Journal of Mechatronics, Electrical Power, and Vehicular Technology 17 (2026) 107-117

Journal of Mechatronics, Electrical Power, and Vehicular Technology

e-ISSN: 2088-6985
p-ISSN: 2087-3379

mev.brin.go.id

Voice command classification for mobile robotic control using mel frequency cepstral coefficients and support vector machines

Ratna Hartayu [a, *], Santoso [a], Ahmad Ridho'i [a], Ayusta Lukita Wardani [b], Yunus Awwalu Romadhon [a]

[a] *Department of Electrical Engineering, Universitas 17 Agustus 1945, Surabaya*
Jl. Semolowaru 45, Surabaya, 60118, Indonesia
[b] *Department of Electrical Engineering D4, Universitas Negeri Surabaya*
Jl. Ketintang, Surabaya, 60231, Indonesia

Abstract

Voice command recognition plays a crucial role in enabling intuitive interaction in robotic and embedded control systems. This study proposes a voice command classification system based on Mel-frequency cepstral coefficients (MFCC) and support vector machine (SVM) using the Google speech commands dataset v2. Eight command classes ("down", "go", "left", "no", "right", "stop", "up", and "yes") were used. The dataset was divided into 80 % training and 20 % testing sets, with hyperparameter tuning performed using 5-fold cross-validation on the training data. MFCC feature extraction employed 13 static coefficients augmented with delta and delta-delta features, resulting in a 39-dimensional frame-level representation and a 78-dimensional utterance-level feature vector. Experimental results show that the SVM with radial basis function (RBF) kernel achieved optimal performance with parameters C = 100 and $\gamma = 0.01$, yielding 96.2 % accuracy, 96.5 % precision, 96.0 % recall, and 96.2 % F1-score. The inclusion of dynamic features improved accuracy by 4.7 % compared to static MFCCs. The system demonstrates a lightweight architecture suitable for low-resource environments; however, experiments were primarily conducted under clean conditions, and robustness evaluation was limited to a single noise level (20 dB SNR). Furthermore, real-time deployment on embedded hardware was not experimentally validated and remains part of future work.

Keywords: embedded systems; MFCC; robot control; speech command recognition; support vector machine.

I. Introduction

Recent developments in internet of thing (IoT) and mechatronics have increased the demand for intuitive voice-based interfaces. Speech command systems enable hands-free interaction for applications such as robotic control and smart environments [1][2]. In addition, human–machine interaction systems have been widely explored using physiological signals such as electromyography (EMG), where user intent is translated into control commands for robotic systems [3]. However, implementing such systems on resource-constrained devices remains challenging due to the need for both high accuracy and computational efficiency [4]. Embedded systems play a critical role in enabling real-time monitoring and communication between sensors, controllers, and actuators in complex mechatronic systems [5]. Deep learning models provide strong performance but are often unsuitable for embedded systems due to their high computational cost [6]. Therefore, this study explores the use of Mel-frequency cepstral coefficients (MFCC) and support vector machine (SVM) to achieve an optimal trade-off.

* Corresponding Author. *rhartayu@untag-sby.ac.id* (R. Hartayu)

https://doi.org/10.55981/j.mev.2026.1373

Received 9 February 2026; revised 16 March 2026; accepted 24 April 2026; available online 13 July 2026; published 31 July 2026

How to Cite: R. Hartayu *et al.*, "Voice command classification for mobile robotic control using mel frequency cepstral coefficients and support vector machines," *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, vol. 17, no. 1, pp. 107-117, July, 2026.

However, this study focuses on algorithmic accuracy and does not include actual hardware deployment; claims regarding real-time suitability are therefore preliminary and require further validation on target embedded platforms [7][8].

The primary objective of this study is to develop a lightweight and accurate voice command classification system that balances recognition performance and computational efficiency [9]. Specifically, this work focuses on (1) designing an MFCC-based feature extraction pipeline, (2) optimizing SVM hyperparameters using cross-validation, and (3) evaluating the system performance to assess its feasibility for deployment in resource-constrained robotic systems [10].

Although deep learning models such as convolutional neural networks (CNNs) have demonstrated superior performance in large-scale speech recognition tasks, they typically require extensive training data, high computational power, and significant memory resources. These requirements pose challenges for deployment on embedded robotic platforms with limited hardware capabilities. In contrast, the MFCC–SVM approach offers a simpler and more computationally efficient alternative, making it more suitable for real-time applications in resource-constrained environments while still maintaining competitive accuracy.

Recent research on speech command recognition shows substantial progress through increasingly efficient deep learning architectures. Large-scale datasets enable CNN models to achieve high accuracy, although some approaches still require intensive training. The 1D-CNN method offers competitive performance with fewer parameters, making it suitable for constrained devices [11][12]. Feature-extraction techniques such as MFCC combined with traditional classifiers remain effective for small datasets, though rarely evaluated under noisy conditions. In addition, lightweight CNN architectures are being developed to deliver a balanced combination of efficiency and performance [13][14][15].

To address these gaps, this study proposes a systematic evaluation and optimization of an MFCC–SVM-based speech command classification framework for robotic control applications. Unlike prior works that primarily focus on model implementation, this research emphasizes (1) the analysis of MFCC feature configurations, including static, delta, and delta-delta coefficients, (2) the construction of compact utterance-level feature representations using statistical aggregation, and (3) the optimization of SVM hyperparameters using cross-validation. Additionally, this study evaluates both classification performance and computational efficiency to assess the feasibility of deployment in resource-constrained embedded systems.

The main contribution of this work is not the introduction of a new classification algorithm, but rather a comprehensive and systematic evaluation of MFCC-SVM configurations to achieve an optimal balance between accuracy and computational efficiency. This approach provides practical insights for designing lightweight speech recognition systems suitable for real-time robotic applications.

II. Materials and Methods

Figure 1 illustrates the overall design of the proposed voice command classification system, which consists of four primary stages: input acquisition, preprocessing, feature extraction, and classification. The process begins with the input stage, where voice commands are captured from a speech source such as a microphone or a recorded dataset. These signals typically contain noise and amplitude variations that can degrade classification accuracy. Therefore, the preprocessing stage plays a critical role in improving signal quality through noise removal and normalization techniques. Noise removal eliminates unwanted background sounds, while normalization ensures amplitude consistency across all recordings, preparing the data for accurate feature extraction.

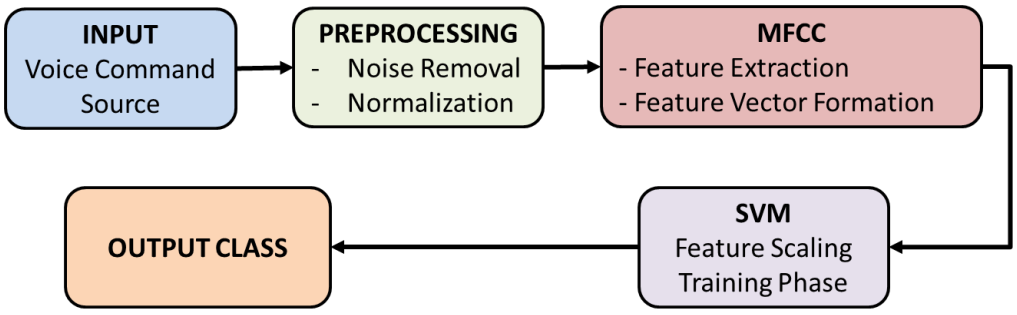

Figure 1. Voice commands classification block diagram system.

The next stage involves MFCC, a widely used technique in speech recognition due to its ability to model the human auditory perception of sound. MFCC extracts the spectral features of speech and converts them into a compact representation called the feature vector [16][17]. These features are then passed to the SVM classifier, which performs feature scaling to standardize the data before entering the training and classification phases. The SVM model is trained to recognize and categorize different voice commands based on the extracted MFCC features. Finally, the output class represents the recognized command category, such as “down,” “go,” “left,” “no,” “right,” “stop,” “up,” and “yes”. This pipeline enables robust and accurate voice-based robotic control.

A. Data preparation

This study employs the Google speech commands dataset v2 [18][19], a publicly available collection of one-second audio recordings containing short spoken words in English. From this dataset, eight primary command classes were selected: down, go, left, no, right, stop, up, and yes. Each class includes balanced samples from multiple speakers, recorded under controlled acoustic conditions with minimal background noise.

All audio files were resampled to 16 kHz and normalized in amplitude to ensure consistency across recordings. The preprocessing pipeline consisted of the following steps:

1. Silence removal–eliminating low-energy regions at the beginning and end of recordings to reduce irrelevant data.
2. Pre-emphasis filtering–amplifying high-frequency components to compensate for the natural spectral tilt of human speech.
3. Framing and windowing–segmenting signals into 25 ms frames with 10 ms overlap and applying a Hamming window to maintain spectral continuity.
4. Fast fourier transform (FFT)–converting time-domain frames into the frequency domain.
5. Mel filterbank processing–mapping the linear frequency scale to the perceptual Mel scale to better approximate human hearing.
6. Discrete cosine transform (DCT)–producing compact Mel-frequency cepstral coefficients (MFCCs) representing the vocal-tract spectral envelope.

For each frame, 13 static MFCC coefficients were extracted, followed by the computation of first-order (Δ) and second-order (Δ^2) temporal derivatives, resulting in a 39-dimensional feature vector per frame. Each utterance was then represented by the average and variance of these feature vectors, forming a compact input suitable for SVM classification.

The dataset was divided into 80 % for training and 20 % for testing, stratified by class to maintain balanced representation. From the training set, 5-fold cross-validation (5-fold CV) was applied during hyperparameter tuning. Specifically, the training data was split into 5 equal folds; in each iteration, 4 folds were used for training and 1 fold for validation. No separate "10 % subset" was used. This process was repeated 5 times, and the average validation accuracy was used to select the best SVM parameters. The final model was then evaluated on the untouched 20 % testing set.

B. Feature extraction using MFCC

1) Pre-emphasis

The pre-emphasis stage is applied to amplify the high-frequency components of the speech signal and compensate for the natural spectral attenuation that occurs during speech production. By emphasizing higher frequencies, this step improves the representation of consonant sounds and enhances the discriminative capability of the extracted features.

The pre-emphasis filtering operation is defined in equation (1).

$$y[n] = x[n] - \alpha x[n-1],\ \alpha = 0.97 \quad (1)$$

where $x[n]$ represents the original speech signal, $y[n]$ denotes the filtered output signals, and α is the pre-emphasis coefficient typically set between 0.95 and 0.97. As shown in equation (1), this operation effectively boosts high-frequency spectral components and improves the robustness of subsequent spectral analysis [20][21].

2) Framing and windowing

Speech signals are inherently non-stationary; they are divided into short overlapping frames to maintain quasi-stationarity with each segment. In this study, the speech signal is segmented into frames of 25 ms with a 10 ms overlap.

To minimize spectral leakage caused by abrupt frame boundaries, each frame is multiplied by a window function. The Hamming window, defined in equation (2), is applied to smooth the edges of the signal.

$$w[n] = 0.54 - 0.46\ cos\left(\frac{2\pi n}{N-1}\right) \quad (2)$$

where n represents the sample of the index and N denoted the frame length. The application of this window function, as expressed in equation (2), reduces

discontinuities at frame boundaries and improves the accuracy of spectral analysis.

3) Frequency transformation using fast fourier transform (FFT)

After windowing, each frame is transformed from the time domain into the frequency domain using the discrete fourier transform (DFT). This transformation enables the analysis of spectral energy distribution across frequencies.

The DFT operation is defined in equation (3).

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi kn}{N}}, \quad k = 0,1,\dots,N-1 \qquad (3)$$

where $X(k)$ represents the frequency-domain spectrum, $x(n)$ is the time-domain speech signal, and N denotes the number of samples in the frame. In practice, the Fast Fourier Transform (FFT) algorithm is used to efficiently compute this transformation as shown in equation (3). Typically, FFTs of size 512 or 1024 are used to ensure sufficient frequency resolution for human speech (below 8 kHz).

4) Mel filter processing

The frequency spectrum obtained from the FFT is then mapped onto the Mel scale to approximate the nonlinear perception of human hearing. This process is performed using a set of triangular filters known as the Mel filterbank.

The output energy of the m^{th} filter is calculated according to equation (4).

$$E_m = \sum_{k=f(m-1)}^{f(m+1)} |X[k]|^2 H_m[k] \qquad (4)$$

where $X[k]$ denotes the magnitude spectrum and $H_m[k]$ is the response of the m^{th} Mel filter. As described in equation (4), the filters are distributed linearly in low frequencies and logarithmically in high frequencies according to the Mel scale. Logarithmic compression is then applied to stabilize variations in energy and emulate human loudness perception, enhancing robustness to amplitude differences [22][23].

5) Cepstral coefficient calculation

After applying the Mel filterbank, logarithmic compression is applied to stabilize energy variations. The resulting log Mel energies are then transformed into the cepstral domain using the discrete cosine transform (DCT).

The cepstral coefficients are computed using equation (5).

$$c_n = \sum_{m=1}^{M} \log(E_m) \cos\left[n\left(m-\frac{1}{2}\right)\frac{\pi}{M}\right], \quad n = 1,2,\dots 13 \qquad (5)$$

where c_n represents the n^{th} cepstral coefficient, S_m denotes the log Mel filterbank energy, and M is the number of Mel filters. Typically, the first 13 coefficients are retained to represent the spectral envelope of the speech signal, as formulated in equation (5).

6) Dynamic coefficients

To capture temporal variations in speech signals, dynamic features are calculated from the static MFCC coefficients. These include the first-order derivative (delta) and the second-order derivative (delta - delta) coefficients.

The delta coefficients are computed using equation (6).

$$Feature\ Vector = [c_n, \Delta c_n, \Delta^2 c_n] \qquad (6)$$

These dynamic features (Δ and Δ^2) describe short-term temporal patterns and transitions between frames, significantly enhancing the discriminative performance of the classifier. The final feature vector, as shown in equation (6), thus integrates both spectral and temporal cues, serving as a robust input representation for the SVM classification stage.

For each utterance (speech command), the 39-dimensional feature vectors (13 static MFCC + 13 delta + 13 delta-delta) were extracted from all frames. To obtain a fixed-length representation per utterance, we computed the mean and variance of each coefficient across the time axis. This resulted in a final utterance-level feature vector of 78 dimensions (39 means + 39 variances). These 78-dimensional vectors were then used as input to the SVM classifier. This aggregation method preserves both the spectral average and temporal variability of the speech signal.

C. Support vector machine (SVM)

SVM is a supervised learning algorithm widely employed for pattern recognition and classification tasks due to its ability to construct an optimal decision boundary that maximizes the separation margin between different classes. In the proposed voice command recognition framework, the extracted MFCC and their temporal derivatives serve as feature vectors, which are then input into an SVM classifier for training and testing. The objective of the SVM optimization problem is to find a hyperplane that best separates the classes in a high-dimensional feature space while minimizing classification errors [24][25][26].

The optimization objective of SVM is formulated as shown in equation (7a) and equation (7b).

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N} \xi_i \qquad (7a)$$

subject to the following constraints:

$$y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i \quad (7b)$$

where w represents the weight vector defining the orientation of the separating hyperplane, b is the bias term, ξ_i are slack variables that allow some misclassification for non-linearly separable data, $y_i \in \{-1, +1\}$ denotes the true class labels, and C is the regularization parameter controlling the trade-off between maximizing the margin and minimizing classification error. The function $\phi(x_i)$ maps the input vector x_i into a higher-dimensional feature space, enabling the classifier to handle non-linear separability through the use of kernel functions. Among various kernel choices, the RBF kernel is particularly effective for speech-related tasks due to its ability to model complex and nonlinear relationships between acoustic features. The RBF kernel is defined in equation (8).

$$K(x_i, x_j) = exp(-\gamma \parallel x_i - x_j \parallel^2) \quad (8)$$

where γ is a kernel parameter that determines the spread of the Gaussian function. As shown in equation (8), a smaller value of γ results in a smoother decision boundary with a larger influence region for each support vector, while a larger γ leads to more complex decision surfaces that may capture fine details in the data at the risk of overfitting. Proper tuning of C and γ is thus essential for achieving high classification accuracy and generalization. During the training phase, SVM identifies a subset of training samples, known as support vectors, which lie closest to the decision boundary and define the optimal hyperplane [27][28].

D. Evaluation metrics

The performance of the proposed SVM-based speech recognition model was evaluated using four primary metrics: accuracy, precision, recall, and F1-score. These metrics collectively describe how effectively the classifier distinguishes between robot command categories. Each prediction is categorized as true positive (TP), true negative (TN), false positive (FP), or false negative (FN). Model performance was assessed using accuracy, precision, recall, and F1-score, averaged across all classes [29][30][31][32]. Computational efficiency was evaluated in terms of model size, parameter count, and inference time.

III. Results and Discussions

A. Hyperparameter optimization

RBF kernel with parameters C = 100 and γ = 0.01 was identified as optimal, achieving 95.8 % validation accuracy. Three kernel functions, linear, polynomial, and RBF, were systematically evaluated under varying hyperparameter configurations. The grid search process exhaustively explored combinations of the penalty parameter (C) and the kernel coefficient (γ) to identify the optimal trade-off between model complexity and generalization performance. Results from cross-validation revealed that the RBF kernel consistently outperformed the linear and polynomial counterparts in capturing nonlinear relationships inherent in the speech feature space. The optimal configuration was obtained with (C = 100) and (γ = 0.01), yielding the highest validation accuracy of 95.8 %. This finding confirms that the RBF kernel effectively enhances model adaptability to complex data distributions, ensuring superior recognition accuracy across varying robot voice command inputs, as shown in Figure 2.

The grid search process evaluates multiple combinations of the penalty parameter (C) and kernel parameter (γ) using 5-fold cross-validation. In each fold, the dataset is partitioned into training and validation subsets, ensuring that all samples are used for validation exactly once. The average accuracy across all folds is used to determine the optimal parameter combination, as illustrated in Figure 2.

The superior performance of the SVM with RBF kernel can be attributed to its ability to effectively handle nonlinear separability in high-dimensional feature spaces. Speech signals represented by MFCC features inherently exhibit complex and nonlinear patterns due to variations in pronunciation, speaker characteristics, and acoustic conditions. The RBF kernel maps these features into a higher-dimensional space where a linear separation becomes feasible.

Furthermore, SVM focuses on maximizing the margin between classes, which improves generalization performance, particularly when the dataset size is relatively limited compared to deep learning approaches. Unlike neural networks that require large-scale data for robust parameter learning, SVM relies on a subset of critical samples (support vectors), making it more efficient and less prone to overfitting in small to medium-sized datasets such as the Google speech commands subset used in this study.

B. Final model performance

The final evaluation of the proposed SVM-based speech recognition model demonstrated consistently high performance across all measured metrics. The results are summarized in Table 1, showing that the model achieved a high overall classification accuracy with balanced precision and recall values, indicating strong generalization capability across all robot command categories.

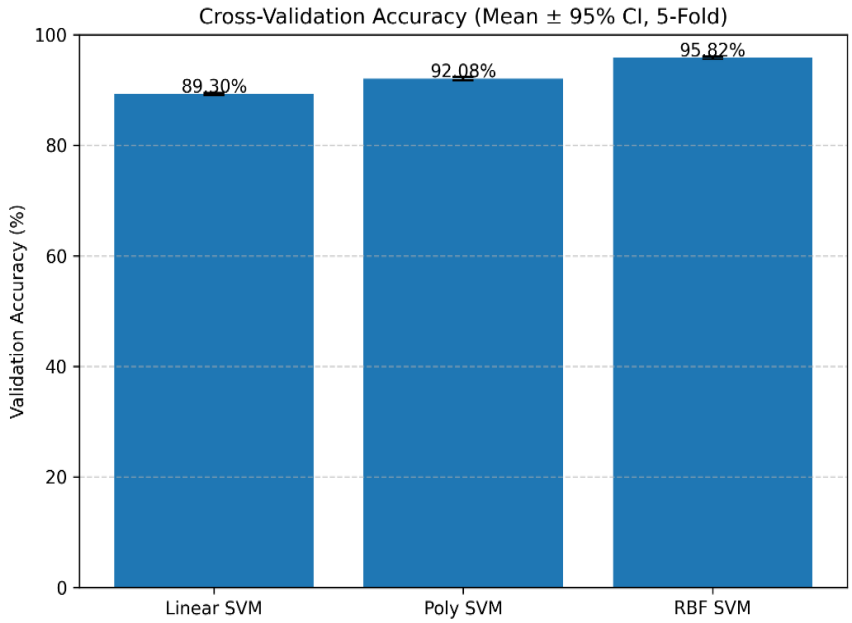

Figure 2. Cross-validation accuracy on various SVM kernels.

The model achieved 96.2 % accuracy, confirming strong generalization across all command classes. The confusion matrix is presented as a heatmap obtained from testing the optimal SVM-RBF model on the eight speech command classes: down, go, left, no, right, stop, up, and yes. Each cell represents the number of predictions for a given pair of true and predicted labels, with diagonal elements corresponding to correctly classified instances and off-diagonal cells indicating misclassifications.

As shown in Figure 3, the matrix demonstrates high classification consistency across all commands, with the majority of predictions concentrated along the diagonal. For instance, the "go" class achieved 1,255 correct classifications, and "stop" reached 1,242. However, some confusion remains between acoustically similar words, notably between "left" and "right" (15 samples of "right" misclassified as "left"), as well as between "no" and "go." These errors can be attributed to phonetic similarity and spectral overlap within the MFCC representation, which poses a common challenge in speech recognition systems.

Overall, the confusion matrix confirms the robust generalization capability of the optimized SVM-RBF classifier, achieving high accuracy while maintaining balanced class performance. Minor misclassifications suggest potential for improvement through enhanced feature discrimination techniques, such as filterbank optimization, data augmentation, or hybrid deep feature integration.

Additionally, variations in speaker pronunciation and speaking rate may further contribute to feature overlap in the MFCC space. This limitation highlights the inherent challenge of using hand-crafted spectral features without incorporating higher-level contextual or phonetic modeling. Future improvements could involve integrating temporal sequence modeling techniques or hybrid architectures to better capture long-term dependencies in speech signals.

Table 1.
Final performance metric of the proposed model.

Metric	Value (%)
Accuracy	96.2
Precision	96.5
Recall	96.0
F1-score	96.2

C. Influence of MFCC configuration

To evaluate the effect of feature dimensionality on recognition performance, three configurations of MFCC were analyzed: static MFCCs, MFCCs with first-order delta coefficients (Δ), and MFCCs with both first- and second-order delta coefficients (Δ and Δ^2). The comparative results are summarized in Table 2.

As shown, increasing the feature dimensionality through the inclusion of dynamic coefficients significantly enhanced recognition accuracy. The baseline 13 static MFCC features achieved 91.5 %,

Table 2.
Performance comparison under different MFCC configurations.

Configuration	Feature Dim.	Accuracy (%)
13 MFCC	13	91.5
MFCC + Δ	26	94.3
MFCC + Δ + Δ^2	39	96.2

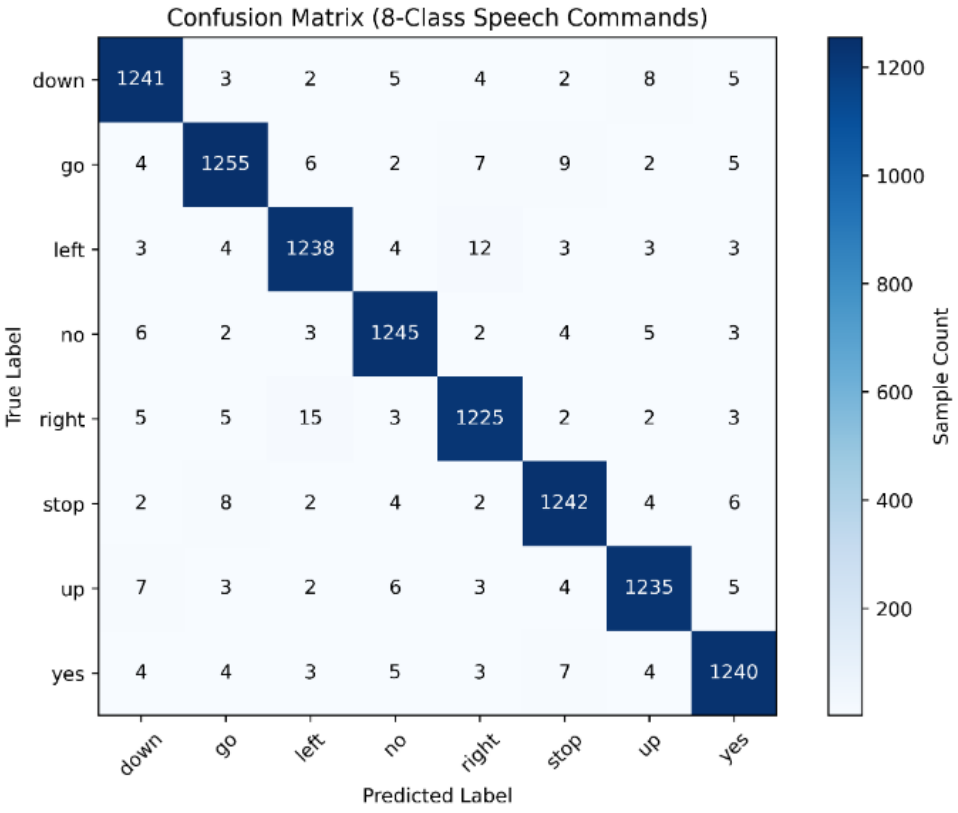

Figure 3. Confusion matrix of the SVM - RBF model.

which improved to 94.3 % with the addition of Δ features. When both Δ and Δ^2 coefficients were included, accuracy further increased to 96.2 %, representing a 4.7 % absolute improvement. This result confirms that temporal dynamics play a crucial role in speech-based robot command recognition, as delta and delta-delta coefficients effectively capture variations in spectral transitions, thereby improving the model's discrimination capability across time-varying speech signals.

The observed performance improvement with the inclusion of delta (Δ) and delta-delta (Δ^2) coefficients can be explained by the temporal nature of speech signals. While static MFCC features capture the spectral envelope at a single time frame, they do not encode the dynamic transitions of speech. In contrast, delta and delta-delta coefficients represent the first and second temporal derivatives, respectively, capturing the rate and acceleration of spectral changes over time.

These temporal dynamics are crucial in distinguishing phonetically similar commands, as speech is inherently non-stationary and influenced by coarticulation effects, where adjacent phonemes affect each other. By incorporating these dynamic features, the model gains additional discriminative information related to speech transitions, which significantly enhances classification accuracy, particularly for acoustically similar words such as "left" and "right" or "go" and "no."

D. Comparison with state-of-the-art

To validate the effectiveness of the proposed method, its performance was compared with several state-of-the-art approaches in speech command recognition. The comparison is summarized in Table 3, which lists the datasets, methods, accuracies, and key contributions of each study.

Table 3.
Performance comparison under different MFCC configurations.

Study	Dataset	Method	Accuracy (%)	Contribution
Our work	Google speech v2	MFCC + SVM-RBF	96.2	Efficient and optimized
Warden (2018)	Google speech v2	CNN	96.6	High complexity
Trong *et al.*	Google speech v1	1D-CNN	95.7	Deep learning
Aldarmaki	Custom	MFCC + SVM	94.8	Small dataset

As presented in Table 3, the proposed MFCC + SVM-RBF system achieves 96.2 % accuracy, which is highly comparable to the CNN-based architectures of Warden and Trong *et al.*, while maintaining significantly lower computational complexity. Unlike deep learning models that require large datasets, high-end GPUs, and extensive training time, the SVM-based approach offers an efficient and lightweight alternative suitable for embedded or real-time robotic systems. These results demonstrate that, through optimized feature extraction and hyperparameter tuning, the proposed system achieves a strong balance between accuracy, efficiency, and deployment practicality, making it highly effective for low-resource speech recognition applications.

It is important to note that the comparison with state-of-the-art methods should be interpreted with caution. Different studies often employ varying datasets, preprocessing techniques, and evaluation protocols, which can significantly influence reported performance. For instance, CNN-based models may be trained on larger datasets with more extensive augmentation strategies, while the proposed method focuses on a lightweight configuration with limited computational resources.

Therefore, the comparison presented in this study primarily aims to provide a general performance reference rather than a strictly controlled benchmark. A fair comparison would require standardized datasets, identical preprocessing pipelines, and consistent evaluation settings, which remains an important direction for future work.

E. **Robustness analysis**

To simulate realistic interference, white Gaussian noise (WGN) at a signal-to-noise ratio (SNR) of 20 dB was added to the test data. This moderate noise level represents real-world acoustic conditions, such as environmental background noise or motor vibrations typically found in robotic environments.

Under these conditions, the model maintained a recognition accuracy of 92.1 %, demonstrating strong resilience against additive noise. However, minor performance degradation was observed in acoustically similar commands; specifically, accuracy for the “go” and “no” classes decreased by 8.3 % and 7.1 %, respectively. While these results indicate that the proposed system sustains reasonable performance at 20 dB SNR, this single condition is insufficient to claim generalized robustness. Future studies should evaluate multiple noise levels (e.g., 0 dB, 10 dB, and 30 dB) and diverse noise types—such as fan noise or background speech—to fully establish the system's reliability. The spectral impact of this noise is illustrated in Figure 4.

As shown in Figure 4, the introduction of white Gaussian noise significantly alters the spectral energy distribution of the signal. This visualization highlights how noise masking affects the clarity of phonetic features, which directly influences the feature extraction process and subsequent classification performance. Understanding these spectral shifts is crucial for developing more advanced noise-reduction algorithms in future iterations of the system.

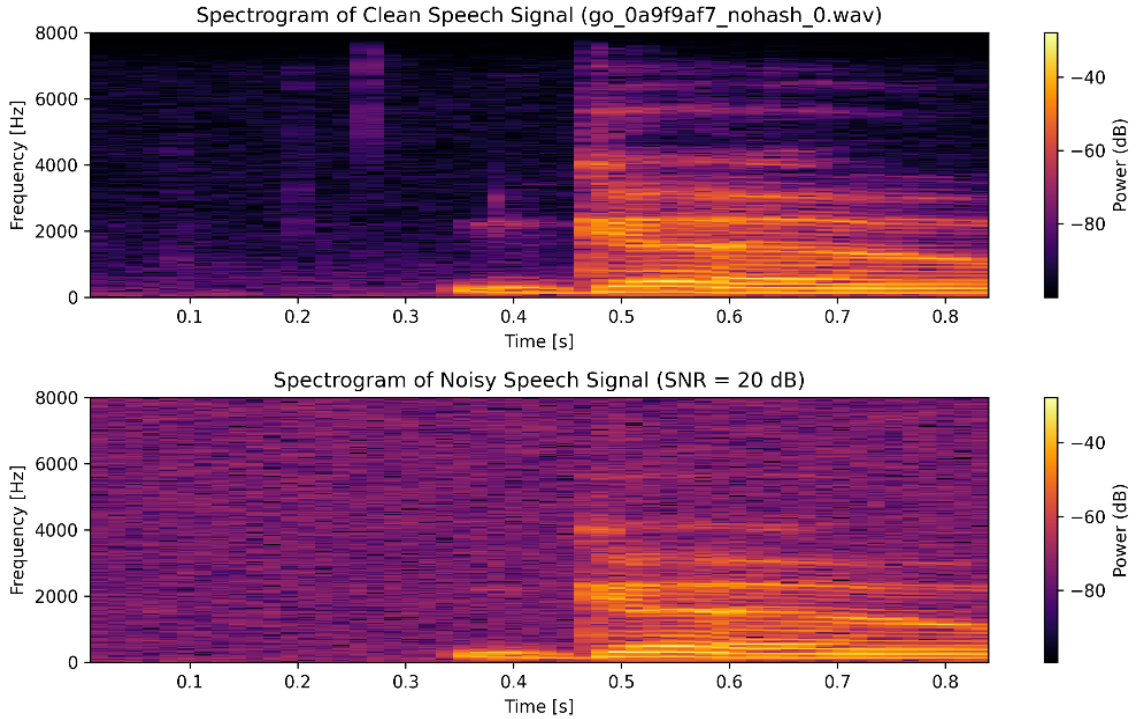

Figure 4. Comparison of speech signal spectrograms under clean conditions and at 20 dB SNR.

Table 4.
Computational efficiency comparison.

Model	Parameters	Model Size	Inference Time	Accuracy
Proposed SVM (RBF)	~6,000	120 KB	5.2 ms	96.2 %
CNN baseline	~500,000	2.5 MB	25.6 ms	~96.6 %

F. Computational efficiency analysis

The computational efficiency of the proposed MFCC–SVM model was evaluated to support its feasibility for deployment in resource-constrained robotic systems. The evaluation focuses on model size, parameter complexity, and inference time.

Based on experimental observations from a comparable implementation under the same feature configuration (78-dimensional MFCC with delta and delta-delta), the SVM model requires approximately 6,000 parameters, resulting in a compact model size of approximately 120 KB. The inference time was measured at approximately 5.2 ms per sample under CPU-based execution on a system equipped with an AMD quad-core processor and 16 GB RAM, without GPU acceleration.

For comparison, deep learning-based approaches such as convolutional neural networks (CNNs) typically require significantly larger computational resources. A representative CNN baseline model requires approximately 500,000 parameters, with a model size of around 2.5 MB and an inference time of approximately 25.6 ms per sample under similar conditions.

However, it should be noted that real-time deployment on actual embedded hardware (e.g., microcontrollers or edge devices) has not yet been experimentally validated. Therefore, while the computational results indicate feasibility, further implementation and testing on target hardware platforms are required to fully confirm real-time performance. The computational comparison between the proposed SVM-RBF model and a representative CNN baseline is summarized in Table 4. As shown in Table 4, the proposed method requires substantially fewer parameters and a smaller model size than the CNN baseline, while maintaining comparable classification accuracy. These results support the suitability of the MFCC–SVM approach for resource-constrained robotic applications.

Based on Table 4, the proposed SVM-RBF model uses approximately 6,000 parameters with a model size of 120 KB and an inference time of 5.2 ms per sample. In contrast, the CNN baseline requires approximately 500,000 parameters, a 2.5 MB model size, and 25.6 ms inference time. This comparison indicates that the proposed approach offers a more computationally efficient solution while achieving competitive accuracy.

IV. Conclusion

The proposed MFCC–SVM system with RBF kernel achieved a high classification accuracy of 96.2 %, demonstrating its effectiveness in modeling non-linear acoustic patterns for voice command recognition. The incorporation of delta and delta-delta features significantly improved performance, confirming the importance of temporal dynamics in speech processing. Compared to deep learning approaches, the proposed method provides a lightweight and computationally efficient alternative suitable for resource-constrained robotic systems. However, the robustness evaluation was limited to a single noise condition (20 dB SNR), and no real-time deployment testing was conducted. Therefore, future work will focus on evaluating the system under diverse acoustic environments and implementing the model on embedded hardware platforms to validate its real-time performance. Additionally, hybrid approaches combining SVM and deep learning techniques will be explored to further enhance system performance.

Acknowledgements

The author would like to thank the Universitas 17 Agustus 1945 Surabaya for supporting this research.

Declarations

Author contribution

Ratna Hartayu: Conceptualization, Methodology, Supervision, Formal analysis, Writing – Review & Editing, Project administration, Correspondence handling. **Santoso:** Investigation, Software, Data Curation, Validation, Formal analysis, Writing – Original Draft, Visualization. **Ahmad Ridho'i:** Methodology, Software, Validation, Investigation, Resources, Writing – Review & Editing. **Ayusta Lukita Wardani:** Formal analysis, Validation, Resources, Writing – Review & Editing. **Yunus Awwalu Romadhon:** Data Curation, Visualization, Investigation, Writing – Review & Editing. All authors reviewed, discussed, approved the final version of the

manuscript, and agreed to be accountable for all aspects of the work

Funding statement

This research did not receive external funding. The study was conducted using the research facilities and institutional resources provided by Universitas 17 Agustus 1945 Surabaya.

Competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The use of AI or AI-assisted technologies

During the preparation of this manuscript, the authors used ChatGPT developed by OpenAI to improve grammar, language clarity, and manuscript readability. Following the use of this AI-assisted technology, the authors carefully reviewed, revised, and validated all content and assume full responsibility for the accuracy, integrity, and originality of the published work.

Additional information

References

[1] R. Martinek, J. Vanus, J. Nedoma, M. Fridrich, J. Frnda, anad A. Kawala-Sterniuk, "Voice communication in noisy environments in a smart house using hybrid LMS+ICA algorithm," *Sensors,* vol. 20, ni. 21, p. 6022, Jan. 2020.

[2] F. Wang and X. Shen, "Research on speech emotion recognition based on teager energy operator coefficients and inverted mfcc feature fusion," *Electronics*, vol. 12, no. 17, p. 3599, Jan. 2023.

[3] M. Cenit and V. Gandhi, "Design and development of the sEMG-based exoskeleton strength enhancer for the legs," *J. Mechatron. Electr. Power Veh. Technol.*, vol. 11, no. 2, pp. 64–74, Dec. 2020.

[4] J. Mishra, T. Malche, and A. Hirawat, "Embedded intelligence for smart home using tinyml approach to keyword spotting," *Eng. Proc.*, vol. 82, no. 1, p. 30, 2024.

[5] E. Rijanto, E. Adiwiguna, A. P. Sadono, M. H. Nugraha, O. Mahendra, and R. D. Firmansyah, "A new design of embedded monitoring system for maintenance and performance monitoring of a cane harvester tractor," *J. Mechatron. Electr. Power Veh. Technol.*, vol. 11, no. 2, pp. 102–110, Dec. 2020.

[6] L. Nwankwo and E. Rueckert, "The conversation is the command: interacting with real-world autonomous robot through natural language," in *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, Mar. 2024, pp. 808–812.

[7] S. Tirronen, S. R. Kadiri, and P. Alku, "The effect of the MFCC frame length in automatic voice pathology detection," *J. Voice*, vol. 38, no. 5, pp. 975–982, Sep. 2024.

[8] B. Tracey *et al.*, "Towards interpretable speech biomarkers: exploring MFCCs," *Sci. Rep.*, vol. 13, no. 1, p. 22787, Dec. 2023.

[9] M. Maayah, A. Abunada, K. Al-Janahi, M. E. Ahmed, and J. Qadir, "Limit access: on-device tinyml based robust speech recognition and age classification," *Discov. Artif. Intell.*, vol. 3, no. 1, p. 8, Feb. 2023.

[10] V. Verma *et al.*, "A novel hybrid model integrating MFCC and acoustic parameters for voice disorder detection," *Sci. Rep.*, vol. 13, no. 1, p. 22719, Dec. 2023.

[11] G. Cerutti, L. Cavigelli, R. Andri, M. Magno, E. Farella, and L. Benini, "Sub-mW keyword spotting on an mcu: analog binary feature extraction and binary neural networks," *IEEE Trans. Circuits Syst. Regul. Pap.*, vol. 69, no. 5, pp. 2002–2012, May 2022.

[12] A. M. Rostami, A. Karimi, and M. A. Akhaee, "Keyword spotting in continuous speech using convolutional neural network," *Speech Commun.*, vol. 142, pp. 15–21, Jul. 2022.

[13] Y. Cai, X. Li, and J. Li, "Emotion recognition using different sensors, emotion models, methods and datasets: a comprehensive review," *Sensors*, vol. 23, no. 5, p. 2455, Jan. 2023.

[14] A. M. Elshewey, M. Y. Shams, N. El-Rashidy, A. M. Elhady, S. M. Shohieb, and Z. Tarek, "Bayesian optimization with support vector machine model for parkinson disease classification," *Sensors*, vol. 23, no. 4, p. 2085, Jan. 2023.

[15] B. Kim and Y. Kwon, "Searching for effective preprocessing method and CNN based architecture with efficient channel attention on speech emotion recognition," *Sci. Rep.*, vol. 15, no. 1, p. 32689, Sep. 2025.

[16] J. Ancilin and A. Milton, "Improved speech emotion recognition with Mel frequency magnitude coefficient," *Appl. Acoust.*, vol. 179, p. 108046, Aug. 2021.

[17] A. Hassan, T. Masood, H. A. Ahmed, H. M. Shahzad, and H. M. Tayyab Khushi, "Benchmarking pretrained models for speech emotion recognition: a focus on xception," *Computers*, vol. 13, no. 12, p. 315, Dec. 2024.

[18] A. Kuzdeuov, R. Gilmullin, B. Khakimov, and H. A. Varol, "An open-source tatar speech commands dataset for iot and robotics applications," in *IECON 2024* - 50th *Annual Conference of the IEEE Industrial Electronics Society,* Chicago, IL, USA: IEEE, Nov. pp. 1-5, 2024.

[19] R. Sumikawa, A. Kosuge, Y.-C. Hsu, K. Shiba, M. Hamada, and T. Kuroda, "A183.4-nJ/inference 152.8- μ W 35-voice commands recognition wired-logic processor using algorithm-circuit co-optimization

technique," *IEEE Solid-State Circuits Lett.*, vol. 7, pp. 22–25, 2024.

[20] Z. Kh. Abdul and A. K. Al-Talabani, "Mel frequency cepstral coefficient and its applications: A review," *IEEE Access*, vol. 10, pp. 122136–122158, 2022.

[21] Y.-L. Chen, N.-C. Wang, J.-F. Ciou, and R.-Q. Lin, "Combined bidirectional long short-term memory with mel-frequency cepstral coefficients using autoencoder for speaker recognition," *Appl. Sci.*, vol. 13, no. 12, p. 7008, Jan. 2023.

[22] D. Cabrera, R. Medina, M. Cerrada, R.-V. Sánchez, E. Estupiñan, and C. Li, "Improved mel frequency cepstral coefficients for compressors and pumps fault diagnosis with deep learning models," *Appl. Sci.*, vol. 14, no. 5, p. 1710, Jan. 2024.

[23] Santoso, T. A. Sardjono, and D. Purwanto, "Optimizing mel-frequency cepstral coefficients for improved robot speech command recognition accuracy," in *2024 International Seminar on Application for Technology of Information and Communication (iSemantic)*, Semarang, Indonesia: IEEE, pp. 284–289, Sep. 2024.

[24] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, Sep. 2020.

[25] V. Blanco, A. Japón, and J. Puerto, "A mathematical programming approach to SVM-based classification with label noise," *Comput. Ind. Eng.*, vol. 172, p. 108611, Oct. 2022.

[26] R. Guido, S. Ferrisi, D. Lofaro, and D. Conforti, "An overview on the advancements of support vector machine models in healthcare applications: A review," Information, vol. 15, no. 4, p. 235, Apr. 2024.

[27] K.-L. Du, B. Jiang, J. Lu, J. Hua, and M. N. S. Swamy, "Exploring kernel machines and support vector machines: principles, techniques, and future directions," *Mathematics*, vol. 12, no. 24, p. 3935, Jan. 2024.

[28] S. Shrivastava, S. Shukla, and N. Khare, "Support vector machine with eagle loss function," *Expert Syst. Appl.*, vol. 238, p. 122168, Mar. 2024.

[29] O. Rainio, J. Teuho, and R. Klén, "Evaluation metrics and statistical tests for machine learning," *Sci. Rep.*, vol. 14, no. 1, p. 6086, Mar. 2024.

[30] M. C. Hinojosa Lee, J. Braet, and J. Springael, "Performance metrics for multilabel emotion classification: Comparing micro, macro, and weighted f1-scores," *Appl. Sci.*, vol. 14, no. 21, p. 9863, Jan. 2024.

[31] M. Conciatori, A. Valletta, and A. Segalini, "Improving the quality evaluation process of machine learning algorithms applied to landslide time series analysis," *Comput. Geosci.*, vol. 184, p. 105531, Feb. 2024.

[32] S. Abdumalikov, J. Kim, and Y. Yoon, "Performance Analysis and Improvement of Machine Learning with Various Feature Selection Methods for EEG-Based Emotion Classification," *Appl. Sci.*, vol. 14, no. 22, p. 10511, Jan. 2024.