Journal of Engineering and Sustainable Development
Vol. 28, No. 06, November 2024
https://jeasd.uomustansiriyah.edu.iq/index.php/jeasd
ISSN 2520-0917
Research Article
https://doi.org/10.31272/jeasd.28.6.8

Enhanced Speech Command Recognition using Convolutional Neural Networks

Inas Jawad Kadhim[1]*, Tawfeeq E. Abdulabbas[2], Riyadh Ali[3], Ali F. Hassoon[4], Prashan Premaratne[5]

[1]Electrical Engineering Technical College, Middle Technical University, Baghdad, Iraq
[2,3,4]Electrical Engineering Department, College of Engineering, Mustansiriyah University, Baghdad, Iraq
[5] School of Electrical and Computer and Telecommunications Engineering, University of Wollongong, NSW, 2522, Australia

[1]https://orcid.org/0000-0001-9404-5653
[2]https://orcid.org/0000-0001-8919-1891
[3]https://orcid.org/0000-0002-1347-1632
[4]https://orcid.org/0000-0002-4981-8407
*Email: *inasjk@mtu.edu.iq*

Article Info

Received	23/11/2023
Revised	01/10/2024
Accepted	19/10/2024

Abstract

In recent years, the growing interest in automatic speech recognition (ASR) has been driven by its wide-ranging applications across various domains. Integrating speech recognition technologies into smart systems highlights the pivotal role of human-machine interaction. This study introduces a robust ASR system that leverages convolutional neural networks (CNNs) in conjunction with Mel-frequency cepstral coefficients (MFCCs). The model's architecture was improved by extensively examining hyperparameters, effectively recognizing ten different spoken commands. The model conducted training and evaluation using the Google Speech dataset, comprising 65,000 audio clips collected from a wide range of speakers across the globe. This dataset accurately represents the natural variations in speech found in real-world scenarios. The design comprises eight storage layers, encompassing convolutional and fully connected layers. It consists of a total of 183,345 weights and utilizes ReLU activation. It is worth mentioning that the average F1-score obtained during the training, validation, and testing stages is 99.06 %, 94.68%, and 95.27%, respectively. Furthermore, the proposed model exhibits about 1.3% improvement in experimental test accuracy over existing methods, confirming its effectiveness in real-world applications.

Keywords: Convolutional Neural Network; Human voice; Machine voice; Mel-frequency cepstral coefficient

1. Introduction

Speech is a crucial mode of communication that allows individuals to articulate their thoughts, concepts, and requirements [1]-[3]. The proliferation of advanced technology and the growing number of smart devices have revolutionized how humans interact with machines, enabling voice to become a highly efficient means of communication [4]. The present research focuses on speech command recognition (SCR), a subset of automatic speech recognition (ASR). SCR involves the recognition of short-spoken commands, an essential process in machine learning to understand and respond to human instructions by associating auditory information with words from language sources [5]. The cognitive system makes communication between humans and machines much easier and more accessible.

The advent of intelligent virtual assistants such as Google Assistant, Amazon Alexa, Apple Siri, and Microsoft Cortana has led to widespread and practical use of speech recognition technology [6]. This technology is utilized in voice dialing, call routing, search, and fundamental data entry tasks [7]. Furthermore, the increasing presence of mobile assistants, sophisticated robots that can understand and respond to voice instructions, and home automation systems like Amazon Echo and Google Home have significantly contributed to the growing need for communication between humans and machines through speech [8]. In addition, SCR has found critical practical applications in various fields, such as medicine [1], education [2], and support for the disabled and visually impaired [4],[9]. Due to the limited resources of small devices, such as those mentioned in references [10]-[12], there is an urgent requirement for lightweight, real-time applications to meet the increasing demand.

Two main classifications of voice recognition techniques exist: classical approaches and deep learning-based methods. Classical methodologies prioritize extracting unique characteristics from unprocessed speech data, employing techniques like the Fourier transform to translate the time-based representation of speech into the frequency-based representation [12],[13]. Afterward, these features are retrieved and represented using phonetic models, such as the hidden Markov model (HMM), to depict the sequence of phonemes in words [14]. Recently, there has been a tendency to merge neural networks with HMM-based approaches to improve performance [15],[16]. Deep learning approaches have become the prevailing trend in speech recognition, exceeding standard machine learning methods in accuracy and ease of application [17]. In [18], researchers trained recurrent deep neural networks (RNNs) with numerous layers to effectively incorporate long-term context. As a result, these networks achieved a test phase error rate of 17.7% on the TIMIT phoneme recognition standard. A separate study [19] devised a deep learning system that exhibited encouraging outcomes in English language acquisition. Deep learning was utilized in [20] to categorize speech, resulting in a 66.22% accuracy rate. As successful applications demonstrate, deep learning techniques have effectively recognized speech emotion [21],[22].

Furthermore, researchers have investigated the modification of deep learning models to accurately identify speech commands in different languages and situations. As an illustration, in reference [23], a model successfully attained high accuracy in identifying Bengali short voice commands. This was accomplished using pre-trained models and extracting Mel-frequency cepstral coefficients (MFCC) characteristics. The study in reference [24] employed a repeated neural network (RNN) to identify Arabic numerals. The RNN utilized long short-term memory (LSTM) cells to handle time-dependent challenges effectively. Another method called CNN-PPG [25] has been developed to combine convolutional neural networks (CNNs) and phonetic posterior gram (PPG) algorithms to recognize verbal commands. This method shows remarkable accuracy and achieves 93.49%. In [26], the authors used a log-mel spectrogram and a deep image classification model, yielding exceptional accuracy on ten commands. However, its performance was slightly affected by the presence of noise.

An important challenge in previous research has been to achieve high accuracy in predicting signals, especially in situations of background noise and time presentation in different languages and speech patterns. Researchers have carefully tried to improve the robustness of speech command recognition devices to maintain their effectiveness in different settings.

The present study proposes a network to improve the accuracy of identifying single-word spoken commands from real-time microphone inputs. The objective is to accurately identify a predefined set of terms from brief audio recordings in quiet and noisy environments. Leveraging the structured design of CNNs, this learning network is trained using the Google speech dataset. The audio spectrum is transformed using MFCC, adding noise to enhance results.

The remainder of this study is organized as follows: Section 2 outlines the design of the convolutional neural network. Section 3 provides details on the materials and methods employed. Section 4 presents the results and subsequence discussion, while Section 5 provides the conclusion.

2. The architecture of convolutional neural networks

The CNN comprises wrapped, subsampled, and fully connected neural layers [16]. The number of layers and how input data is passed between layers varies among networks. The wide range of parameters CNN explores goes beyond traditional voice feature recognition methods and can be learned through training examples. Voice recognition, ranking, and performance evaluation have become major topics with the advancement of CNN. In image processing, the image is used as the direct input, eliminating the need for pre-processing and automatically extracting features. CNN can recognize two-dimensional objects by designing a multi-layered grid, and the network structure is robust to image translation, zoom, and distortion [14],[15],[27].

In sound processing, it is more complex than in image processing because sound has one dimension. Using an audio signal's waveform or time domain directly for processing operations is challenging. Using waveforms to detect and identify acoustic events is almost only possible if they occur in a dynamic media, such as a loud noise in a quiet environment [28]. The audio signal must be transformed into the frequency domain and represented in 2D through one of the conversion methods between the frequency and time domains, such as the auditory spectrum.

Fig. 1 illustrates the basic steps in the deep learning process. Regardless of the network structure design or the problem addressed by deep learning, the process is always iterative. If the network fails to achieve the desired resolution, its architecture must be modified to improve it [29],[30].

In a supervised deep learning approach, the network is trained to extract features from speech commands and classify them based on a speech command classification dataset. At the testing stage, speech recognition is performed by linking a series of convolutional layers with pooling and a fully connected layer, forming a traditional CNN. The network consists of an input layer and an output layer connected by a series of intermediate layers.

The convolution and pooling layers in Fig. 2 represent the feature extraction and learning process. Meanwhile, the fully connected layer and Softmax represent the classification stage. The next subsection will briefly discuss CNN's geometric layout and its most commonly used features for speech command recognition.

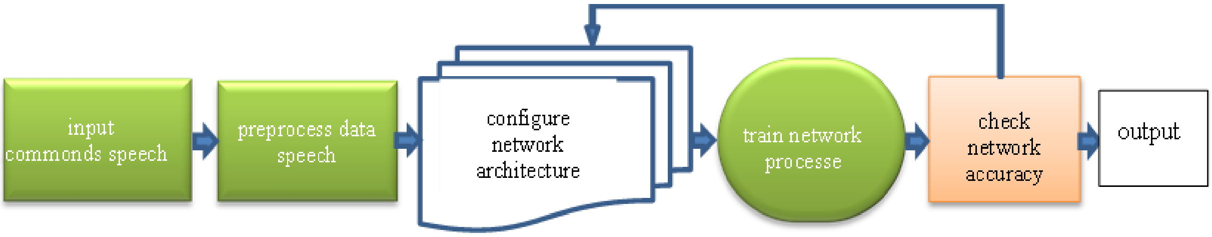


Figure 1. The main steps for the deep learning workflow

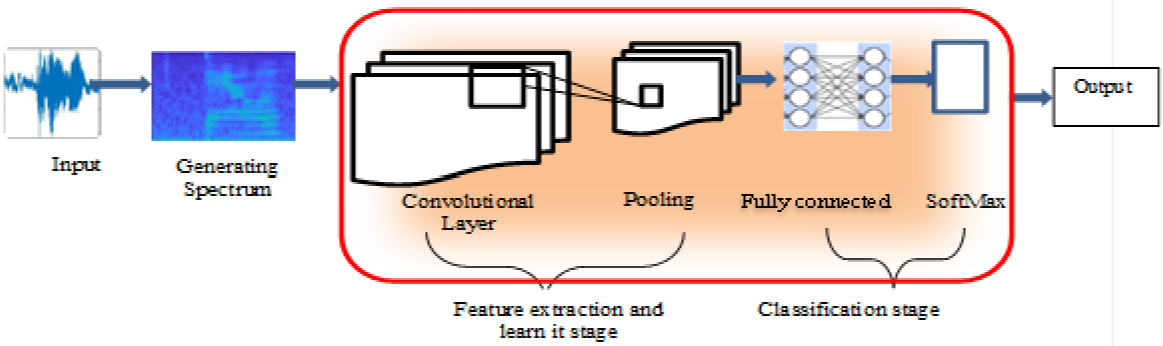


Figure 2. Convolutional Neural Network

2.1. Convolution layer

Once the map of the input layer is realized, the convolution layer becomes a hidden layer and handles two-dimensional data. By applying certain random filters, a feature map is created that allows a CNN to understand the local aspects of the data. Each layer only processes a portion of the input elements instead of the total features from the previous layer.

2.2. Poolling layer

The pooling layer, a second layer, may be connected to the convolution layer. A stride, represented as an arbitrary-sized window, extracts feature maps and reduces their dimensions. The maximum, average, or total of windows can be extracted. This study used the maximum aggregation to obtain the highest possible values from each feature map window.

2.3. Utilizing a rectified linear unit to activate (ReLU)

Several nonlinear activation functions exist in the deep structure of CNNs. This research used rectified linear units (ReLUs) as a common alternative to the sigmoid tangent and logistic hyperbolic functions. ReLUs replace negative values in the matrix with zeros to produce the desired outcome. They are positioned between the pooling layer and the convolution layer.

2.4. Fully connected layer with softmax

The output feature maps of a CNN are connected to a fully connected layer, where each input and output is connected to a learnable weight. After the features are transformed by the stacking of layers and reduced by the grouping of layers, the final output of the grid is assigned, producing probabilities for each category in classification tasks. The number of output nodes in the fully connected final layer is typically equal to the number of classes.

The softmax function activates the output unit and is connected to a fully connected layer for multi-class classification. The geometric structure of this network has a strong ability to recognize speech with high accuracy.

3. Resources and techniques.

Section 3.1 discusses the techniques and tools used for deep learning speech recognition and describes the dataset. Section 3.2 showcases a CNN model for converting and classifying spectrograms.

3.1. Dataset

The dataset used in our research is Google Speech, which consists of 65,000 audio clips of speech commands taken from thousands of speakers from various parts of the world, recorded in varying quality [31]. Each audio file is a short word lasting 1 second. Our study will identify ten specific terms: "Yes", "No", "Up", "Down", "Left", "Right", "On", "Off", "Stop", and "Go". Any other words will be classified as "Unknown". The dataset also includes audio clips containing only background noise, which we have named "Background." The audio data has a frequency of 16 kHz and a length of 1 second. Before creating the spectrogram, noise was added to the speech commands to simulate real-world conditions. 80% of the data was used for training, 10% for validation, and 10% for testing.

3.2. Proposed architecture

The exploratory study used MATLAB 2021a, an Intel Core i7 5600 processor, 8 GB of RAM, SSD, and the Windows 10 operating system, 64-bit edition.

The raw audio signals were transformed into waveforms to determine the appropriate acoustic features and then processed into spectrograms of various sizes [32]. Time and frequency are assigned as the axes of the spectra, forming 2D representations of the spectrogram.

Spectral models provide more information than conventional hand-crafted features for audio analysis and have fewer dimensions than raw audio. Automatic voice recognition employs features from the MFCC approach. The Hann window used for analysis was 400 samples, with 50 filters in the auditory spectrogram. A Fast Fourier Transform (FFT) length of 512 samples was used with a window overlap of 240 samples and 16 kHz speech samples without zero padding. The spectral matrix over time was obtained, as shown in Fig. 3.

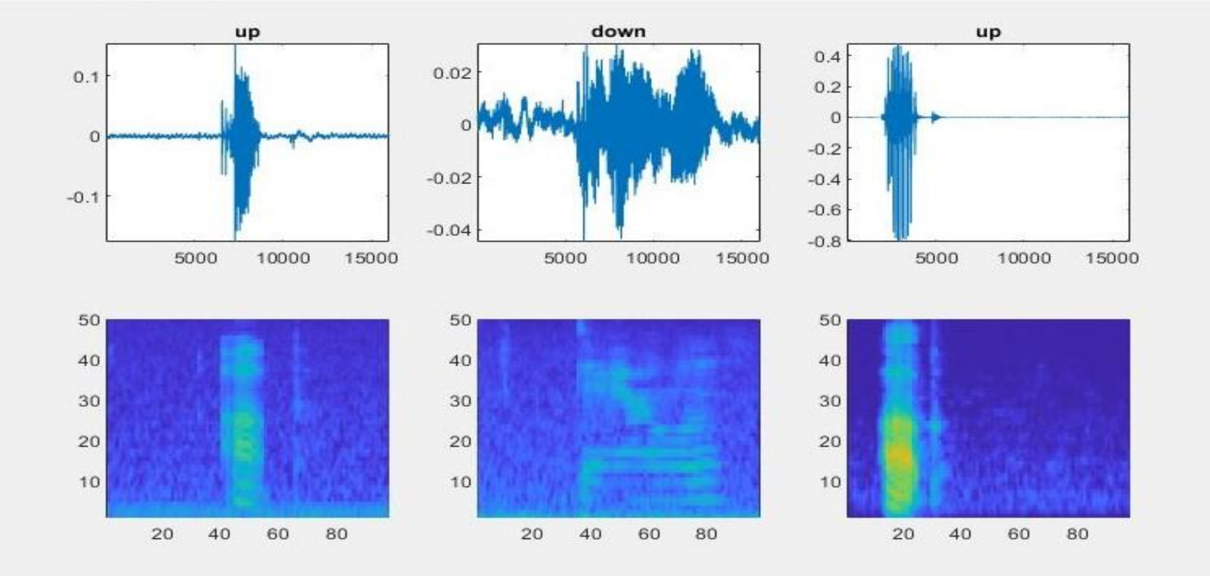


Figure 3. Speech signal with auditory spectrogram

Another objective of this work is to minimize the number of architectural layers to simplify the model and reduce runtime. This is achieved using convolutional layers with a wide enough convolution window, stride, and pooling layers to detect patterns and decrease dimensionality between neural layers. The results of updating the underlying CNN model using the backpropagation approach are shown in Table 1.

There are eight hidden layers of (((([3 4] x15, [3 2] x 15) + batch, normalization layer + RELU activation function in each layer). Moreover, there are four max-pooling layers and one layer fully connected with the softmax layer. The number of weights is 183345.

4. Results and discussion

To assess the performance of the proposed model, the precision, recall, accuracy, and F1-score were evaluated for training, validation, and testing datasets. The SCR dataset, consisting of 10 English speech commands, was used in this study, employing CNN architecture. Fig. 4 illustrates the progress of accuracy and loss during the CNN training process. The curve in the figure shows the training process, while the black curve represents the validation process. The validation process curve comprises epochs, represented by columns of gray and white colors with black nodes.

Table 1. Implemented CNN model configuration

Parameter	***Range***
Hidden Layers for CNN	8 layers
Validation data	10 %
Max Epochs	20
Mini batch size	128
Learning Option	Adam
Shuffle	every-epoch
Initial Learn Rate	3×10^{-4}
Learn Rate Drop Factor	0.1
Learn Rate Drop Period	20
Learn Rate Schedule	piecewise
Max Iteration	3620
Iteration per Epoch	181
Input Nodes	98 × 50
Output Nodes	1

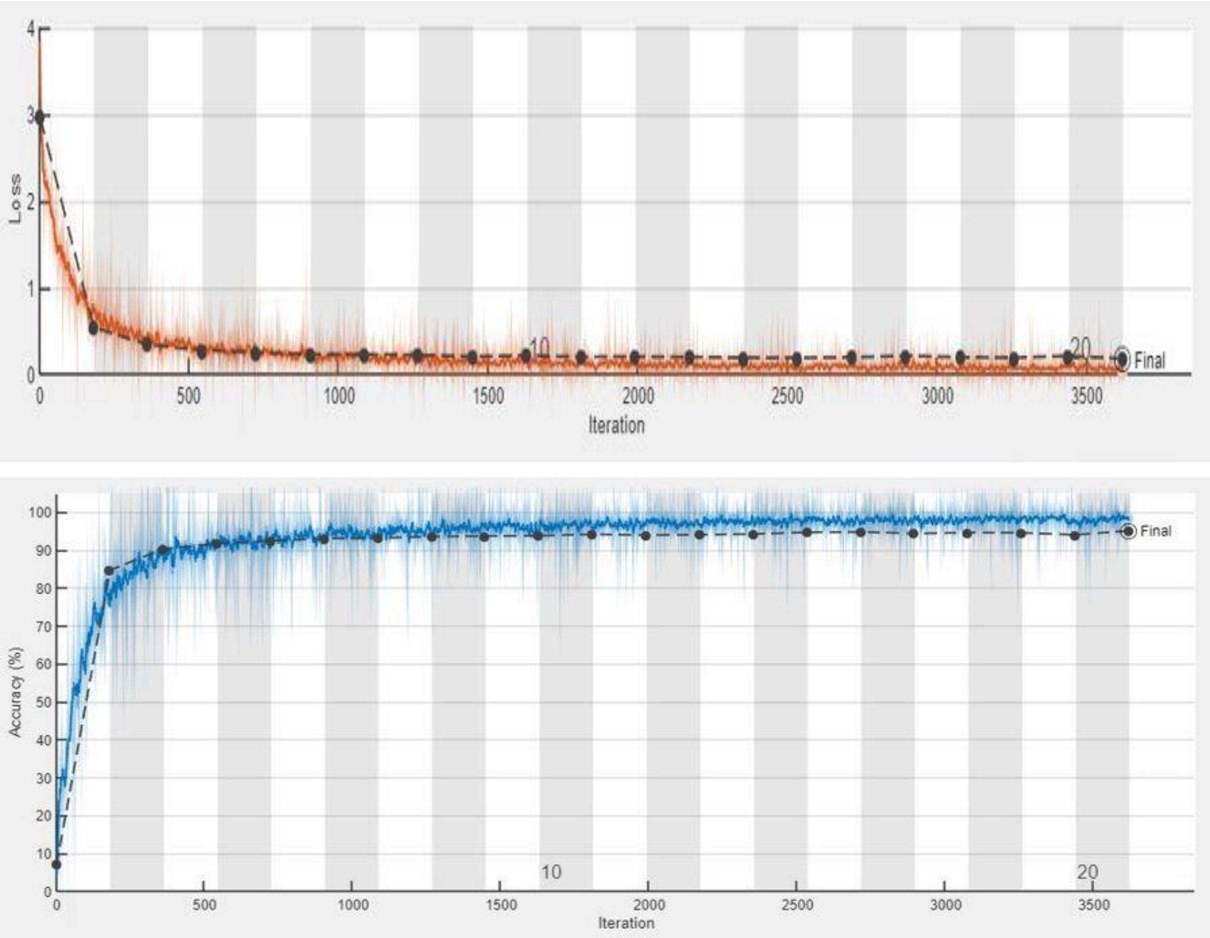
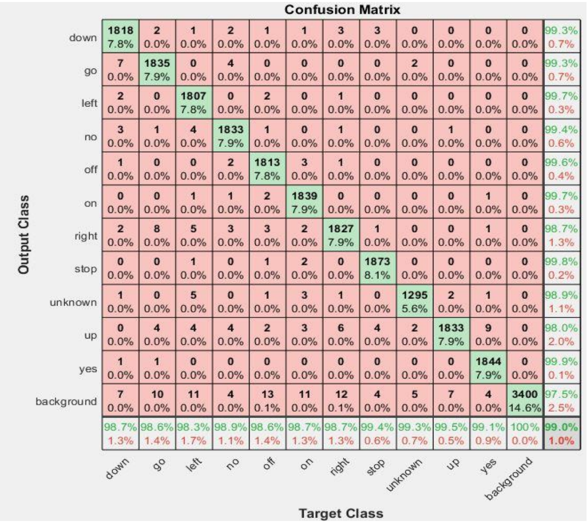


Figure 4. Accuracy and loss vs iteration in training progress

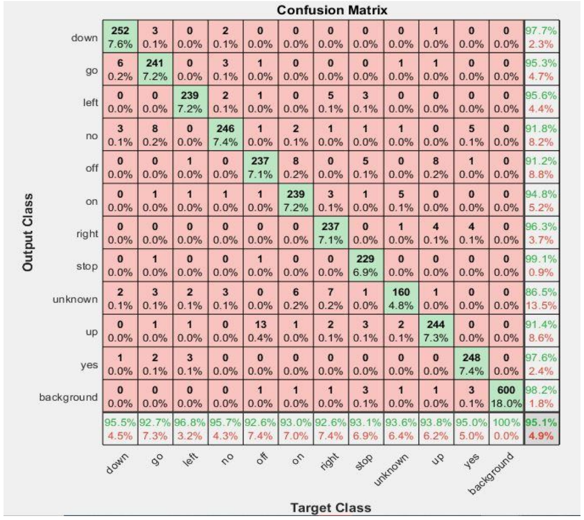
Fig. 5 and 6 display the training and validation data confusion matrix. The confusion matrix provides a visual representation of the performance of the speech recognition system. The matrix shows the number of samples from each true class that were predicted to belong to each class. The true class refers to the actual spoken command, while the predicted class represents the command the neural network recognizes and outputs as its prediction. The diagonal parts in the matrix correspond to the accurately categorized samples, whereas the off-diagonal elements reflect the incorrectly classified samples.

Upon analyzing the confusion matrix of the training datasets, it is clear that the speech recognition system had outstanding results across different classes, displaying high accuracy rates. The "background" class, which included 3400 samples, obtained perfect categorization. The "yes" and "no" classifications demonstrated remarkable accuracy, correctly classifying 1844 and 1833 samples.


Confusion Matrix
Output Class
Target Class

	down	go	left	no	off	on	right	stop	unknown	up	yes	background	
down	1818 7.8%	2 0.0%	1 0.0%	2 0.0%	1 0.0%	1 0.0%	3 0.0%	3 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.3% 0.7%
go	7 0.0%	1835 7.9%	0 0.0%	4 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2 0.0%	0 0.0%	0 0.0%	0 0.0%	99.3% 0.7%
left	2 0.0%	0 0.0%	1807 7.8%	0 0.0%	2 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.7% 0.3%
no	3 0.0%	1 0.0%	4 0.0%	1833 7.9%	1 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	99.4% 0.6%
off	1 0.0%	0 0.0%	0 0.0%	2 0.0%	1813 7.8%	3 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.6% 0.4%
on	0 0.0%	0 0.0%	1 0.0%	1 0.0%	2 0.0%	1839 7.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	99.7% 0.3%
right	2 0.0%	8 0.0%	5 0.0%	3 0.0%	3 0.0%	2 0.0%	1827 7.9%	1 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	98.7% 1.3%
stop	0 0.0%	0 0.0%	1 0.0%	0 0.0%	1 0.0%	2 0.0%	0 0.0%	1873 8.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.8% 0.2%
unknown	1 0.0%	0 0.0%	5 0.0%	0 0.0%	1 0.0%	3 0.0%	1 0.0%	0 0.0%	1295 5.6%	2 0.0%	1 0.0%	0 0.0%	98.9% 1.1%
up	0 0.0%	4 0.0%	4 0.0%	4 0.0%	2 0.0%	3 0.0%	6 0.0%	4 0.0%	2 0.0%	1833 7.9%	9 0.0%	0 0.0%	98.0% 2.0%
yes	1 0.0%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1844 7.9%	0 0.0%	99.9% 0.1%
background	7 0.0%	10 0.0%	11 0.0%	4 0.0%	13 0.1%	11 0.0%	12 0.1%	4 0.0%	5 0.0%	7 0.0%	4 0.0%	3400 14.6%	97.5% 2.5%
	98.7% 1.3%	98.6% 1.4%	98.3% 1.7%	98.9% 1.1%	98.6% 1.4%	98.7% 1.3%	98.7% 1.3%	99.4% 0.6%	99.3% 0.7%	99.5% 0.5%	99.1% 0.9%	100% 0.0%	99.0% 1.0%

Figure 5. Confusion matrix for training data



	down	go	left	no	off	on	right	stop	unknown	up	yes	background	
down	252 7.6%	3 0.1%	0 0.0%	2 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	97.7% 2.3%
go	6 0.2%	241 7.2%	0 0.0%	3 0.1%	1 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	1 0.0%	0 0.0%	0 0.0%	95.3% 4.7%
left	0 0.0%	0 0.0%	239 7.2%	2 0.1%	1 0.0%	0 0.0%	5 0.1%	3 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	95.6% 4.4%
no	3 0.1%	8 0.2%	0 0.0%	246 7.4%	1 0.0%	2 0.1%	1 0.0%	1 0.0%	1 0.0%	0 0.0%	5 0.1%	0 0.0%	91.8% 8.2%
off	0 0.0%	0 0.0%	1 0.0%	0 0.0%	237 7.1%	8 0.2%	0 0.0%	5 0.1%	0 0.0%	8 0.2%	1 0.0%	0 0.0%	91.2% 8.8%
on	0 0.0%	1 0.0%	1 0.0%	1 0.0%	1 0.0%	239 7.2%	3 0.1%	1 0.0%	5 0.1%	0 0.0%	0 0.0%	0 0.0%	94.8% 5.2%
right	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	237 7.1%	0 0.0%	1 0.0%	4 0.1%	4 0.1%	0 0.0%	96.3% 3.7%
stop	0 0.0%	1 0.0%	0 0.0%	0 0.0%	1 0.0%	0 0.0%	0 0.0%	229 6.9%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	99.1% 0.9%
unknown	2 0.1%	3 0.1%	2 0.1%	3 0.1%	0 0.0%	6 0.2%	7 0.2%	1 0.0%	160 4.8%	1 0.0%	0 0.0%	0 0.0%	86.5% 13.5%
up	0 0.0%	1 0.0%	1 0.0%	0 0.0%	13 0.4%	1 0.0%	2 0.1%	3 0.1%	2 0.1%	244 7.3%	0 0.0%	0 0.0%	91.4% 8.6%
yes	1 0.0%	2 0.1%	3 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	248 7.4%	0 0.0%	97.6% 2.4%
background	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.0%	1 0.0%	1 0.0%	3 0.1%	1 0.0%	1 0.0%	3 0.1%	600 18.0%	98.2% 1.8%
	95.5% 4.5%	92.7% 7.3%	96.8% 3.2%	95.7% 4.3%	92.6% 7.4%	93.0% 7.0%	92.6% 7.4%	93.1% 6.9%	93.6% 6.4%	93.8% 6.2%	95.0% 5.0%	100% 0.0%	95.1% 4.9%

Figure 6. Confusion matrix for validation data

The confusion matrix for the validation dataset demonstrates the voice recognition system's strong overall performance, with high accuracy rates for most classes. The "background" class achieved flawless classification, accurately identifying all 600 samples. The "yes" and "no" classes demonstrated impressive accuracy, correctly classifying 248 and 246 samples, respectively. There was some misclassification in other classes, but the number is very small relative to the total sample size.

The confusion matrix of the test data set reveals high accuracy rates of the ASR system for a wide range of classes, highlighting its proficiency in recognizing specific speech commands but particularly pointing to areas of improvement in dealing with background noise and misclassification of unknown commands. This emphasizes the necessity for increased adaptability to enhance the accuracy and robustness of the system, hence ensuring dependable performance in real-life situations. To evaluate the performance of the proposed performance, the accuracy, recall, precision, and F1-score of the CNN can be calculated using the equations [33]:

$$Precision = \frac{T_p}{T_p+F_p} \quad (1)$$

$$Recall = \frac{T_p}{T_p+F_n} \quad (2)$$

$$\text{Accuracy} = \frac{\mathrm{T_p+T_n}}{\mathrm{T_p+T_n+F_p+F_n}} \quad (3)$$

$$\text{F1}-\text{score} = \frac{2\times precision\times recall}{precision+recall} \quad (4)$$

Where $\mathrm{T_p}$ is an accurate positive prediction, $\mathrm{T_n}$ is a true negative prediction, $\mathrm{F_p}$ is a false positive prediction, and $\mathrm{F_n}$ is a false negative prediction.

Fig. 7, 8, and 9 show each speech group's precision, recall, and F1-score metrics during the training, validation, and testing phases. Inspection of Fig. 7 shows consistently high performance on the training dataset, with precision values ranging from 98.3% to 100% and 98.98% on average, and recall rates ranging from 97.5% to 99.9%, with an average of 99.15 %. These results apply exactly class instances to reduce false negatives, demonstrating the model's assessment capabilities. Similarly, F1-scores ranged from 98.7% to 99.59%, averaging 99.06%. In the validation set, Fig. 8, precision and recall metrics range from 91.2% to 99.1%, with an average F1-score of 94.68%, indicating a competitive performance of the model even when facing new observations. Fig. 9 presents the testing set assessment, revealing good precision and recall metrics ranging from 92.6% to 95.5%, with an average F1-score of 95.27%, demonstrating consistent performance across various classes. Table 2 summarizes the proposed models' accuracy, recall, precision, and F1-score rates on the training, validation, and test datasets. The suggested model has achieved high-performance metrics on all three datasets, indicating reliability and accuracy. The slight difference between the training, validation, and testing results demonstrates that the model is generalizing well to new data and is not overfitting the training data.

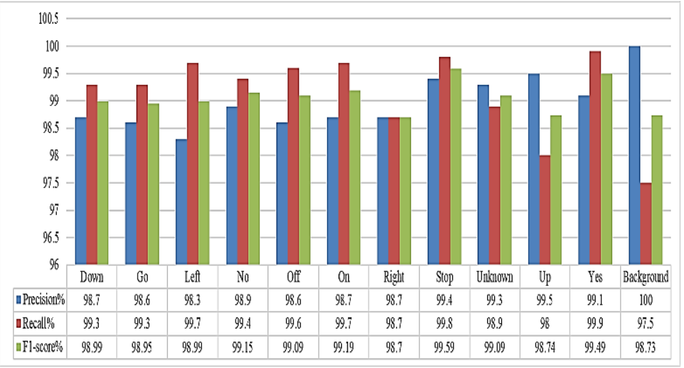
Table 3 compares the suggested method's performance in speech command recognition to that of the state-of-the-art techniques to evaluate its overall effectiveness. The assessment of these techniques throughout the testing phase considered variables such as the frequency of commands, background noise, and instances of encountering unknown words. The results demonstrate that the suggested model outperformed existing techniques, attaining a testing accuracy of 94.8%. In comparison, the log-mel spectrogram and CNN-PPG approaches produced accuracy rates of 93.71% and 93.49%, respectively.

Table 2. Overall Performance Metrics for the Proposed Model

	Precision %	**Recall %**	**Accuracy %**	**F1-score%**
Training Data	98.98	99.15	99.0	99.06
Validation Data	94.53	94.88	95.1	94.68
Testing Data	94.67	95.93	94.8	95.27

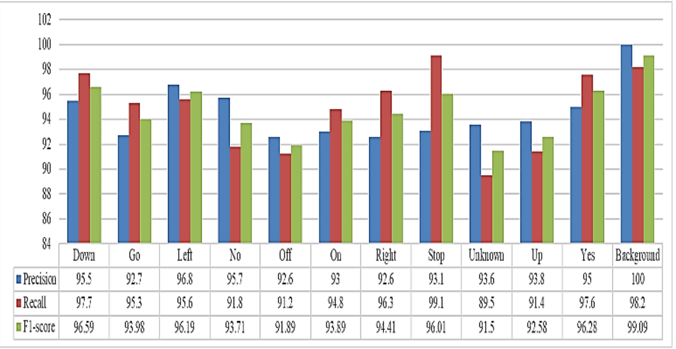
Table 3. Comparison of the proposed model with state-of-the-art techniques

Methods	**No. of the speech command**	**Presence of noise**	**Unknown commands**	**Test accuracy %**
Log-mel spectrogram deep image classification [28]	10	yes	yes	93.71
CNN-PPG [26]	10	no	no	93.49
Proposed model	10	yes	yes	94.8



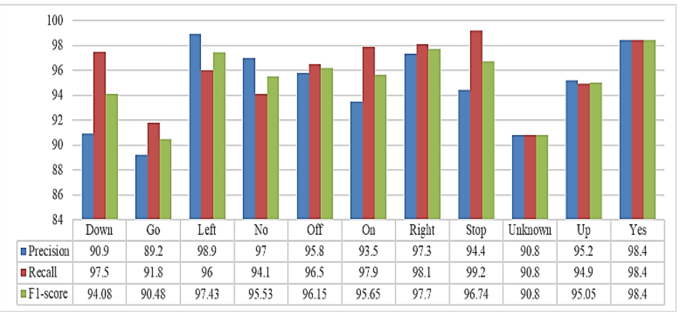
	Down	Go	Left	No	Off	On	Right	Stop	Unknown	Up	Yes	Background
■ Precision%	98.7	98.6	98.3	98.9	98.6	98.7	98.7	99.4	99.3	99.5	99.1	100
■ Recall%	99.3	99.3	99.7	99.4	99.6	99.7	98.7	99.8	98.9	98	99.9	97.5
■ F1-score%	98.99	98.95	98.99	99.15	99.09	99.19	98.7	99.59	99.09	98.74	99.49	98.73

Figure 7. Performance metrics for the training dataset



	Down	Go	Left	No	Off	On	Right	Stop	Unknown	Up	Yes	Background
■ Precision	95.5	92.7	96.8	95.7	92.6	93	92.6	93.1	93.6	93.8	95	100
■ Recall	97.7	95.3	95.6	91.8	91.2	94.8	96.3	99.1	89.5	91.4	97.6	98.2
■ F1-score	96.59	93.98	96.19	93.71	91.89	93.89	94.41	96.01	91.5	92.58	96.28	99.09

Figure 8. Performance metrics for the validation dataset



	Down	Go	Left	No	Off	On	Right	Stop	Unknown	Up	Yes
■ Precision	90.9	89.2	98.9	97	95.8	93.5	97.3	94.4	90.8	95.2	98.4
■ Recall	97.5	91.8	96	94.1	96.5	97.9	98.1	99.2	90.8	94.9	98.4
■ F1-score	94.08	90.48	97.43	95.53	96.15	95.65	97.7	96.74	90.8	95.05	98.4

Figure 9. Performance metrics for the testing dataset

5. Conclusion

This study effectively implemented a convolutional neural network (CNN) integrated with Mel-frequency cepstral coefficients (MFCC) to develop an automated system for recognizing spoken commands. The efficacy of this approach was demonstrated using the Google Speech commands dataset, which comprises 65,000 audio recordings. The model exhibited proficiency in recognizing 10 predetermined spoken commands, including classifying unfamiliar commands and accurately categorizing background noise. Exceptional accuracy was achieved across all three datasets, with a training accuracy of 99.0%, a validation accuracy of 95.1%, and a testing accuracy of 94.8%. A significant breakthrough was achieved by using MFCC for feature extraction, coupled with the subsequent implementation of a CNN model for additional feature learning and classification. This approach increased precision and reduced the overall complexity of the model. While the study primarily focused on identifying spoken commands lasting one second, there is potential for improving the model's effectiveness with longer commands by refining feature selection. Future research will explore other CNN network architectures to enhance the effectiveness of the proposed model further.

Conflict of interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Author Contribution Statement

Inas Jawad Kadhim and Ali F. Hassoon proposed the research problem, methodology, investigation, software, and writing – original draft and editing. Taufeeq E. Abdulabbas and Riyadh A. Abdulhussein Al-Hilali verified the analytical method and review. All authors discussed the results and contributed to the final manuscript.

References

[1] M. A. Grasso, "The long-term adoption of speech recognition in medical applications," *16th IEEE Symposium Computer-Based Medical Systems*, 2003. Proceedings., doi: https://doi.org/10.1109/cbms.2003.1212798.

[2] A. Cocciolo, "Using speech recognition technology in the classroom," *Proceedings of the 9th international conference on Computer supported collaborative learning - CSCL'09*, 2009, doi: https://doi.org/10.3115/1599503.1599538.

[3] F. S. Hassen, "Performance of Discrete Wavelet Transform (DWT) Based Speech Denoising in Impulsive and Gaussian Noise," *Journal of Engineering and Sustainable Development*, vol. 10, no. 2, pp. 175–193, Jun. 2006

[4] S. A. E. Mohamed, A. S. Hassanin, and M. T. B. Othman, "Educational System for the Holy Quran and Its Sciences for Blind and Handicapped People Based on Google Speech API," *Journal of Software Engineering and Applications*, vol. 07, no. 03, pp. 150–161, 2014, doi: https://doi.org/10.4236/jsea.2014.73017.

[5] H. Sak, F. Beaufays, K. Nakajima, and C. Allauzen, "Language model verbalization for automatic speech recognition," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, doi: https://doi.org/10.1109/icassp.2013.6639276.

[6] V. Kepuska and G. Bohouta, "Next-generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa, and Google Home)," *2018 IEEE 8th Annual Computing and Communication Workshop and Conference* (CCWC), Jan. 2018, doi: https://doi.org/10.1109/ccwc.2018.8301638.

[7] S. Husnjak, D. Perakovic, and I. Jovovic, "Possibilities of Using Speech Recognition Systems of Smart Terminal Devices in Traffic Environment," *Procedia Engineering*, vol. 69, pp. 778–787, 2014, doi: https://doi.org/10.1016/j.proeng.2014.03.054.

[8] M. Soltanian, J. Malik, J. Raitoharju, A. Iosifidis, S. Kiranyaz, and M. Gabbouj, "Speech Command Recognition in Computationally Constrained Environments with a Quadratic Self-Organized Operational Layer," *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2021, doi: https://doi.org/10.1109/ijcnn52387.2021.9534232.

[9] S. Okada, Y. Tanaba, H. Yamauchi, and S. Sato, "Single-surgeon thoracoscopic surgery with a voice-controlled robot," *The Lancet*, vol. 351, no. 9111, p. 1249, Apr. 1998, doi: https://doi.org/10.1016/s0140-6736(98)24017-7.

[10] B. Dal and M. Askar, "Fixed-point FPGA Implementation of ECG Classification using Artificial Neural Network," *2022 Medical*

Technologies Congress (TIPTEKNO), Oct. 2022, doi: https://doi.org/10.1109/tiptekno56568.2022.9960216.

[11] A. N. Azhiimah, K. Khotimah, M. S. Sumbawati, and A. B. Santosa, "Automatic Control Based on Voice Commands and Arduino," *Proceedings of the International Joint Conference on Science and Engineering (IJCSE 2020)*, 2020, doi: https://doi.org/10.2991/aer.k.201124.006.

[12] G. H. Shakoory, "FPGA Implementation of Multilayer Perceptron for Speech Recognition," *Journal of Engineering and Sustainable Development*, vol. 17, no. 6, pp. 175–185, Dec. 2013

[13] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 11, pp. 1641–1648, 1989, doi: https://doi.org/10.1109/29.46546.

[14] A. Ouisaadane and S. Safi, "A comparative study for Arabic speech recognition system in noisy environments," *International Journal of Speech Technology*, vol. 24, no. 3, pp. 761–770, Apr. 2021, doi: https://doi.org/10.1007/s10772-021-09847-7.

[15] H.-P. Lin, Y.-J. Zhang, and C.-P. Chen, "Systems for Low-Resource Speech Recognition Tasks in Open Automatic Speech Recognition and Formosa Speech Recognition Challenges," *Interspeech* 2021, Aug. 2021, doi: https://doi.org/10.21437/interspeech.2021-358.

[16] D. Palaz, M. Magimai-Doss, and R. Collobert, "End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition," *Speech Communication*, vol. 108, pp. 15–32, Apr. 2019, doi: https://doi.org/10.1016/j.specom.2019.01.004.

[17] A. S. Dhanjal and W. Singh, "A comprehensive survey on automatic speech recognition using neural networks," *Multimedia Tools and Applications*, vol. 83, no. 8, pp. 23367–23412, Aug. 2023, doi: https://doi.org/10.1007/s11042-023-16438-y.

[18] H. F. Pardede, P. Adhi, V. Zilvan, A. Ramdan, and D. Krisnandi, "Deep convolutional neural networks-based features for Indonesian large vocabulary speech recognition," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 12, no. 2, p. 610, Jun. 2023, doi: https://doi.org/10.11591/ijai.v12.i2.pp610-617.

[19] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, doi: https://doi.org/10.1109/icassp.2013.6638947.

[20] L. Meng, P. Kuppuswamy, J. Upadhyay, S. Kumar, S. V. Athawale, and M. A. Shah, "Nonlinear Network Speech Recognition Structure in a Deep Learning Algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–7, Mar. 2022, doi: https://doi.org/10.1155/2022/6785642.

[21] P. Lakkhanawannakun and C. Noyunsan, "Speech Recognition using Deep Learning," *2019 34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, Jun. 2019, doi: https://doi.org/10.1109/itc-cscc.2019.8793338.

[22] F. R. Jr. Arnel Fajardo, "Convolutional Neural Network for Automatic Speech Recognition of Filipino Language," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 9, no. 1.1 S I, pp. 34–40, Feb. 2020, doi: https://doi.org/10.30534/ijatcse/2020/0791.12020.

[23] L. Khurana, A. Chauhan, M. Naved, and P. Singh, "Speech Recognition with Deep Learning," *Journal of Physics: Conference Series*, vol. 1854, no. 1, p. 012047, Apr. 2021, doi: https://doi.org/10.1088/1742-6596/1854/1/012047.

[24] S. Ahmed Sumon, J. Chowdhury, S. Debnath, N. Mohammed, and S. Momen, "Bangla Short Speech Commands Recognition Using Convolutional Neural Networks," *2018 International Conference on Bangla Speech and Language Processing (ICBSLP)*, Sep. 2018, doi: https://doi.org/10.1109/icbslp.2018.8554395.

[25] A. S. Mahfoudh BA WAZIR and J. Huang CHUAH, "Spoken Arabic Digits Recognition Using Deep Learning," *2019 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, Jun. 2019, doi: https://doi.org/10.1109/i2cacis.2019.8825004.

[26] Y.-Y. Lin et al., "A Speech Command Control-Based Recognition System for Dysarthric Patients Based on Deep Learning Technology," *Applied Sciences*, vol. 11, no. 6, p. 2477, Mar. 2021, doi: https://doi.org/10.3390/app11062477.

[27] A. H. Shah, A. H. Miry, and T. M. Salman, "Automatic Modulation Classification Using Deep Learning Polar Feature," *Journal of Engineering and Sustainable Development*, vol. 27, no. 4, pp. 477–486, Jul. 2023, doi: https://doi.org/10.31272/jeasd.27.4.5.

[28] A. Patra, C. Pandey, K. Palaniappan, and P. K. Sethy, "Convolutional Neural Network-Enabling Speech Command Recognition," *Lecture Notes on Data Engineering and Communications Technologies*, pp. 321–332, Oct. 2022, doi: https://doi.org/10.1007/978-981-19-3035-5_25.

[29] R. Serizel, V. Bisot, S. Essid, and G. Richard, "Acoustic Features for Environmental Sound Analysis," *Computational Analysis of Sound Scenes and Events*, pp. 71–101, Sep. 2017, doi: https://doi.org/10.1007/978-3-319-63450-0_4.

[30] S. Ajibola Alim and N. Khair Alang Rashid, "Some Commonly Used Speech Feature Extraction Algorithms," *From Natural to Artificial Intelligence - Algorithms and Applications*, Dec. 2018, doi: https://doi.org/10.5772/intechopen.80419.

[31] G. Sharma, K. Umapathy, and S. Krishnan, "Trends in audio signal feature extraction methods," *Applied Acoustics*, vol. 158, p. 107020, Jan. 2020, doi: https://doi.org/10.1016/j.apacoust.2019.107020.

[32] P. Warden, (2018). Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*.

[33] A. Adnan Alnawas, M. Al-Jawad, and H. Alharbi, "A Prediction Model Based On Students's Behavior In E-Learning Environments Using Data Mining Techniques," *Journal of Engineering and Sustainable Development*, vol. 26, no. 5, pp. 115–126, Sep. 2022, doi: https://doi.org/10.31272/jeasd.26.5.11.