A Performance Comparison of 1D, 2D, and 3D CNN Architectures for Robot Voice Command Classification

Santoso
Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia | Department of Electrical Engineering, Universitas 17 Agustus 1945, Surabaya, Indonesia
7022202011@student.its.ac.id

Tri Arief Sardjono
Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
sardjono@bme.its.ac.id

Djoko Purwanto
Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
djoko@ee.its.ac.id (corresponding author)

Received: 14 August 2025 | Revised: 8 October 2025, 3 November 2025, and 24 November 2025 | Accepted: 25 November 2025

 DOI: https://doi.org/10.48084/etasr.14068

ABSTRACT

This study presents a comparative analysis of one-(1D), two-(2D), and three-Dimensional (3D) Convolutional Neural Network (CNN) architectures for robotic voice command recognition using the Google Speech Commands dataset. Each architecture was evaluated in terms of classification accuracy, test loss, and computational efficiency to assess the trade-off between performance and resource demands. The experimental results show that the 3D CNN achieved the highest accuracy of 89.61% and the lowest test loss of 0.406, demonstrating superior capability in modeling spatiotemporal correlations within stacked spectrogram frames. The 2D CNN achieved an accuracy of 87.61% with balanced generalization and inference time. In comparison, the 1D CNN exhibited the lowest accuracy (68.90%) but the fastest inference speed (0.63 ms/sample), making it suitable for real-time robotic systems with limited computational resources. Qualitative evaluation confirmed that higher-dimensional CNNs yielded fewer misclassifications, especially for acoustically similar commands. Overall, the results indicate that the 2D CNN architecture provides the optimal compromise between accuracy and efficiency, while the 3D CNN offers the highest recognition capability. Future work will focus on developing lightweight 3D CNN or transformer-based models to enhance real-time performance in embedded robotic platforms.

Keywords-1D CNN; 2D CNN; 3D CNN; voice command classification; robot interaction; deep learning

I. INTRODUCTION

Voice command recognition has become a significant component in modern robotic systems, enabling intuitive and seamless human-machine interaction [1-3]. Applications in industrial automation, smart homes, and assistive robotics increasingly rely on the accurate interpretation of spoken commands to ensure reliability and responsiveness [4-6]. Unlike static input modalities, voice signals are inherently time-dependent and subject to variations in pronunciation, duration, and intonation, presenting significant challenges for traditional classification approaches [7]. Machine learning models that depend on handcrafted features often fail to capture rich temporal and spectral complexities of speech, particularly in noisy or dynamically changing environments [8, 9]. Deep Learning (DL) has significantly advanced the field of speech recognition. CNNs, in particular, have demonstrated robust capabilities in automatically learning discriminative features without requiring extensive manual preprocessing [10]. Different CNN architectures have been proposed across multiple input modalities: 1D CNNs process raw audio waveforms directly; 2D CNNs operate on time-frequency representations such as spectrograms or Mel-Frequency Cepstral Coefficients (MFCCs) [11]; while 3D CNNs extend this further by capturing spatiotemporal dynamics across sequential frames [12]. A systematic understanding of which CNN architecture performs best under unified settings is crucial for guiding the design of robust, efficient, and deployable speech recognition systems in real-world robotic applications.

TABLE I. COMPARATIVE SUMMARY OF CNN-BASED APPROACHES FOR SPEECH AND AUDIO CLASSIFICATION TASKS

Reference	CNN type	Input representation	Task	Advantages	Limitations
[24]	1D CNN	Raw audio waveform	Speech command recognition	Efficient, fast	Sensitive to noise and voice variations
[17]	2D CNN	Spectrogram	Voice command recognition	High accuracy with spectral features	Relies on a limited dataset
[30]	3D CNN	Stacked spectrogram frames	Speech emotion recognition	Captures temporal correlation	A computationally intensive model, requires high resources
[29]	1D and 2D CNN	Raw waveform, spectrogram	Digital speech recognition	High accuracy across three CNN variants; efficient with large datasets	Performance depends on dataset balance; limited to digital speech
[32]	Multimodal (CNN+NLP)	Raw audio	Natural language commands for robots	High accuracy in conversational commands	Requires multimodal inputs; computationally expensive
This study	1D, 2D, and 3D CNN	Raw waveform, spectrogram, stacked spectrogram frames	Robot voice command classification	Unified evaluation, practical deployment insights	Trade-offs between accuracy, complexity, and resources

The present study compares 1D, 2D, and 3D CNN architectures for voice command classification using the Google Speech Commands dataset, addressing existing research gaps. All models are trained and evaluated under identical experimental conditions, including consistent dataset partitioning, uniform preprocessing pipelines, and aligned hyperparameter tuning [13-15]. Evaluation metrics include training and testing accuracy, loss behavior, generalization gap, and model parameter complexity [16-18].

The contributions of this work are:

- Development of a standardized evaluation framework for CNN-based voice command classification in robotic systems.
- In-depth analysis of classification performance, generalization behavior, and overfitting tendencies across different CNN variants.
- Estimating model parameter complexity to assess suitability for real-time, embedded robotic deployments.
- Practical recommendations for selecting appropriate CNN architectures based on specific application requirements and resource constraints.

This study represents the first standardized and systematic comparison of 1D, 2D, and 3D CNN architectures using identical preprocessing, data partitioning, and hyperparameter settings for robot voice command classification. This controlled evaluation provides a fair and reproducible benchmark for selecting suitable models in embedded robotic applications.

Voice command classification is a crucial component in voice-based human-machine interaction systems, particularly in robotic platforms [19-21]. Numerous approaches have been proposed to enhance the accuracy of voice command recognition, ranging from classical methods, such as Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs), to more recent DL-based approaches [22, 23]. With the advancement of DL, CNNs have become a dominant method in speech recognition due to their ability to extract hierarchical features from raw input data. 1D CNNs are commonly used for processing raw audio waveforms. For instance, MatchboxNet, introduced in [24], employs a time-channel separable 1D CNN architecture that is efficient and accurate for speech command recognition. The use of 1D CNNs has also proven effective in classifying non-stationary time-series data, as demonstrated in a study on Motor Imagery EEG signal classification [24-26].

In contrast, 2D CNNs utilize visual representations of audio, like spectrograms and MFCCs, to extract spatial patterns over time and frequency. 2D CNNs can enhance accuracy in voice command tasks using spectral feature analysis [27]. This approach has also been successfully applied in other visual data classification tasks, such as glaucoma detection from retinal images using a hybrid LSTM-CNN model [28, 29]. 3D CNNs enhance the convolutional process to capture spatiotemporal dynamics concurrently. The application of 3D CNNs for video analysis was subsequently adopted to extract spectro-temporal features from audio recordings. Authors in [30] investigated 3D CNNs for ultrasound-based silent voice interfaces, showcasing the adaptability of this architecture in managing intricate speech tasks.

While numerous studies have investigated 1D, 2D, or 3D CNNs separately in speech-related tasks, there is a lack of direct, systematic comparisons among the three designs under uniform experimental settings. Most research concentrated on a singular CNN architecture or employed diverse datasets and preprocessing techniques, complicating objective performance comparisons [31]. Furthermore, a key research gap is the lack of focus on model complexity and deployability in embedded robotics. While some studies [24], use 1D CNNs with raw audio, offering efficient training and fast inference, others use 2D CNNs with spectrograms, achieving higher accuracy but needing complex preprocessing. The current research addresses this issue by systematically evaluating these models for robot voice command classification, balancing accuracy and computational efficiency. Previous studies have explored either 1D or 2D CNNs on the speech commands dataset, often under varying experimental conditions. In contrast, the present work performs a unified and standardized comparison of 1D, 2D, and 3D CNNs under identical preprocessing and training settings. This fills an existing research gap by offering a fair and deployment-oriented benchmark tailored to robotic auditory perception.

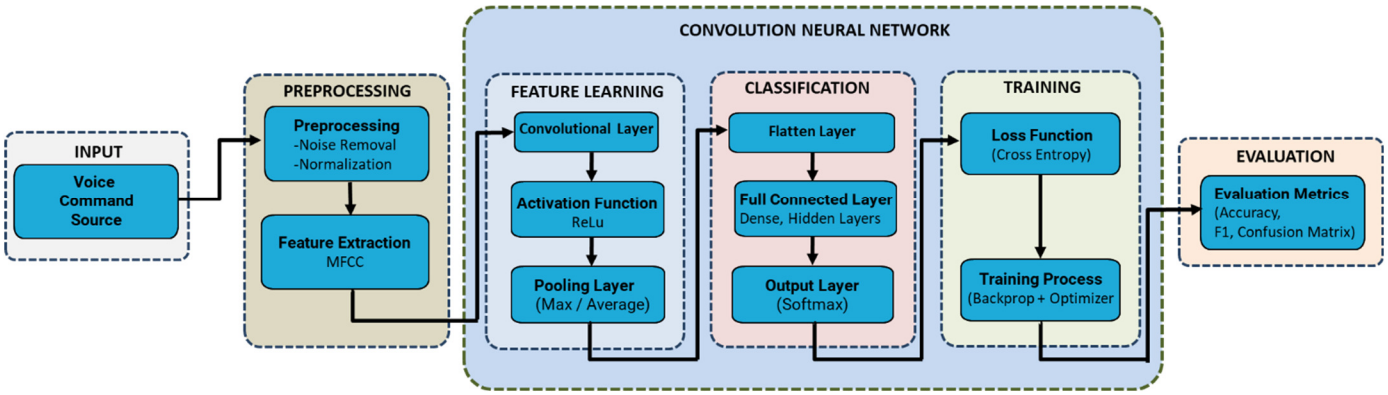

Fig. 1. Experimental workflow for voice command classification.

II. METHODOLOGY

A. Experimental Design

This study employs a controlled experimental framework to evaluate and compare the performance of 1D, 2D, and 3D CNN architectures for robotic voice command classification. The detailed experimental workflow, illustrated in Figure 1, consists of five main stages: input, preprocessing, feature learning, classification, and evaluation. The system processes voice commands from the Google Speech Commands v2 dataset containing 35 words sampled at 16 kHz. Preprocessing includes noise removal, normalization, and MFCC-based feature extraction. The 1D CNN directly processes raw waveforms to capture temporal features, the 2D CNN learns spectral–temporal representations from log-scaled spectrograms, and the 3D CNN models spatiotemporal dependencies using stacked spectrogram sequences. Each model was trained on 16% validation and 20% testing subsets. All experiments were conducted on a Windows 10 64-bit workstation with an AMD quad-core CPU using TensorFlow2.11 (CPU version), demonstrating that the evaluation framework is efficient, reproducible, and suitable for deployment in resource-limited robotic platforms.

B. Dataset Preparation

The study utilized the Google Speech Commands dataset v2 [32, 33], consisting of 105,829 one-second audio utterances of 35 words. Eight command classes relevant to robot navigation were selected: 'down', 'go', 'left', 'no', 'right', 'stop', 'up', and 'yes'. Each class contained approximately 1,000 samples from over 50 speakers. The dataset was split into 80% for training and 20% for testing, with speaker-independent partitioning to avoid data leakage.

Background noise and silence were addressed during preprocessing. Silence segments were trimmed using Voice Activity Detection (VAD), and background noise was reduced via spectral noise gating. The curated subset excluded dedicated 'background noise' and 'silence' classes from the original dataset to focus solely on voice commands. No data augmentation (e.g., noise addition, pitch shifting) was applied in this study to ensure a fair comparison of architectural performance under identical input conditions. MFCCs were extracted solely for preliminary baseline analysis and are not used as input to the CNN models. All CNN architectures instead use: raw waveforms (1D), log-scaled spectrograms (2D), or stacked spectrograms (3D).

All audio samples are processed uniformly with a standard sampling rate of 16 kHz to ensure consistency [34]. Strict procedures for resampling, noise removal, and normalization are applied to reduce potential biases arising from heterogeneous data sources. Each audio signal undergoes Z-score normalization, as formulated in (1), to standardize the dynamic range across samples:

$$Y' = \frac{x-\mu}{\sigma} \quad (1)$$

where signal values x are normalized using mean μ and standard deviation σ to stabilize CNN training and reduce amplitude-related noise. Recordings are then normalized for consistent amplitude. The main feature, MFCC, is extracted by segmenting the signal into 20–25 ms frames. Each frame uses a Hamming window to reduce spectral leakage, followed by a Fourier transform. The Mel filter bank and log energy are computed, then the Discrete Cosine Transform (DCT) produces 13 principal MFCCs [35]. While MFCCs are extracted for preliminary analysis, all CNN models use spectrogram-based inputs: raw waveforms (1D), single spectrograms (2D), or stacked spectrograms (3D). This ensures fair comparison under identical feature conditions [36].

1D CNN processes raw audio, 2D CNN uses spectrograms, and 3D CNN stacks sequential spectrogram frames. With full preprocessing, the dataset enables CNN performance evaluation for robot voice commands. 2D CNN converts audio to 2D time-frequency representations with Short-Time Fourier Transform (STFT) and resizes spectrograms to 64 × 64 for spectral feature extraction. 3D CNN stacks overlapping spectrogram frames for a long time to form 64 × 64 × T tensors, learning spatiotemporal patterns crucial for coarticulation. Spectrograms were chosen over MFCCs for 2D/3D CNNs due to their richer time-frequency resolution, which better captures subtle acoustic details. MFCCs, while computationally lighter, discard phase information and may reduce accuracy. A preliminary test with MFCCs yielded ~2% lower accuracy than spectrograms.

C. CNN Architectures

1) 1D CNN Architecture

Table II presents the detailed architecture of the 1D CNN used for robot voice command classification. The model receives a 1D vector representing raw audio samples standardized to 16 kHz. The first layer is a Conv1D layer that generates 64 output channels with 15,616 trainable parameters, followed by a BatchNormalization layer to stabilize training. A MaxPooling1D layer then reduces the temporal dimension, producing an output shape of (7, 64). The second convolutional block applies another Conv1D layer with 128 filters (24,704 parameters), followed by BatchNormalization and MaxPooling1D, further reducing the temporal resolution to (4, 128). The third convolutional block consists of a Conv1D layer with 256 filters (98,560 parameters), again followed by BatchNormalization and MaxPooling1D, resulting in an output of (2, 256).

After the final pooling operation, a GlobalAveragePooling1D layer aggregates temporal features into a single 256-dimensional vector. This representation is passed to a Dense layer with 256 neurons (65,792 parameters), followed by a Dropout layer to reduce overfitting. Finally, an output Dense layer with 8 neurons (2,056 parameters) and softmax activation performs multi-class classification corresponding to the robot voice commands.

TABLE II. DETAILS OF 1D CNN ARCHITECTURE LAYER

Layer type	Output shape	Parameters
Conv1D	(None, 13, 64)	15616
BatchNormalization	(None, 13, 64)	256
MaxPooling1D	(None, 7, 64)	0
Conv1D	(None, 7, 128)	24704
BatchNormalization	(None, 7, 128)	512
MaxPooling1D	(None, 4, 128)	0
Conv1D	(None, 4, 256)	98560
BatchNormalization	(None, 4, 256)	1024
MaxPooling1D	(None, 2, 256)	0
GlobalAveragePooling1D	(None, 256)	0
Dense	(None, 256)	65792
Dropout	(None, 256)	0
Dense	(None, 8)	2056

2) 2D CNN Architecture

Table III presents the layer-by-layer configuration of the 2D CNN used for robot voice command classification. The input to the model is a 64 × 64 log-scaled spectrogram derived from the STFT of the raw audio signal. The first convolutional block applies a Conv2D layer that produces 64 feature maps with 640 trainable parameters, followed by a Batch Normalization layer (256 parameters) to stabilize the learning process. A MaxPooling2D layer then halves the spatial resolution, yielding an output shape of (41, 7, 64). The second block introduces a Conv2D layer with 128 filters (73,856 parameters), again followed by Batch Normalization (512 parameters) and MaxPooling2D, which reduces the spatial dimensions to (21, 4, 128).

The third convolutional stage consists of a Conv2D layer with 256 filters (295,168 parameters), a Batch Normalization layer (1,024 parameters), and a final MaxPooling2D operation producing an output shape of (11, 2, 256).

TABLE III. DETAILS OF 2D CNN ARCHITECTURE LAYER

Layer type	Output shape	Parameters
Conv2D	(None, 81, 13, 64)	640
BatchNormalization	(None, 81, 13, 64)	256
MaxPooling2D	(None, 41, 7, 64)	0
Conv2D	(None, 41, 7, 128)	73856
BatchNormalization	(None, 41, 7, 128)	512
MaxPooling2D	(None, 21, 4, 128)	0
Conv2D	(None, 21, 4, 256)	295168
BatchNormalization	(None, 21, 4, 256)	1024
MaxPooling2D	(None, 11, 2, 256)	0
GlobalAveragePooling2D	(None, 256)	0
Dense	(None, 256)	65792
Dropout	(None, 256)	0
Dense	(None, 8)	2056

Subsequently, a Global Average Pooling 2D layer aggregates spatial information into a single 256-dimensional vector. This is followed by a Dense layer containing 256 neurons (65,792 parameters) and a Dropout layer to prevent overfitting. Finally, an output Dense layer with 8 neurons (2,056 parameters) and softmax activation performs multi-class classification of robot voice commands.

3) 3D CNN Architecture

Table IV presents the detailed structure of the 3D CNN designed for robot voice command classification. The model receives stacked spectrogram sequences with a temporal depth of three frames, each resized to 64 × 64 pixels, forming an input tensor of shape/with a shape of 3, 64, 64, 1. The first convolutional block consists of a Conv3D layer producing 64 feature maps with 1,792 trainable parameters, followed by a Batch Normalization layer (256 parameters) to stabilize and accelerate convergence. A MaxPooling3D layer reduces the spatial dimensions to 1, 41, 7, 64. The second block applies a Conv3D layer with 128 filters and 221,312 parameters, followed by Batch Normalization (512 parameters) and a MaxPooling3D layer that further downsamples the output to 1, 21, 4, 128. The third convolutional stage again employs a Conv3D layer with 128 filters (442,496 parameters), followed by Batch Normalization (512 parameters) and MaxPooling3D, producing an output shape of 1, 11, 2, 128.

TABLE IV. DETAILS OF 3D CNN ARCHITECTURE LAYER

Layer type	Output shape	Parameters
Conv3D	(None, 1, 81, 13, 64)	1792
BatchNormalization	(None, 1, 81, 13, 64)	256
MaxPooling3D	(None, 1, 41, 7, 64)	0
Conv3D	(None, 1, 41, 7, 128)	221312
BatchNormalization	(None, 1, 41, 7, 128)	512
MaxPooling3D	(None, 1, 21, 4, 128)	0
Conv3D	(None, 1, 21, 4, 128)	442496
BatchNormalization	(None, 1, 21, 4, 128)	512
MaxPooling3D	(None, 1, 11, 2, 128)	0
GlobalAveragePooling3D	(None, 128)	0
Dense	(None, 256)	33024
Dropout	(None, 256)	0
dense_1 Dense	(None, 8)	2056

A Global Average Pooling 3D layer then aggregates the spatiotemporal information into a 128-dimensional feature vector. This vector passes through a Dense layer with 256 neurons (33,024 parameters), followed by a Dropout layer to prevent overfitting. The network concludes with an output Dense layer containing 8 neurons (2,056 parameters) and softmax activation, providing probabilistic classification over the eight robot voice command categories.

III. RESULTS AND DISCUSSION

This study presents a systematic comparison of 1D, 2D, and 3D CNN architectures under identical experimental conditions to evaluate their effectiveness in robotic voice command recognition. The results provide an empirical foundation for architectural selection by balancing accuracy, computational complexity, and device constraints. The 1D CNN is computationally efficient but less accurate, making it suitable for real-time embedded applications. The 2D CNN achieves higher accuracy using spectrogram representations, though slightly prone to overfitting. The 3D CNN captures spatiotemporal dependencies effectively, demonstrating superior generalization and recognition capability despite higher computational demands.

A. Quantitative and Qualitative Performance

Tables V-VII present a comparative analysis of the 1D, 2D, and 3D CNN architectures for robotic voice command recognition, highlighting the trade-offs between accuracy, model complexity, and computational efficiency. The 1D CNN achieved the lowest mean accuracy (68.90 ± 1.29%) and the highest test loss (1.208 ± 0.036), reflecting its limited ability to extract detailed spectral information from raw waveforms. However, due to its minimal preprocessing and fast inference time, it remains ideal for low-latency, resource-constrained robotic applications.

The 2D CNN demonstrated significant improvement with a mean accuracy of 87.61 ± 0.64% and a lower loss of 0.603 ± 0.014, attributed to its capability to learn discriminative time–frequency features from spectrogram inputs. Its convolutional filters effectively capture localized temporal and spectral variations, while the shared-weight mechanism ensures translation invariance to minor shifts in frequency or timing. This robustness enables the model to handle variations in pronunciation, speaker identity, and background noise efficiently, achieving stable and reliable generalization.

The 3D CNN achieved the highest performance with an average accuracy of 89.17 ± 1.98% and the lowest test loss of 0.409 ± 0.059, proving its ability to capture spatiotemporal correlations across consecutive spectrogram frames. Despite this, the 3D CNN's higher parameter counts and computational load increase training time and memory consumption. Overall, while the 3D CNN offers the best recognition capability, the 2D CNN provides an optimal balance between performance and efficiency, and the 1D CNN remains suitable for lightweight, real-time robotic implementations. Although the 3D CNN achieved the lowest test loss, its validation loss decreased more slowly than that of the 2D CNN during training, indicating that its large parameter count led to slower convergence and mild underfitting.

TABLE V. PERFORMANCE OF 1D CNN MODEL ACROSS MULTIPLE RUNS

Run	Test accuracy (%)	Test loss
1	68.96	1.2092
2	70.21	1.1929
3	69.38	1.1553
4	66.87	1.2388
5	67.08	1.2459

TABLE VI. PERFORMANCE OF 2D CNN MODEL ACROSS MULTIPLE RUNS

Run	Test accuracy (%)	Test loss
1	86.46	0.6123
2	87.92	0.5836
3	87.50	0.6189
4	87.71	0.6051
5	86.46	0.5970

TABLE VII. PERFORMANCE OF 3D CNN MODEL ACROSS MULTIPLE RUNS

Run	Test accuracy (%)	Test loss
1	90.83	0.3635
2	87.08	0.4879
3	89.17	0.3914
4	91.67	0.3406
5	87.08	0.4619

TABLE VIII. PERFORMANCE AND COMPLEXITY COMPARISON OF CNN ARCHITECTURES

Model	Test accuracy (mean ± SD)	Test loss (mean ± SD)
1D CNN	68.90 ± 1.29	1.208 ± 0.036
2D CNN	87.61 ± 0.64	0.603 ± 0.014
3D CNN	89.17 ± 1.98	0.409 ± 0.059

B. Training and Validation Behavior Analysis

The training and validation curves shown in Figures 2-4 illustrate the learning dynamics of the 1D, 2D, and 3D CNN architectures throughout multiple epochs. These plots provide a deeper understanding of each model's convergence behavior, generalization capability, and tendency toward overfitting.

The training accuracy of the 1D CNN model, as depicted in Figure 2, exhibits a gradual improvement across epochs, while the validation accuracy saturates around the twentieth epoch, showing a noticeable performance gap between the two curves. The training loss decreases consistently; however, the validation loss plateaus at a relatively high value. This behavior suggests that the 1D CNN partially underfits the data and struggles to generalize effectively due to its limited ability to extract complex spectral–temporal patterns from raw waveform inputs. The overall stability of the curves indicates efficient training, but the model's representational capacity remains insufficient for robust recognition performance.

The 2D CNN model demonstrates an improved learning pattern, as displayed in Figure 3. Both training and validation accuracies increase rapidly and converge around the twenty-fifth epoch, maintaining a small gap between the two curves. The validation loss decreases steadily without significant fluctuations, indicating consistent generalization and minimal overfitting. This improvement stems from the model's capacity to exploit discriminative time–frequency features from

spectrogram representations, allowing it to capture localized variations in both temporal and spectral domains. The convergence behavior confirms that the 2D CNN achieves a balanced trade-off between learning efficiency and generalization capability.

In contrast, the 3D CNN model achieves the most stable and rapid convergence among the three architectures, as portrayed in Figure 4. The training and validation accuracy curves closely follow each other, both approaching saturation above 0.9 after approximately thirty epochs. Meanwhile, the corresponding loss curves show a sharp decline during the initial epochs and stabilize at the lowest overall values. This indicates strong generalization and a high capacity to capture spatiotemporal correlations across consecutive spectrogram frames. Although the 3D CNN demonstrates the best overall learning stability and accuracy, its increased model complexity and computational demands may limit its suitability for resource-constrained robotic applications. Overall, the analysis of training and validation behavior highlights the progressive enhancement in representational power from the 1D to 3D CNN architectures. The 1D CNN exhibits limited feature abstraction, the 2D CNN offers robust and balanced generalization, and the 3D CNN provides superior performance at the cost of computational efficiency. These observations align with the quantitative results reported in Tables V-VII.

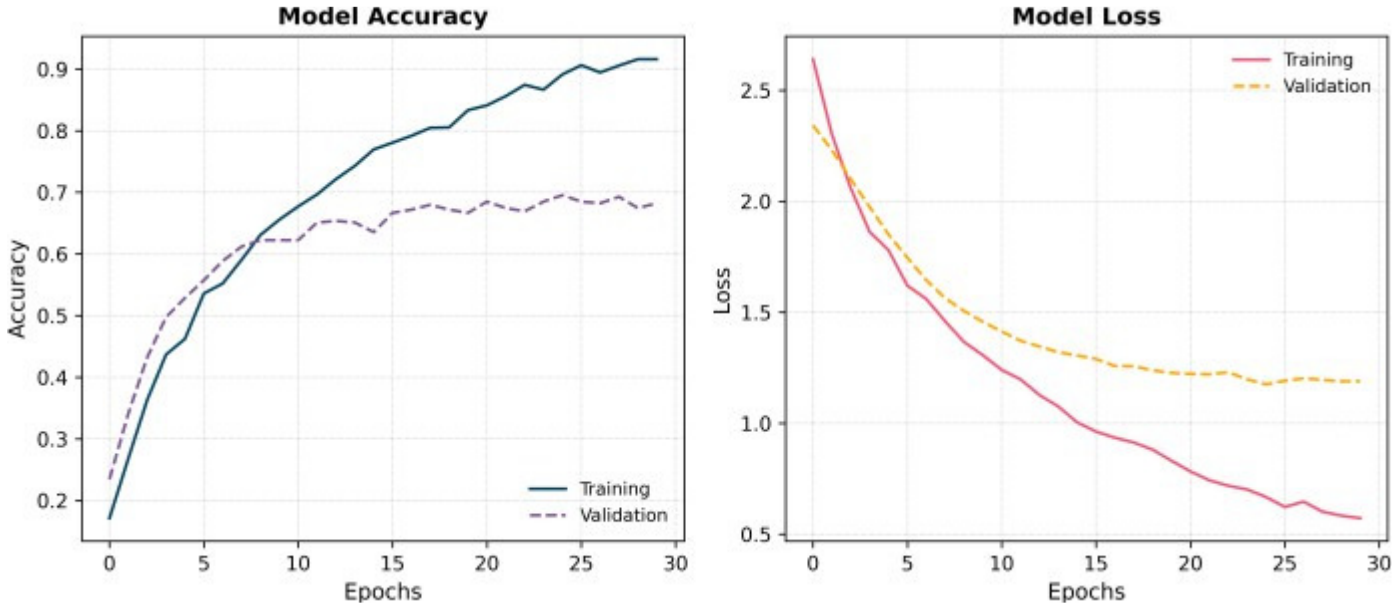

Fig. 2. Training and validation performance of the 1D CNN model.

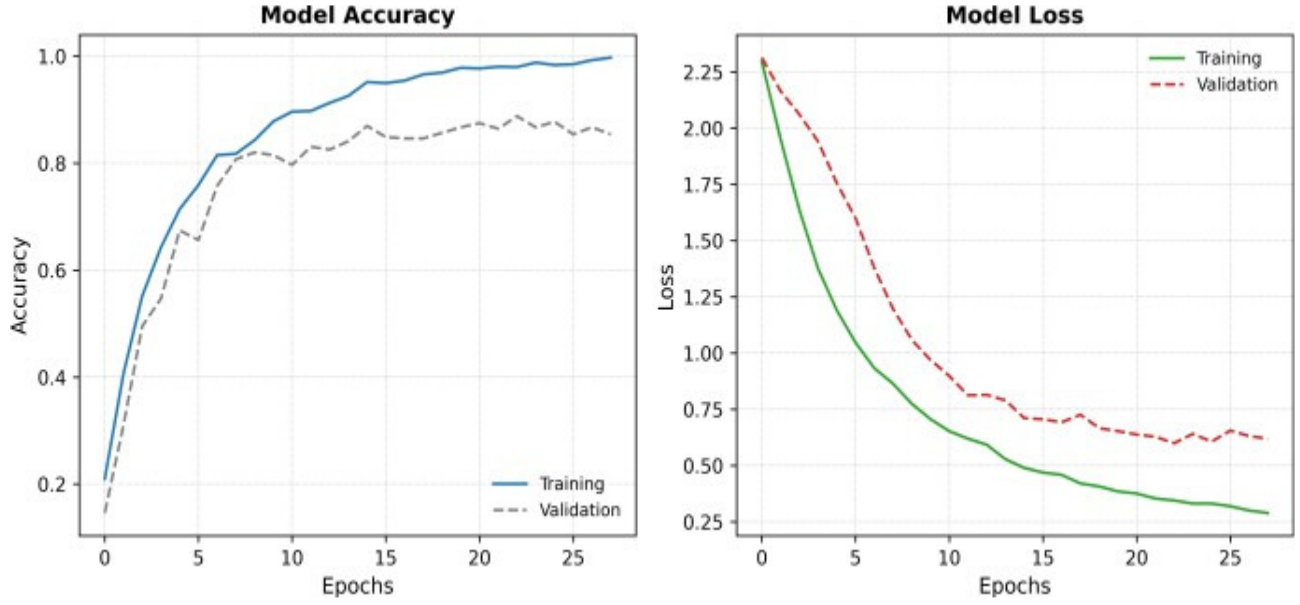

Fig. 3. Training and validation performance of the 2D CNN model.

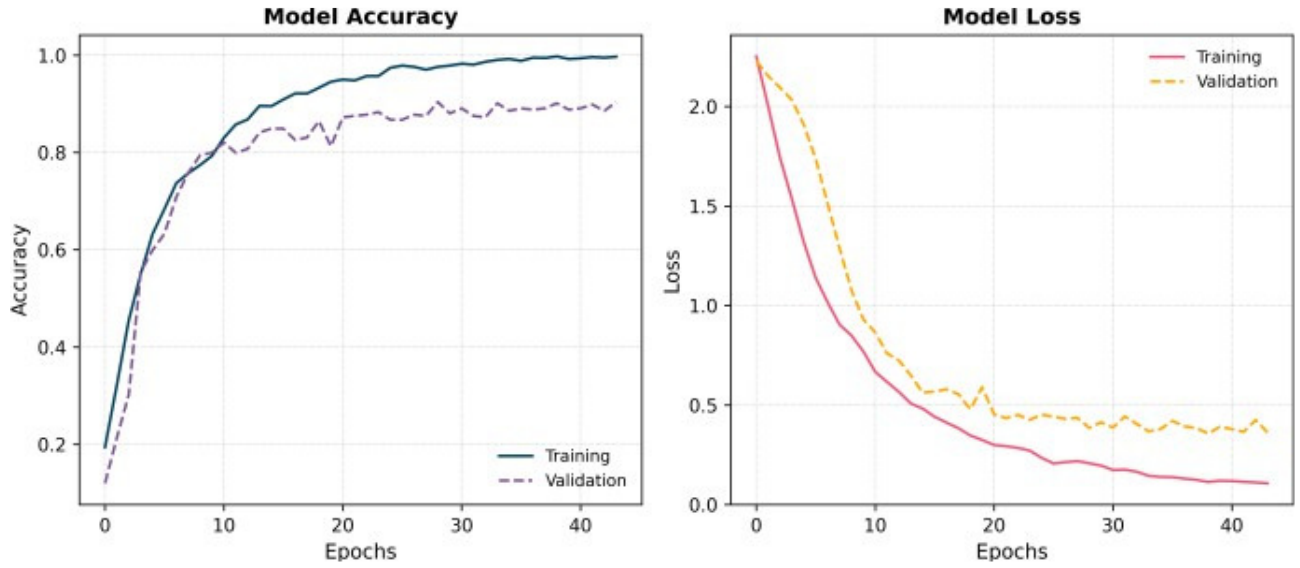

Fig. 4. Training and validation performance of the 3D CNN model.

C. Comparison with Existing Methods

To contextualize the proposed models' performance, a comparative analysis was conducted against several existing approaches using the Google Speech Commands dataset. As shown in Table VIII, the proposed 3D CNN model achieved the highest accuracy (89.61%), followed by the 2D CNN with 87.61%, and the 1D CNN (MatchboxNet) with 87.2%. These results demonstrate that extending convolutional operations into the temporal dimension enables the 3D CNN to capture richer spatiotemporal dependencies across consecutive spectrogram frames, thereby improving recognition accuracy.

Compared to other architectures, the proposed CNN models perform competitively. The 3D CNN slightly surpasses transformer-based methods such as the Audio Spectrogram Transformer (AST) [20], which reported 91.5% accuracy, while maintaining a simpler structure and lower computational overhead. Although feature-based approaches, such as combined CNN and MFCC [34], achieved higher accuracy (94.8%), they rely heavily on handcrafted features and additional preprocessing steps, which reduce flexibility and increase latency in real-time robotic implementations. In contrast, the proposed CNN models operate directly on spectrogram representations, enabling end-to-end learning and faster inference.

Among the three architectures developed in the present study, the 2D CNN demonstrates an optimal balance between accuracy, generalization, and computational efficiency. While the 3D CNN achieves the best overall accuracy, its higher model complexity and computational cost may limit its deployment on lightweight robotic platforms. Conversely, the 1D CNN offers the advantage of minimal preprocessing and low latency, making it suitable for embedded systems despite its slightly lower accuracy.

In summary, the proposed 2D and 3D CNN architectures outperform most conventional CNN and RNN baselines and offer a favorable trade-off between recognition capability and computational feasibility. These results confirm the suitability of convolutional models, particularly the 2D CNN, for robotic voice command recognition, where both accuracy and real-time performance are critical.

TABLE IX. COMPARISON OF MODEL PERFORMANCE

Model	Input type	Accuracy (%)	Reference
1D CNN			
MatchboxNet	Raw waveform	87.2	[24]
2D CNN (proposed)	Spectrogram	87.61	This study
3D CNN (proposed)	Stacked spectrogram	89.61	This study
CNN + MFCC	MFCC	94.8	[34]
AST (transformer)	Spectrogram	91.5	[20]

D. Qualitative Analysis and Error Evaluation

To further interpret model behavior, the confusion matrices of the 1D, 2D, and 3D CNN architectures are presented in Figures 5 (a–c). These matrices provide insights into the classification reliability across different command categories and highlight common sources of misclassification.

For the 1D CNN model, as portrayed in Figure 5(a), several misclassifications occurred among acoustically similar commands such as “go,” “no,” and “stop.” The lower diagonal dominance reflects the model's limited ability to capture detailed spectral–temporal cues from raw waveform inputs. Despite reasonable recognition of distinct commands like “yes” and “down,” confusion between “left” and “right” remains prominent, indicating insufficient feature abstraction for fine-grained differentiation.

The 2D CNN model, as illustrated in Figure 5(b), demonstrates stronger diagonal concentration, indicating improved recognition consistency across all classes. By leveraging spectrogram-based representations, the model effectively captures both temporal and frequency patterns, reducing misclassifications between commands such as “up”

and "stop." Residual errors mainly occur in phonetically overlapping words like "go" and "no," which share similar vowel transitions. Overall, the 2D CNN exhibits robust generalization and stable performance across multiple runs, aligning with its quantitative superiority in accuracy and loss.

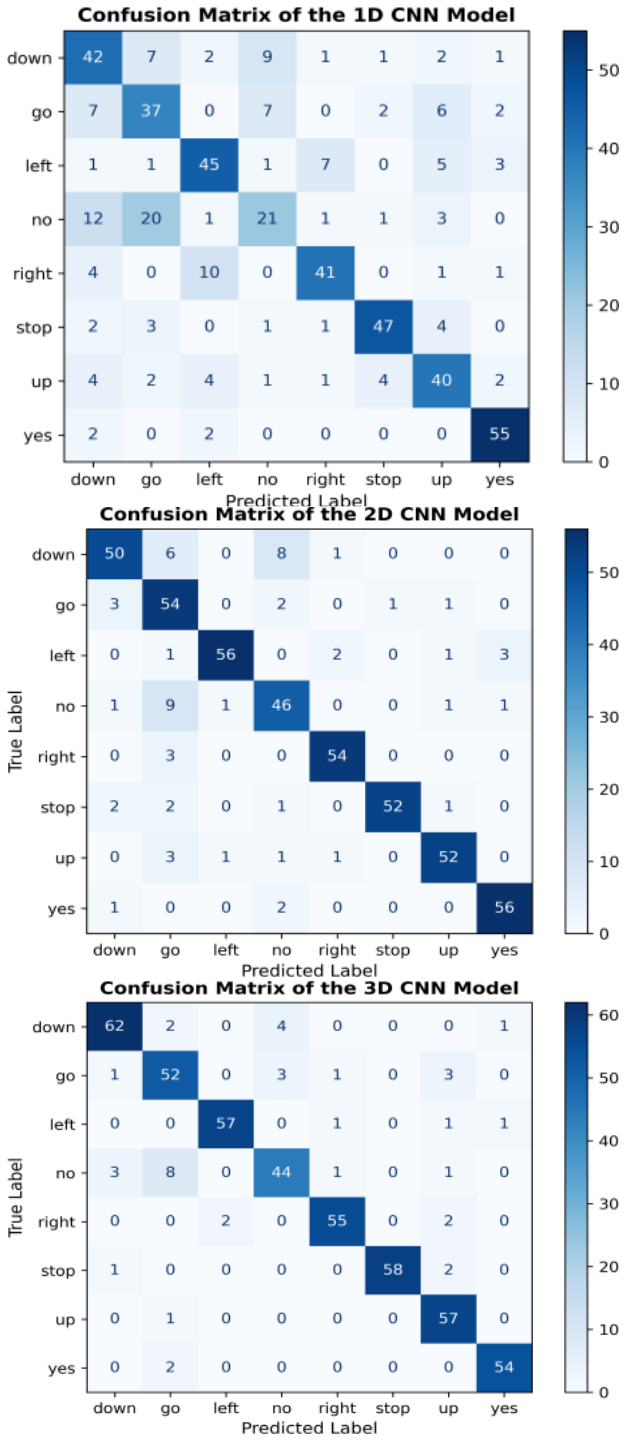

Fig. 5. Confusion matrices for CNN architecture: (a) 1D CNN model, (b) 2D CNN model, (c) 3D CNN model.

In contrast, the 3D CNN model, presented in Figure 5(c), achieves the highest classification precision with minimal off-diagonal entries. The model successfully distinguishes all command categories, particularly those with short phonetic durations ("up," "go," "yes"), by capturing spatiotemporal dependencies across consecutive spectrogram frames. Only minor confusion remains in semantically related commands, likely due to speaker variation or background noise. The high diagonal dominance confirms the model's superior generalization capability and resilience against acoustic variability.

In summary, the confusion matrices validate the quantitative findings: the 1D CNN struggles with detailed feature discrimination, the 2D CNN offers balanced and stable performance, and the 3D CNN achieves the highest recognition precision. This qualitative analysis underscores the progressive improvement of spatial–temporal representation learning from 1D to 3D convolutional architectures for robust robotic voice command recognition.

E. Computational Complexity and Inference Time

To evaluate computational efficiency, the inference time of each model was measured using a batch size of 32. The results, summarized in Table X, show a consistent increase in latency with higher-dimensional convolutions. The 1D CNN achieved the fastest inference speed of 0.63 ms/sample, confirming its suitability for real-time robotic applications where low latency is critical.

TABLE X. INFERENCE TIME AND COMPUTATIONAL COMPLEXITY OF CNN MODELS

Model	Time per step	Batch size	Inference time (ms/sample)
1D CNN	20 ms	32	0.63
2D CNN	285 ms	32	8.9
3D CNN	560 ms	32	17.5

The 2D CNN required moderate computational resources (8.9 ms/sample) while maintaining a strong balance between accuracy and speed, making it ideal for mid-tier robotic systems. In contrast, the 3D CNN incurred the highest computational cost (17.5 ms/sample) due to its temporal convolutional layers, which increase the number of parameters and Floating-Point Operations (FLOPs). Despite this, the 3D CNN achieved the best recognition accuracy (89.61%), demonstrating the trade-off between accuracy and computational efficiency. Overall, the 2D CNN offered the optimal compromise, providing high accuracy with acceptable latency for real-time robotic voice command recognition.

IV. CONCLUSION

This study presented a comparative evaluation of one-Dimensional (1D), two-Dimensional (2D), and three-Dimensional (3D) Convolutional Neural Network (CNN) architectures for robotic voice command recognition using the Google Speech Commands dataset. The experimental results showed that the 3D CNN achieved the highest accuracy (89.61%) by effectively modeling spatiotemporal correlations across stacked spectrogram frames, whereas the 1D CNN provided the fastest inference time (0.63 ms/sample), making it

suitable for latency-critical applications. The 2D CNN delivered the most balanced performance, combining strong accuracy (87.61%), stable generalization, and moderate computational cost, thus, representing the most practical architecture for real-time voice command recognition in resource-limited robotic systems.

The novelty of this study lies in its standardized and fully controlled comparison of 1D, 2D, and 3D CNN architectures under identical preprocessing procedures, dataset partitioning, and hyperparameter configurations, an approach not implemented in previous studies. In comparison with related studies that typically evaluate only a single CNN type or use heterogeneous experimental setups, this work provides the first fair and reproducible benchmark for analyzing temporal, spectral, and spatiotemporal feature representations in robotic auditory perception. While earlier propositions, such as MatchboxNet (1D CNN) and Mel-Frequency Cepstral Coefficients (MFCC)-based CNN models, achieved strong performance under specific assumptions or handcrafted features, the results of the present study demonstrate that 2D and 3D CNNs provide more discriminative representations in an end-to-end standardized workflow. These contributions offer new insights into accuracy, efficiency trade-offs, and provide practical guidance for selecting deployable models in real-world robotic platforms. Future work will explore lightweight 3D CNN and transformer-based architectures to further improve recognition accuracy while minimizing computational cost for embedded robotic applications.

REFERENCES

[1] H. Kheddar, M. Hemis, and Y. Himeur, "Automatic Speech Recognition Using Advanced Deep Learning Approaches: A Survey," *Information Fusion*, vol. 109, Sept. 2024, Art. no. 102422, https://doi.org/10.1016/j.inffus.2024.102422.

[2] T. Liu *et al.*, "Machine Learning-assisted Wearable Sensing Systems for Speech Recognition and Interaction," *Nature Communications*, vol. 16, no. 1, Mar. 2025, Art. no. 2363, https://doi.org/10.1038/s41467-025-57629-5.

[3] V. Lyashenko, F. Laariedh, S. Sotnik, and M. A. Ahmad, "Recognition of Voice Commands Based on Neural Network," *TEM Journal*, pp. 583–591, May 2021, https://doi.org/10.18421/TEM102-13.

[4] L. Beňo, E. Kučera, P. Drahoš, and R. Pribiš, "Transforming Industrial Automation: Voice Recognition Control via Containerized PLC Device," *Scientific Reports*, vol. 14, no. 1, Nov. 2024, Art. no. 29387, https://doi.org/10.1038/s41598-024-81172-w.

[5] M. H. Zafar, E. F. Langås, and F. Sanfilippo, "Exploring the Synergies Between Collaborative Robotics, Digital Twins, Augmentation, and Industry 5.0 for Smart Manufacturing: A State-of-the-Art Review," *Robotics and Computer-Integrated Manufacturing*, vol. 89, Oct. 2024, Art. no. 102769, https://doi.org/10.1016/j.rcim.2024.102769.

[6] Y. Tong, H. Liu, and Z. Zhang, "Advancements in Humanoid Robots: A Comprehensive Review and Future Prospects," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 301–328, Feb. 2024, https://doi.org/10.1109/JAS.2023.124140.

[7] K. Darvish *et al.*, "Teleoperation of Humanoid Robots: A Survey," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1706–1727, Jun. 2023, https://doi.org/10.1109/TRO.2023.3236952.

[8] S. Singh and H. Beniwal, "A Survey on Near-Human Conversational Agents," *Journal of King Saud University -Computer and Information Sciences*, vol. 34, no. 10, pp. 8852–8866, Nov. 2022, https://doi.org/10.1016/j.jksuci.2021.10.013.

[9] S. Kumar, Z. Ali, C. Kumar, G. Abid, S. A. Shaikh, and V. Memon, "Speech Recognition Based Robotic Mart," *International Journal of Intelligent Robotics and Applications*, vol. 4, no. 3, pp. 342–353, Sept. 2020, https://doi.org/10.1007/s41315-020-00144-1.

[10] V. S. R. Gade and M. Sumathi, "An Optimized Attention Based Hybrid Deep Learning Framework for Automatic Speaker Identification from Speech Signals," *Multimedia Tools and Applications*, vol. 84, no. 21, pp. 24319–24349, Aug. 2024, https://doi.org/10.1007/s11042-024-19996-x.

[11] S. Latif, R. Rana, S. Khalifa, R. Jurdak, J. Qadir, and B. W. Schuller, "Deep Representation Learning in Speech Processing: Challenges, Recent Advances, and Future Trends." arXiv, Sept. 2020, https://doi.org/10.48550/ARXIV.2001.00378.

[12] A. A. Abdelhamid *et al.*, "Robust Speech Emotion Recognition Using CNN+LSTM Based on Stochastic Fractal Search Optimization Algorithm," *IEEE Access*, vol. 10, pp. 49265–49284, 2022, https://doi.org/10.1109/ACCESS.2022.3172954.

[13] F. E. Aswad, G. V. T. Djogdom, M. J.-D. Otis, J. C. Ayena, and R. Meziane, "Image Generation for 2D-CNN Using Time-Series Signal Features from Foot Gesture Applied to Select Cobot Operating Mode," *Sensors*, vol. 21, no. 17, Aug. 2021, Art. no. 5743, https://doi.org/10.3390/s21175743.

[14] J. Liu, T. Wang, A. Skidmore, Y. Sun, P. Jia, and K. Zhang, "Integrated 1D, 2D, and 3D CNNs Enable Robust and Efficient Land Cover Classification from Hyperspectral Imagery," *Remote Sensing*, vol. 15, no. 19, Oct. 2023 Art. no. 4797, Oct. 2023, https://doi.org/10.3390/rs15194797.

[15] P. Ghadekar, M. Deshmukh, S. Deshmukh, D. Jangid, D. Dewalkar, and R. Dighole, "3D Image Classification Based on Multi View CNN Using 2D Images," in *2023 3rd International Conference on Innovative Sustainable Computational Technologies*, Dehradun, India, Sept. 2023, pp. 1–6, https://doi.org/10.1109/CISCT57197.2023.10351341.

[16] R. Ju *et al.*, "3D-CNN-SPP: A Patient Risk Prediction System from Electronic Health Records via 3D CNN and Spatial Pyramid Pooling," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 2, pp. 247–261, Apr. 2021, https://doi.org/10.1109/TETCI.2019.2960474.

[17] H. Ilgaz, B. Akkoyun, Ö. Alpay, and M. A. Akcayol, "CNN Based Automatic Speech Recognition: A Comparative Study," *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 13, Aug. 2024, Art. no. e29191, https://doi.org/10.14201/adcaij.29191.

[18] S. A. A. Jeevakumari and K. Dey, "LipSyncNet: A Novel Deep Learning Approach for Visual Speech Recognition in Audio-Challenged Situations," *IEEE Access*, vol. 12, pp. 110891–110904, 2024, https://doi.org/10.1109/ACCESS.2024.3436931.

[19] K. Koutini, H. Eghbal-zadeh, and G. Widmer, "Receptive Field Regularization Techniques for Audio Classification and Tagging with Deep Convolutional Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1987–2000, 2021, https://doi.org/10.1109/TASLP.2021.3082307.

[20] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio Spectrogram Transformer," in *Interspeech 2021*, Brno, Czechia, Aug. 2021, pp. 571–575, https://doi.org/10.21437/Interspeech.2021-698.

[21] A.-H. Jo and K.-C. Kwak, "Classification of Speech Emotion State Based on Feature Map Fusion of TCN and Pretrained CNN Model from Korean Speech Emotion Data," *IEEE Access*, vol. 13, pp. 19947–19963, 2025, https://doi.org/10.1109/ACCESS.2025.3534176.

[22] K. Seaborn, N. P. Miyake, P. Pennefather, and M. Otake-Matsuura, "Voice in Human–Agent Interaction: A Survey," *ACM Computing Surveys*, vol. 54, no. 4, pp. 1–43, May 2022, https://doi.org/10.1145/3386867.

[23] D. Liu, A. Honoré, S. Chatterjee, and L. K. Rasmussen, "Powering Hidden Markov Model by Neural Network based Generative Models," in *24th European Conference on Artificial Intelligence*, Santiago de Compostela, Spain, 2020, https://doi.org/10.48550/ARXIV.1910.05744.

[24] S. Majumdar and B. Ginsburg, "MatchboxNet: 1D Time-Channel Separable Convolutional Neural Network Architecture for Speech Commands Recognition," in *Interspeech 2020*, Shanghai, China, Oct. 2020, pp. 3356–3360, https://doi.org/10.21437/Interspeech.2020-1058.

[25] F. Jia, S. Majumdar, and B. Ginsburg, "MarbleNet: Deep 1D Time-Channel Separable Convolutional Neural Network for Voice Activity Detection," in *ICASSP 2021 -2021 IEEE International Conference on Acoustics, Speech and Signal Processing*, Toronto, ON, Canada, Jun. 2021, pp. 6818–6822, https://doi.org/10.1109/ICASSP39728.2021.9414470.

[26] S. Baghel, M. Bhattacharjee, S. R. M. Prasanna, and P. Guha, "Shouted and Normal Speech Classification Using 1D CNN," in *Pattern Recognition and Machine Intelligence*, vol. 11942, B. Deka, P. Maji, S. Mitra, D. K. Bhattacharyya, P. K. Bora, and S. K. Pal, Eds. Cham, Switzerland: Springer International Publishing, 2019, pp. 472–480.

[27] I. Djemai, S. A. Fezza, W. Hamidouche, and O. Déforges, "Extending 2D Saliency Models for Head Movement Prediction in 360-Degree Images Using CNN-Based Fusion," in *2020 IEEE International Symposium on Circuits and Systems*, Seville, Spain, Oct. 2020, pp. 1–5, https://doi.org/10.1109/ISCAS45731.2020.9181229.

[28] J. G. García Pardo, "Machine Learning Strategies for Diagnostic Imaging Support on Histopathology and Optical Coherence Tomography," Universitat Politècnica de València, Valencia, Spain, 2022.

[29] Q. B. Diep, H. Y. Phan, and T.-C. Truong, "Crossmixed Convolutional Neural Network for Digital Speech Recognition," *PLOS ONE*, vol. 19, no. 4, Apr. 2024, Art. no. e0302394, https://doi.org/10.1371/journal.pone.0302394.

[30] N. Hajarolasvadi and H. Demirel, "3D CNN-Based Speech Emotion Recognition Using K-Means Clustering and Spectrograms," *Entropy*, vol. 21, no. 5, May 2019, Art. no. 479, https://doi.org/10.3390/e21050479.

[31] S. A. Ahmed, E. H. Khalifa, M. Nawaz, F. A. Abdalla, and A. F. A. Mahmoud, "Enhancing Cloud Data Center Security through Deep Learning: A Comparative Analysis of RNN, CNN, and LSTM Models for Anomaly and Intrusion Detection," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 20071–20076, Feb. 2025, https://doi.org/10.48084/etasr.9445.

[32] 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, Nov. 2024, pp. 1–5, https://doi.org/10.1109/IECON55916.2024.10905876.

[33] 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 Letters*, vol. 7, pp. 22–25, 2024, https://doi.org/10.1109/LSSC.2023.3334625.

[34] L. Nwankwo and E. Rueckert, "The Conversation is the Command: Interacting with Real-World Autonomous Robots Through Natural Language," in *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, Boulder, CO, USA, Mar. 2024, pp. 808–812, https://doi.org/10.1145/3610978.3640723.

[35] 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, Sept. 2024, pp. 284–289, https://doi.org/10.1109/iSemantic63362.2024.10762627.

[36] I. J. Kadhim, Tawfeeq E. Abdulabbas, R. Ali, Ali F. Hassoon, and P. Premaratne, "An Enhanced Speech Command Recognition using Convolutional Neural Networks," *Journal of Engineering and Sustainable Development*, vol. 28, no. 6, pp. 754–761, Nov. 2024, https://doi.org/10.31272/jeasd.28.6.8.