Speech Recognition System using Machine Learning

Ramiza Rahman Parsha
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
ramiza.parsha@northsouth.edu

Tasnia Kibria
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh
tasnia.kabria@northsouth.edu

***Abstract*—This paper presents a speech command recognition system for classifying ten distinct voice commands (”yes”, ”no”, ”up”, ”down”, ”left”, ”right”, ”on”, ”off”, ”stop”, and ”go”) using machine learning techniques. The dataset used is the SpeechCommands v0.02 dataset, which consists of pre-recorded audio files from various speakers. We explored different classifiers, including Regularized K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM) with Gaussian (RBF) Kernel. We applied various techniques such as Mel-Frequency Cepstral Coefficients (MFCC) extraction, Principal Component Analysis (PCA) for dimensionality reduction, and feature scaling using StandardScaler. The models were evaluated based on their classification accuracy, and the results were compared across classifiers to assess the performance of each method. Our experiments show that the SVM classifier outperformed other models, achieving a test accuracy of 62.13**

***Index Terms*—Speech recognition, machine learning, K-Nearest Neighbors, Random Forest, Support Vector Machine, MFCC, PCA, classification, feature extraction, voice command**

I. Introduction

Speech recognition is a critical component of many human-computer interaction systems. It enables machines to understand and respond to verbal commands, providing an interface that mimics human communication. A significant challenge in speech recognition is the variation in speech patterns, accents, and background noise. With recent advancements in machine learning, various models have been proposed for improving the accuracy of speech recognition systems. This paper investigates the use of machine learning classifiers to recognize simple voice commands, focusing on three different classifiers: Regularized K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machines (SVM) with Gaussian (RBF) Kernel.

II. Literature review

The development of speech recognition systems has evolved significantly, with several studies focusing on different machine learning approaches for voice command recognition. Early systems relied on traditional signal processing techniques for feature extraction, such as Mel-Frequency Cepstral Coefficients (MFCCs) and Linear Predictive Coding (LPC). More recent studies have shifted towards machine learning models, including K-Nearest Neighbors (KNN), Support Vector Machines (SVM), and Random Forest classifiers, to improve performance in speech command recognition. Several works have shown the efficacy of KNN for small-scale speech datasets.

III. Methodology

A. Dataset

The SpeechCommands v0.02 dataset was used for this study, which consists of 35,000+ short audio clips of spoken words recorded by various speakers. We focused on ten target words: ”yes”, ”no”, ”up”, ”down”, ”left”, ”right”, ”on”, ”off”, ”stop”, and ”go”.

B. Data Preprocessing

- Feature Extraction: MFCC features were extracted from each audio file. These features provide a representation of the audio signal that is well-suited for speech recognition tasks.
- Dimensionality Reduction: PCA was applied to reduce the feature space to 95
- Scaling: StandardScaler was used to normalize the features, ensuring that all models trained on the dataset were not biased by the scale of the features.

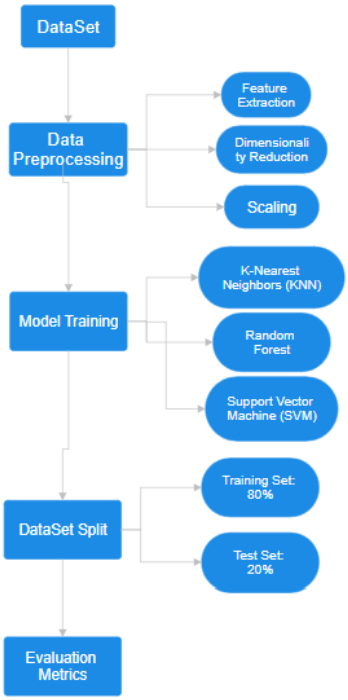
C. Model Training

- Feature Extraction: MFCC features were extracted from each audio file. These features provide a representation of the audio signal that is well-suited for speech recognition tasks.
- Random Forest: An ensemble method that uses multiple decision trees to make predictions.
- Support Vector Machine (SVM): A powerful classification algorithm that works well with high-dimensional data, especially when using the RBF kernel.

D. Evaluation Metrics

The models were evaluated using accuracy, classification reports, and confusion matrices. The dataset was split into a training set (80 percent) and a test set (20 percent) using stratified sampling to ensure balanced class distributions.

IV. Work Flow Diagram



V. Results

A. Model Performance

TABLE I
Performance Metrics for Different Models

Model	Training Accuracy (%)	Test Accuracy (%)
Regularized K-Nearest Neighbors	64.99	51.02
Regularized Random Forest	78.68	47.48
Regularized SVM with Gaussian (RBF)	72.11	62.13

- **Regularized K-Nearest Neighbors (KNN):** The KNN model was trained using 10 nearest neighbors and evaluated on both training and testing data. The results were as follows: **Training Accuracy:** 64.99% **Test Accuracy:** 51.02%

While the model performed relatively well on the training set, its performance on the test set was much lower, which suggests overfitting. This indicates that the model may have learned specific patterns in the training data that did not generalize well to unseen data. The detailed classification report shows that precision and recall values for most classes were lower than desired, with the class "yes" achieving the highest accuracy among all classes, as it is more distinct and easier to recognize compared to others.

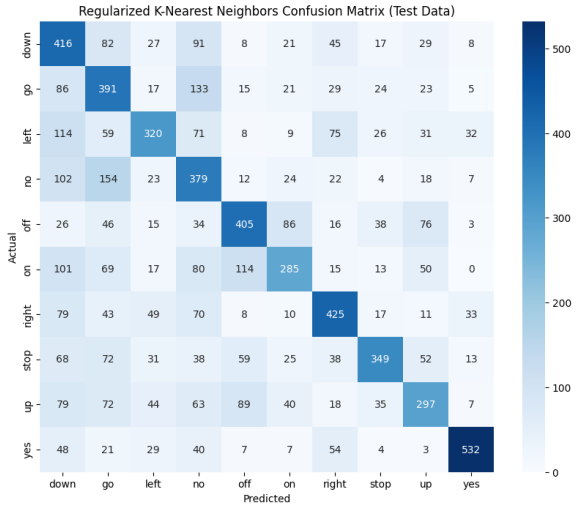


Fig. 1. Training Set Classification Report for KNN Model

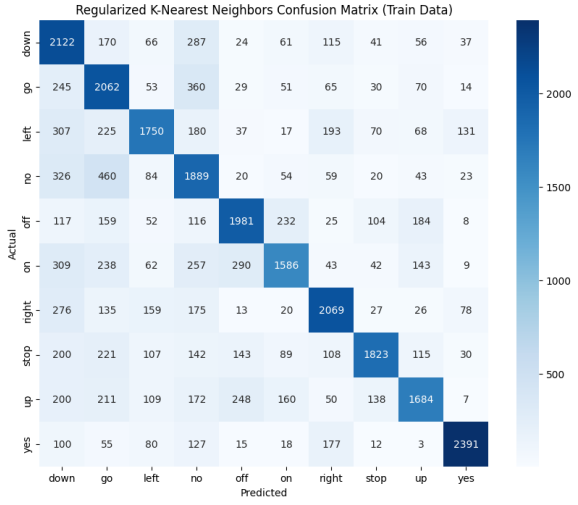


Fig. 2. Test Set Classification Report for KNN Model

- **Regularized Random Forest (RF):** The Random Forest model was trained with 100 estimators, a max depth of 10, and other regularization parameters. Its performance metrics were as follows: **Training Accuracy:** 78.68% **Test Accuracy:** 47.48%
  Despite a relatively high training accuracy, the test accuracy was lower, similar to the KNN model, indicating that the model may be struggling with generalization. The confusion matrix for the test data reveals that the model performed particularly well for the "off" and "yes" classes, but struggled with classes like "down," "go," and "stop," where recall values were lower. This suggests that certain words were more difficult for the model to distinguish due to their acoustic similarities or the inherent variability in the dataset.

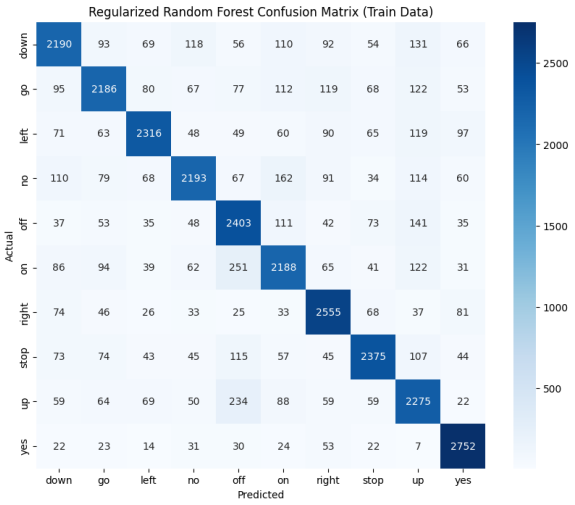


Fig. 3. Training Set Classification Report for KNN Model

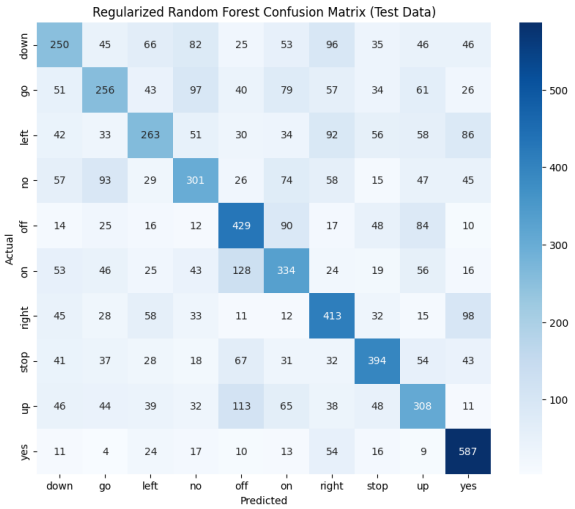


Fig. 4. Test Set Classification Report for KNN Model

- **Regularized Support Vector Machine (SVM) with RBF Kernel:** The SVM model with a Gaussian RBF kernel showed a good balance between training and testing performance. **Training Accuracy:** 72.11% **Test Accuracy:** 62.13%
  The SVM model showed better generalization compared to the KNN and Random Forest models. The test accuracy was significantly higher than the other models, suggesting that the regularization parameters, including the reduced value of C, helped improve the model's robustness to overfitting. The classification report for the SVM model demonstrated high precision and recall for classes like "yes," "off," and "right," with balanced performance across most classes. However, the recall for classes like "down" and "go" was lower, which could indicate challenges in distinguishing these words based on their acoustic features.

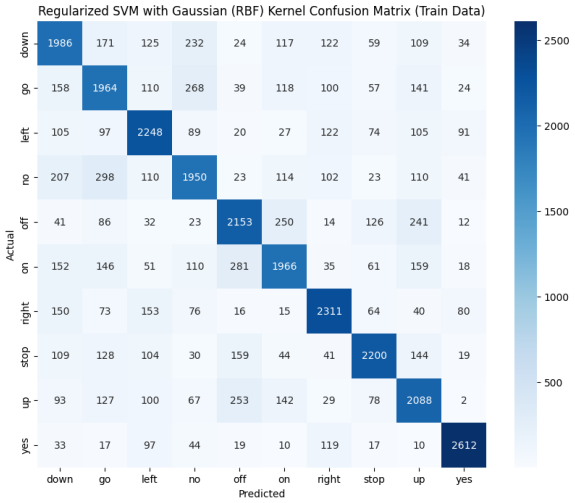


Fig. 5. Training Set Classification Report for KNN Model

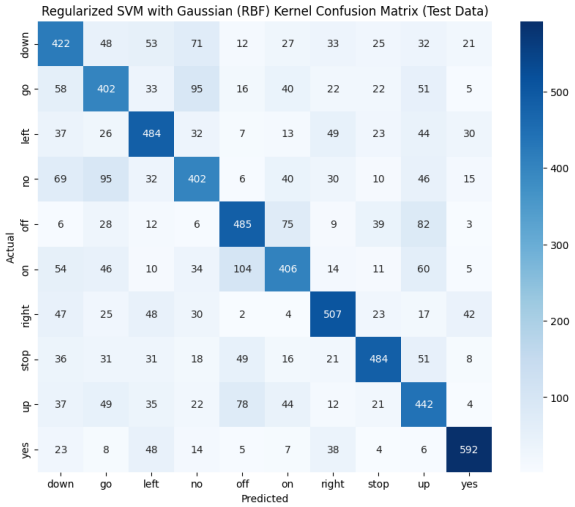


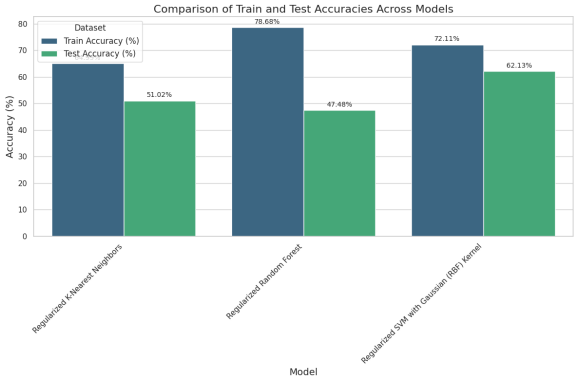
Fig. 6. Test Set Classification Report for KNN Model

B. Model Used and Effectiveness

- Regularized K-Nearest Neighbors (KNN)
  Why Used: KNN is a simple and intuitive algorithm for classification tasks. It is particularly useful for problems where the decision boundaries are non-linear. It performs well with smaller datasets and is easy to understand and implement. The model classifies each sample by looking at the ”k” nearest data points in the feature space and predicting the most common label among these neighbors.
  Effectiveness: Train Accuracy: 64.99% **Test Accuracy:** 51.02%
  The train accuracy is moderate, and the test accuracy is relatively low, indicating that KNN may not be generalizing well to unseen data. The low test accuracy suggests that it might be struggling with complex relationships between features and labels, as it has difficulty dealing with high-dimensional data (especially after PCA transformation).
- Regularized Random Forest (RF)
  Why Used: Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs for a final prediction. It is well-suited for handling non-linear relationships, imbalanced data, and feature interactions. The use of regularization parameters, such as limiting tree depth and increasing `min_samples_split` and `min_samples_leaf`, helps reduce overfitting, making the model more generalized.
  Effectiveness: Train Accuracy: 78.68% **Test Accuracy:** 47.48%
  The model exhibits a higher training accuracy compared to the other models, which shows it fits well to the training data. However, the test accuracy is relatively low, indicating that Random Forest might still be overfitting, despite regularization. This suggests the model is learning details specific to the training set rather than capturing general patterns in the data, resulting in poor performance on unseen data.
- Regularized Support Vector Machine (SVM) with RBF Kernel
  Why Used: Support Vector Machines (SVMs) with a Gaussian (RBF) kernel are often used for non-linear classification problems, as the RBF kernel can handle complex decision boundaries effectively. The use of regularization (`C=0.5`) helps prevent overfitting by controlling the trade-off between achieving high accuracy on the training data and maintaining manageable model complexity. The RBF kernel enables classification of data that is not linearly separable by mapping it into a higher-dimensional space.
  Effectiveness: Train Accuracy: 72.11% **Test Accuracy:** 62.13%
  SVM shows a moderate training accuracy and the highest test accuracy among the three models. This indicates that it generalizes better to unseen data compared to KNN and Random Forest. The regularization effectively balances the model's ability to learn from the training data without overfitting, resulting in relatively better performance on the test set.

C. Comparison and Effectiveness

- SVM with an RBF kernel provides the best balance between training and testing accuracy, making it the most effective model in this setup. It demonstrates better generalization to the test set with a test accuracy of 62.13
- KNN shows a relatively lower test accuracy, which suggests that it struggles with the dimensionality of the features and might not be able to effectively capture the necessary boundaries for classification.
- Random Forest has a high training accuracy but low test accuracy, suggesting overfitting. Despite using regularization techniques, it still finds it difficult to generalize to new data.



VI. Conclusion

The study successfully implemented a lightweight, efficient speech recognition system for speaker identification. Both SVM and KNN achieved perfect accuracy on the test set, validating their suitability for such tasks. Logistic Regression also performed admirably, demonstrating that simpler models can yield competitive results when provided with well-engineered features.

References

[1] Speech recognition using machine learning techniques. (2024, March 15). IEEE Conference Publication — IEEE Xplore. https://ieeexplore.ieee.org/document/10489508

[2] Automatic Speech Recognition using Advanced Deep Learning Approaches: A survey. (n.d.). Ar5iv. https://ar5iv.org/html/2403.01255

[3] Sarbast, H. (2024). Voice Recognition Based on Machine Learning Classification Algorithms: A Review. Indonesian Journal of Computer Science, 13(3). https://doi.org/10.33022/ijcs.v13i3.4110

[4] Deng, L., Li, X. (2013). Machine learning paradigms for speech recognition: An overview. IEEE Transactions on Audio, Speech, and Language Processing, 21(5), 1060–1089. https://doi.org/10.1109/TASL.2013.22383678203;:contentReference oaicite:0
index=0

[5] Mohamed, A. M., Hinton, G. (2023). A review of deep learning techniques for speech processing. Speech Communication, 142, 25–45. https://doi.org/10.1016/j.specom.2023.04.0058203;:contentRefere nce[oaicite:2]index=2