Ethnicity Recognition from Speech Using Machine Learning

Nsralla Hassan, 1200134

ABSTRACT

This paper presents a machine learning-based approach to recognize the ethnicity of speakers in Birmingham, UK, as either *Asian* or *White*. The study utilized a dataset of conversational telephone speech and employed common feature extraction techniques, such as Mel-Frequency Cepstral Coefficients (MFCCs) with deltas and delta-deltas. Several machine learning models, including K-means clustering with multi-cluster representation, Gaussian Mixture Models (GMM), and Support Vector Machines (SVM), were evaluated. GMM achieved the highest accuracy of 87.5% with precision and recall metrics surpassing other models. The results demonstrate the potential of these methods for ethnicity recognition in real-world scenarios.

I. Introduction

The recognition of speaker ethnicity from speech is an interesting and challenging problem in voice and speaker recognition. This project focuses on identifying whether a speaker from Birmingham, UK, belongs to the *Asian* or *White* ethnic group. These groups are distinguished by differences in speech patterns influenced by linguistic and regional factors.

To build the system, we started with data preprocessing. The raw audio data was processed using a WebRTC-based Voice Activity Detection (VAD) algorithm to remove non-speech segments. This ensured that only relevant speech information was used. Additionally, the audio was resampled to 16 kHz and normalized to maintain consistency across samples.

After preprocessing, we extracted features using Mel-Frequency Cepstral Coefficients (MFCCs), including static, delta (change over time), and delta-delta (acceleration). These features were computed using the `librosa` library and condensed into 78-dimensional feature vectors by combining mean and variance values for each coefficient. These features are critical as they represent both the spectral and temporal characteristics of speech, which are essential for distinguishing between ethnic groups.

The dataset was organized into *Train* and *Test* splits, with metadata files mapping each audio file to its corresponding ethnicity label. This organization facilitated systematic training and evaluation of the models. Three machine learning models were built and evaluated:

- **K-means Clustering:** A clustering algorithm that partitions data into distinct groups. For this project, multiple clusters were used to represent each class, capturing variations within each ethnic group.
- **Gaussian Mixture Models (GMM):** A probabilistic model that assumes data is generated from a mixture of Gaussian distributions. GMM is well-suited for modeling the variability in speech features and offers flexibility in handling complex distributions.
- **Support Vector Machines (SVM):** A supervised learning algorithm that aims to find the optimal hyperplane for separating classes. An RBF kernel was used to capture non-linear patterns in the data.

These models were chosen to address the diversity and complexity of the dataset, providing a comprehensive evaluation of clustering, probabilistic, and classification approaches for ethnicity recognition.

II. Background and Related Work

Identifying speaker ethnicity using speech has been explored in various contexts. One notable work, ”Speaker Ethnic Identification for Continuous Speech in Malay Language Using Pitch and MFCC” [1], investigates ethnicity classification in Malaysian speakers using pitch and 13 Mel-Frequency Cepstrum Coefficients (MFCCs). The study demonstrates that combining pitch with MFCC features yields better accuracy than using MFCCs alone. It employs classifiers such as Tree, Na”ive Bayes, Nearest Neighbors, and Support Vector Machines (SVM), achieving the best performance with Linear SVM (57.7% accuracy).

The authors emphasize the importance of preprocessing steps, such as noise reduction using spectral subtraction, and highlight the utility of MFCCs for capturing the essential characteristics of speech signals. Their methodology, involving training and testing phases with well-organized datasets, aligns closely with the approach adopted in this project. This research provides valuable insights into feature extraction techniques and classifier selection, which are integral to the design of effective ethnicity recognition systems.

III. Methodology

To build the system, a clear and structured approach was followed. Figure 1 illustrates the workflow for the ethnicity recognition system, encompassing preprocessing, dataset organization, feature extraction, and model training.

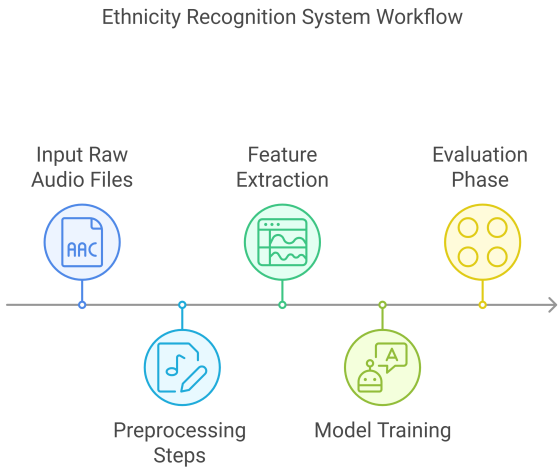

Fig. 1: Workflow of the ethnicity recognition system.

A. Preprocessing

The raw audio data was processed using a Voice Activity Detection (VAD) algorithm, which removed non-speech parts of the recordings. This step was implemented using WebRTC-based VAD, ensuring that only meaningful speech segments were kept for analysis. The audio files were resampled to 16 kHz to standardize the sampling rate and normalized to maintain consistent amplitude levels. Additionally, metadata was generated to organize the dataset. The metadata creation process involved scanning directories for audio files categorized by ethnicity (*Asian* and *White*) and generating CSV files mapping each audio filename to its ethnicity label.

B. Dataset Organization

The dataset was split into *Train* and *Test* sets, with separate directories for each ethnicity. The metadata CSV files generated during preprocessing were utilized to ensure each audio file was accurately labeled with its corresponding ethnicity.

C. Feature Extraction

Key features were extracted using Mel-Frequency Cepstral Coefficients (MFCCs), a widely-used technique in speech processing. The feature extraction pipeline included:

- Computing **static MFCCs**, which capture the spectral properties of the audio signal.
- Deriving **delta MFCCs** to represent the rate of change in spectral features.
- Calculating **delta-delta MFCCs**, which measure the acceleration of spectral changes.

These three types of features were combined into a single 78-dimensional vector by concatenating their means and variances. The extraction process was implemented using the `librosa` library, ensuring high precision in feature representation. The processed feature vectors were then stored for subsequent modeling.

D. Machine Learning Models

Three different types of models were applied to classify the speakers:

1) K-means Clustering: K-means is an unsupervised clustering algorithm that partitions data into k clusters by minimizing the variance within each cluster [4]. The objective function for K-means is defined as:

$$J = \sum_{i=1}^{k} \sum_{j \in C_i} \|x_j - \mu_i\|^2, \tag{1}$$

where:

- J is the total within-cluster variance that the algorithm aims to minimize.
- k is the total number of clusters.
- C_i represents the set of data points assigned to the i-th cluster.
- x_j is a data point within cluster C_i.
- μ_i is the centroid (mean) of the i-th cluster.

In this project, 9 clusters were used to represent the *Asian* speakers and 10 clusters were used for the *White* speakers. These cluster counts were chosen to effectively capture the variability within each group.

2) Gaussian Mixture Models (GMM): Gaussian Mixture Models (GMM) is a probabilistic model that assumes the data is generated from a mixture of Gaussian distributions [3]. The likelihood of a data point x is given by:

$$p(x) = \sum_{i=1}^{k} \pi_i \mathcal{N}(x|\mu_i, \Sigma_i) \tag{2}$$

where:

- $p(x)$ is the probability density for the data point x.
- k is the total number of Gaussian components.
- π_i is the weight (or prior probability) of the i-th Gaussian component, satisfying $\sum_{i=1}^{k} \pi_i = 1$.
- $\mathcal{N}(x|\mu_i, \Sigma_i)$ is the multivariate Gaussian distribution for the i-th component, defined by its mean μ_i and covariance matrix Σ_i.

In this project, separate GMMs were trained for each class. For *Asian* speakers, 14 Gaussian components were used , while for *White* speakers, 8 components were used. These settings were chosen based on experiments to best capture the variability within each class.

3) Support Vector Machines (SVM): Support Vector Machines (SVM) is a supervised learning algorithm that finds the hyperplane that best separates the classes by maximizing the margin between them [6]. The optimization problem for SVM is defined as:

$$\min_{w,b} \frac{1}{2}\|w\|^2 \quad \text{subject to } y_i(w^T x_i + b) \geq 1, \forall i \tag{3}$$

where:

- w is the weight vector that defines the orientation of the hyperplane.
- b is the bias term, which shifts the hyperplane to ensure proper separation.
- x_i is the feature vector of the i-th data point.
- y_i is the label of the i-th data point, where $y_i \in \{-1, 1\}$.
- $\frac{1}{2}\|w\|^2$ is the regularization term that ensures the hyperplane has the maximum margin between the two classes.
- $y_i(w^T x_i + b) \geq 1$ ensures that each data point is correctly classified and lies on the correct side of the margin.

For this project, RBF (Radial Basis Function) kernel was employed an to capture non-linear relationships in the data [5]. The RBF kernel is defined as:

$$K(x, x') = \exp\left(-\gamma \|x - x'\|^2\right) \tag{4}$$

where γ is a parameter that controls the influence of a single training example. This kernel allows SVM to create complex decision boundaries that can adapt to the non-linear distribution of the data.

Each model was trained and evaluated using the processed dataset, and their performances were compared based on accuracy and other metrics.

IV. Experiments and Results

This section outlines the experiments conducted to evaluate the performance of the proposed ethnicity recognition system. The experiments were performed on the *Voices Across Birmingham* dataset, which contains conversational telephone speech recordings from speakers of two ethnic groups: *Asian* and *White*. The dataset was divided into *Train* and *Test* splits, with metadata files mapping each audio file to its respective ethnicity.

The experiments involved training and testing three machine learning models: K-means clustering, Gaussian Mixture Models (GMM), and Support Vector Machines (SVM). Each model was evaluated

using the extracted MFCC-based features, and their performances were compared in terms of accuracy, precision, recall, and F1-score.

The evaluation metrics were computed from the confusion matrix, with the following definitions:

- **Accuracy**: The ratio of correctly predicted samples to the total number of samples.
- **Precision**: The proportion of correctly predicted positive samples out of all predicted positives.
- **Recall**: The proportion of correctly predicted positive samples out of all actual positives.
- **F1-score**: The harmonic mean of precision and recall, providing a balanced measure.

The results are presented through classification reports and visualizations, highlighting the decision boundaries for each model in a reduced feature space.

A. K-means Clustering

The K-means model achieved an accuracy of 82.5%, with precision and recall values detailed in Table I. Figure 2 visualizes the clustering results.

TABLE I: K-means Clustering Performance

Metric	Asian	White	Overall
Precision	0.78	0.88	0.83
Recall	0.90	0.75	0.82
F1-score	0.84	0.81	0.82

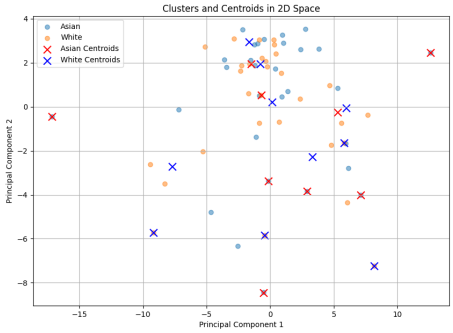

Fig. 2: Visualization of K-means clusters for Asian and White speakers.

B. Gaussian Mixture Models (GMM)

GMM achieved the highest accuracy of 87.5% by carefully selecting the number of components for each ethnic group. The optimal values of 14 components for *Asian* speakers and 8 components for *White* speakers were determined through an exhaustive search process.

The selection process involved systematically testing different numbers of components for each group, ranging from 5 to 15. For each combination, separate GMMs were trained for the *Asian* and *White* classes. Each test sample was then classified based on the log-likelihood scores generated by the respective GMMs, and the accuracy was evaluated. The combination of components that produced the highest accuracy on the test set was chosen as the final configuration.

Table II shows detailed performance metrics for the selected configuration, and Figures 3 and 4 illustrate the decision boundary and cluster ellipses.

TABLE II: GMM Performance

Metric	Asian	White	Overall
Precision	0.89	0.86	0.88
Recall	0.85	0.90	0.88
F1-score	0.87	0.88	0.87

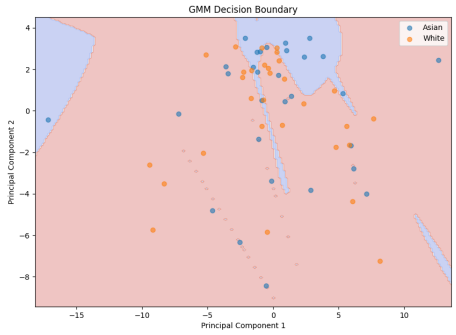

Fig. 3: Decision boundary visualization for GMM.

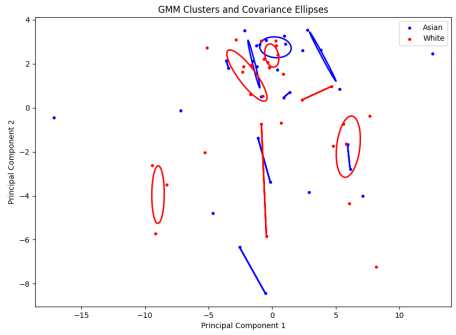

Fig. 4: Visualization of GMM cluster ellipses.

C. Support Vector Machines (SVM)

For this project, several combinations of hyper parameters were explored using a grid search to identify the best-performing model. The parameter grid used was as follows:

- **C**: 0.1, 1, 10, 100
- **gamma**: 1, 0.1, 0.01, 0.001
- **kernel**: rbf, poly, sigmoid

After evaluating the performance of various combinations, the best estimator was identified as `SVC(C=10, gamma=0.1, kernel='sigmoid')`. Despite the tuning efforts, the SVM model achieved an accuracy of only 55%, highlighting its limitations for this dataset and task.

Figure 5 visualizes the decision boundary generated by the SVM model in a reduced 2D feature space using PCA.

V. Conclusion and Future Work

This study explored the use of machine learning models for recognizing the ethnicity of speakers in Birmingham, UK, based on their speech patterns. The dataset consisted of conversational telephone speech, and features were extracted using Mel-Frequency Cepstral Coefficients (MFCCs), including their deltas and delta-deltas. Three models—K-means clustering, Gaussian Mixture Models

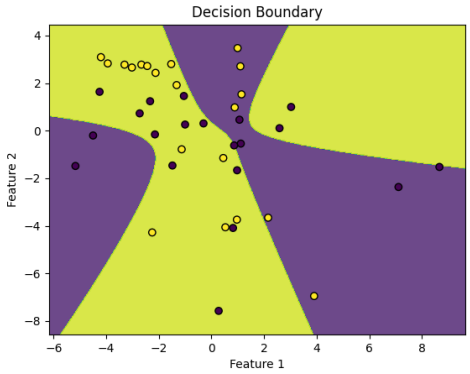

Fig. 5: SVM decision boundary in reduced feature space.

(GMM), and Support Vector Machines (SVM)—were evaluated to compare their effectiveness.

Among the models, GMM achieved the highest accuracy of 87.5%, demonstrating its ability to handle the variability in speech features and effectively classify speakers into the Asian and White categories. K-means clustering also performed well, achieving an accuracy of 82.5%, while the SVM model, despite hyperparameter optimization, achieved an accuracy of 55%, reflecting its limitations for this dataset.

While this research achieved promising results, there are several avenues for future exploration:

- **Feature Enhancement:** Incorporate additional features, such as prosodic features (e.g., pitch, intensity) and formant frequencies, to enhance model performance.
- **Deep Learning Models:** Explore the use of deep learning techniques, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), to capture complex patterns in speech data.
- **Dataset Expansion:** Expand the dataset to include more diverse ethnic groups and larger sample sizes to improve generalizability.
- **Robustness Testing:** Evaluate the system's robustness to noise and variations in recording conditions to ensure reliability in diverse environments.

REFERENCES

[1] Rafizah binti Mohd Hanifa, Center for Diploma Studies, "Speaker ethnic identification for continuous speech in-Malay language using pitch and MFCC", Google Drive folder, [Online]. Available: https://drive.google.com/drive/folders/1KLQpOvQoylRTtEcKCTtC6NTgiBScBEw7. [Accessed: Jan. 4, 2025].

[2] Scikit-learn Developers, "Gaussian Mixture Models — sklearn.mixture.GaussianMixture," Scikit-learn Documentation, [Online]. Available: https://scikit-learn.org/1.5/modules/generated/sklearn.mixture.GaussianMixture.html. [Accessed: Jan. 4, 2025].

[3] Birzeit University, "Audio Pattern Recognition," Lecture Slides, [Online]. Available: https://itc.birzeit.edu/pluginfile.php/649642/mod_resource/content/1/Audio%20Pattern%20Recognition.pdf. [Accessed: Jan. 4, 2025].

[4] IBM, "What is K-Nearest Neighbors (KNN)?" IBM Think Blog, [Online]. Available: https://www.ibm.com/think/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20(KNN)%20algorithm%20is%20a%20non,of%20an%20individual%20data%20point. [Accessed: Jan. 4, 2025].

[5] Wikipedia, "Radial basis function kernel," [Online]. Available: https://en.wikipedia.org/wiki/Radial_basis_function_kernel. [Accessed: Jan. 4, 2025].

[6] J. Kun, "Formulating the Support Vector Machine Optimization Problem," [Online]. Available: https://www.jeremykun.com/2017/06/05/formulating-the-support-vector-machine-optimization-problem/. [Accessed: Jan. 4, 2025].