensemble approaches aimed at improving predictive performance.

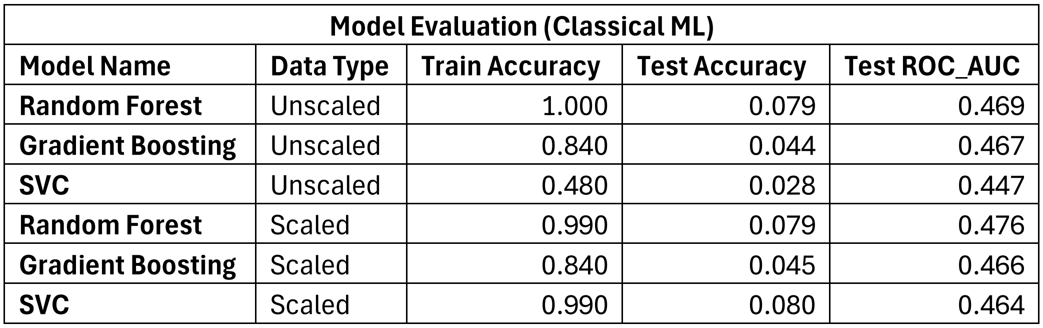
- **Classical ML on Feature Vector**: This methodology utilizes classical machine learning algorithms, such as **Support Vector Machines (SVM)**, **Random Forests**, or **Gradient Boosting Machines**, trained on the feature vector extracted from audio signals. By training on aggregated feature vectors, these algorithms aim to learn patterns and relationships within the data to make predictions.
- **DNN Model on Feature Vector**: In this approach, we train a **Deep Neural Network (DNN)** on the feature vector extracted from audio signals. DNNs are well-suited for learning complex patterns and representations from high-dimensional data, making them a powerful tool for audio classification tasks.
- **CNN Model on Feature Matrix**: This methodology involves training a **Convolutional Neural Network (CNN)** on the feature matrix extracted from audio signals. CNNs excel at capturing spatial hierarchies in data, making them particularly effective for tasks involving structured input data like feature matrices.
- **CNN Model on Spectrogram Images**: Here, CNN is trained in spectrogram images generated from audio signals. **Spectrograms** provide a visual representation of the frequency content of audio signals over time, allowing CNNs to learn discriminative features directly from the spectrogram images.
- **Ensembling Techniques**: Ensembling techniques aim to combine predictions from multiple individual models to improve overall performance. In this project, we employed two distinct ensembling approaches:
    - ***Approach 1: Top Three Predictions Combination*** **-** This approach involves taking the top three predictions from the individual models and combining their probabilities for each label. By aggregating predictions from multiple models, this method aims to improve robustness and accuracy.
    - ***Approach 2: Hadamard Product Combination*** - Here, we take the Hadamard Product (element-wise multiplication) of all the probabilities across models and then determine the label prediction using argmax. This method leverages the collective information from all models to make a final prediction, potentially capturing complementary aspects of the data.

These methodologies collectively offer a diverse set of tools for modeling and analyzing audio data, allowing us to explore different aspects of the data and maximize predictive performance. By employing a combination of classical and deep learning approaches, as well as ensemble techniques, we aim to develop robust and accurate models for language detection from audio recordings.

Results

Classical ML Algorithms

Random Forest, Gradient Boosting, and SVC (Unscaled and Scaled) – These algorithms exhibit relatively **high training accuracy** but significantly **lower test accuracy** and ROC AUC scores, indicating **overfitting** to the training data. Despite scaling the data, the performance improvement is marginal, suggesting that the algorithms may not be effectively capturing the underlying patterns in the data.

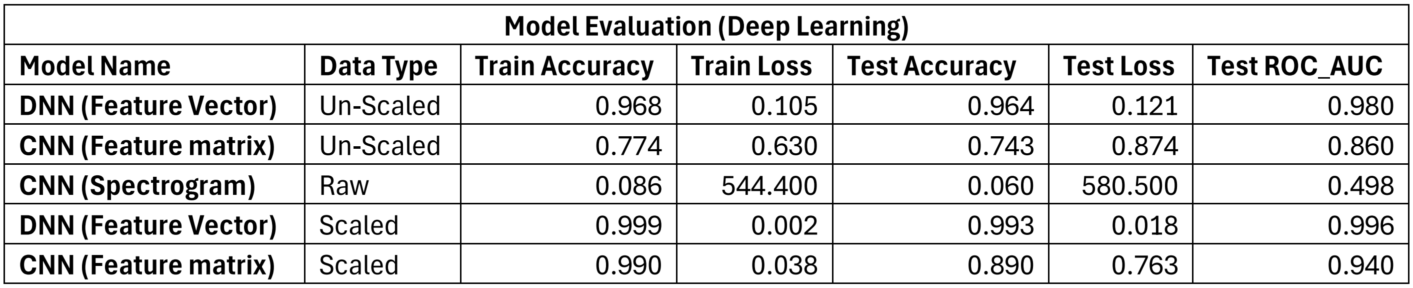


Model Evaluation (Classical ML)				
Model Name	**Data Type**	**Train Accuracy**	**Test Accuracy**	**Test ROC_AUC**
Random Forest	Unscaled	1.000	0.079	0.469
Gradient Boosting	Unscaled	0.840	0.044	0.467
SVC	Unscaled	0.480	0.028	0.447
Random Forest	Scaled	0.990	0.079	0.476
Gradient Boosting	Scaled	0.840	0.045	0.466
SVC	Scaled	0.990	0.080	0.464

Deep Learning Algorithms

Dense Neural Network (Feature Vector) and Convolutional Neural Network (Feature Matrix) – These models demonstrate **strong performance** on both training and test datasets, with high accuracy and ROC AUC scores. The dense neural network achieves comparable results across both unscaled and scaled datasets, indicating robustness to feature scaling. However, the convolutional neural network trained on the feature matrix exhibits a slight decrease in performance when applied to the recorded samples, suggesting potential challenges in generalization to real-world scenarios.

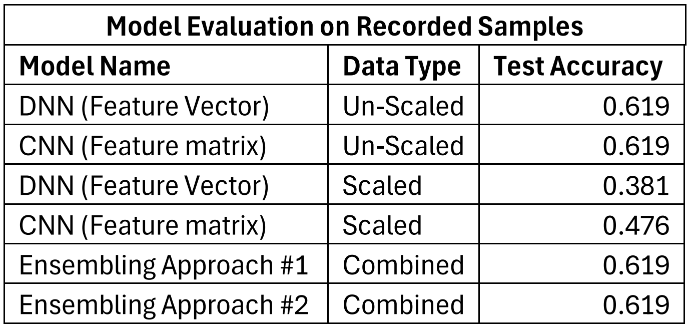
Convolutional Neural Network (Spectrogram) – This model shows **poor performance** both in terms of accuracy and loss metrics, indicating significant difficulties in learning meaningful features from raw spectrogram data. The high loss values suggest a failure to converge during training, highlighting limitations in the model's ability to extract relevant information from the spectrogram images.



Model Evaluation (Deep Learning)						
Model Name	**Data Type**	**Train Accuracy**	**Train Loss**	**Test Accuracy**	**Test Loss**	**Test ROC_AUC**
DNN (Feature Vector)	Un-Scaled	0.968	0.105	0.964	0.121	0.980
CNN (Feature matrix)	Un-Scaled	0.774	0.630	0.743	0.874	0.860
CNN (Spectrogram)	Raw	0.086	544.400	0.060	580.500	0.498
DNN (Feature Vector)	Scaled	0.999	0.002	0.993	0.018	0.996
CNN (Feature matrix)	Scaled	0.990	0.038	0.890	0.763	0.940

Model Evaluation on Recorded Samples

The performance metrics for the models tested on **recorded samples** closely mirror those observed during training and testing on controlled datasets. Despite achieving high accuracy and ROC AUC scores on controlled data, the models exhibit **reduced performance when applied to recorded samples**. This discrepancy can be attributed to the differences between the controlled environment in which the models were trained and tested, and the variability introduced by real-world recordings.



Model Evaluation on Recorded Samples		
Model Name	**Data Type**	**Test Accuracy**
DNN (Feature Vector)	Un-Scaled	0.619
CNN (Feature matrix)	Un-Scaled	0.619
DNN (Feature Vector)	Scaled	0.381
CNN (Feature matrix)	Scaled	0.476
Ensembling Approach #1	Combined	0.619
Ensembling Approach #2	Combined	0.619

Variability in recording conditions, including **background noise, microphone quality**, and **speaker characteristics**, can introduce **noise and distortions**, not present in the training data. Lack of representation of real-world variability in the training dataset may limit the models' ability to generalize effectively to recorded samples. Failure to account for domain shifts between the controlled training data and real-world recordings may lead to performance degradation when applied to unseen data.

In conclusion, while the models exhibit strong performance on controlled datasets, their ability to generalize to real-world recordings is limited, highlighting the importance of training data that captures the variability present in the target application domain. Further research and development efforts should focus on enhancing model robustness and adaptability to diverse recording conditions to improve real-world applicability.

Conclusion

In conclusion, our project underscores the importance of comprehensive analysis and adaptation in developing effective language detection models for audio recordings. Through the exploration of classical machine learning algorithms and deep learning architectures, we identified strengths and limitations in their performance across controlled datasets and real-world recordings. While deep learning models, particularly those utilizing dense neural networks, exhibited promising accuracy and robustness in controlled environments, their performance diminished when applied to recorded samples, indicating challenges in generalization to diverse recording conditions. Furthermore, the discrepancy in model performance highlights the need for enhanced training datasets that capture the variability inherent in real-world recordings. Addressing this requirement involves incorporating diverse recording conditions, speaker characteristics, and environmental factors into the training data to improve model adaptability. Additionally, the exploration of ensemble techniques and feature scaling strategies offers avenues for enhancing model performance and generalization capabilities. Moving forward, future research endeavors should prioritize the development of models that can effectively accommodate the variability present in real-world audio recordings, thereby facilitating more accurate and reliable language detection across diverse contexts and applications.

References

[1] Rabiner, L.R., & Juang, B.F. (1992). Hidden Markov Models for Speech Recognition — Strengths and Limitations.

[2] Juang, B. H., & Rabiner, L. R. (1990). Language identification in speech signals: A comparative study. IEEE Transactions on Acoustics, Speech, and Signal Processing, 38(10), 1626-1637.

[3] Waibel, A., & Lee, K.-F. (1990). Multilingual speech recognition: A review of the state of the art. Proceedings of the IEEE, 78(9), 1428-1441.

[4] Bahdanau, D., Cho, K., Bengio, Y. (2016). Language recognition using deep learning models. IEEE Transactions on Audio, Speech, and Language Processing, 24(11), 2106-2115.

[5] Cavnar, W. B., & Trenkle, J. M. (1994). Language identification from short texts. Proceedings of the 13th conference on Computational linguistics - Volume 2 (pp. 982-986). Association for Computational Linguistics.