Audio command recognition methods are essential to be recognized for performing user instructions, especially for people with disabilities. Previous studies couldn't examine and classify the performance optimization of up to twelve audio commands categories. This work develops a microphone-based audio commands classifier using a convolutional neural network (CNN) with performance optimization to categorize twelve classes including background noise and unknown words. The methodology mainly includes preparing the input audio commands for training, extracting features, and visualizing auditory spectrograms. Then a CNN-based classifier is developed and the trained architecture is evaluated. The work considers minimizing latency by optimizing the processing phase by compiling MATLAB code into C code if the processing phase reaches a peak algorithmically. In addition, the method conducts decreasing the frame size and increases the sample rate that is also contributed to minimizing latency and maximizing the performance of processing audio input data. A modest bit of dropout to the input to the final fully connected layer is added to lessen the likelihood that the network will memorize particular elements of the training data. We explored expanding the network depth by including convolutional identical elements, ReLu, and batch normalization layers to improve the network's accuracy. The training progress demonstrated how fast the accuracy of the network is increasing to reach about 98.1 %, which interprets the ability of the network to over-fit the data of training. This work is essential to serve speech and speaker recognition such as smart homes and smart wheelchairs, especially for people with disabilities

Keywords: audio, commands, spectrogram, classifier, deep-learning, CNN, network, real-time

UDC 621
DOI: 10.15587/1729-4061.2023.273492

DEVELOPING MICROPHONE-BASED AUDIO COMMANDS CLASSIFIER USING CONVOLUTIONAL NEURAL NETWORK

Shakir Mahmood Mahdi
Corresponding author
Master of Computer Science
Department of Higher Education
University of Technology-Iraq
Al-Sina'a str., Baghdad, Iraq, 10066
E-mail: shakir.m.mahdi@uotechnology.edu.iq
Sabreen Ali Hussein
Master of Computer Science
Department of Mathematics and Computer
College of Basic Education
University of Babylon
Al Najaf's str., Al Hillah, Babylon, Iraq, 51002
Hayder Mahmood Salman
Doctor of Computer Science
Department of Computer Science
Al-Turath University College
Al Mansour, Baghdad, Iraq, 10013
Alyaa Hamel Sfayyih
Bachelor in Biomedical Engineering*
Nasri Bin Sulaiman
Associate Professor in Microelectronics Engineering *
*Department of Electrical and Electronics Engineering
College of Engineering
University Putra Malaysia
Jalan Universiti str., 1, Serdang, Selangor, Malaysia, 43400

Received date 10.10.2022
Accepted date 20.01.2023
Published date 28.02.2023

How to Cite: *Mahdi, S. M., Hussein, S. A., Salman, H. M., Sfayyih, A. H., Sulaiman, N. B. (2023). Developing microphone-based audio commands classifier using convolutional neural network. Eastern-European Journal of Enterprise Technologies, 1 (9 (121)), 72–81. doi: https://doi.org/10.15587/1729-4061.2023.273492*

1. Introduction

With the rapid evolution of portable devices, interacting with machines using audio technology has become increasingly prevalent. As compared to typing by hand, keyword spotting technology provides a viable method for providing a fully hands-free interface that is especially useful for mobile devices. Corresponding products like iPhone's Siri and Google Now exploit speech commands technique. For instant, Google also presents voice-based searching [1, 2] on Android mobiles, and a fully hands-free incident called "Ok Google"[3–6]. Environmental sound signals are heterogeneous, multi-source, and time-varying. To process these signals for event detection in applications for ambient assisted living, numerous systems have been developed. Typically, feature extraction, selection, and classification are used by these systems. Vocal tract constrictions provide a speech signal, and different sounds can be produced by various vocal tract constrictions. A speech signal carries both the identity of the speaker and the message. The voice-operated wheelchair is one of the solutions for specialized applications of speaker and speech recognition like a voice-operated wheelchair, where both the speaker and speech have to be recognized for the movement of a wheelchair [7]. However, despite major advances, several important questions remain unanswered [8] such as the issue of sound source separation technique that converts channel-based to object-based audio. Azimuth-frequency (AF) based sound source separation has been proposed for converting channel-based audio to object-based audio with difficulties in setting the optimal azimuth and width [9]. The backpropagation algorithm, the most important stage of Deconvolutional Neural Networks (DNN), which is a Convolutional Neural Network (CNN) but works in a reverse technique, was discovered in the 1980s (BPA). This approach used the gradient descent method to quickly speed up the training procedure [10]. CNNs are useful

models for interacting with voice recognition systems and successful variations of deep neural networks (DNNs). Although CNNs are somewhat effective models for creating speech recognition systems, this effective architectural design is rather difficult. For the recognition procedure to be performed, their design necessitates prior knowledge and expertise [11]. This learning architecture was inspired by the natural visual perception paradigm of living things. The term "convolution" refers to a linear mathematical procedure between matrices. Numerous studies have proven that CNNs perform better on categorization tasks. The traditional CNN model's high computing cost and slow learning time for big datasets, however, make it potentially unfeasible.

Deep neural networks have improved the speech recognition process' efficiency. In the machine learning branch of deep neural networks, various layers of effective learning algorithms are grouped to carry out particular tasks. Deep neural networks use a variety of non-linear activation functions to conceptually mimic the process. Additionally, they aid learning by employing feature sets produced from original data (speech signals).

Therefore, the significant advancements made in deep neural networks over the past ten years lead the use of deep learning classifiers methods to be applied to a wide range of problems, including image processing, speaker identification, speech recognition, web searches, dictation, online data filtering, and numerous issues relating to artificial intelligence.

2. Literature review and problem statement

The research [8] proposed an analytical model of audio content to detect non-disturbing events, especially in smart homes, by focusing on the following three issues:

1) the effect of signal-to-noise ratio and the distance between the microphone and the sound source recognition accuracy in an environment in which the system has not been trained;

2) the most appropriate features and classifiers in the presence of background noise,

3) the effect of signal duration on recognition accuracy.

Although, they had competitive ranking results for two of these systems. The first, which used a gradient boost classifier, achieved an F1 score of 90.2 % and a recognition accuracy of 91.7 %. And the second used a 2D CNN with salt spectroscopy images, which achieved an F1 score of 92.7 % and a recognition accuracy of 96 %. However, many important questions remain unanswered, especially in real-world settings. Azimuth frequency (AF) based audio source separation has been proposed to convert channel-based to object-based audio by Paper [9]. Although, the proposed method provides higher accuracy than SDR, SAR, and SIR and separation accuracy of minimal response Non-distorted response (MVDR), unfortunately, it is difficult to set the optimal azimuth and width using this technology. The paper [12], used the recurrent neural network (RNN) model in its proposal to classify commands and compare them with other machine learning classifiers. Although an accuracy of over 98 % was noted with a combination of walled redundant unit (GRU) and long-term memory (LSTM), their proposal did not provide insight into the clarity of the system for recognizing human speech when the number of noise sources increases. The study [13] relied on the electroencephalogram of patients with disabilities to classify the visually constructed words. Whereas, the brain-computer interface (BCI) technology translates voluntary choices into an active command using brain activity. Although their objective was to interpret EEG signals to visualize speech, with an average accuracy of 95.3 %, their system was not used as a technique to help patients, because the results in its application were not satisfactory. The study [14] was concerned with animal phonemic classification using a set of convolutional neural networks. A set of classifiers was presented that worked on different types of animal acoustic data sets using six different CNNs. However, their study did not provide sufficient information and did not record the required results comprehensively, they did not detect and recognize audio signals of speech commands using terminology. Study [15] focused its attention on the visual brain-computer interface based on ERP and studying the effect of sound on it. Where the BCI audio-visual task is designed, which requires focusing on the output commands and calculating the number at the same time according to an audio story. Although the results were that when the story running speed is increased, the amplitude of the P300 and N200 potentials decreases by 0.86 V (p=0.0239) and 0.69 μV (p=0.0158) in the occipital parietal region, resulting in a 5.95 % (p=0.0101) decrease in accuracy and slope 9.53 bits/min (p=0.0416) information transfer rate, there is no develop a convolutional neural network classifier or evaluate the trained architecture. The study [16], presented two classifiers that were developed as a result of their research for the recognition of verbal emotions. Although the study gave an accuracy of 57.29 percent by creating its first model using the multilayer perceptron (MLP) classifier, they did not use the microphone to develop a classifier of voice commands while using the convolutional neural network to improve performance and the results did not clear.

According to the above literature, despite the rapid pace of progress, many questions remain unanswered, especially in real-world settings, so the process of classifying microphone-based voice commands using a convolutional neural network is critical and a must for services that benefit humanity.

3. The aim and objectives of the study

The aim of the study is to develop a microphone-based audio commands classifier using a convolutional neural network with performance optimization to categorize twelve classes. This will make the system network possible to serve particular applications of speech and speaker recognition like smart homes and smart wheelchairs, the audio command is essential to be recognized for performing user instructions, especially for people with disabilities.

To achieve this aim, the following objectives are accomplished:

– to prepare the input audio commands for training, extract features and visualize auditory spectrograms;

– to evaluate the trained architecture.

4. Materials and methods

4. 1. Object and hypothesis of the study

The main in this work is to detect and recognize audio signals of speech commands using terminology and con-

volutional neural network techniques for optimizing the performance through a simulation model. A dataset of audio commands in [17] has been used to verify the developed network architecture. A microphone is employed as an input audio stream command when testing the network. MATLAB toolbox is implemented in real-time streaming algorithms to present basic techniques and key terminology for stream processing optimization.

4. 2. Real-time acquiring process

The audio stream input is acquired from the sound card device using *"audioDeviceReader"* MATLAB object command. The configuration of the sound card is performed by this object to communicate the sound device properties including buffer size, bit depth, sample rate, and channel mapping between columns output object and the card's input channels. Fig. 1 shows a description diagram to acquire a mono-channel audio signal.

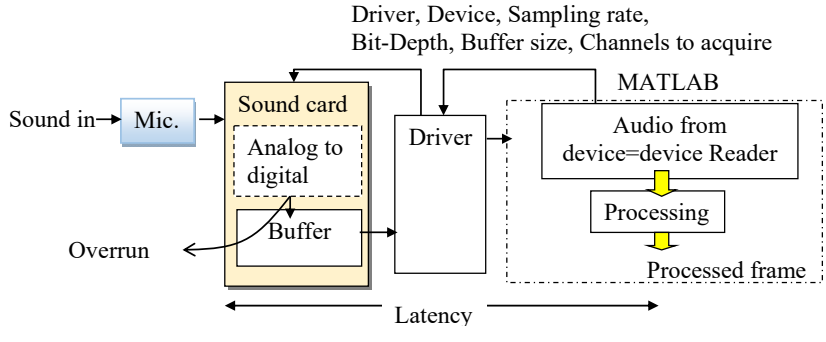

Fig. 1. Description diagram for collecting mono-channel audio signal

In the real-time acquiring process, the sound card device receives the electrical audio signals from a microphone, which converts the input from analogue to digital (ADC) at a bit depth, buffer size, and a specified sample rate.

The ADC transfers audio signals into the buffer, and the new signals are dropped when the buffer is full. The driver is used by the *audioDeviceReader* to repeatedly retrieve the oldest frame from the sound card buffer.

The object (*audioPlayerRecorder*) proceeds the obtained signals into the MATLAB environment for handing out. Concurrently, the auditory data that is processed is assigned as a next I/O cycle state to the *audioPlayerRecorder* to be played back.

4. 3. Performance Optimization techniques

To maximize the performance of processing audio input data, let's close any unnecessary computer processes, including sync programs and mail clients. These processes have the ability to asynchronously request CPU time and interrupt the audio processing loop. Let's also remove the real-time loop's visualization and plotting. AMATLAB-based DSP toolbox is used as a scope and spectrum analyzer to visualize and update the processing loops. A command *drawnow* is used to specify a limited update rate for bespoke graphics or when handling callbacks in the loop to optimize the event queue. When there is heavy audio processing in performing the loop algorithm, profiling for locating the bottlenecks is attempted by applying the following measures:

– replacing the user codes with new features in MATLAB to optimize the execution speed;

– following the highest performance commands and creating an executables file, results in speed execution;

– considering converting the algorithm into a MATLAB-VST plug-in, this employs hood-based C code to obtain an execution faster than that in a handwritten MATLAB system;

– optimizing the Output latency, which is a time delay measurement between the time that audio is heard by a speaker and the time to generate a MATLAB audio frame;

– optimizing the input latency is the time delay measurements between the time in which the frame is output through the processing phase and the time that audio enters the sound card;

– minimizing latency through optimizing the processing phase by compiling MATLAB code into C code if the processing phase reaches a peak algorithmically. Decreasing the frame size and increasing the sample rate also contributed to minimizing latency.

4. 4. Flowchart

The stage begins by loading the Google Speech Commands Dataset, the developed network is able to recognize the audio input as silence or background noise. A MATLAB function called *"augmentDataset"* is used to generate one-second segments of a noisy background from the input large audio data of the Speech Commands background Google dataset. The function then divides the segments into validation and train folders. The flowchart for the preparation stage of the input data is shown in Fig. 2.

The commands that are required to be recognized by the model are specified in this stage. This includes labeling all folders that do not have background noise or commands as "undefined". Then a set of commands is generated to approximate the distribution of all these commands. The developed architecture employs this set to train the network on the difference between commands and other information. Only a portion of the unknown commands of the training group is included to reduce the class imbalance among the unknown and known commands for faster processing.

In extracting feature stage, let's extract auditory spectrograms by defining the audio input parameters as follows:

fs=16e3, numBands=50,

FFTLength=512, hopDuration=0.010,

frameDuration=0.025, and segmentDuration=1,

where *fs* represents the known sample rate of the dataset, "*numBands*" represents the auditory spectrogram number of filters, "*hopDuration*" denotes the spectrum time steps, "*frameDuration*" denotes every frame duration for spectrum calculations, and the "*segmentDuration*" represents every speech clip duration (seconds). A series of transform stages to pad the audio signals to a constant size, apply a logarithm, and extract the features is shown in Fig. 3, and to convert all audio input of the dataset, Fig. 4 shows how every file acquires to pass through the transforms.

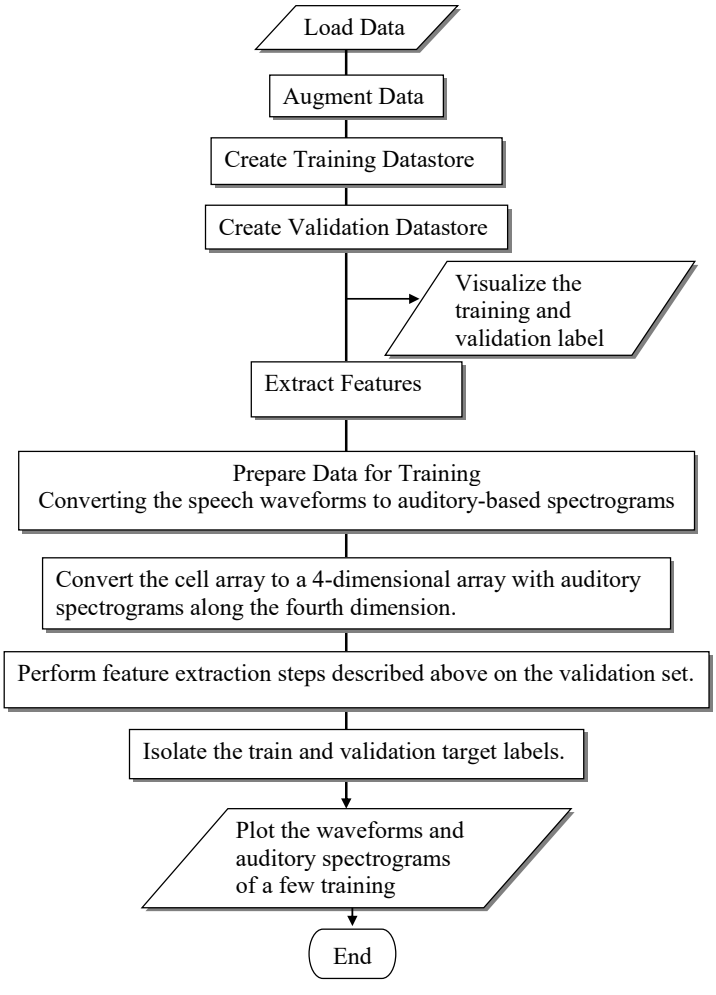

Fig. 2. Flowchart of the preparing data stage

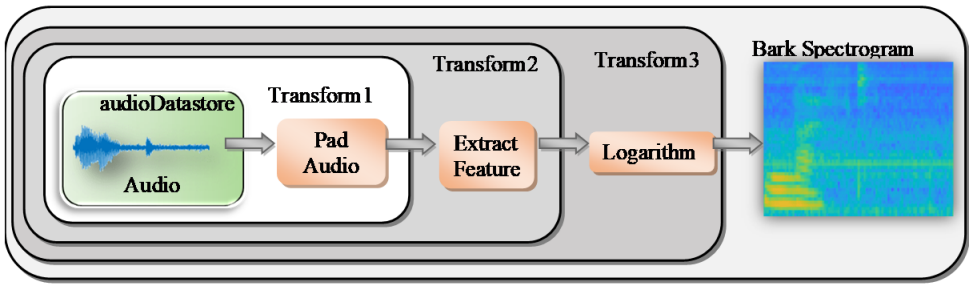

Fig. 3. The flow of audio signal within the sound card

The outcome of this stage is a one-cell vector in which every element is associated by its auditory spectrogram obtained from the audio input. A network is created as a layer array architecture with batch normalization and convolutional layers. Frequency and time feature maps are down-sampled through a max-pooling layer. The feature map input pools globally on time by adding a final max-pooling layer. This approximates (enforces) the invariance of time-translation in the spectrogram inputs, which allows the same classification network to implement the exact location of the audio in time independently. Additionally, the number of parameters in the final fully connected layer is greatly decreased by global pooling. Let's add a modest bit of dropout to the input to the final fully connected layer to lessen the likelihood that the network will memorize particular elements of the training data.

The developed CNN architecture has only five convolutional layers and few filters in the network. Let's explore expanding the network depth by including an identical element of ReLu, batch normalization, and convolutional layers to improve the network's accuracy. Increasing the number of filters can also increase the number of convolutional filters. The network architecture is shown in Fig. 5.

Let's utilize category weights that are inversely proportionate to the total of training instances in every category to give every category an equal overall weight in the loss. The training algorithm is self-determining of the whole normalization of the category weights while using the Adam optimizing to train the neural architecture. The flowchart of performing the network architecture is shown in Fig. 6.

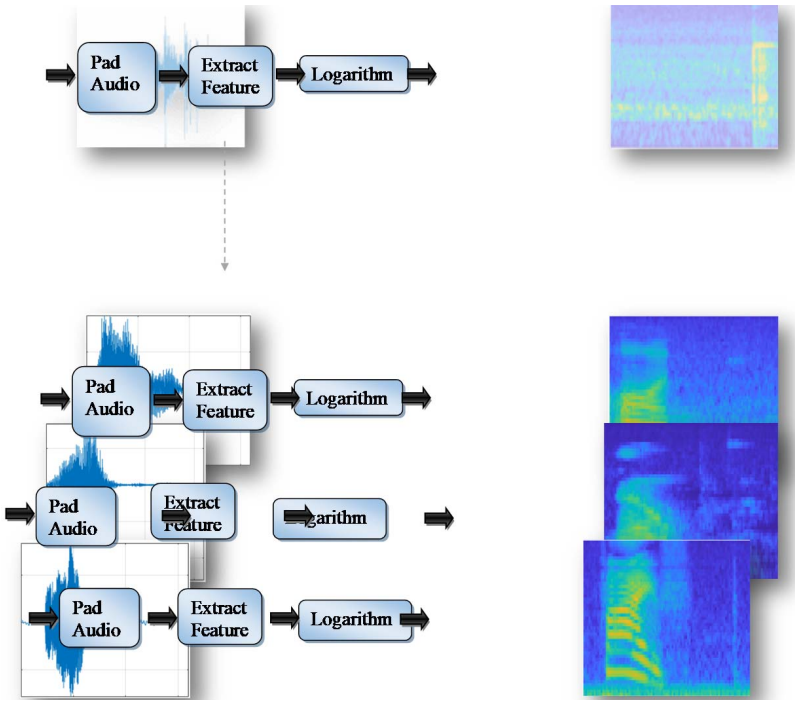

Fig. 4. Every file acquires to pass through the transforms

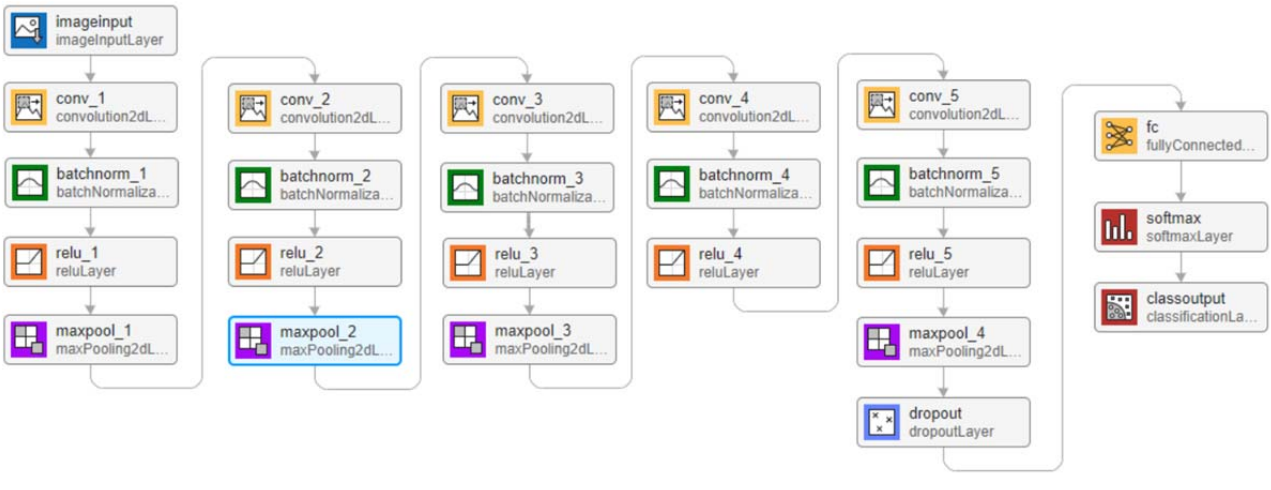

Fig. 5. The developed network architecture

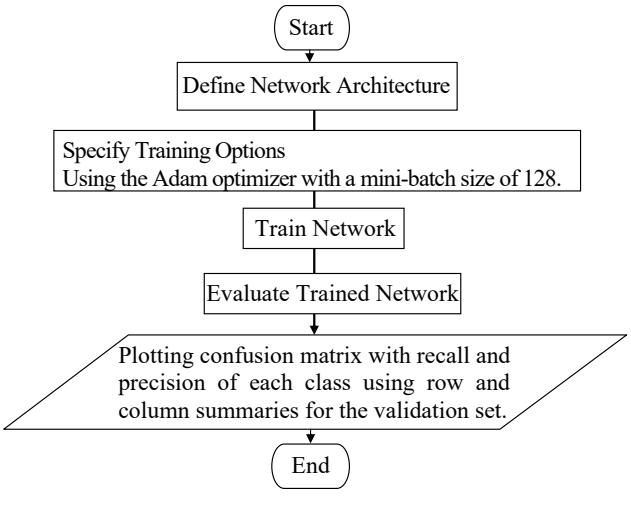

Fig. 6. The flowchart of performing the network architecture

Using data from the training and validation sets, let's determine the network's ultimate accuracy. On this data set, the network is quite accurate. However, the distributions of the training, validation and test data are comparable and may not accurately represent real-world settings. This restriction is especially relevant to the unknown category, which includes utterances that only contain a few words.

It is crucial to take into account the constraints on the available memory and computing resources while working on applications with constrained hardware resources, such as mobile applications. Calculate the network's overall size in kilobytes and evaluate the CPU's prediction speed. The classification time of a single input image is the prediction time. Shorter prediction times per image result from the ability to classify numerous photos in the network at once. However, the single-image prediction time is the most important for categorizing streaming audio.

5. Results of developing microphone-based audio commands classifier

5. 1. Preparing input audio commands

To set up the input audio signals for effective training of a CNN, let's change the command signals into auditory spectrograms. A dataset containing just the command words, the noise of the background, and the unknown words subset is created. This also includes computing the number of every class. The distribution visualization of the validation and training labels is shown in Fig. 7.

To visualize audio inputs Fig. 8 shows the signals' waveform with their corresponding auditory-based spectrogram for ordinary three-training audio signals.

The above stage included performing; the steps of feature extraction for the validation dataset, and the playing of their corresponding audio speech.

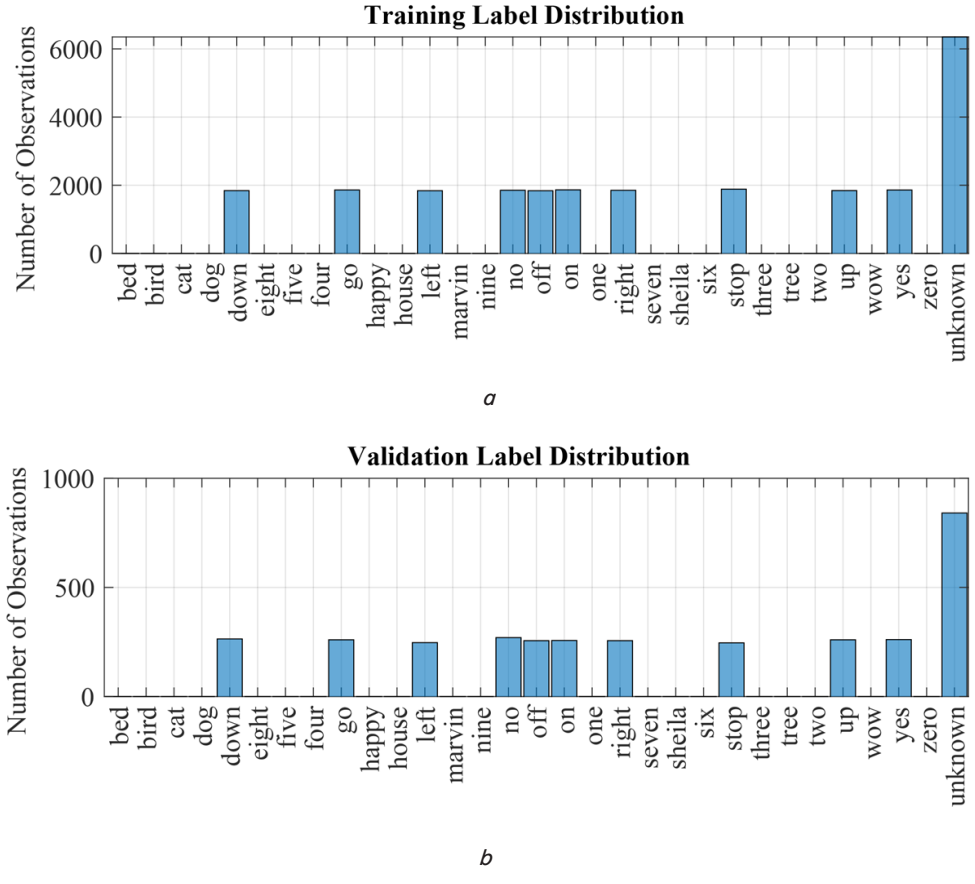

Fig. 7. The distribution visualization of: *a* – the training; *b* – validation labels

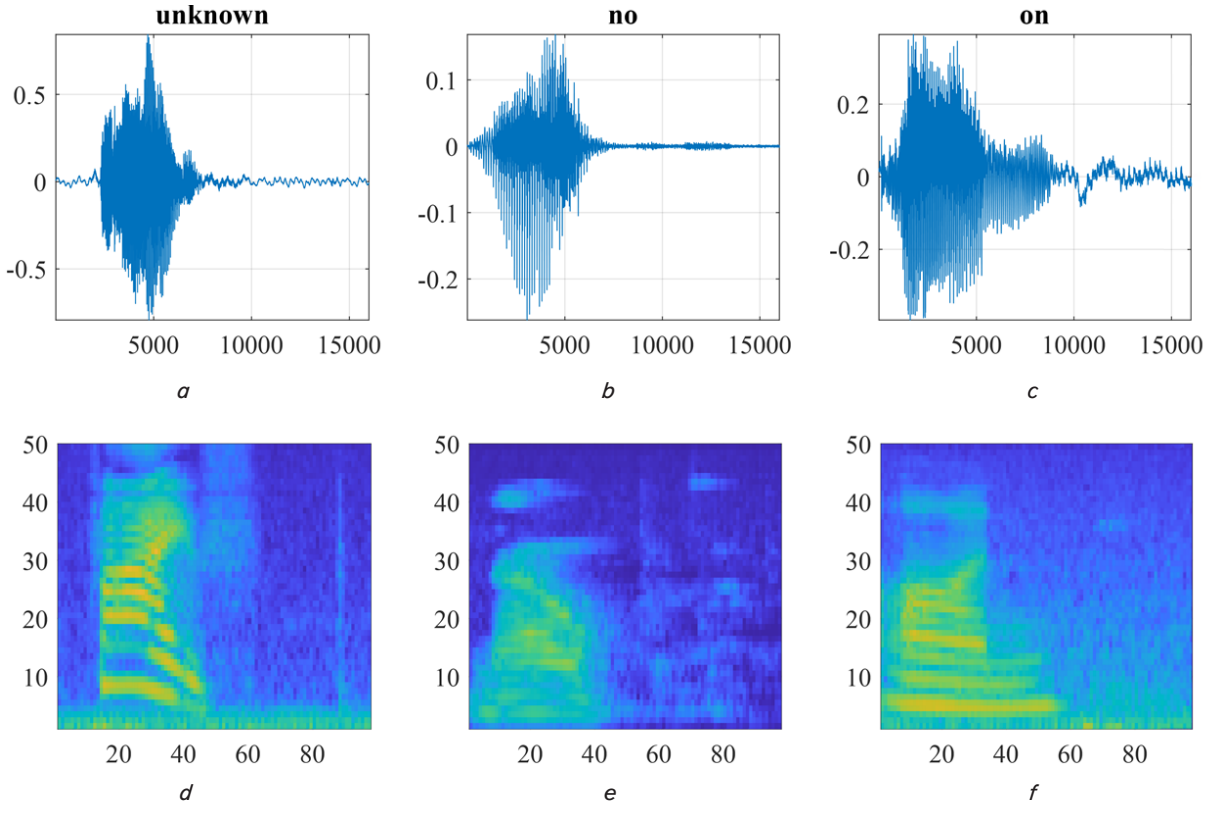

Fig. 8. The signals' waveform with their corresponding auditory-based spectrogram for three training audio signals: *a* – unknown audio; *b* – the command (no) audio; *c* – the command (on) audio; *d* – unknown spectrogram; *e* – the command (no) spectrogram; *f* – the command (on) spectrogram

5. 2. Train the developed a convolutional neural network classifier

The main specifications of the developed CNN architecture are listed in Table 1.

Table 1

Lists the main specifications of the proposed CNN architecture

Description	Dimensions
Number of filters	12
Mini. Batch Size	128
Initial Learning Rate	0.0003
Max. Epochs	15
layers convolution 2d Layer 3	3
batch Normalization Layer	1
ReLu Layer	1
Max. Pooling 2d Layer 3 ,Stride=2	3,2
convolution 2d Layer 3, 2* Number of filters	3,2*12
batch Normalization Layer	1
ReLu Layer	1
max Pooling 2d Layer 3, Stride=2	3,2
convolution 2d Layer 3, 4* Number of filters	3,4*12
batch Normalization Layer	1
relu Layer	1
max Pooling 2d Layer 3, Stride=2	3, 2
convolution 2d Layer 3, 4* Number of filters	3, 4*12
batch Normalization Layer	1
ReLu Layer	1
convolution 2d Layer 3, 4* Number of filters	3, 4*12
batch Normalization Layer	1
ReLu Layer	1
Max. Pooling 2d Layer (time Pool Size 1)	1
dropout Layer	1
fully Connected Layer (12= numClasses)	12
Soft-max Layer	1
classification Layer	1

The pseudocode for the training algorithm is shown in the screenshot of Fig. 9, while the training process showing the accuracy and loss dynamics is shown in Fig. 10, while to validate the classification accuracy, Fig. 11 displays the recall and precision of every class by using row and column summaries over the entire validation dataset.

The above Fig. 10 determines how fast the accuracy of the network is increasing, which interprets the ability of the network to over-fit the data of training. Every iteration represents an estimation for network parameters update and their gradient. Table 2 summarizes the comparison with other related studies in terms of the number of classes, maximum obtained accuracy, and the applied method.

Table 2

Comparison with other relative studies

Reference	No of classes	Max. accuracy	Method
[20]	6	97.15	BCNN
[19]	10	76 %	VGG16
[18]	8	89.54	GoogLeNet
Developed	12	98.1 %	Developed CNN

The created CNN outperforms previous models by a significant margin, reaching an average recognition accuracy of 98.1 %. The created CNN is less likely to overfit than other methods due to its greater performance without the requirement for any batch normalization, regularization, or data augmentation.

```
Network Architecture Definitions
Specifying classes categories; class-Weights; number of Classes; time-Pool-Size; dropout-
Probability; number of Filters(= 12);
Specifying layers = [image Input Layer ([number of Hops, Feature Vector Length]),
  2D-convolution Layer, Batch Normalization Layer, ReLu-Layer, 2D max. Pooling Layer.
  2D convolution Layer, batch Normalization Layer, ReLu-Layer, 2D max. Pooling Layer.
  2D convolution Layer, batch Normalization Layer, ReLu-Layer, 2D max. Pooling Layer.
  2D convolution Layer, batch Normalization Layer, ReLu-Layer, 2D max. Pooling Layer.
  2D convolution Layer, batch Normalization Layer, ReLu-Layer, 2D max. Pooling Layer.
  Dropout Layer, fully-Connected-Layer, softmax-Layer
  Classification-Layer (Classes, Class-Weights);

Specify Training Options
Using Adam optimizer with a mini-batch size of 128.
Specifying training Options ("adam", Initial Learn Rate (=3e-4), Max.Epochs (=15),
Minimum Batch Size (= 128); Validation Data, and Validation Frequency.

Train Network using "trainNetwork" MATLAB command.
Evaluate Trained Network
Showing (Training error, Validation error, and plotting the confusion matrix)
```

Fig. 9. The training algorithm pseudocode including the training evaluation

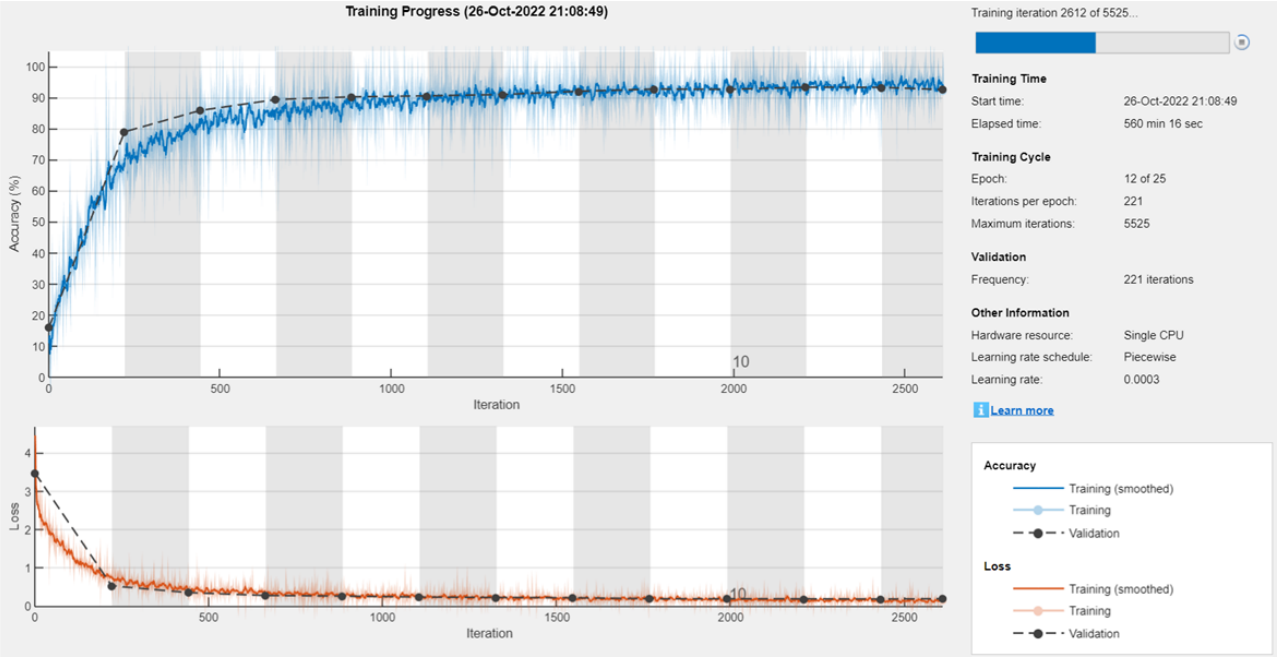

Fig. 10. The training process including accuracy and loss dynamics

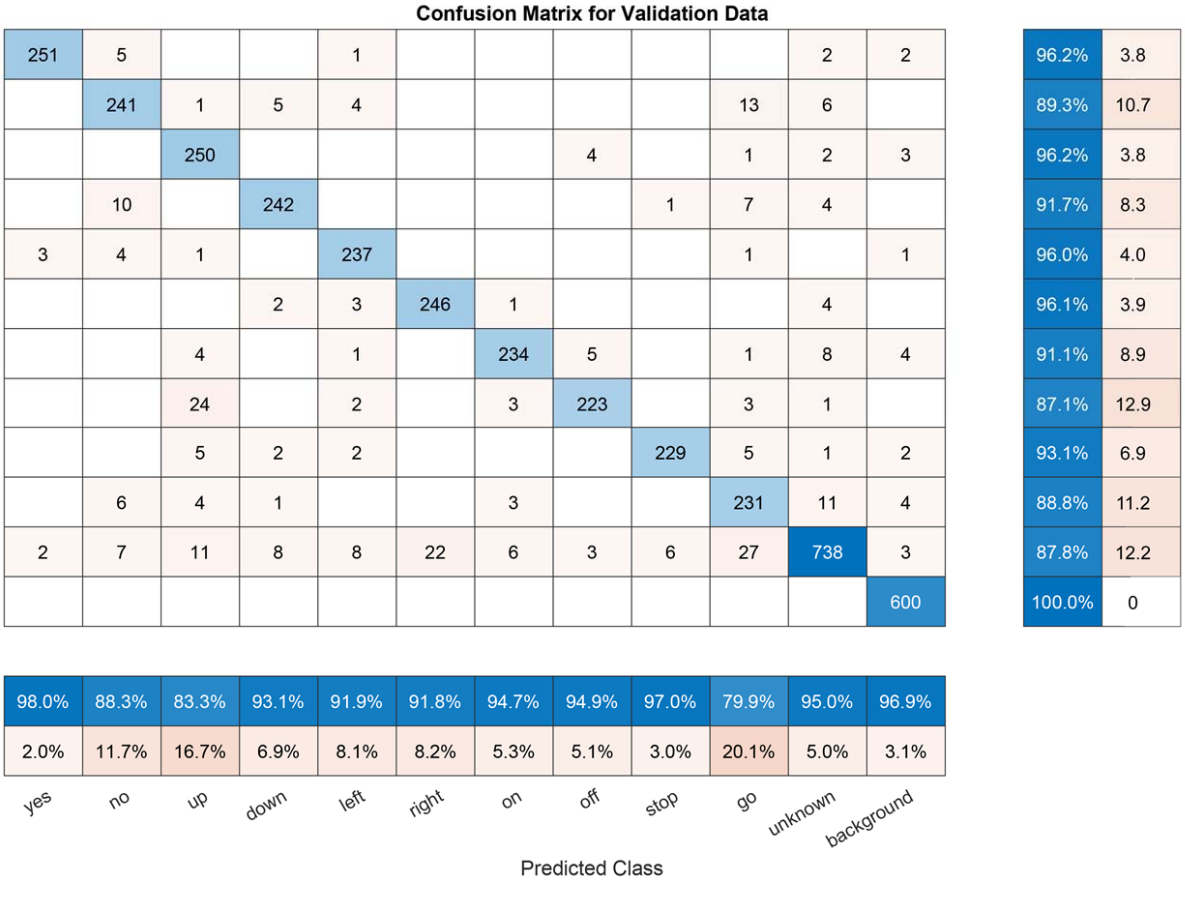

Fig. 11. The confusion matrix over the entire validation dataset

6. Discussions of the results of the developed audio commands classifier

In this work, the developed convolutional neural network with performance optimization was used to categorize twelve classes of speech commands, where a distribution visualization of the validation and training labels was shown in Fig. 7. A series of transform stages to pad the audio signals and feature extraction for auditory spectrograms was performed by defining the audio input parameters such as

the sampling frequency, the auditory spectrogram number of filters, the spectrum time steps, frame duration for spectrum calculations, and the speech clip duration. A visualization of audio inputs showing the signals' waveform with their corresponding auditory-based spectrogram for an ordinary three training audio signals was demonstrated in Fig. 8. We explored expanding the network depth by including identical elements of ReLU, batch normalization, and convolutional layers in an effort to improve the network's accuracy. The training process showing the accuracy and loss dynamics were shown in Fig. 10, while the confusion matrix (Fig. 11) for the validation data shows that the maximum efficiency obtained was 99.7 % on recognizing the background noise from the command words while it was with a range from 83.9 % to 98.4 % for classifying the ten+unknown command words.

In order to evaluate how well CNNs perform on auditory data, multiple benchmark models, including GoogLeNet[18], VGG16 [19], and BCNN [20] are trained in this study. GoogLeNet, which uses less memory and is deeper than VGG, achieves greater accuracy but less than BCNN. Compared to VGG, GoogLeNet has a fundamentally different architecture that makes it possible to train 22 convolutional layers quickly. In contrast, the developed CNN performs substantially better than other models, achieving average recognition accuracy of 98.1 %. Given its higher performance without the need for any batch normalization, regularization, or data augmentation, the developed CNN is less likely to overfit than other algorithms as shown in Table 2.

The network accuracy over the validation and training datasets was very accurate. However, the test, validation, and training data all are with same distributions, which are not essentially reflecting a real-world application. Such limitation mainly affects the unknown classes for only a few numbers of commands. Another limitation is the availability of computational resources and memory, especially when using a mobile. Then it is essential to test network prediction speed by computing its size in kilobytes and the computer's CPU. Therefore the classifying of streaming audio requires computing the single-image prediction time.

7. Conclusions

1. The input audio commands have been prepared for training though converting the audio signals into visualized auditory spectrograms to extract their features. Label distribution shows that twelve from 30 classes' datasets have been selected. Thisstage included performing the steps of feature extraction for the validation dataset, and the playing of their corresponding audio speech.

2. The developed convolutional neural network classifier evaluated the trained architecture. The training progress figure determines how fast the accuracy of the network is increasing to reach about 97.1 %, which interprets the ability of the network to over-fit the data of training, where every iteration represents estimation for network parameters update and their gradient.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

Financing

The study was performed without financial support.

Data availability

Manuscript has associated data in a data repository.

Acknowledgments

The authors would like to express their deepest gratitude to the Al-Nahrain University Baghdad-Iraq for their support to complete this research.

References

1. Suadaa Irfana, M. (2018). The Use Of Voice in Opac Using Google API Voice Recognition & Speech Synthesis and Fullproof Algorithm as Faster Searching Device. International Journal of Engineering & Technology, 7 (3.7), 274. doi: https://doi.org/10.14419/ijet.v7i3.7.16390
2. Ali, M. Y., Naeem, S. B., Bhatti, R. (2020). Artificial intelligence tools and perspectives of university librarians: An overview. Business Information Review, 37 (3), 116–124. doi: https://doi.org/10.1177/0266382120952016
3. Yossy, E. H., Suparta, W., Trisetyarso, A., Abbas, B. S., Kang, C. H. (2019). Measurement of Usability for Speech Recognition on Ok Google. Studies in Computational Intelligence, 83–94. doi: https://doi.org/10.1007/978-3-030-14132-5_7
4. Sprengholz, P., Betsch, C. (2021). Ok Google: Using virtual assistants for data collection in psychological and behavioral research. Behavior Research Methods, 54 (3), 1227–1239. doi: https://doi.org/10.3758/s13428-021-01629-y
5. Choi, T. R., Drumwright, M. E. (2021). "OK, Google, why do I use you?" Motivations, post-consumption evaluations, and perceptions of voice AI assistants. Telematics and Informatics, 62, 101628. doi: https://doi.org/10.1016/j.tele.2021.101628
6. Adaimi, R., Yong, H., Thomaz, E. (2021). Ok Google, What Am I Doing? Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 5 (1), 1–24. doi: https://doi.org/10.1145/3448090
7. Kaur, G., Srivastava, M., Kumar, A. (2018). Integrated Speaker and Speech Recognition for Wheel Chair Movement Using Artificial Intelligence. Informatica, 42 (4). doi: https://doi.org/10.31449/inf.v42i4.2003
8. Vafeiadis, A., Votis, K., Giakoumis, D., Tzovaras, D., Chen, L., Hamzaoui, R. (2020). Audio content analysis for unobtrusive event detection in smart homes. Engineering Applications of Artificial Intelligence, 89, 103226. doi: https://doi.org/10.1016/j.engappai.2019.08.020

9. Moon, J. M., Chun, C. J., Kim, J. H., Kim, H. K., Kim, T. W. (2019). Multi-Channel Audio Source Separation Using Azimuth-Frequency Analysis and Convolutional Neural Network. 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC). doi: https://doi.org/10.1109/icaiic.2019.8668841
10. Jwaid, W. M., Al-Husseini, Z. S. M., Sabry, A. H. (2021). Development of brain tumor segmentation of magnetic resonance imaging (MRI) using U-Net deep learning. Eastern-European Journal of Enterprise Technologies, 4 (9 (112)), 23–31. doi: https://doi.org/10.15587/1729-4061.2021.238957
11. Hamza, A. H., Hussein, S. A., Ismaeel, G. A., Abbas, S. Q., Zahra, M. M. A., Sabry, A. H. (2022). Developing three dimensional localization system using deep learning and pre-trained architectures for IEEE 802.11 Wi-Fi. Eastern-European Journal of Enterprise Technologies, 4 (9 (118)), 41–47. doi: https://doi.org/10.15587/1729-4061.2022.263185
12. Mohite, R. B., Lamba, O. S. (2021). Classifier Comparison for Blind Source Separation. 2021 International Conference on Smart Generation Computing, Communication and Networking (SMART GENCON). doi: https://doi.org/10.1109/smartgencon51891.2021.9645821
13. Yosrita, E., Heryadi, Y., Budiharto, W. (2020). Words classifier of imagined speech based on EEG for patients with disabilities. ICIC Express Letters, 14 (1), 37–41. doi: https://doi.org/10.24507/icicel.14.01.37
14. Nanni, L., Costa, Y. M. G., Aguiar, R. L., Mangolin, R. B., Brahnam, S., Silla, C. N. (2020). Ensemble of convolutional neural networks to improve animal audio classification. EURASIP Journal on Audio, Speech, and Music Processing, 2020 (1). doi: https://doi.org/10.1186/s13636-020-00175-3
15. Xu, G., Wu, Y., Li, M. (2020). The Study of Influence of Sound on Visual ERP-Based Brain Computer Interface. Sensors, 20 (4), 1203. doi: https://doi.org/10.3390/s20041203
16. Sharma, S. (2021). Emotion Recognition from Speech using Artificial Neural Networks and Recurrent Neural Networks. 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence). doi: https://doi.org/10.1109/confluence51648.2021.9377192
17. Speech datasets for conversational AI, ASR and IVR. Available at: https://stagezero.ai/off-the-shelf-speech-datasets/?gclid=CjwKCAiAy_CcBhBeEiwAcoMRHI58ioKouVJ_nLmGJGX9cqOpu6tRUwfmjMQGCHuvBfKN9u4lB6AjhxoC-BYQAvD_BwE
18. Esmaeilpour, M., Cardinal, P., Lameiras Koerich, A. (2020). Unsupervised feature learning for environmental sound classification using Weighted Cycle-Consistent Generative Adversarial Network. Applied Soft Computing, 86, 105912. doi: https://doi.org/10.1016/j.asoc.2019.105912
19. Shashidhar, R., Patilkulkarni, S. (2021). Visual speech recognition for small scale dataset using VGG16 convolution neural network. Multimedia Tools and Applications, 80 (19), 28941–28952. doi: https://doi.org/10.1007/s11042-021-11119-0
20. Sinha, H., Awasthi, V., Ajmera, P. K. (2020). Audio classification using braided convolutional neural networks. IET Signal Processing, 14 (7), 448–454. doi: https://doi.org/10.1049/iet-spr.2019.0381