

Comparative Evaluation of Random Forest and 1D CNN Using MFCC Features for Embedded Voice-Command Recognition

Helmi Aprianto¹, Dessy Irmawati^{1,*}

¹Undergraduate Program in Applied Electronics Engineering, Universitas Negeri Yogyakarta, Indonesia

E-mail: dessy.irmawati@uny.ac.id *

*Corresponding Author

ABSTRACT

Comparative studies of Random Forest (RF) and Convolutional Neural Network (CNN) classifiers for voice-command recognition typically report only which model is more accurate, without explaining why the weaker model underperforms or assessing whether either model can be deployed on a resource-constrained microcontroller. This study addresses that gap by comparing RF and a one-dimensional CNN for Mel-Frequency Cepstral Coefficient (MFCC)-based voice classification, diagnosing the specific cause of RF's underperformance, and evaluating estimated deployment feasibility on an ESP32-S3 microcontroller. Using 15,400 augmented recordings from the Google Speech Commands dataset across eleven command classes, both models were evaluated on a dedicated test set. An RF classifier evaluated on time-averaged MFCC features reached only 49% test accuracy; the confusion matrix, feature-importance, principal component analysis, and centroid-distance analyses converged on one explanation: averaging across time erases the temporal structure of speech, collapsing acoustically similar classes into overlapping feature-space regions. Retaining the complete, un-averaged MFCC sequence raised the RF test accuracy to 73%, but the resulting model far exceeded a typical microcontroller's flash budget. Conversely, a CNN trained on the identical full-sequence representation reached 93% test accuracy. Based on hardware profiling estimates, this CNN requires only a kilobyte-scale flash memory footprint, replacing the massive multi-megabyte arrays required by RF. These results demonstrate that preserving temporal structure is critical for both classification accuracy and compact model size in embedded voice-command recognition. Future work will address current study limitations by incorporating silence and unknown background classes, and by ensuring strictly speaker-independent evaluation protocols for robust real-world deployment.

This is an open access article under the CC-BY-SA license.



ARTICLE INFO

Article history

Received:

06 August 2026

Revised:

15 September 2026

Accepted:

22 September 2026

Keywords

Convolutional Neural Network
feature extraction

MFCC

model comparison

Random Forest

voice classification

1. Introduction

Choosing a suitable machine-learning classifier is one of the most consequential decisions in building a voice-command recognition system intended for deployment on embedded hardware, because the choice affects not only recognition accuracy but also memory footprint, inference latency, and training-data requirements. Random Forest (RF) is a widely used classical baseline for audio and speech tasks due to its simplicity, interpretability, and comparatively modest data requirements[1], [2], [3]; Yerramreddy et al. [4] compared RF against GMM, CNN, RNN, and KNN for speaker identification using MFCC features, Durgam et al. [5] applied RF to age estimation based on speech features, and Sabuj et al. [6] combined RF with linear predictive coding features for general audio classification. Convolutional Neural Networks (CNN), in contrast, have become the dominant approach for tasks where the temporal or spatial structure of a signal carries discriminative information: Abdel-Hamid et al. [7] demonstrated that CNNs applied to speech spectrograms



outperform earlier statistical models, Avabratha et al. [8] analyzed multimodal speech and facial emotion recognition using both CNN and RF, Clemency et al. [9] applied RF for noise-robust speech emotion recognition using MFCC features, and CNN and LSTM architectures have also been compared using MFCC and GTCC features for pathological voice detection, while multitask audio analysis has been explored for emotion, gender, and speaker recognition [10], [11].

Table 1. Overview of previous comparative research regarding Random Forest and CNN models in audio and voice classification

Study	Classifier(s) Compared	Feature Representation	Task / Domain	Explains Cause of Weaker Model	Embedded Deployment Considered
Yerramreddy et al. [4]	RF, GMM, CNN, RNN, KNN	MFCC	Speaker identification	No	No
Durgam et al. [5]	RF	MFCC	Speaker recognition	No	No
Sabuj et al. [6]	RF	Linear predictive coding	General audio classification	No	No
Abdel-Hamid et al. [7]	CNN	Spectrogram	Speech recognition	No	No
Avabratha et al. [8]	CNN, RF	Speech and facial data	Emotion recognition	No	No
Clemency et al. [9]	CNN	MFCC	Emotion recognition	No	No
Boualoulou et al. [10]	CNN, LSTM	MFCC, GTCC	Pathological voice detection; emotion/gender/speaker recognition	No	No
This study	RF vs. one-dimensional CNN	MFCC (averaged and full-sequence)	Embedded voice-command recognition (ESP32-S3)	Yes	Yes

As summarized in Table 1, the prior studies reviewed above generally report only which classifier achieves higher accuracy: none of them both diagnose the specific cause of the weaker model's underperformance and evaluate whether the resulting model can be deployed on embedded hardware. This study is distinguished from all of them by addressing both aspects explicitly, as summarized in the final row of Table 1. Despite this substantial body of comparative work, many studies report only the final accuracy achieved by each architecture without diagnosing why a weaker model underperforms at the level of feature geometry, and few explicitly quantify the storage cost of each candidate model against the flash and RAM constraints of a target embedded platform, a practical concern that is often decisive in real deployments even when accuracy differences are already well documented in the literature. This omission matters because a system designer choosing between RF and CNN for a microcontroller-based product needs to know not only which model is more accurate, but concretely why one architecture succeeds and the other fails, and whether either candidate can physically fit within the target device's memory budget.

This study addresses that gap directly, motivated by the practical need to select a classifier for an ESP32-S3-based, hands-free presentation-navigation voice interface. Using a common dataset, feature extraction pipeline, and evaluation protocol, RF and a one-dimensional CNN are trained and compared on an eleven-class MFCC-based voice command task. The specific contributions of this paper are threefold: (1) a multi-angle diagnosis of RF's limited accuracy combining confusion-matrix analysis, Gini feature importance, principal component analysis (PCA), and Euclidean distance between class centroids, tracing the underperformance to the loss of temporal information inherent in time-averaged MFCC features; (2) a quantified accuracy-versus-model-size trade-off across two RF feature representations and a CNN, evaluated explicitly against the flash-storage constraints of an embedded target; and (3) a seven-configuration CNN architecture ablation that empirically justifies the final network design.

2. Method

2.1 Dataset and Augmentation

The dataset used in this study was obtained from the Google Speech Commands v0.02 corpus downloaded from Kaggle. Eleven command-relevant classes were manually selected, consisting of *forward*, *backward*, and the digits *one* through *nine*. From each selected class, 1,400 recordings were used, resulting in a total of 15,400 audio recordings.

The selected recordings were manually organized into separate training, validation, and test folders. The dataset was divided into training, validation, and test sets using a 70:20:10 ratio, resulting in 10,780 training recordings, 3,080 validation recordings, and 1,540 test recordings. Each partition contained 980, 280, and 140 recordings per class, respectively. Before feature extraction, the training and validation recordings were standardized to a duration of 1 s. Recordings shorter than 1 s were zero-padded exclusively at the beginning of the waveform. White Gaussian noise with zero mean and a standard deviation of 0.005 was subsequently added to the training and validation recordings. The test recordings were not subjected to augmentation and were kept separate throughout the model development process.

For the CNN, the training, validation, and test partitions were imported into Edge Impulse, where the training and validation sets were used for model development and the test set was reserved for final evaluation. No five-fold cross-validation was applied to the CNN. For the Random Forest (RF) models, five-fold stratified cross-validation was applied during model development. The same independently separated test set used for the CNN was also reserved for the final evaluation of the RF models. Therefore, although the RF and CNN used different model-development procedures, their final test performance was evaluated using the same test recordings.

The dataset partitioning was not explicitly based on speaker identity. Therefore, speaker-independent separation between the training, validation, and test sets could not be guaranteed. This limitation is considered when interpreting the results, and speaker-independent evaluation is recommended for future work.

2.2 Feature Extraction Pipeline

A common preprocessing pipeline was applied identically ahead of both classifiers, so that any accuracy difference could be attributed to the classifier itself rather than to inconsistent feature engineering [12]. Each 16 kHz recording was first passed through a pre-emphasis high-pass filter, then divided into overlapping frames and smoothed with a Hamming window and finally normalized in amplitude.

Table 2 lists the exact MFCC configuration used, and the amplitude-normalization step is specified as Cepstral Mean Normalization (CMN): the mean of each cepstral coefficient is subtracted across all frames of a recording to reduce channel and background-level variability.

Two representations of this feature set were evaluated for RF: a 13-value vector obtained by averaging each coefficient across all frames, and the full 1,274-value sequence (98 frames $\times$ 13 coefficients) with no temporal averaging; the CNN was trained exclusively on the latter, full-sequence representation, reshaped to preserve frame order.

Table 2. MFCC feature-extraction configuration

Parameter	Value	Justification
Pre-emphasis coefficient (α)	0.98	Compensates for high frequencies suppressed during human speech production
Frame length	25 ms	Captures phonetic characteristics within short command words
Frame stride (hop size)	10 ms	Standard overlap for MFCC analysis; yields ~98 frames per 1 s utterance
Window function	Hamming	Minimizes discontinuity at frame boundaries
FFT size	256	Adequate frequency resolution with low computational cost
Number of Mel filters	32	Perceptually motivated frequency representation
Frequency range	0–8,000 Hz	Full audible range up to the Nyquist frequency for 16 kHz audio
Cepstral coefficients retained	13	Standard MFCC coefficient count for speech classification
Amplitude normalization	Cepstral Mean Normalization (CMN)	Reduces channel/background-level variability across recordings

2.3 Random Forest Configuration

Random Forest (RF) was used as the classification model to evaluate the impact of MFCC feature representation on voice command recognition performance. Two forms of feature representation were compared: a 13-value vector (time-averaged MFCCs) and a full sequence consisting of 1,274 features (98 frames $\times$ 13 coefficients) without averaging.

In both experiments, the RF was configured using identical fixed parameters, as summarized in Table 3. This configuration included 100 decision trees ($n_estimators = 100$), $max_depth = 25$, $min_samples_split = 5$, $min_samples_leaf = 2$, $max_features = \log_2$, bootstrap enabled, and $random_state = 42$ to ensure reproducibility. No hyperparameter tuning was performed for the number of trees. Both RF variants strictly used the exact same 100 trees to ensure a fair comparison. Therefore, the difference in model storage size purely reflects the structural complexity and the actual number of nodes in each tree, rather than a difference in the number of trees. Model selection and performance estimation used five-fold stratified cross-validation [13], [14], which was applied identically to both the averaged and the full-sequence feature representations.

Table 3. Random Forest hyperparameter configuration

Parameter	Value
$n_estimators$	100
Max_depth	25
$Min_samples_split$	5
$Min_samples_leaf$	2
$Max_features$	$\log_2$
Bootstrap sampling	True
$random_state$	42
Cross-validation scheme	5-fold, stratified

2.4 CNN Architecture Search

Seven distinct CNN setups were tested to determine the optimal architecture by adjusting convolutional block counts, filter sizes per block, and dropout rates following each pooling layer (Table 4), following established design principles for one-dimensional convolutional networks applied to sequential sensor and audio data [15], [16], [17], [18]. Every setup underwent 100 epochs of

training utilizing the Adam optimizer (0.0005 learning rate) and a batch size of 32, applying the identical 1,274-dimension MFCC sequence as the full-feature RF model. Training and validation accuracy were compared using a paired significance test to check whether the gap between the two was statistically meaningful at each configuration.

Table 4. CNN architecture ablation across seven candidate configurations

Exp.	Conv1D Blocks (filters)	Dropout	Training Acc.	Loss
1	3 (8-16-32)	-	84.7%	0.7
2	3 (32-64-128)	-	89%	0.4
3	3 (32-64-128)	0.2	92%	0.33
4	3 (32-64-128)	0.3	92.8%	0.3
5	3 (32-64-128)	0.4	92.7%	0.28
6	5 (32-64-64-128-128)	0.4	94.4%	0.19
7	5 (32-64-64-128-128)	0.6	93%	0.25

Experiment 6, as depicted in Fig. 1, comprising five convolutional blocks with 32-64-64-128-128 filters and a dropout rate of 0.4, Experiment 6, comprising five convolutional blocks with 32-64-64-128-128 filters and a dropout rate of 0.4, was selected as the optimal architecture because it demonstrated the strongest generalization capability. Rather than relying on training metrics, the selection was strictly based on validation performance; Experiment 6 achieved the highest validation accuracy (94.4%) and the lowest validation loss (0.19) among the evaluated configurations. Increasing the dropout rate further to 0.6 (Experiment 7) reduced the validation accuracy to 93.0%, indicating that excessive regularization began to discard discriminative temporal features, while architectures with fewer blocks (Experiments 1-4) consistently underfit the dataset.

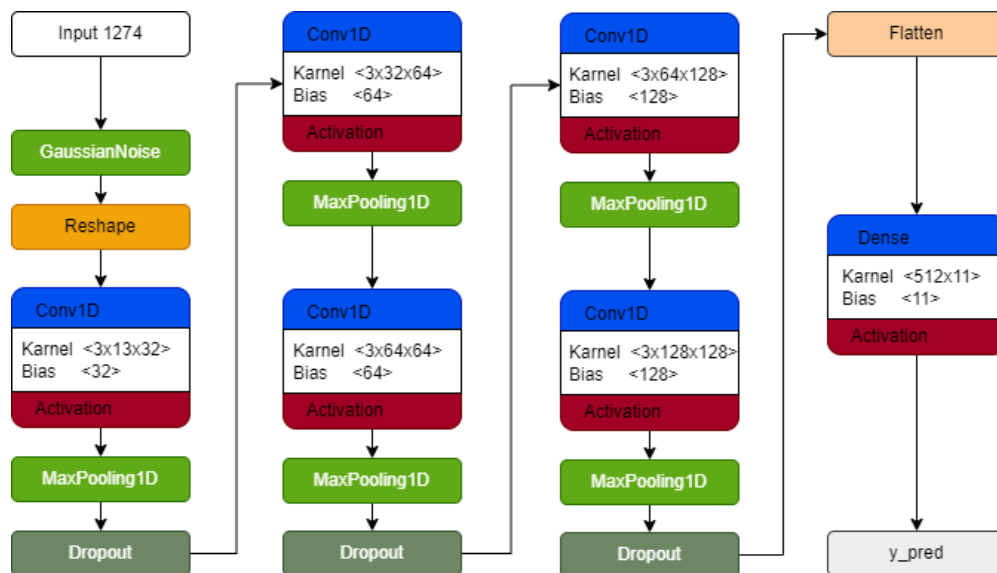


Fig. 1. Final CNN architecture (Experiment 6): an input of 1,274 MFCC values passes through five Conv1D-MaxPooling-Dropout blocks (32-64-64-128-128 filters) before a dense soft-max classification layer over eleven command classes

Table 5. CNN configurations experiment 6

Layer (Type)	Configuration	Output Shape	Activation
Input	1274 MFCC values	(1274)	-
GaussianNoise	stddev = 0.2	(1274)	-
Reshape	target_shape = [98, 13]	(98, 13)	-
Conv1D_1	32 filters, kernel size = 3, padding = same	(98, 32)	ReLU
MaxPooling1D_1	pool size = 2, strides = 2, padding = same	(49, 32)	-
Dropout_1	rate = 0.4	(49, 32)	-
Conv1D_2	64 filters, kernel size = 3, padding = same	(49, 64)	ReLU
MaxPooling1D_2	pool size = 2, strides = 2, padding = same	(25, 64)	-
Conv1D_3	64 filters, kernel size = 3, padding = same	(25, 64)	ReLU
MaxPooling1D_3	pool size = 2, strides = 2, padding = same	(13, 64)	-
Dropout_2	rate = 0.4	(13, 64)	-
Conv1D_4	128 filters, kernel size = 3, padding = same	(13, 128)	ReLU
MaxPooling1D_4	pool size = 2, strides = 2, padding = same	(7, 128)	-
Conv1D_5	128 filters, kernel size = 3, padding = same	(7, 128)	ReLU
MaxPooling1D_5	pool size = 2, strides = 2, padding = same	(4, 128)	-
Dropout_3	rate = 0.4	(4, 128)	-
Flatten	-	(512)	-
Dense (Output)	11 units	(11)	Softmax

2.5 Evaluation Metrics

Performance for both classifiers was assessed through overall accuracy, per-class metrics (precision, recall, and F1-score), alongside confusion matrices. To clarify, rather than merely report, RF's behavior, three additional analyses were applied to the averaged-feature RF model: Gini feature importance across the 13 MFCC coefficients, two-dimensional Principal Component Analysis (PCA) of the feature space, and the pairwise Euclidean distance between the centroid (mean feature vector) of each of the eleven command classes.

3. Results and Discussion

3.1 Random Forest on Time-Averaged MFCC Features

The RF classifier trained using time-averaged 13-dimensional MFCC values achieved an average five-fold cross-validation accuracy of 47%. The confusion matrix shown in Fig. 2 reveals a faint yet distinct diagonal pattern: the "six" class was the best recognized, with 59 validation samples correctly classified, whereas classes such as "one" (26 correct) and "five" (34 correct) were frequently confused with multiple neighboring classes rather than a single dominant alternative a result consistent with the classifier's difficulty in establishing sharp decision boundaries within the feature space.

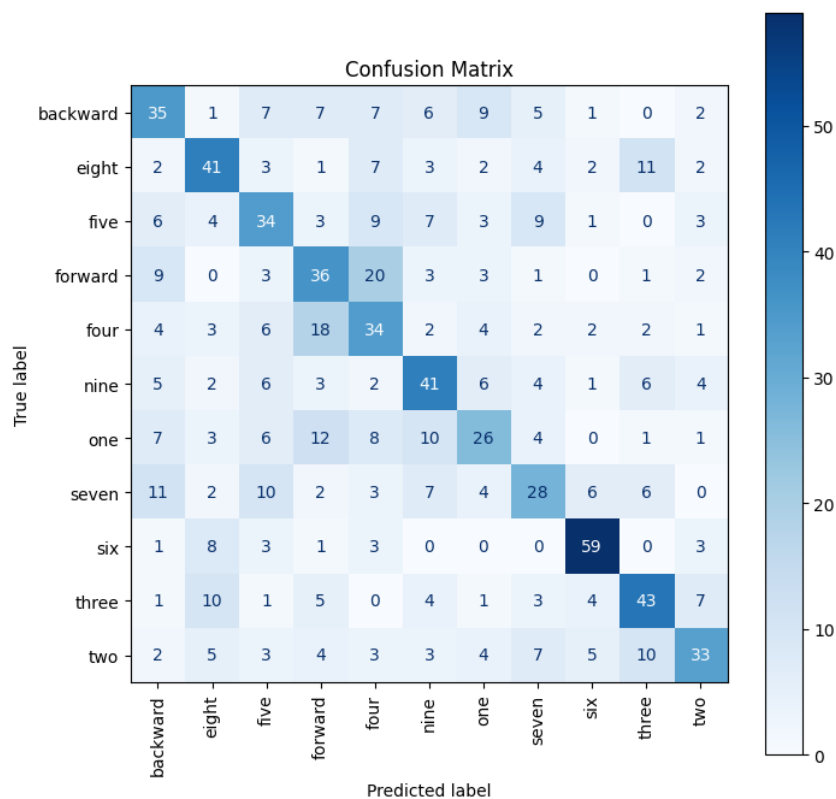


Fig. 2. Confusion matrix of the Random Forest model on time-averaged 13-value MFCC features (validation data)

Fig. 3 shows the Gini feature-importance scores for the 13 MFCC coefficients used by this model. Every coefficient contributes to classification, but contributions are nearly uniform: the first and third coefficients are only marginally more important than the rest, and the single highest importance value recorded is approximately 12%. This near-uniform distribution indicates that no individual MFCC coefficient carries dominant discriminative power once the temporal dimension has been averaged away, and that class-distinguishing information is instead spread thinly across many coefficients, none of which RF's axis-aligned decision trees can exploit strongly on its own.

Once it was confirmed that all features had been utilized, the analysis of characteristic similarities between classes was reinforced by visualizing the distribution of MFCC values. This visualization was achieved by projecting the high-dimensional MFCC data onto principal components using the PCA technique, allowing the data distribution patterns to be observed visually.

The visualization results revealed that most classes did not form clearly distinct clusters; instead, they overlapped within the feature space. This indicates that many samples from different classes share similar feature representations. Such overlap suggests similarities in spectral characteristics across certain classes such as words with similar phonetic structures, so that reducing the capability of standard models to differentiate them properly, which in turn results in low classification accuracy.

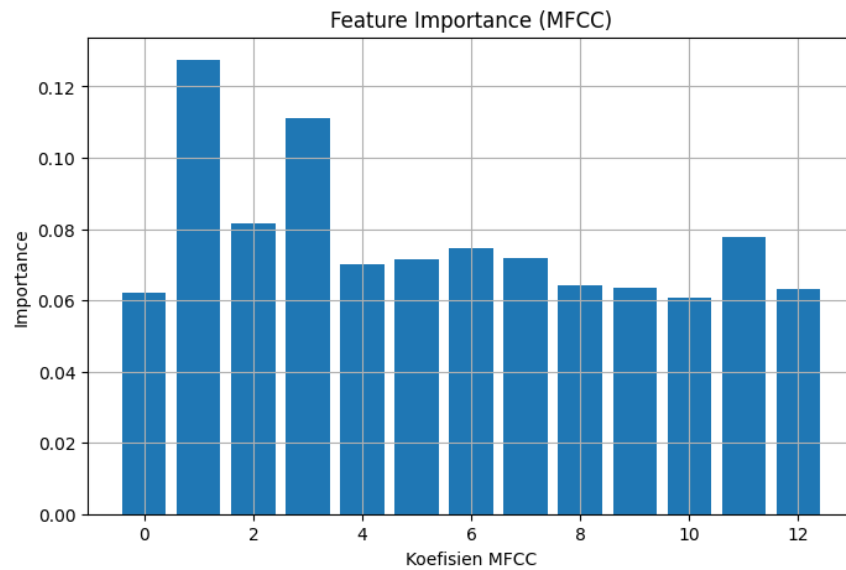


Fig. 3. Feature importance MFCC

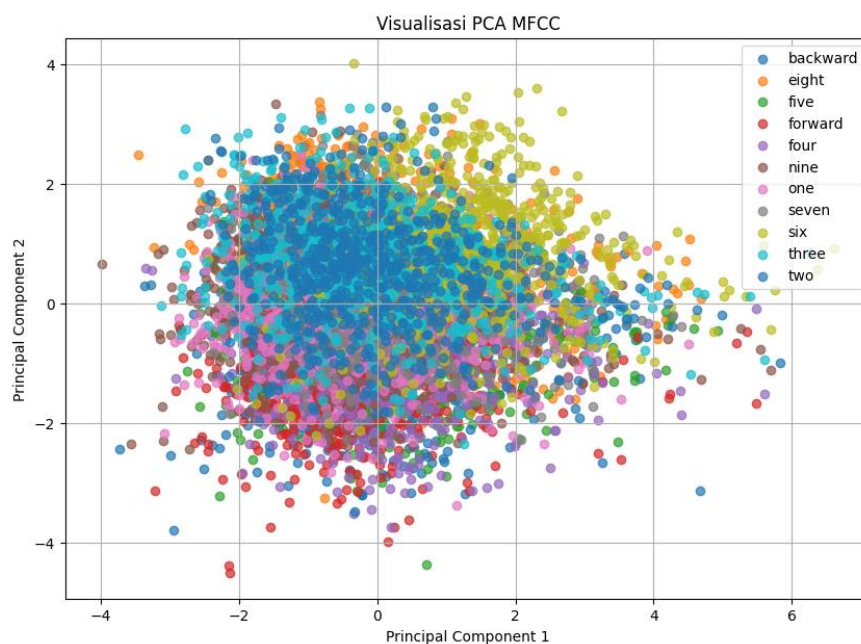


Fig. 4. Visualization PCA MFCC

A two-dimensional PCA projection of the averaged MFCC feature space (Fig. 4) shows that most classes do not form clearly separated clusters and instead overlap substantially. To quantify this overlap, as requested by both reviewers, the pairwise Euclidean distance between class centroids was computed and is reported in Table 6.

Table 6. Pairwise Euclidean distance between class centroids (13 time-averaged MFCC features)

	back.	eight	five	forw.	four	nine	one	seven	six	three	two
backward	0.00	2.23	1.45	1.19	1.22	1.46	1.32	1.61	2.78	1.96	1.73
eight	2.23	0.00	2.13	2.22	2.17	1.76	1.87	1.66	1.94	0.75	1.02
five	1.45	2.13	0.00	1.98	1.44	1.52	1.47	1.36	2.33	2.16	2.04
forward	1.19	2.22	1.98	0.00	0.83	1.99	1.55	2.11	2.84	2.04	1.70
four	1.22	2.17	1.44	0.83	0.00	1.94	1.40	1.81	2.46	2.16	1.74
nine	1.46	1.76	1.52	1.99	1.94	0.00	0.87	1.40	2.73	1.51	1.54
one	1.32	1.87	1.47	1.55	1.40	0.87	0.00	1.47	2.65	1.70	1.38
seven	1.61	1.66	1.36	2.11	1.81	1.40	1.47	0.00	1.63	1.57	1.47
six	2.78	1.94	2.33	2.84	2.46	2.73	2.65	1.63	0.00	2.19	2.00
three	1.96	0.75	2.16	2.04	2.16	1.51	1.70	1.57	2.19	0.00	0.77
two	1.73	1.02	2.04	1.70	1.74	1.54	1.38	1.47	2.00	0.77	0.00

To explicitly quantify the impact of this feature-space overlap on the model's final generalization, the Random Forest classifier was evaluated on the held-out test set. The detailed predictive performance encompassing precision, recall, and F1-score for each of the eleven command classes is summarized in the classification report (Table 7). Furthermore, the specific misclassification patterns resulting from these acoustic similarities are illustrated in the corresponding test-set confusion matrix (Fig. 5).

Table 7. Classification of testing data using the average of 13 MFCCs

Class	Precision	Recall	F1-Score	Support
backward	0.42	0.50	0.46	192
eight	0.53	0.50	0.51	187
five	0.52	0.54	0.53	199
forward	0.44	0.49	0.47	191
four	0.42	0.43	0.43	194
nine	0.48	0.48	0.48	200
one	0.45	0.36	0.40	190
seven	0.48	0.46	0.47	190
six	0.70	0.75	0.72	194
three	0.46	0.43	0.44	196
two	0.44	0.41	0.42	197
accuracy			0.49	2130
macro avg	0.49	0.49	0.48	2130
weighted avg	0.49	0.49	0.48	2130

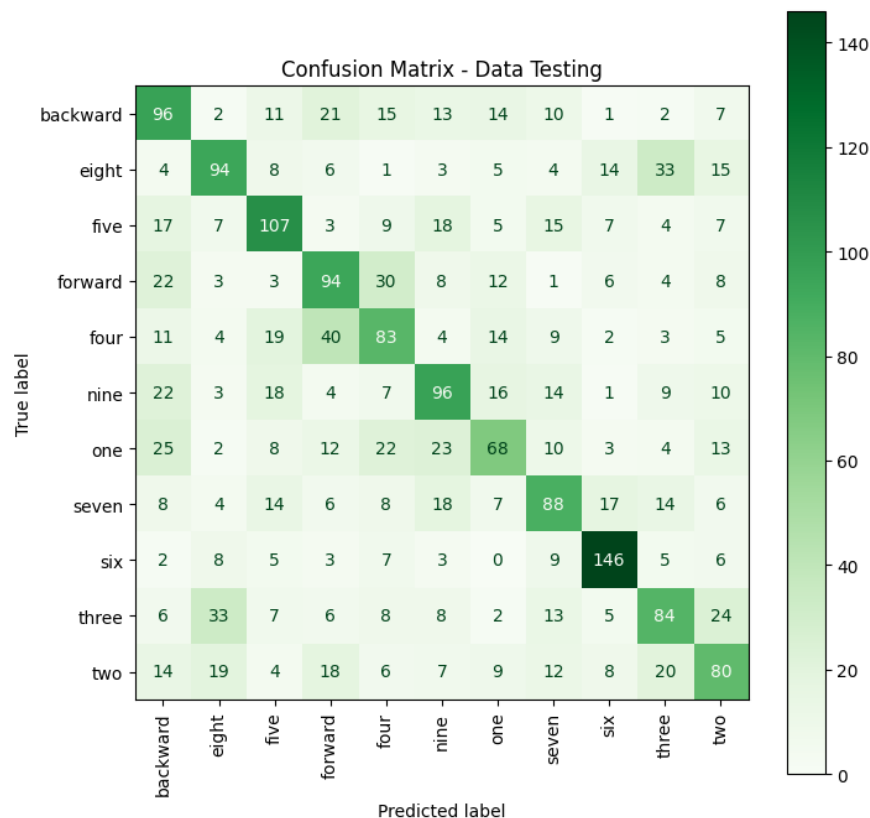


Fig. 5. Confusion matrix of testing data using the average of 13 MFCCs

3.2 Random Forest on Full, Un-Averaged MFCC Sequences

Retaining the full MFCC sequence consisting of 1,274 values instead of averaging them down to 13 values raised the 5-fold cross-validation accuracy of the RF model to an average of 72% across the five folds; this represents an increase of approximately 25 percentage points compared to the model using averaged features. This confirms that the initial performance limitations of the RF model were primarily due to the loss of temporal information, rather than inherent characteristics of the classifier family itself.

To validate this improvement on unseen data, the full-sequence RF model was subsequently evaluated on the held-out test set, achieving 73% overall accuracy. As detailed in the test-set classification report (Table 8), preserving the temporal sequence allowed the model to significantly elevate its macro F1-score to 73%. Furthermore, the corresponding test-set confusion matrix (Fig. 6) visually confirms a substantial reduction in misclassification clusters compared to the time-averaged model, demonstrating that the full temporal structure is essential for correctly differentiating acoustically similar command words.

Table 8. Classification of testing data using the average of full MFCCs

Class	Precision	Recall	F1-Score	Support
backward	0.76	0.79	0.78	192
eight	0.77	0.82	0.80	187
five	0.69	0.62	0.66	199
forward	0.63	0.65	0.64	191
four	0.69	0.65	0.67	194
nine	0.70	0.69	0.70	200
one	0.64	0.63	0.63	190
seven	0.88	0.79	0.83	190
six	0.78	0.86	0.82	194
three	0.83	0.69	0.75	196
two	0.66	0.80	0.72	197
accuracy			0.73	2130
macro avg	0.73	0.73	0.73	2130
weighted avg	0.73	0.73	0.73	2130

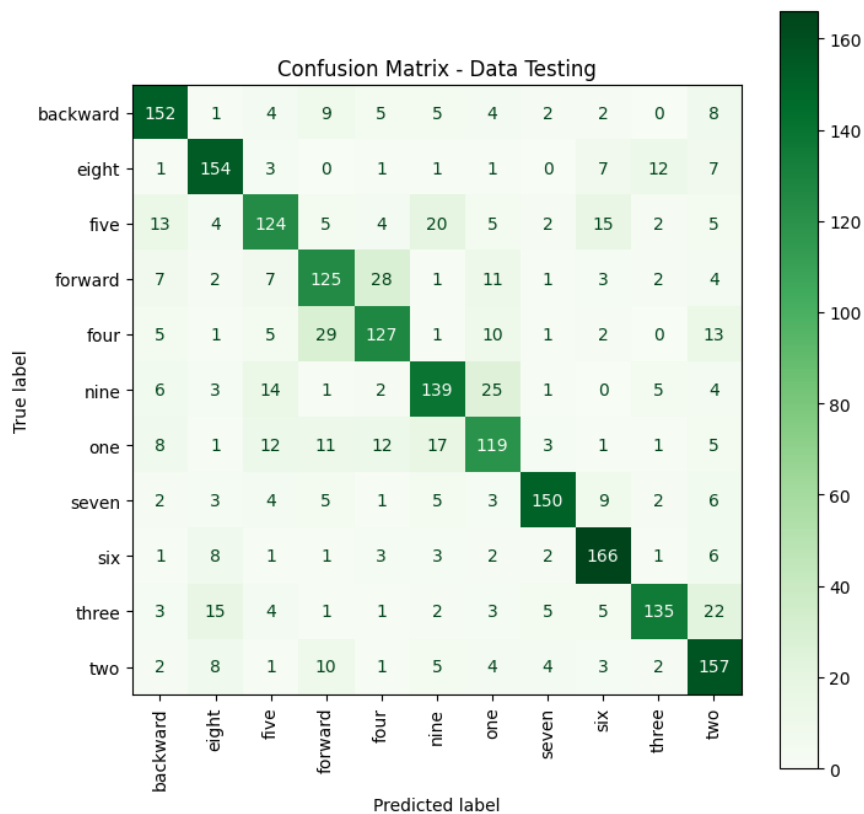


Fig. 6. Confusion matrix of testing data using the average of full MFCCs

This gain, however, came at a steep storage cost: the serialized model grew to 32.5 MB, and to roughly 113.8 MB once converted into a C-header array suitable for firmware embedding. The C-

header representation clearly indicates that the RF models impose a substantially larger storage requirement than the CNN model. It should be noted that while the header size itself should not be interpreted as the exact compiled firmware flash usage, it reflects the massive memory footprint inherent to the ensemble of deep decision trees.

3.3 CNN Performance

The final CNN architecture (Experiment 6 in Table 4) was trained for the full 100 epochs, with illustrated by the training and validation curves in Fig. 7 and Fig. 8. Both training and validation accuracy rise sharply within the first ten epochs and continue improving gradually thereafter, converging to 94% validation accuracy with a small and stable gap between the training and validation curves throughout training, indicating that the model generalizes well rather than memorizing the training set.

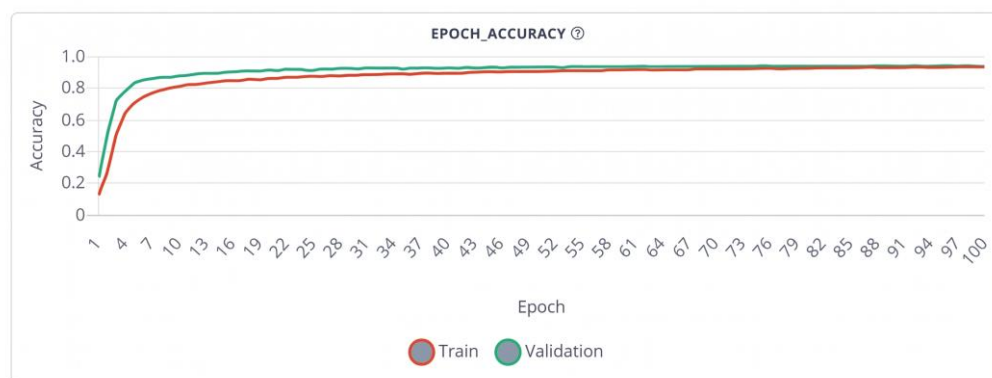


Fig. 7. Accuracy curve of the CNN model across 100 training epochs



Fig. 8. Loss curve of the CNN model across 100 training epochs

On the held-out test set, the CNN reached an overall accuracy of 93% (Fig. 8). Every class exceeded 120 correct predictions out of approximately 140 test samples. The previously most problematic class for RF, “four”, still shows its largest confusion with “forward” (14 of 148 samples). The fact that this specific pair was also heavily confused and exhibited substantial overlap in the RF analysis confirms that this misclassification stems from a genuine acoustic similarity in the dataset rather than an artifact of either classifier. Critically, “six”, which was already identified as the best-recognized class in the RF confusion matrix, remains one of the strongest-performing classes for the CNN as well, with 136 of its test samples correctly classified.

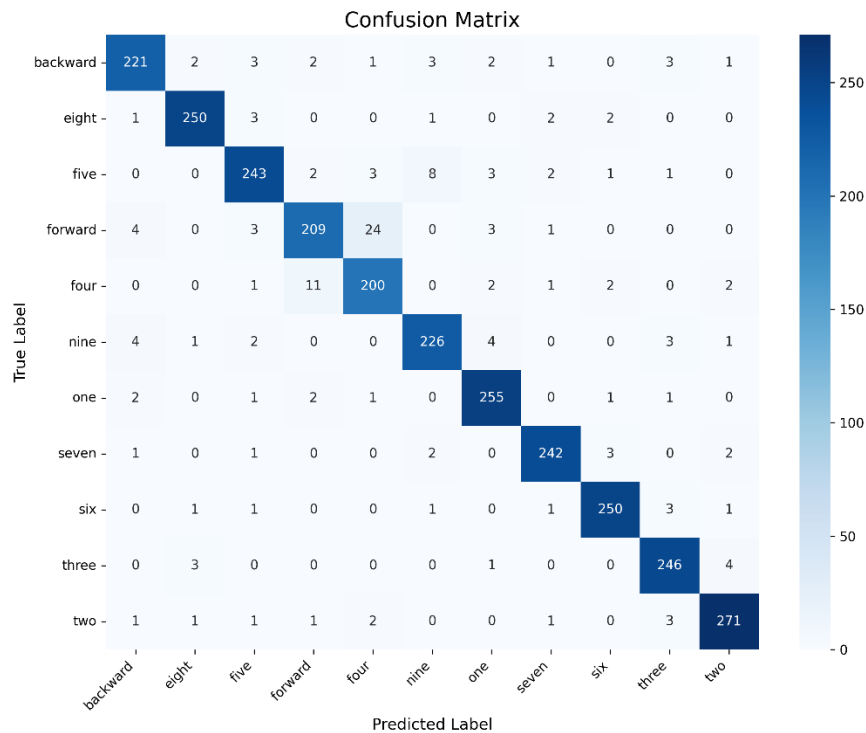


Fig. 9. Performance evaluation of the final CNN model on the test dataset, showing a 93% overall accuracy via confusion matrix

To further evaluate this robust performance across individual commands, detailed metrics are presented in the test-set classification report (Table 9). Consistent with the visual evidence from the confusion matrix as shown in Fig. 9, the CNN achieved macro-averaged precision, recall, and F1-scores of 93%. Unlike the Random Forest variant which struggled to balance false positives and false negatives for words with temporal characteristics the CNN maintained high and stable F1-scores across all eleven classes. This consistency demonstrates the CNN's superior ability to fully leverage sequential MFCC representations to establish precise decision boundaries, thereby successfully overcoming the feature space overlap that posed a challenge for classical baseline methods.

Table 9. Classification report of testing data using CNN

Class	Precision	Recall	F1-Score	Support
backward	0.91	0.94	0.93	140
eight	0.92	0.95	0.94	140
five	0.93	0.94	0.93	140
forward	0.90	0.86	0.88	140
four	0.91	0.86	0.88	140
nine	0.93	0.92	0.93	140
one	0.94	0.99	0.96	140
seven	0.96	0.96	0.96	140
six	0.94	0.97	0.95	140
three	0.95	0.94	0.94	140
two	0.96	0.92	0.94	140
accuracy			0.93	1540
macro avg	0.93	0.93	0.93	1540
weighted avg	0.93	0.93	0.93	1540

3.4 Comparative Synthesis

Table 10 summarizes the three models evaluated in this study. To ensure a fair comparison, all models were evaluated on the identical held-out test set. On this test data, the CNN (93% test accuracy) outperforms the best RF variant trained on full-sequence features (73% test accuracy) by 20 percentage points, and outperforms the baseline averaged-feature RF (49% test accuracy) by 44 percentage points, while requiring only a small fraction of the RF's storage footprint. This pattern is consistent with, and directly explained by, the geometric evidence gathered from the RF analysis: because command classes such as “forward”/“four” and “eight”/“three” occupy overlapping regions once their MFCC sequences are averaged into a single vector, no amount of additional trees can recover the discriminative information that averaging destroys.

The CNN's convolutional filters, by contrast, operate directly on the full 98-frame sequence and can learn local, time-localized patterns, such as the precise timing and shape of a phoneme transition, that distinguish these acoustically similar words [19]. This architecture explains both its higher accuracy and its markedly smaller deployable footprint: based on compilation estimates for the target hardware, a compact set of learned filters requires only a small amount of estimated flash memory, replacing the massive, multi-megabyte C-header arrays required by the deep decision trees of the RF models and demonstrating strong estimated deployment feasibility on the ESP32-S3.

Table 10. Comparison of the three evaluated models

Model	Feature Representation			Accuracy	Estimated Flash	Deployable on ESP32-S3?
Random Forest	13-value averaged MFCC			49% test	44.3 MB (~155.2 MB as C header)	No
Random Forest	1,274-value sequence	full	MFCC	73% test	32.5 MB (~113.8 MB as C header)	No
CNN (1D, 5 layers)	1,274-value sequence	full	MFCC	93% test	219.1 KB	Yes

4. Conclusion

This study compared Random Forest and a one-dimensional CNN for MFCC-based voice command classification and traced the source of RF's underperformance using confusion-matrix, feature-importance, PCA, and Euclidean centroid-distance analyses, all of which converged on the same explanation: averaging MFCC features across time destroys the temporal structure that distinguishes acoustically similar command words. Restoring that temporal structure by retaining the full 1,274-value MFCC sequence raised the RF test accuracy from 49% to 73%, but produced model files far too large for a microcontroller's flash budget. A CNN trained on the same full-sequence representation, selected after a seven-configuration architecture ablation, reached 94% validation and 93% test accuracy.

Based on hardware profiling estimates, this CNN demonstrates strong estimated deployment feasibility for resource-constrained microcontrollers like the ESP32-S3. These findings indicate that, for embedded voice-command classification, an architecture capable of exploiting temporal structure is more decisive for both accuracy and deployability than either additional raw features or additional trees. However, this study is limited by the absence of 'unknown' or silence background classes which are critical for continuous real-world keyword spotting and the lack of a strictly speaker-independent evaluation protocol. Future work should address these limitations, examine quantized and pruned CNN variants to further reduce memory footprint, and extend the comparison to additional classical baselines such as gradient-boosted trees and support vector machines under the same feature-extraction pipeline.

Acknowledgment

The authors thank the Electronics Engineering Study Program, Faculty of Vocational Studies, Universitas Negeri Yogyakarta, for the facilities and support provided during model development and evaluation.

References

- [1] M. M. Rezapour Mashhadi and K. Osei-Bonsu, "Speech emotion recognition using machine learning techniques: Feature extraction and comparison of convolutional neural network and random forest," *PLoS One*, vol. 18, no. 11, p. e0291500, Nov. 2023, doi: 10.1371/journal.pone.0291500.
- [2] B. Charbuty and A. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning," *Journal of Applied Science and Technology Trends*, vol. 2, no. 01, pp. 20–28, Mar. 2021, doi: 10.38094/jastt20165.
- [3] P. Taneja, R. Bhatia, S. Gaur, and J. Priyani, "Beyond Words: Extracting Emotions from Speech with AI Techniques," in *2023 IEEE 4th Annual Flagship India Council International Subsections Conference (INDISCON)*, IEEE, Aug. 2023, pp. 1–6. doi: 10.1109/INDISCON58499.2023.10270809.
- [4] D. R. Yerramreddy, J. Marasani, P. S. V. Gowtham, S. Yashwanth, P. S. S., and A. K., "Speaker Identification Using MFCC Feature Extraction: A Comparative Study Using GMM, CNN, RNN, KNN and Random Forest Classifier," in *2023 Second International Conference on Trends in Electrical, Electronics, and Computer Engineering (TEECCON)*, IEEE, Aug. 2023, pp. 287–292. doi: 10.1109/TEECCON59234.2023.10335892.
- [5] L. kantham Durgam and R. kumar Jatoth, "Age Estimation based on MFCC Speech Features and Machine Learning Algorithms," in *2022 IEEE International Symposium on Smart Electronic Systems (iSES)*, IEEE, Dec. 2022, pp. 398–401. doi: 10.1109/iSES54909.2022.00087.
- [6] H. H. Sabuj *et al.*, "A Comparative Study of Machine Learning Classifiers for Speaker's Accent Recognition," in *2023 IEEE World AI IoT Congress (AIIoT)*, IEEE, Jun. 2023, pp. 0627–0632. doi: 10.1109/AIIoT58121.2023.10174511.
- [7] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional Neural Networks for Speech Recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 22, no. 10, pp. 1533–1545, 2014, doi: 10.1109/TASLP.2014.2339736.
- [8] V. V. Avabratha, S. Rana, S. Narayan, S. Y. Raju, and S. S., "Speech and Facial Emotion Recognition using Convolutional Neural Network and Random Forest: A Multimodal Analysis," in *2024 Asia Pacific Conference on Innovation in Technology (APCIT)*, IEEE, Jul. 2024, pp. 1–5. doi: 10.1109/APCIT62007.2024.10673495.
- [9] C. A. D. D. Clemency, B. Lakshmi Samatha Reddy, and B. A. Chakri, "Noise Robust Speech Emotion Recognition Using MFCC Features and Random Forest Classifier," in *2026 4th International Conference on Knowledge Engineering and Communication Systems (ICKECS)*, IEEE, Apr. 2026, pp. 1–5. doi: 10.1109/ICKECS70176.2026.11527600.
- [10] N. Boualoulou, T. B. Drissi, and B. Nsiri, "CNN and LSTM for the classification of parkinson's disease based on the GTCC and MFCC," *Applied Computer Science*, vol. 19, no. 2, pp. 1–24, 2023.
- [11] P. Deb, "Multitask Audio Analysis for Emotion, Gender, and Speaker Recognition in Bangla Speech Comparing Features and Models," in *2023 International Conference on Advanced Computing Technologies and Applications (ICACTA)*, IEEE, Oct. 2023, pp. 1–6. doi: 10.1109/ICACTA58201.2023.10393734.
- [12] K. Jangra, D. Ghai, and S. Kumar, "Exploring the Effectiveness of Advanced Machine Learning Models in Speech Emotion Recognition," in *2024 International Conference on*

- Communication, Computer Sciences and Engineering (IC3SE)*, IEEE, May 2024, pp. 507–512. doi: 10.1109/IC3SE62002.2024.10593399.
- [13] J. M. Gorriz, R. M. Clemente, F. Segovia, J. Ramirez, A. Ortiz, and J. Suckling, “Is K-fold cross validation the best model selection method for Machine Learning?,” *arXiv preprint arXiv:2401.16407*, 2024.
- [14] R. R. N, S. H. Gebremariam, S. N, K. G, P. C. R. Alla, and S. Chauhan, “Enhanced Speaker Accent Classification: An Experimental Study of Diverse Machine Learning Methods,” in *2025 International Conference on Intelligent Systems and Computational Networks (ICISCN)*, IEEE, Jan. 2025, pp. 1–6. doi: 10.1109/ICISCN64258.2025.10934549.
- [15] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” *Mech. Syst. Signal Process.*, vol. 151, p. 107398, 2021.
- [16] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J. Big Data*, vol. 8, no. 1, 2021.
- [17] K. Sudhakar, T. J. P. Singh, M. Thanuj, and B. Praveen, “MFCC-Guided Convolutional Autoencoders: A Hybrid Framework for Noise-Resilient, Edge-Optimized Keyword Spotting,” in *2025 9th International Conference on Inventive Systems and Control (ICISC)*, IEEE, Aug. 2025, pp. 1688–1693. doi: 10.1109/ICISC65841.2025.11187868.
- [18] Y. Izotov and A. Velichko, “Speech Command Recognition Using LogNet Reservoir Computing for Embedded Systems,” Aug. 2025.
- [19] M. Yan, “RESSCR: An Edge-Friendly 1D Residual CNN with Systematic Enhancements for Speech Command Recognition,” in *2025 IEEE 3rd International Conference on Electrical, Automation and Computer Engineering (ICEACE)*, IEEE, Dec. 2025, pp. 1760–1765. doi: 10.1109/ICEACE67491.2025.11439102.