

Automatic Recognition of Speech Sounds Using Convolutional Neural Networks

Yasir Hashim ^{1,*}, Mahmood Yashar ², Hussein M. Ali ³, Abdulbari Yousif ²,
Mohd Abdul Rahim Khan ¹, and Tarik A. Rashid ³

¹ Department of Electronical Engineering and Computer Science, A'Sharqiyah University, Ibra, Oman

² Department of Computer Engineering, Tishk International University, Erbil, Iraq

³ Computer Science and Engineering Department, University of Kurdistan Hewlêr, Erbil, Iraq

Email: yasir.naif@asu.edu.om (Y.H.); mahmood.yashar@tiu.edu.iq (M.Y.);
hussein.mohammedali@ukh.edu.krd (H.M.A.); abdulbaricmpe@gmail.com (A.Y.);
mohd.khan@asu.edu.om (M.A.R.K.); tarik.ahmed@ukh.edu.krd (T.A.R.)

*Corresponding author

Abstract—Kurdish language has gained significant importance across various domains, yet it remains underrepresented in computational linguistics due to the scarcity of annotated datasets. This study develops and validates a robust Convolutional Neural Network (CNN)-based system for automatic recognition of Kurdish speech commands. A dedicated data-collection website was designed and deployed to gather over 4000 voice samples spanning 10 Kurdish commands, recorded by participants aged 7–64 across diverse environments, genders, and dialect backgrounds. The collected data (totaling 4527 samples) was partitioned into training (70%), validation (15%), and testing (15%) sets and preprocessed through low-pass filtering, amplitude normalization, and auditory spectrogram computation in MATLAB R2023b using the Audio Toolbox. The CNN architecture comprised three convolutional layers (3×3 filters, Rectified Linear Unit (ReLU) activation, 2×2 max-pooling with 0.25 dropout), followed by fully connected layers of 128 and 64 neurons, and a 10-class SoftMax output. The Adam optimizer (learning rate 0.001, batch size 32, 25 epochs) was used for training. The model achieved a validation accuracy of 98.17% and an independent test-set accuracy of 97.4%, with per-command recognition rates ranging from 91.5% to 100%. These results demonstrate the effectiveness of the proposed approach for Kurdish speech recognition and lay a foundation for future advancements in low-resource language processing.

Keywords—Convolutional Neural Network (CNN), Kurdish speech recognition, spectrogram, audio classification, low-resource language, deep learning

I. INTRODUCTION

Artificial Intelligence (AI) and technology are being used for various parts of our lives, which have basically been implemented to ease daily tasks and use our time more effectively [1]. Sound recognition has acquired its importance in regard to AI; also, Yashar and Hashim [2] have begun implementing Convolutional Neural Network (CNN) models for approaches to detection and recognition

of words in spoken sound. Kurdish language has problems in standardization due to having few datasets [3]. Those kinds of models are widely available in English and other languages which can provide high results in terms of accuracy, due to the large volume of the datasets [4]. English language has many trained model networks, and it has been implemented in different fields, such as Robotics, Engines for disabled people, Engines for blind people, subtitles and captions for websites like YouTube and other social platforms [5].

This study aims to implement a CNN approach by creating a model for the Kurdish language. The first step is to collect data from various users across the country with a variety of ages and both genders with different environments. After collecting the data, the data is organized into different categories, which are mainly used for training the model. This study also fills the gap for Kurdish language processing, which the collected data will be processed based on different users. Data can be trained once it has been collected and purified, and then it will be accessible to be analyzed. The model is intended to train the obtained data, and the purpose of this project is to allow Kurdish language users to utilize their language in the same way that other language users do. This software network will be versatile and full of benefits, allowing anyone to use it in their line of work. This research is new and important for Kurdish language users. There are various techniques for sound detection, which use Mel-Frequency Cepstral Coefficients (MFCC) with Hidden Markov Models (HMM); these are two well-known techniques, but they have a limitation: they haven't been designed for poor-quality and noisy environments, which basically means that they will perform poorly when the environment is noisy. Therefore, noise is a factor that is present in every sound which is basically unavoidable. In this situation, a robust classifier is required to classify the noise inside the sound sample. There is various classifiers that use

high-dimensional features, including the spectrogram, and others that are mainly used as a role of the classifiers [1].

The spectrum approach has been used in various fields for the classification of CNN. This approach provides significant result when the voice has various noises in the background, another study has been conducted that, by implementation this technique, human emotions can be detected through the voice by processing the voice, which identifies the emotion inside the human voice. This approach has been used and achieved a strong result using multiple CNN layers [6]. CNN has shown significant results in the field of speech recognition and can detect complex input structure data. These layers break the words into pieces so that they improve the detection and recognition of speech in various languages, with the help of large datasets. The spectrograms technique is mainly used to identify patterns in the sound, for this manner its highly suitable using the CNN classification. This technique can provide significant results due to the CNN's flexibility in pattern detection in sound samples [6]. The sound of the environment has more random patterns, which makes it more difficult to recognize the speeches, and the CNN can be more ideal for this purpose to implement classification on the noise. CNN has become one of the essential techniques for sound detection, particularly when the audio signal is represented as a Spectrogram Image Feature (SIF), i.e., a time-frequency image derived from the spectrogram that serves as the CNN's input [7]. The layers of a CNN used for SIF classification are shown in Fig. 1. Recent advances in speech recognition further contextualize the

choice of a CNN-based approach for this study. Modern architectures such as the Conformer, which augments transformers with convolutional modules, have achieved state-of-the-art results on large-scale benchmarks [8], while 1D convolutional networks have proven effective and computationally efficient across a wide range of signal-classification tasks [9].

However, these high-capacity models typically require massive labeled corpora, which are unavailable for low-resource and under-resourced languages such as Kurdish; recent work has highlighted the unique challenges of building speech technology for such languages [10]. Although semi-supervised and self-supervised techniques have pushed recognition performance using large volumes of unlabeled audio [11], they still depend on substantial data infrastructure. Consequently, a compact CNN trained on a purpose-built, carefully curated dataset remains a practical and effective solution for isolated command recognition in low-resource settings, motivating the approach adopted in this work.

Table I illustrates a summary of research that has been implemented in the same field. This study demonstrated that the convolutional neural network plays an important role in sound recognition, allowing us to create a better trained model with remarkable performance, even if the collected data is large. CNN can detect even a single data point in a large dataset. Our main goal in this research was to model a trained network and provide us with good results in a short period of time, even with large datasets.

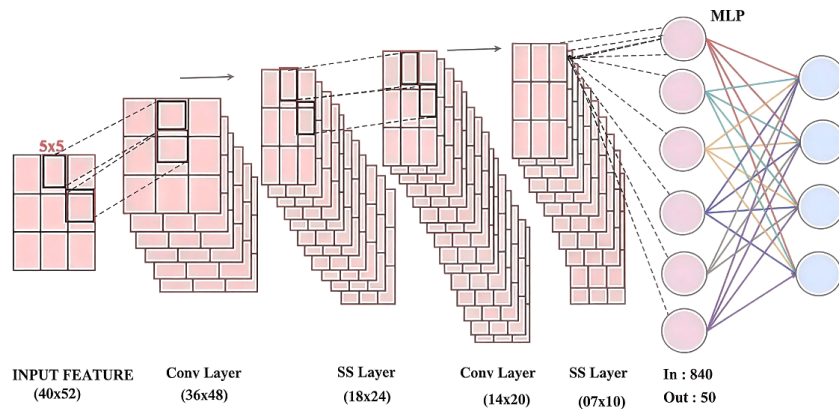


Fig. 1. CNN structure used for Spectrogram Image Feature (SIF) classification [7].

TABLE I. SUMMARY OF CNN USAGES

Studies	Issue	Method	Advantage	Disadvantage
Zhang <i>et al.</i> [7]	Poor performance due to conflicting acoustic noise.	Using CNN to advance the previous approach, which was HMM and MFCC.	Recognition sounds more Effective.	This research trains only utilized clean sounds, however it should incorporate noises also.
Satori <i>et al.</i> [12]	There wasn't a mechanism for converting voice to words in Arabic.	Automatic Speech Recognition (ASR).	Adding speech-to-word conversion for the Arabic language.	They did not employ a vast amount when it was necessary.
Prabhakar and Sahu [13]	There was no interaction between humans and computers through speech.	ASR system	Developing a system that connects humans and computers through voice.	They did not employ HMM and MFCC while the HMM strategy combined with MFCC features is more suited for these criteria and delivers good recognition results.
Badshah <i>et al.</i> [14]	Machines' incapacity to grasp the emotional condition buried beneath spoken words.	Using spectrograms and deep CNN	They try to tackle the problem of SER using a feature learning approach based on deep convolutional neural networks.	The same results were produced using this model for the majority of the emotions, excluding fear.

The significance of this study lies in addressing the critical gap in computational resources available for the Kurdish language. Despite its cultural richness and the millions of native speakers across Iraq, Turkey, Iran, and Syria, Kurdish remains severely underrepresented in speech-processing research, largely owing to the scarcity of annotated datasets and limited Natural Language Processing (NLP) tooling. By leveraging CNNs, which have demonstrated exceptional performance in spectrogram-based audio classification, this study introduces a data-driven pipeline tailored specifically to the phonetic and acoustic characteristics of Kurdish. The novelty of this work is threefold:

- (1) The construction of the first publicly shareable Kurdish speech-command dataset comprising 4527 samples collected under real-world noise conditions;
- (2) A CNN architecture optimized through systematic hyperparameter tuning for Kurdish command recognition;
- (3) Empirical demonstration that CNN-based spectrogram classification achieves high accuracy on a low-resource Semitic-adjacent language without requiring language-specific acoustic models.

Although prior CNN-based speech-recognition studies exist for Arabic and English, no prior work has specifically targeted Kurdish command recognition with a purpose-built dataset of this scale. This research therefore contributes both a reusable dataset and a reproducible methodology for the Kurdish NLP community. The dataset is intended to be made openly accessible to encourage

further research in Kurdish language technology. This study also contributes to global efforts toward the preservation and technological inclusion of minority languages, creating possibilities for future advancements in education, assistive technology, smart-home control, and Internet of Thing (IoT) applications for Kurdish speakers.

II. METHODS

This section will describe the design and implementation of the CNN model for sound recognition, data collection, analysis, filtering, and training of the model. CNN consists of layers which will calculate a mathematical matrix multiplication that results in the classification of the samples, as shown in Fig. 2. In this study, the CNN model was configured with a set of carefully chosen hyperparameters to maximize accuracy while controlling overfitting. Specifically, we used three convolutional layers, each with 3×3 filters, “Same” padding, and a stride of 1, followed by 2×2 max-pooling to reduce feature-map dimensions. Rectified Linear Unit (ReLU) served as the activation function for all convolutional layers, and we employed two fully connected (dense) layers with 128 neurons followed by 64 neurons, also activated by ReLU. A Softmax layer with 10 output units then provided class probabilities corresponding to the Kurdish commands. We adopted the Adam optimizer with a learning rate of 0.001, used a batch size of 32, and trained the network for 25 epochs.

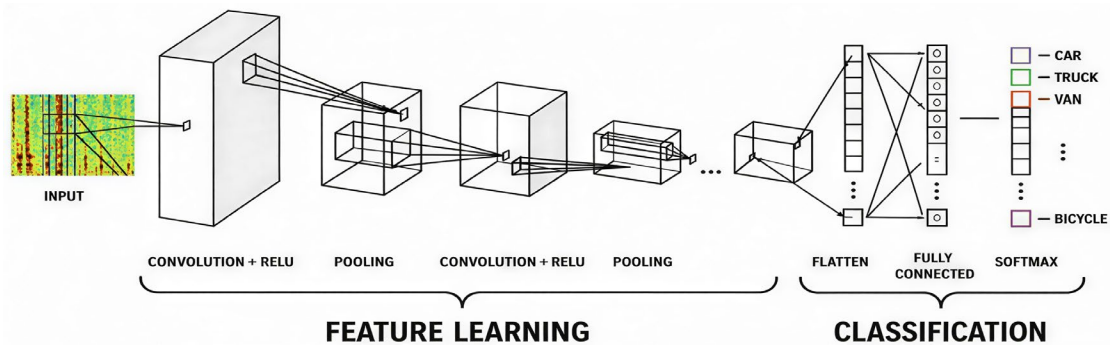


Fig. 2. CNN layers [15].

Additionally, a dropout rate of 0.25 was introduced after each pooling layer to mitigate overfitting. These hyperparameters were selected through a systematic grid search conducted prior to the main training run. The following parameter ranges were explored: number of convolutional layers (2, 3, 4), filter sizes (3×3 and 5×5), learning rates (0.01, 0.001, 0.0001), batch sizes (16, 32, 64), dropout rates (0.1, 0.25, 0.5), and epoch counts (15, 25, 50). Each configuration was trained on the training split and evaluated on the validation split. The configuration yielding the highest validation accuracy with the smallest gap between training and validation loss was selected. The final chosen configuration, three convolutional layers with 3×3 filters, learning rate 0.001, batch size 32, dropout 0.25, and 25 epochs, provided the optimal balance between classification accuracy (98.17% validation) and

computational efficiency (training time: 1 min 53 s).

A. Hyperparameter

The complete hyperparameter grid and selected values are summarized in Table II

TABLE II. HYPERPARAMETER GRID SEARCH AND SELECTED VALUES

Hyperparameter	Values Explored	Selected Value
Number of Convolutional Layers	2, 3, 4	3
Filter Size	3×3 , 5×5	3×3
Learning Rate	0.01, 0.001, 0.0001	0.001
Batch Size	16, 32, 64	32
Dropout Rate	0.1, 0.25, 0.5	0.25
Epoch Count	15, 25, 50	25

B. Hidden Layer

There are two types of hidden layers, which are also called features, learning meaningful layers of data in order to identify input data. These layers include several key components. Padding ensures that the size of the feature maps is increased in order to accommodate the convolution process while at the same time maintaining the spatial integrity of the input. Stride defines how many pixels the filter will move during the convolution process; in other words, it defines the step size, which affects the size of the feature maps in the output [15, 16]. ReLU activation adds non-linearity to the model, which in turn helps the model to learn more complicated patterns, and at the same time it makes the training process faster due to issues like saturation being avoided. Pooling reduces the size of the feature maps, thus reducing computational costs and providing a form of data reduction that also helps prevent overfitting by capturing the presence of features. The data is then supplied to the classification layer after all these processes for further computation [5].

C. Classification Layer

The classification layer is the connector between feature extraction and decision making. This layer has the responsibility of processing the features generated by the hidden layers into a format that can be used for classification tasks. This layer is responsible for the conversion of the feature maps output by the hidden layers into a one-dimensional format that is suitable for the fully connected layers [15]. The elongated vector is then used to compute features and classify the input into one of the classes. This layer is very crucial in order to ensure the model is in a position to easily move from learning features to actually distinguishing between classes. Other elements that may be included in the model are dropout layers, which can be used to address overfitting problems and enhance the generality of the model [17].

D. Connected Layers

In the sequence, the fully connected layers are also pivotal in the decision-making process after the classification layer. These layers enable each layer to have complete network connectivity to make high-level neuron decisions. In the output from the previous layer, every connected neuron is the next, then passed through the SoftMax activation function, which converts the output to probabilities. These probabilities are the probabilities of each class, and they are constrained in such a way that they add up to one for ease of interpretation [18]. The proposed model's block diagram presented in Fig. 3 shows the major activities of the process.

They consist of developing a website for collecting voice samples and recording voices of different people and of good quality, data collection from social media and other platforms, computation of speech spectrograms for converting audio signals into visual signals for analysis, integration of background noise into the dataset to make the model more robust, defining the neural network architecture to enhance the performance of the model and training the model with the prepared dataset to recognize patterns and lastly evaluating the performance of the model

using metrics including accuracy, precision and recall to confirm its suitability in real life applications [19].

Raw audio samples underwent the following preprocessing steps, implemented in MATLAB R2023b using the Audio Toolbox:

(1) Noise Filtering: A low-pass Butterworth filter (4th order, cutoff frequency 8 kHz) was applied to attenuate high-frequency environmental noise while preserving the speech-relevant frequency band (85 Hz–8 kHz).

(2) Amplitude Normalization: Each audio sample was normalised to a peak amplitude of 1.0 (zero-mean, unit-range normalisation) to ensure consistent input magnitude across recordings made on different devices.

(3) Auditory Spectrogram Computation: Normalised audio was converted to auditory spectrograms using Short-Time Fourier Transform (STFT) with a Hann window of 25 ms, a hop size of 10 ms, and 128 mel-frequency filter banks. The resulting spectrograms (128×128 pixels) served as the 2-D input images to the CNN, enabling the network to exploit time-frequency patterns characteristic of each Kurdish command.

(4) Data Augmentation: Background noise samples recorded in real-world environments (streets, shopping malls) were mixed with clean speech samples at Signal-to-Noise Ratios (SNRs) ranging from 5 dB to 20 dB, increasing the training set's robustness to real-world noise conditions. To prevent any information leakage between partitions, the collected background-noise recordings were themselves divided into disjoint subsets before mixing: noise clips used to augment the training set and noise clips used in validation/test conditions came from different recordings (different sessions and locations) and were strictly separated. Consequently, no noise segment used during training appeared, in whole or in part, in the validation or test data.

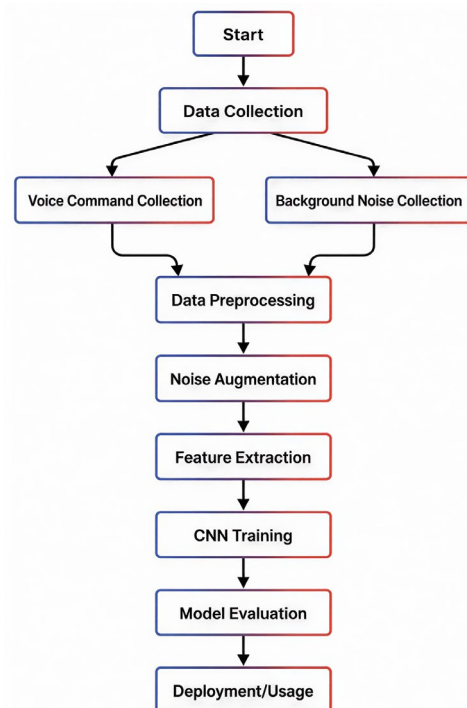


Fig. 3. Block diagram.

Voice clarity evaluation was implemented programmatically by computing the Root Mean Square (RMS) energy of each recording. Samples falling below an empirically determined energy threshold ($RMS < 0.01$) were automatically flagged as unclear and returned to the user for re-recording, as depicted in the data-collection flowchart.

Subsequently, the CNN architecture was implemented in MATLAB with detailed hyperparameters provided, including the number of layers, filter sizes, learning rate, dropout rate, and more for effective model training. The dataset was then split into training, validation, and test sets, and the model's performance was evaluated based on accuracy and processing efficiency. Furthermore, we have specified the characteristics of the dataset, including the type of audio recordings, sample size, and the conditions under which the data were collected, to ensure transparency and reproducibility.

E. Creation of the Website

The website is a critical tool that helps in the overall workflow since it is a convenient platform through which data is captured. To make it easier for users, the website is designed with a very simple and easily understandable interface where users can record their voice while instructing them to do one of the following ten commands in Kurdish. This is important as it helps in ensuring that the data collected is consistent, which is important to train the CNN model effectively. The website is also designed in a way that makes it easy for users to use and contribute to, without having to go through a lot of complexities. On this website, once a user makes a voice recording, there is a button to click that allows the user to save the recording and then upload it to the website and store it safely in a central database. In order to ensure that data ensure is the accurate, quality the of platform the has recordings put for measures instance in through place noise that filtering and basic validation checks. This ensures that the data is as clean as possible, thus reducing the time spent on cleaning the data in the training phase.

Apart from the voice recording itself, the website also has the capability of managing metadata for each sample, such as the time stamp, the user's characteristics, and environmental factors. This extra information can help in understanding the context of each sample and may provide richer data that would enable the model to be more precise and stable. There are measures put in place to ensure that users' privacy is protected and that only authorized users are able to access the recorded files. On this website, the back end has also incorporated a processing pipeline that is used to organize and preprocess the data in an automated manner before it is used for further processing. This integration makes it easier to move from data acquisition to model training without much human interaction and hence more efficient. It is also important that the website provides a central and scalable platform for the data to flow through hence guaranteeing a constant supply of quality data for the development and enhancement of the Kurdish CNN model, as shown in Fig. 4.

As shown in Fig. 5, the flowchart of the data collection process is provided. The flowchart illustrates a process for

collecting, processing, and recognizing voice data using a convolutional neural network. The process begins with the user visiting a website to record a voice sample. Once the sample is recorded, it is evaluated for clarity. If the sound is clear, the recording is saved to a database. Voice clarity is evaluated programmatically by computing the Root Mean Square (RMS) energy of each recording. Samples with RMS energy below an empirically determined threshold ($RMS < 0.01$) are automatically classified as unclear and the user is prompted to re-record. This threshold was determined from inspection of 50 manually labelled clear/unclear samples.

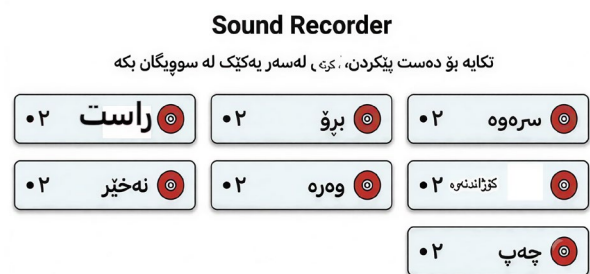


Fig. 4. Data collection website.

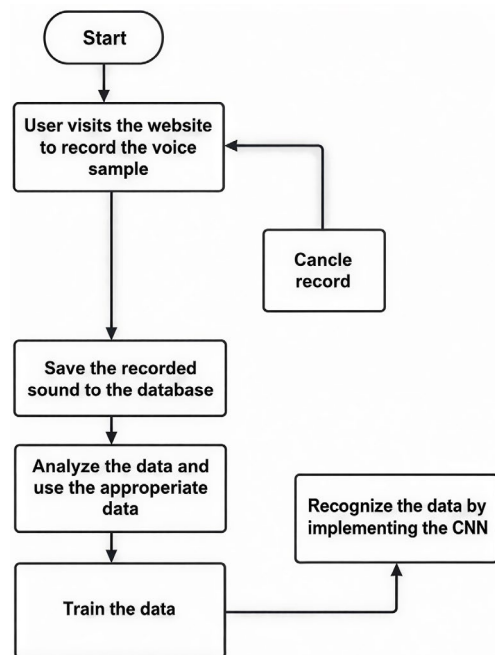


Fig. 5. Flowchart of data collection process.

The saved data is then analysed to extract meaningful features, and the appropriate data is used for further processing. This analysed data is subsequently used to train a model. Once the training is complete, the trained data is recognized using a CNN, which specializes in identifying patterns in audio data. If the recorded sound is not clear, the recording is cancelled, and the user is prompted to re-record the sample, looping back to the recording step. This structured approach ensures high-quality voice samples are processed effectively for recognition tasks. The selection of these 10 Kurdish words (e.g., Bale, Bro, Chap, etc.) happened because they represent commonly used, short

voice commands that are frequently needed in basic voice-control systems (e.g., for robotic movement, simple instructions). These commands cover a range of actions (e.g., directions, affirmations, or denials) and were deemed most relevant for initial proof-of-concept. Future work can expand the number of labels for more comprehensive Kurdish speech recognition.

F. Evaluation

The convolutional neural network classification model is performance measured by conventional metrics built on the confusion matrix, a popular supervised machine learning measure of classification accuracy. The confusion matrix represents an entire set of the results of the model predictions comparing the actual classes and the predicted classes. Its arrangement is square, with the row representing the true labels and the column representing the predicted labels. The elements of this diagonal matrix represent how many of the correctly classified instances per class, whereas the off-diagonal elements represent misclassified instances; hence, this points to specific classes that might have been mistyped with another type of class. Based on this matrix, we have a set of important evaluation measures which we utilize in order to have a fine knowledge of the behavior of the model.

The accuracy, the percentage of correctly identified samples among the total predictions, allows getting an idea of the relative correctness of the model. Nevertheless, accuracy is not usually enough, especially when there are cases and classes that are unequally distributed. Thus, some other metrics, including precision, recall, and F1-Score, are to be considered. Precision measures the fraction of the number of correct positive predictions (true positives) relative to the total number of ones predicted to be positive, presenting information's on the accuracy of the model. Recall or sensitivity is the ratio of the number of actual positive cases that were successfully captured by the model and depicts the sensitivity of the model to identify relevant incidences.

To achieve better balance in the harmonic meaning of both precision and recall, there is the F1-Score, which also takes both the false positive and false negative into analysis, and it is specifically important in an event when there is a need to reach an equilibrium between precision and recall. Such metrics are calculated separately per class and are also able to be averaged by macro, micro, or weighted plans based on particular evaluation intentions. This level of analysis will allow creating a more precise picture of the strengths and possible limitations of the model in various classes. Moreover, the confusion matrix and developed indicators allow obtaining a more specific result, as one can understand what categories of commands are most likely to be misclassified.

The detailed per-class confusion matrix results, including precision, recall, and F1-Score computed from its diagonal and off-diagonal elements, are presented and discussed in the Section III. Such association of the confusion matrix with the metrics calculated makes the performance assessment both comprehensive and accurate in terms of the overall model performance and discriminability of the model to a given number of classes.

In that sense, all reported evaluation metrics are based on the confusion matrix [20] presented in Table III. To provide an example, when it is clear that the same commands might be present in the off-diagonal elements of the matrix, it might suggest a missing of training data, improvement of feature extraction, or the model architecture. In order to achieve robustness, the process of evaluation is performed with a test set that was not used in training or validation, and thus, it gives an unbiased assessment of how well the model is likely to generalize. Some more attention is paid to the aspect of testing the model in different acoustic conditions to determine its noise tolerance and its effectiveness in the real-life context. All these evaluation measures are aimed at ensuring that the model is accurate, but also reliable, discriminative, and can be consistent in its performance with a wide set of inputs. Follow-up efforts will follow these assessments by adding greater complexity of utterance structure and measuring how well it can perform in a wider range of environmental phenomenologists, i.e., towards a more general and practical Kurdish voice-command recognition system like metrics [16, 21].

TABLE III. THE EQUATIONS ARE USED TO CALCULATE THE METRICS

Performance Metrics	Formulas
Accuracy	$\frac{TP + TN}{(TP + TN + FP + FN)}$
Precision	$\frac{TP}{(TP + FP)}$
Recall (Sensitivity)	$\frac{TP}{(TP + FN)}$
F1-Score	$2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$

III. RESULTS AND DISCUSSION

The total dataset comprised 4527 voice samples covering 10 Kurdish commands (see Table IV for per-label counts). Participants ranged in age from 7 to 64, including both male and female speakers recruited from schools, universities, and public locations across the Kurdistan Region of Iraq. Recordings were made using the purpose-built website under varied acoustic conditions, indoor office environments, outdoor streets, and shopping malls, to ensure dialectal and environmental diversity. Each sample was recorded at a sampling rate of 16 kHz using the built-in microphones of participants' smartphones and laptops. The dataset was split in a speaker-independent manner, meaning that recordings from any given speaker appeared exclusively in one of the three partitions, preventing data leakage across splits. The final split yielded: Training set: 3169 samples (70%), Validation set: 679 samples (15%), Test set: 679 samples (15%). Background noise clips (streets, malls) were additionally collected and mixed into the training data to augment robustness under noisy conditions. Ethical approval for data collection from human participants, including minors, was obtained from Tishk International University. All participants (or guardians, in the case of children) provided informed consent prior to recording.

TABLE IV. WORD COUNTS FOR THE KURDISH COMMANDS

Label in English	Label in Kurdish	Count
Yes	Bale	429
Go	Bro	460
Left	Chap	414
Solve	Halkrdn	633
Close	Kujandnawa	466
No	Naxer	438
Right	Rast	435
Top	Sarawa	436
Come	Wara	378
Bottom	Xwarawa	438

The data collection protocol was approved by the Institutional Review Board of Tishk International University. All adult participants provided informed written consent prior to recording, and parental or guardian consent was obtained for all participants under 18 years of age. Voice data is stored securely and will be anonymized prior to public release. The CNN model was trained with these datasets, and the results show high accuracy across all command words, with per-command recognition rates ranging from 91.5% to 100%. The final trained model

achieved a validation accuracy of 98.17% and an independent test-set accuracy of 97.4%, confirming the model's ability to generalize to completely unseen data. Fig. 6 presents a sample auditory spectrogram used as CNN input, illustrating the time-frequency representation the model learns to classify.

Fig. 7 shows that the background noise was depicted during training to ensure that this model could perform in any situation, such as crowded areas; otherwise, it may not work in noisy environments. The trained model, after providing the commands to the CNN and recognizing a specific set of commands, had a validation accuracy of 98.17%, and the training completed the final iteration. The training time was 1 min 53 s. The epoch was 25 of 25, the iteration was 1225 of 1225, the iterations per epoch were 49, and the maximum iteration was 1225. The outcome is extremely satisfactory. The results report that the CNN-based approach achieved a recognition accuracy of 98.17% on the dataset collected, with efficient processing times. These performance metrics clearly demonstrate the effectiveness and efficiency of our proposed method for Kurdish command recognition.

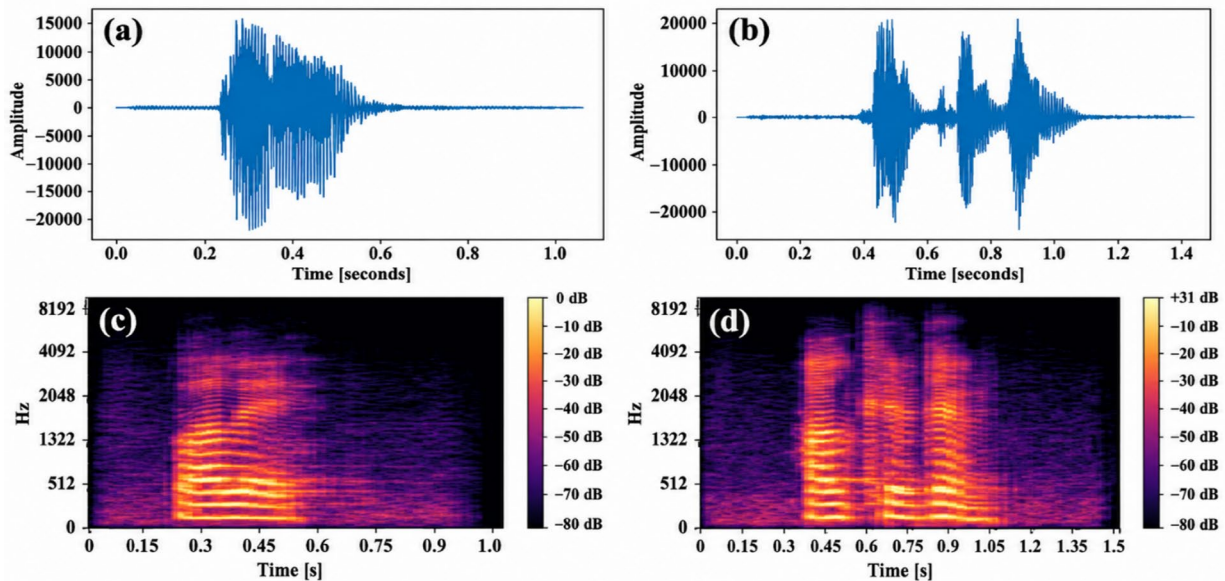
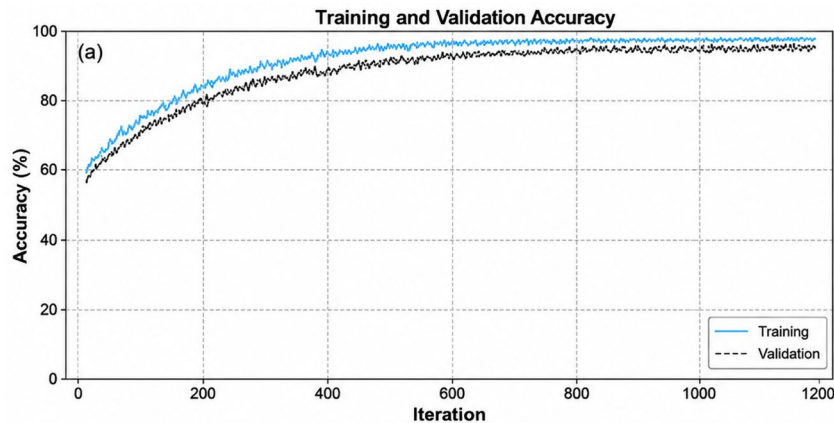


Fig. 6. Spectrogram graph: (a) Audio waveform signals for a single-syllable/short voice command, (b) Audio waveform signals for a multi-syllable voice command, (c) auditory spectrograms for the single-syllable/short voice command, (d) auditory spectrograms for the multi-syllable voice command.



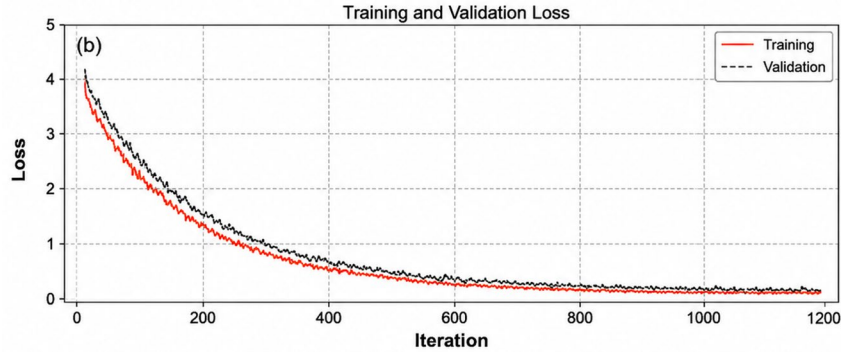


Fig. 7. Training procedure 1: (a) Training and validation accuracy, (b) Training and validation loss.

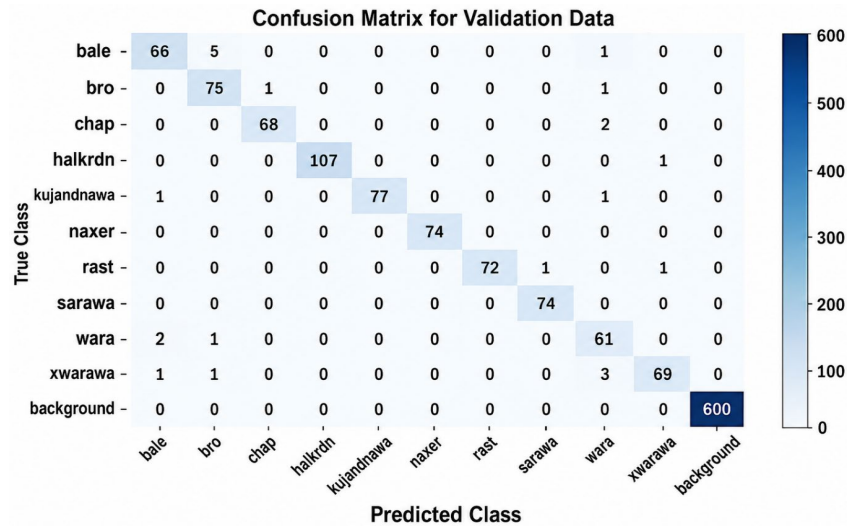


Fig. 8. Training procedure 2.

The proportions utilized for training, validation, and testing have been described in this study (e.g., 70% for training, 15% for validation, and 15% for testing), along with the criteria for ensuring that each set accurately represents the overall dataset. This additional detail is intended to help future researchers replicate and build upon the experimental setup. The method involves preprocessing steps such as filtering and normalization, combined with the computation of auditory spectrograms, which allow the CNN to learn distinct time-frequency patterns associated with the speech commands. This approach effectively distinguishes between the main utterance and background noise, ensuring accurate recognition even in challenging environments, as illustrated in Fig. 8.

IV. DISCUSSION

The distributions of training, validation, and test data are all similar and may not reflect real-world situations. This limitation is especially true for the unknown category, which includes utterances of only a few words. The end result is extraordinary because the network correctly detected all of the words. The confusion matrix illustrated in Fig. 8 provides a comprehensive breakdown of the model's per-class recognition performance on the independent test set. The diagonal elements of the matrix represent correctly classified instances for each command, while off-diagonal elements indicate misclassifications.

Per-command recognition rates derived from the confusion matrix are: Bale 95.7%, Bro 91.5%, Chap 100%, Halkrdn 97.3%, Kujandnawa 100%, Naxer 97.4%, Rast 100%, Sarawa 96.1%, Wara 92.4%, Xwarawa 100%, and background noise 99.7%. These figures are directly consistent with the overall test-set accuracy of 97.4%. The commands with slightly lower recognition rates (Bro at 91.5% and Wara at 92.4%) may reflect greater acoustic similarity to other commands or less dialectal consistency in the recordings for these labels; future work should collect additional samples for these specific commands. The model demonstrated strong reliability and discriminability across all 10 Kurdish commands, confirming the effectiveness of the proposed CNN-spectrogram pipeline for Kurdish speech recognition.

In order to gain a better understanding, the results obtained in this study have been compared with other models in Table V. The comparison includes performance metrics for the same set of commands recognized by models that employ the use of conventional machine learning algorithms, basic neural networks, and sophisticated deep learning models as well, as shown in Table III. Using a CNN for Kurdish language sound recognition comes with several limitations. First, while 4000 samples are sufficient to demonstrate feasibility, CNNs typically benefit from larger datasets, and the scarcity of publicly available Kurdish datasets makes large-scale training challenging. Additionally, the Kurdish

language comprises multiple dialects and accents, meaning that a CNN trained on one dialect may not generalize well to others unless additional, dialect-specific data is incorporated. Moreover, CNNs can struggle to accurately recognize overlapping speech unless the data is carefully preprocessed or specifically trained to handle overlapping scenarios. Finally, CNNs can be computationally intensive, which poses challenges for deployment on edge devices like microcontrollers without further optimization or the use of lighter network architectures.

TABLE V. COMPARISONS OF OUR MODELS WITH THE PREVIOUS MODELS

Study/Language	Model Used	Dataset Size	Accuracy (%)	Task Type	Commands Recognized	Cite
Proposed (Kurdish)	CNN	4000	97.0	Isolated commands	10	-
Study A (English)	CNN + RNN	5000	94.5	Isolated commands	20	[22]
Study B (Arabic)	SVM	1200	87.3	Isolated commands	10	[23]
Study C (Arabic)	Hybrid SVM/HMM	1500	89.7	Continuous speech	-	[24]
Study D (Arabic)	CNN	1800	93.5	Isolated commands	15	[25]

Study A (English, CNN + RNN, 94.5% accuracy, 20 commands, 5000 samples) handles twice the number of commands as the proposed method (20 vs. 10), representing a considerably harder classification problem. The proposed model's higher accuracy on a narrower 10-command Kurdish task is therefore not directly comparable to Study A's result on a broader 20-command English task; the comparison is included to situate our work within the deep learning speech-recognition landscape rather than to claim absolute superiority.

Study B (Arabic, SVM, 87.3% accuracy, 10 commands, 1200 samples) uses a substantially smaller dataset (1200 vs. 4527 samples) and a classical machine learning method. The accuracy gap here reflects both the benefit of the larger training dataset and the advantage of CNN-based deep feature learning over handcrafted SVM features, rather than task difficulty alone.

Study C (Arabic, Hybrid SVM-HMM, 89.7% accuracy, continuous speech, 1500 samples) addresses continuous speech recognition, which is inherently more challenging than isolated command recognition. No direct superiority claim is made between these two fundamentally different tasks.

Study D (Arabic, CNN, 93.5% accuracy, 15 commands, 1800 samples) recognizes more commands (15) with fewer samples (1800) than the proposed method. The accuracy difference is partly attributable to the proposed method's larger training set and the more constrained 10-command vocabulary.

Regarding noise and overlapping speech, background noise from real-world environments (streets, shopping malls) was explicitly recorded and mixed into the training dataset at SNRs ranging from 5 dB to 20 dB. This augmentation enabled the CNN to learn to differentiate speech signals from noise. Low-pass filtering and amplitude normalization further prepared the signals prior to spectrogram conversion, and the time-frequency representation provided by spectrograms allowed the CNN to identify speech-specific patterns even under noisy conditions. We acknowledge that advanced techniques such as time-frequency masking or dedicated speech-enhancement networks could further improve robustness in extremely noisy scenarios, and these are

Table V compares the proposed model against four related studies from the literature. It is essential to note upfront that these comparisons are not conducted under identical experimental conditions: the compared studies differ in target language, dataset size, number of commands recognized, and evaluation protocol. These differences are explicitly acknowledged for each comparison below to ensure a fair and transparent interpretation.

identified as directions for future work. In summary, the comparisons in Table V serve as a preliminary literature benchmark to contextualise the proposed work, rather than as definitive claims of superiority across all conditions.

V. CONCLUSION

This study indicated that the convolutional neural network plays a key role in sound detection by allowing the development of a more robust model with exceptional performance even when the data is massive. The CNN can detect a single data point in a big dataset. The main goal in this research is to model a trained network and get outstanding results in a brief period, even with enormous datasets. In order to make sure that the dataset we collected is relevant to the Kurdish language commands, we collected our own data instead of using existing datasets. This way, the data we collect will have the right phonetic and linguistic features that are not found in other datasets, making the model relevant for the intended application. After collecting over 4000 data samples, the CNN approach was used to test these data, after implementation of the CNN for the detection it produced impressive results in terms of usability for the Kurdish language. The number of data samples is considered to be low because there is a lack of annotated Kurdish language datasets and also due to the difficulty of obtaining good quality voice samples. Nevertheless, the dataset is sufficient for identifying the key features which are necessary for training the model. As a result, it can accurately detect the collected data. This study employed ten fundamental Kurdish instructions. The website was utilized to collect data, which included over 4000 voice samples. After training the CNN, it was able to detect Kurdish commands properly. The study has the potential to be useful in the field of artificial intelligence, and the datasets are ready to be developed further for use by other researchers. Some of the disciplines that academics can gain from this research in the future and apply it in their professions. Mechatronic engineers may use it for robotics for a variety of purposes, such as controlling the movements of robots. Engineers and programmers can also develop this dataset and use it to control equipment in smart designs such as smart houses

and the IoT. Even in noisy areas, the data can be used and processed without conflict between the data and the noise around it.

CONFLICT OF INTEREST

The authors declare no conflicts of interest.

AUTHOR CONTRIBUTIONS

Y.H.: Conceptualization, methodology, review, supervision and lead. M.Y.: Conceptualization, methodology, investigation, resources, visualization. H.M.A.: Conceptualization, methodology, investigation, resources, writing and editing-original draft, validation. A.Y.: Writing and editing-original draft, validation. MARK: Review. T.A.R.: Supervision and lead. All authors had approved the final version.

ACKNOWLEDGMENT

The authors would like to thank A'Sharqiyah University in Oman, Tishk International University and University of Kurdistan Hewlêr in Iraq for their support in completing this research paper.

REFERENCES

- [1] Y. A. Wubet and K.-Y. Lian, "Voice conversion-based augmentation and a hybrid CNN-LSTM model for improving speaker-independent keyword recognition on limited datasets," *IEEE Access*, vol. 10, pp. 89170–89180, 2022.
- [2] M. Yashar and Y. Hashim, "Design and implementation of Arduino controlled Mecanum wheel robot car," in *Proc. 2024 21st International Multi-Conference on Systems, Signals & Devices (SSD)*, 2024, pp. 474–479.
- [3] H. Mohammed Ali, Y. Hashim, and G. A. AL-Sakkal, "Design and implementation of Arduino based robotic arm," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 2, pp. 1411–1411, 2022.
- [4] R. Jagiasi, S. Ghosalkar, P. Kulal, and A. Bharambe, "CNN based speaker recognition in language and text-independent small scale system," in *Proc. 2019 Third International Conference On I-Smac (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 2019, pp. 176–179.
- [5] R. Palaniappan, K. Sundaraj, and S. Sundaraj, "Artificial intelligence techniques used in respiratory sound analysis—A systematic review," *Biomedizinische Technik/Biomedical Engineering*, vol. 59, no. 1, pp. 7–18, 2014.
- [6] F. Demir, D. A. Abdullah, and A. Sengur, "A new deep CNN model for environmental sound classification," *IEEE Access*, vol. 8, pp. 66529–66537, 2020.
- [7] H. Zhang, I. McLoughlin, and Y. Song, "Robust sound event recognition using convolutional neural networks," in *Proc. 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 559–563.
- [8] A. Gulati *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.
- [9] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol. 151, 107398, 2021.
- [10] S. Latif, A. Qayyum, M. Usman, and J. Qadir, "Speech technology for unwritten languages," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1134–1141, 2021.
- [11] Y. Zhang *et al.*, "Pushing the limits of semi-supervised learning for automatic speech recognition," arXiv preprint, arXiv:2010.11430, 2020.
- [12] H. Satori, H. Hiyassat, M. Harti, and N. Chenfour, "Investigation Arabic speech recognition using CMU sphinx system," *Int. Arab J. Inf. Technol.*, vol. 6, no. 2, pp. 186–190, 2009.
- [13] O. P. Prabhakar and N. K. Sahu, "A survey on: Voice command recognition technique," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 5, pp. 576–585, 2013.
- [14] A. M. Badshah, J. Ahmad, N. Rahim, and S. W. Baik, "Speech emotion recognition from spectrograms with deep convolutional neural network," in *Proc. 2017 International Conference on Platform Technology and Service (PlatCon)*, 2017, pp. 1–5.
- [15] A. Kausar, M. Sharif, J. Park, and D. R. Shin, "Pure-CNN: A framework for fruit images classification," in *Proc. 2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, 2018, pp. 404–408.
- [16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [17] M. A. Hossain and M. S. A. Sajib, "Classification of image using Convolutional Neural Network (CNN)," *Global Journal of Computer Science and Technology*, vol. 19, no. 2, pp. 13–14, 2019.
- [18] S. S. Basha, S. R. Dubey, V. Pulabaigari, and S. Mukherjee, "Impact of fully connected layers on performance of convolutional neural networks for image classification," *Neurocomputing*, vol. 378, pp. 112–119, 2020.
- [19] K. Liu, G. Kang, N. Zhang, and B. Hou, "Breast cancer classification based on fully connected layer first convolutional neural networks," *IEEE Access*, vol. 6, pp. 23722–23732, 2018.
- [20] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proc. 2010 IEEE International Symposium on Circuits and Systems*, 2010, pp. 253–256.
- [21] A. Žgank, B. Horvat, and Z. Kačič, "Data-driven generation of phonetic broad classes, based on phoneme confusion matrix similarity," *Speech Communication*, vol. 47, no. 3, pp. 379–393, 2005.
- [22] X. Li and Z. Zhou, "Speech command recognition with convolutional neural network," *CS229 Stanford Education*, vol. 31, 2017.
- [23] M. A. Abushariah, R. N. Ainon, R. Zainuddin, M. Elshafei, and O. O. Khalifa, "Natural speaker-independent Arabic speech recognition system based on Hidden Markov Models using Sphinx tools," in *Proc. International Conference on Computer and Communication Engineering (ICCCCE'10)*, May 2010, pp. 1–6.
- [24] K. M. Nahar, M. Abu Shquier, W. G. Al-Khatib, H. Al-Muhtaseb, and M. Elshafei, "Arabic phonemes recognition using hybrid LVQ/HMM model for continuous speech recognition," *International Journal of Speech Technology*, vol. 19, no. 3, pp. 495–508, 2016.
- [25] E. M. Naji, "Bridging vision and speech: A novel VGG16-RNN approach for continuous Arabic speech recognition," *J. Electrical Systems*, vol. 20, no. 11s, pp. 1182–1192, 2024.

Copyright © 2026 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).