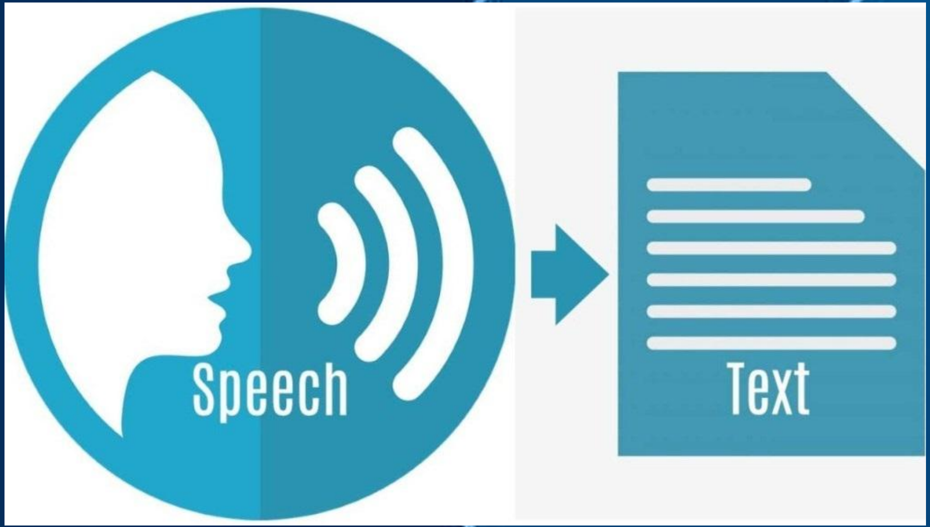
Project Title: Speech Recognition System

Group Members:

1. Ramiza Rahman Parsha (2122370642)
2. Tasnia Kibria (2013553642)

Introduction

- Goal: Build a model to recognize spoken commands.
- Dataset: Google Speech Commands v0.02.
- Focus: Comparing performance of KNN, Random Forest, and SVM models.



Dataset

- **Source:** TensorFlow Speech Commands dataset.
- **Classes:** 10 target words (e.g., Yes, No, Up, Down, Left, Right, On, Off, Stop, Go).
- **Dataset Size:** 38,546 samples initially, balanced to 37,230 samples
- **File Format:** WAV audio files

Data Preprocessing

Audio Features Extracted:

- MFCC (Mel-frequency Cepstral Coefficients)
- Delta MFCC (first derivative)
- Delta-delta MFCC (second derivative)

Normalization: StandardScaler to normalize features

Dimensionality Reduction: PCA to retain 95% variance

Model

Models Tested:

- **K-Nearest Neighbors (KNN):** Regularized with k=10
- **Random Forest:** Regularized with max depth=10, min samples=2
- **SVM (Support Vector Machine):** Regularized with Gaussian (RBF) kernel

Evaluation Metric: Accuracy, Precision, Recall, F1-Score, Confusion Matrix

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).

Model Used and Effectiveness

Regularized Random Forest

- **Why Used**:
  - Random Forest is an ensemble learning method that builds multiple decision trees and combines their outputs for a final prediction. It's well-suited for handling non-linear relationships, imbalanced data, and feature interactions.
  - The use of regularization parameters like limiting tree depth and increasing `min_samples_split` and `min_samples_leaf` helps in reducing 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 means that the model is learning details specific to the training set rather than capturing general patterns in the data, which results in poor performance on new, unseen data.

Model Used and Effectiveness

Regularized SVM with Gaussian (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 in preventing overfitting by controlling the trade-off between achieving a high accuracy on training data and keeping the model's complexity manageable.
  - The RBF kernel makes it possible to classify 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 train accuracy and the highest test accuracy among the three models. This indicates that it is better at generalizing to unseen data compared to KNN and Random Forest. The regularization helps in balancing the model's ability to learn from the training data without overfitting, resulting in a relatively better performance on the test set.

Training and Testing

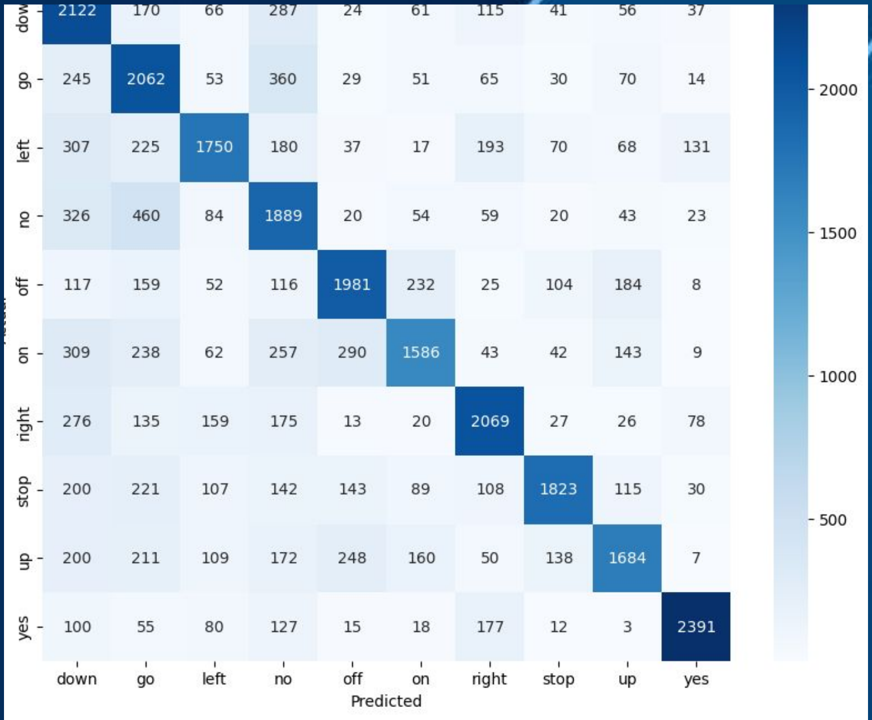
- **Training/Testing Split:** 80/20 (29,784 train, 7,446 test)
- **Feature Scaling:** StandardScaler applied to training and test data
- **PCA Applied:** Reduced feature space to 45 dimensions
- **Model Evaluation:** Accuracy scores and confusion matrices

Results

K-Nearest Neighbors (KNN):

- **Train Accuracy:** 64.99%
- **Test Accuracy:** 51.02%
- **Key Observations:**
  - High recall for words like "yes"
  - Lower accuracy for more difficult words like "down" and "off"
- **Confusion Matrix:** Visualized performance for test/train data


Regularized K-Nearest Neighbors Confusion Matrix (Test Data)
Actual
Predicted
down go left no off on right stop up yes
down 416 82 27 91 8 21 45 17 29 8
go 86 391 17 133 15 21 29 24 23 5
left 114 59 320 71 8 9 75 26 31 32
no 102 154 23 379 12 24 22 4 18 7
off 26 46 15 34 405 86 16 38 76 3
on 101 69 17 80 114 285 15 13 50 0
right 79 43 49 70 8 10 425 17 11 33
stop 68 72 31 38 59 25 38 349 52 13
up 79 72 44 63 89 40 18 35 297 7
yes 48 21 29 40 7 7 54 4 3 532
500
400
300
200
100
0

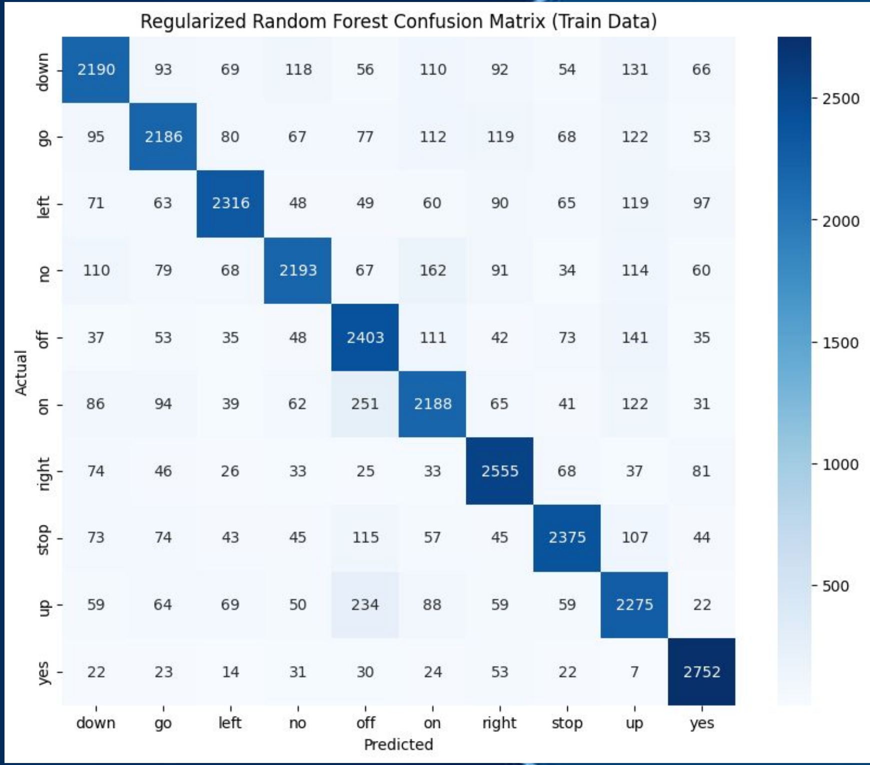

2122 170 66 287 24 61 115 41 56 37
go 245 2062 53 360 29 51 65 30 70 14
left 307 225 1750 180 37 17 193 70 68 131
no 326 460 84 1889 20 54 59 20 43 23
off 117 159 52 116 1981 232 25 104 184 8
on 309 238 62 257 290 1586 43 42 143 9
right 276 135 159 175 13 20 2069 27 26 78
stop 200 221 107 142 143 89 108 1823 115 30
up 200 211 109 172 248 160 50 138 1684 7
yes 100 55 80 127 15 18 177 12 3 2391
down go left no off on right stop up yes
Predicted
2000
1500
1000
500

Results

Random Forest:

- **Train Accuracy:** 78.68%
- **Test Accuracy:** 47.48%
- **Key Observations:**
  - High accuracy on training data, lower on testing data
  - Misclassification between similar sounding commands
- **Confusion Matrix:** Visualized performance for test/train data

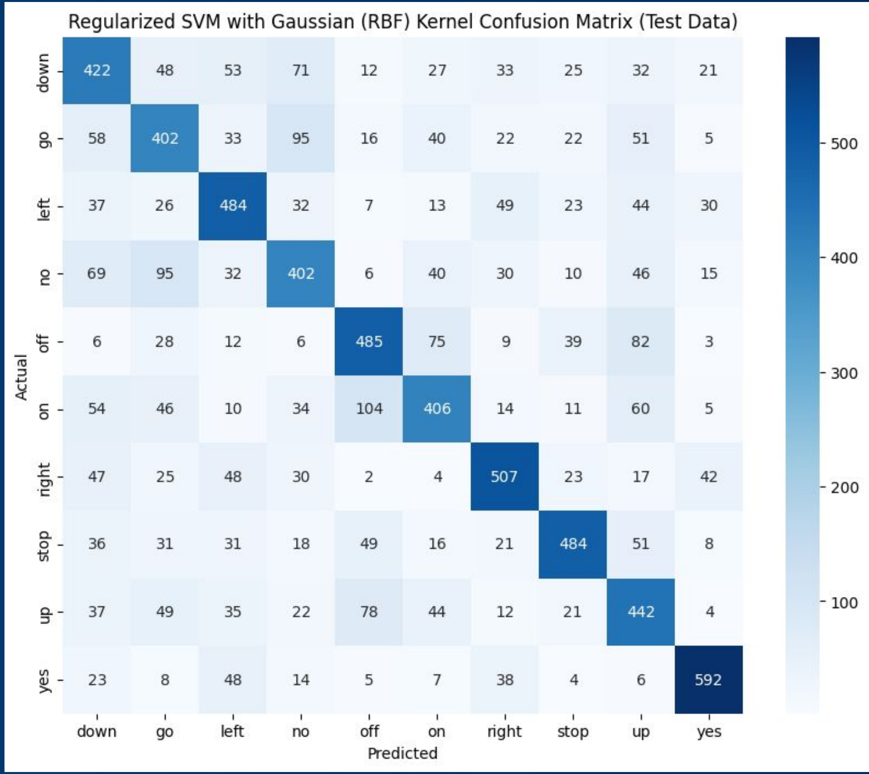

Regularized Random Forest Confusion Matrix (Test Data)
Actual
Predicted
down go left no off on right stop up yes
down 250 45 66 82 25 53 96 35 46 46
go 51 256 43 97 40 79 57 34 61 26
left 42 33 263 51 30 34 92 56 58 86
no 57 93 29 301 26 74 58 15 47 45
off 14 25 16 12 429 90 17 48 84 10
on 53 46 25 43 128 334 24 19 56 16
right 45 28 58 33 11 12 413 32 15 98
stop 41 37 28 18 67 31 32 394 54 43
up 46 44 39 32 113 65 38 48 308 11
yes 11 4 24 17 10 13 54 16 9 587
500
400
300
200
100

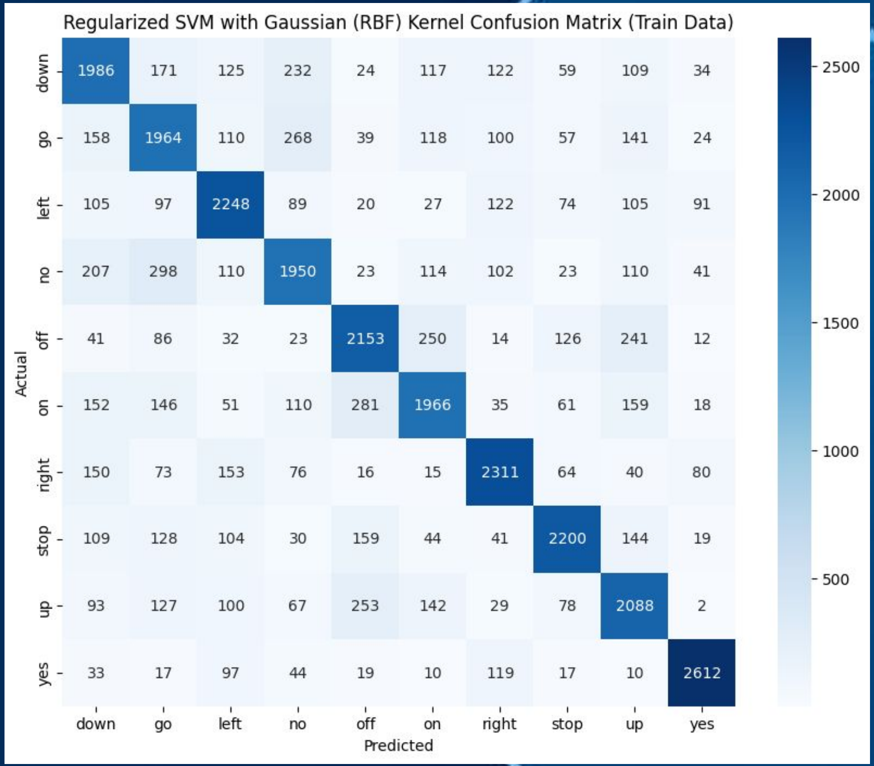

Regularized Random Forest Confusion Matrix (Train Data)
Actual
Predicted
down go left no off on right stop up yes
down 2190 93 69 118 56 110 92 54 131 66
go 95 2186 80 67 77 112 119 68 122 53
left 71 63 2316 48 49 60 90 65 119 97
no 110 79 68 2193 67 162 91 34 114 60
off 37 53 35 48 2403 111 42 73 141 35
on 86 94 39 62 251 2188 65 41 122 31
right 74 46 26 33 25 33 2555 68 37 81
stop 73 74 43 45 115 57 45 2375 107 44
up 59 64 69 50 234 88 59 59 2275 22
yes 22 23 14 31 30 24 53 22 7 2752
2500
2000
1500
1000
500

Results

Support Vector Machine (SVM):

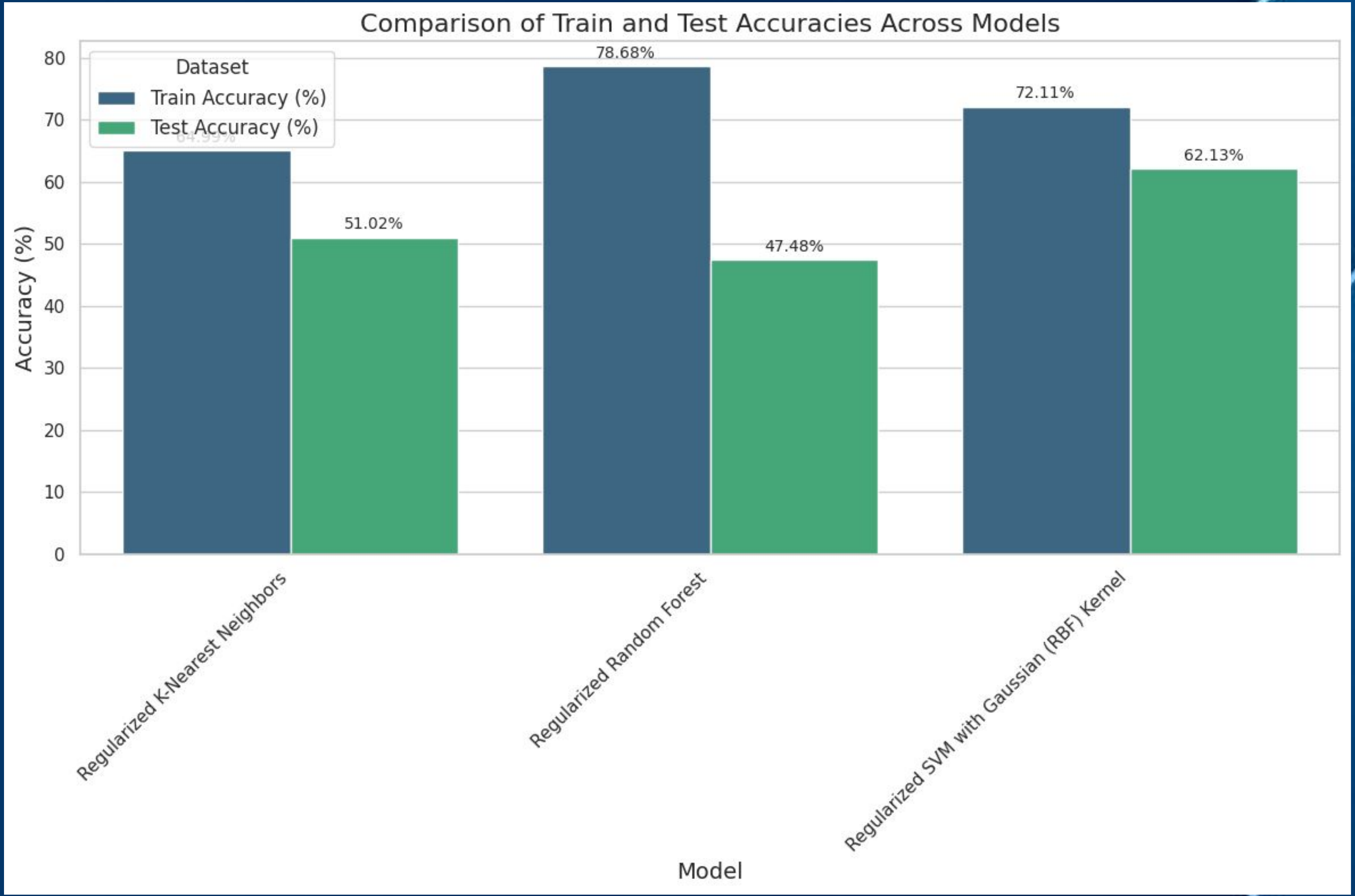
- **Train Accuracy:** 72.11%
- **Test Accuracy:** 62.13%
- **Key Observations:**
  - SVM performs better on the test set compared to Random Forest
  - Stronger performance on commands like "right" and "yes"
- **Confusion Matrix:** Visualized performance for test/train data


Regularized SVM with Gaussian (RBF) Kernel Confusion Matrix (Test Data)
Actual
Predicted
down go left no off on right stop up yes
down 422 48 53 71 12 27 33 25 32 21
go 58 402 33 95 16 40 22 22 51 5
left 37 26 484 32 7 13 49 23 44 30
no 69 95 32 402 6 40 30 10 46 15
off 6 28 12 6 485 75 9 39 82 3
on 54 46 10 34 104 406 14 11 60 5
right 47 25 48 30 2 4 507 23 17 42
stop 36 31 31 18 49 16 21 484 51 8
up 37 49 35 22 78 44 12 21 442 4
yes 23 8 48 14 5 7 38 4 6 592
500
400
300
200
100


Regularized SVM with Gaussian (RBF) Kernel Confusion Matrix (Train Data)
Actual
Predicted
down go left no off on right stop up yes
down 1986 171 125 232 24 117 122 59 109 34
go 158 1964 110 268 39 118 100 57 141 24
left 105 97 2248 89 20 27 122 74 105 91
no 207 298 110 1950 23 114 102 23 110 41
off 41 86 32 23 2153 250 14 126 241 12
on 152 146 51 110 281 1966 35 61 159 18
right 150 73 153 76 16 15 2311 64 40 80
stop 109 128 104 30 159 44 41 2200 144 19
up 93 127 100 67 253 142 29 78 2088 2
yes 33 17 97 44 19 10 119 17 10 2612
2500
2000
1500
1000
500

Comparison of results

- **Model Comparison:**
  - **KNN:** Highest train accuracy, but poor test performance
  - **Random Forest:** Good on training but overfits on test set
  - **SVM:** Best test accuracy overall
- **Bar Chart:** Comparison of Train vs Test accuracy for each model
- **Key Insight:** SVM offers the best balance between training and test performance.


Comparison of Train and Test Accuracies Across Models
Dataset
Train Accuracy (%)
Test Accuracy (%)
Accuracy (%)
80
70
60
50
40
30
20
10
0
64.99%
51.02%
78.68%
47.48%
72.11%
62.13%
Regularized K-Nearest Neighbors
Regularized Random Forest
Regularized SVM with Gaussian (RBF) Kernel
Model

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. *Speech Recognition Using Machine Learning Techniques*. (2024, March 15). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/10489508
4. 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
5.
6.