Fuzzy Systems and Soft Computing
ISSN : 1819-4362

PARALLEL CONVOLUTIONAL NEURAL NETWORKS FOR SOUND RECOGNITION USING TEMPORAL AND FREQUENCY FEATURES

Pratibha Rashmi Department of Computer Science, Dr. Bhimrao Ambedkar University, Agra
Manu Pratap Singh Department of Computer Science, Dr. Bhimrao Ambedkar University, Agra

Abstract.
Sound recognition is a critical task in numerous machine learning applications such as speech recognition, voice-controlled assistants, music recognition, and environmental sound classification. During the last decade, Convolutional Neural Networks (CNNs) have emerged as a powerful tool for sound recognition tasks. In the proposed work, we consider a novel approach using parallel convolutional neural networks (CNNs) to process temporal and frequency features separately. Short-time energy is used to extract temporal characteristics, whereas Mel-frequency cepstral coefficients (MFCCs) are used to extract frequency features. The parallel convolutional neural networks (CNNs) showcase the effective integration of these characteristics to enhance sound recognition performance.

Keywords:
Parallel CNNs, Short-time energy, MFCC, 1D-CNN, 2D-CNN, Sound Classification

1. Introduction

Sound recognition of the spoken digit, words and environment has received enormous attention in recent years. As compared to visual images, sound recognition of spoken digits and words can be challenging due to different types of acoustic scenes and background noise. There is limited research on the recognition of spoken digits. Thus, there are very few labeled databases available for this task. Therefore, it is challenging for researchers to accurately identify and classify spoken digits and words in various acoustic environments. Effective digit recognition has various applications, including automated telephone systems, voice-controlled interfaces, secure access systems, and digit-based data entry in diverse environments. In the past few years, verities of sound classification techniques have been proposed. The most widely used machine learning techniques includes support vector machine (SVM), HMM, and GMM [2]. Apart from these traditional sound classification algorithms, Deep Convolutional Neural Networks (DCNNs) has become a promising solution to handle the sound classification's problems. One of the most difficult aspects in the sound classification problem is feature extraction. Most of literature considered the feature extraction of the pre-processing step prior to classification. It has been reported that it is very difficult to present the feature of sounds in the pattern vector form. Some hand-designed techniques of feature extraction are used for the sound samples. In early works on sound classification relied on manually engineered features that have been widely used in sound classification, such as Amplitude Envelope, Zero-Crossing Rate, Root Mean Square Energy, Spectral Centroid, or other customized temporal and spectral acoustic features. These features combined with some traditional classifier like SVM, k-NN, GMM, random forest, etc., then classified into their corresponding sound classes. On the other hand Deep Convolutional Neural Networks (DCNNs) are well suited to the sound classification's problems because they have the ability to transform the sound data from temporal to spectral and then take benefits of spectrogram features as input which has been demonstrated to be an important characteristic for differentiate between various sounds like spoken digits and words. By using convolutional filters, the CNN should able to be to learn the spectro-temporal pattern of sound and differentiate them for the classification purpose. Thus, Convolutional Neural Network (CNN) has demonstrated its superior performance over previous techniques to handle the sound classification issues. Convolutional Neural Network requires less pre-processing and hand-engineered requirements as compared to other traditional pre-processing techniques which make it independent from prior knowledge. Traditional methods often process temporal and frequency features sequentially or in a combined manner, potentially overlooking intricate details unique to each type of feature. In the proposed work, we consider a parallel processing approach using CNNs to separately handle temporal and frequency features, thereby enhancing the recognition accuracy and computational efficiency. Short-time energy is used to extract temporal characteristics, whereas Mel-frequency cepstral coefficients (MFCCs) are used to extract frequency features. In the first experiment, we considered one-dimensional convolutional neural network (1D-CNN) with temporal features as input. In the second experiment, we considered MFCC based spectral features for two-dimensional convolutional neural network (2D-CNN), in the last experiment we consider a parallel processing method using CNNs to separately handle temporal and frequency features. The parallel processing method using CNNs performs more effectively over the proposed 1D-CNN with temporal features as input and 2D-CNN with frequency features.

This paper is organized as follows: section 2 consists of recent spoken digit classification work. Section 3 presents the pre-processing and feature extraction. Section 4 of the paper discusses about the implementation of proposed model of convolution neural network followed by simulation results. It also includes the discussion of performance analysis and state-of-art performance of CNNs for the classification. Finally, the conclusion is presented in the last section followed by the references.

2. Related Work

The success of Convolutional Neural Networks (CNNs) in image identification has led to the popularity of CNNs in sound recognition research. CNNs have been the subject of increasing interest for sound recognition in recent years. The research results on CNN-based sound recognition from current studies are compiled in this review of the literature. A Deep Convolution Neural Network was suggested in [3] for the recognition of Pashto isolated digits. Utilizing Mel frequency cepstral coefficients (MFCC), characteristics were taken out of the speech signal. The accuracy of the suggested work using the MFCC feature is 84.17%, indicating a 7.32% improvement over the current. For the spoken English digit dataset, a multiclass classification model utilizing random forest (RF), K-nearest neighbor (KNN), and support vector machine (SVM) was suggested in [4]. With 97.50%, Random Forest outperformed SVM and KNN. The audio data underwent one hot encoding after features was retrieved using the Short Time Fourier Transform (STFT). The Adam optimization technique produced the best accuracy of 99.65%. Based on Long Short-Term Memory (LSTM), an end-to-end method for learning to deal with a non-uniform sequence length of voice utterances has been proposed in [5]. The Mel Frequency Cepstral Coefficients technique (MFCC) was utilized by the system to extract features, which were further processed by a deep neural network. To encode MFCC feature sequences like a fixed size vector and feed them into a multilayer perceptron network for classification, they employed a recurrent LSTM or GRU architecture. Li and Lerch [6] emphasize the importance of temporal and frequency feature extraction in sound recognition in their thorough review. They go over several techniques for obtaining Mel-frequency cepstral coefficients (MFCC) and short-time energy (STE). The article emphasizes how MFCCs offer a reliable representation of the audio's spectral characteristics, whereas STE aids in capturing the energy changes of the sound signal. Jane Oruh and Serestina Viriri [7] have presented a multiclass classification model. It was proposed for the spoken English digit dataset using support vector machine (SVM), K-nearest neighbor (KNN), and random forest (RF). Random Forest performed better with 97.50% than the SVM and KNN. Features were extracted through Short Time Fourier Transform (STFT) from audio data and one hot encoding was performed on the audio data. The optimal accuracy of 99.65% was achieved by using Adam optimization algorithm. To sum up, these new studies show how to use a variety of machine learning (ML) and deep learning methods, including CNNs, SVM, RF, residual networks, and combinations of multiple neural network architectures, to improve spoken digit classification. In order to improve this field's accuracy, resilience, and efficiency, researchers are still looking into novel strategies.

3. Preprocessing and Feature Extraction

In our proposed system, we considered the sound samples of spoken English digits dataset. The dataset Speech Commands v1 (SCV1) [8] is used to collect the sound samples. It consists of 64,727 one-second .wav audio file of 30 common speech commands. The audio files are arranged into the folders based on the word they contain. So we have used only the folders that contain the digits. We have considered the 13323 audio clips for the training of the CNN models, a training-testing data split of 80%–20% was used. Out of 13323 samples, we constructed 10658 sound samples for the training set and rest 2665 sound samples were used for the test. Recent developments in Deep Neural Networks have promoted feature learning from the time-domain features to reduce the pre-processing task. In a sound recognition task, the performance of the model is significantly affected by the type of input. So, in the first experiment, we have considered temporal features as inputs which are extracted using short-time energy. This involves calculating the root mean square (RMS) energy of the signal over short frames. In the second experiment we have considered frequency features as input which extracted using MFCC. This involves transforming the signal into the frequency domain and then computing the cepstral coefficients that capture the signal's spectral properties, and in last experiment we have considered both temporal and frequency features.

3.1. Short-Term Energy

The short time energy is the energy of a short phrase [9]. Short time energy is an easy and useful way to tell the difference between voiced and unvoiced parts. Energy can also be used to find the end of a sentence [10]. The signal energy can be representing as:

$$E = \sum_{m=-\infty}^{\infty} x^2(m) \qquad (1)$$

$$E_n = \sum_{m=n-N+1}^{n} x^2(m) = x^2(n-N+1) + \cdots \ldots . + x^2(n) \qquad (2)$$

In above equation E represents energy of the signal $x(m)$. There is a very small or almost no efficacy of this definition for time-varying signals. Thus, n^{th} frame window is applied on short-term speech signal:

$$x_n(m) = x(m)w(n-w) \qquad n-N+1 \leq m \leq n \qquad (3)$$

Here, $n = 0, IT, 2T, \ldots, N$ is considered as window length and T is the frame shift.

The short time energy (STE) of above signal can be calculated by the following equation:

$$E_n = \sum_{m=n-N+1}^{n} x\,[(m)w(n-m)]^2 \qquad (4)$$

Here, $w(n-m)$is the window, n is the speech signal sample that the analysis window is centred on and N is the window length. In this, high energy would be classified as voiced and lower energy as unvoiced.

3.2. Mel-frequency cepstral coefficients (MFCCs)

Mel-Frequency Cepstral Coefficients (MFCC) is one of the sound feature extraction techniques that are often used to distinguish one sound from other sounds. This may be attributed because MFCCs models [11] the human auditory perception with regard to frequencies which in return can represent sound better. The general procedure of the MFCC is shown in Figure 1.

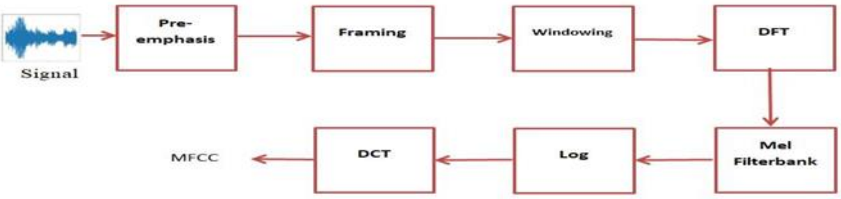


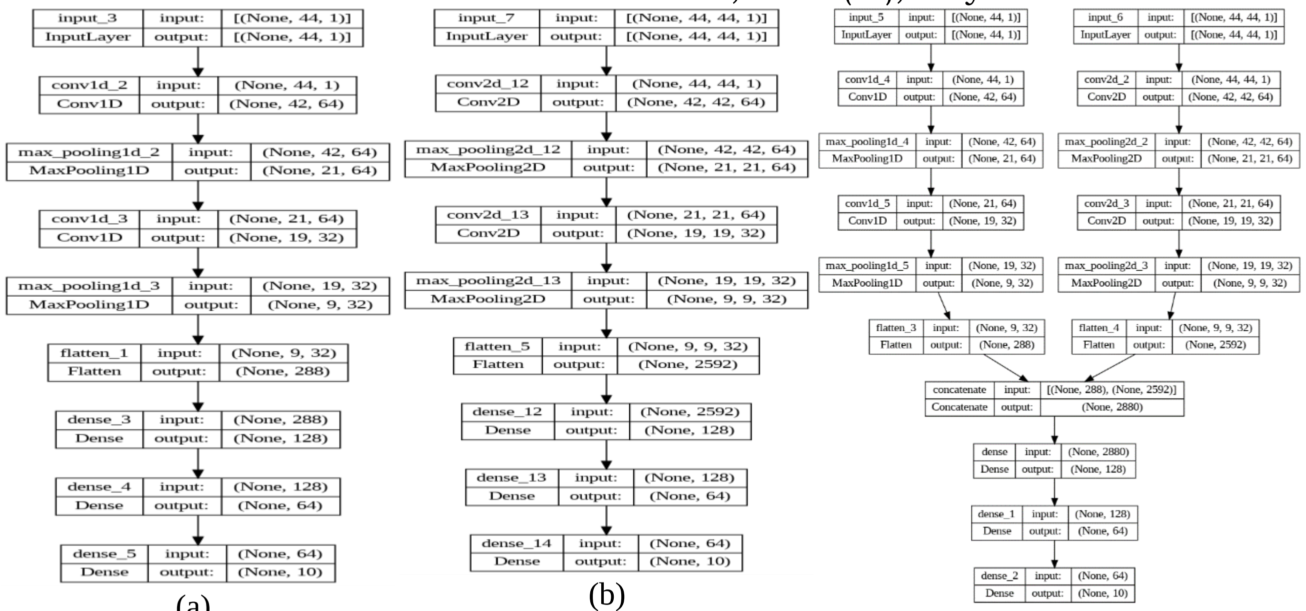
Fig.1. Block diagram of the MFCC processor.

It depicts that MFCC consists of seven blocks and each block has its function and mathematical approaches. The mapping between real frequency (Hz) and Mel frequency is given by the following Eq. 5 as:

$$mel(f) = 2595 * log10(1 + f/700) \qquad (5)$$

4. Implementation and Experimental Models

The proposed CNN architecture consists of two convolutional blocks with max-pooling layer stacked together in a deep architecture. The two convolutional layers with same size of filters are of 3×3. The different number of filters is 64 and 32 respectively in the convolution layers. All the convolution layers use the rectified linear unit (ReLU) activation function for lower computational cost. The each convolution layer is followed by 2×2 pool size of max-pooling layer over the obtained feature maps. After flattening layers, we have also used two dense layers with 128 and 64 neurons respectively followed by output layer. In the third experiment the outputs of the two CNNs are concatenated and passed through fully connected layers to produce the final classification output. The CNN model is trained for the input samples and the parameters of the network are optimized using Adam stochastic gradient learning and sparse categorical cross entropy methods to minimize the error. The architecture of experiment 1, 2, and 3 can be shown in Figure 2.


input_3 | InputLayer | input: [(None, 44, 1)] | output: [(None, 44, 1)]
conv1d_2 | Conv1D | input: (None, 44, 1) | output: (None, 42, 64)
max_pooling1d_2 | MaxPooling1D | input: (None, 42, 64) | output: (None, 21, 64)
conv1d_3 | Conv1D | input: (None, 21, 64) | output: (None, 19, 32)
max_pooling1d_3 | MaxPooling1D | input: (None, 19, 32) | output: (None, 9, 32)
flatten_1 | Flatten | input: (None, 9, 32) | output: (None, 288)
dense_3 | Dense | input: (None, 288) | output: (None, 128)
dense_4 | Dense | input: (None, 128) | output: (None, 64)
dense_5 | Dense | input: (None, 64) | output: (None, 10)
input_7 | InputLayer | input: [(None, 44, 44, 1)] | output: [(None, 44, 44, 1)]
conv2d_12 | Conv2D | input: (None, 44, 44, 1) | output: (None, 42, 42, 64)
max_pooling2d_12 | MaxPooling2D | input: (None, 42, 42, 64) | output: (None, 21, 21, 64)
conv2d_13 | Conv2D | input: (None, 21, 21, 64) | output: (None, 19, 19, 32)
max_pooling2d_13 | MaxPooling2D | input: (None, 19, 19, 32) | output: (None, 9, 9, 32)
flatten_5 | Flatten | input: (None, 9, 9, 32) | output: (None, 2592)
dense_12 | Dense | input: (None, 2592) | output: (None, 128)
dense_13 | Dense | input: (None, 128) | output: (None, 64)
dense_14 | Dense | input: (None, 64) | output: (None, 10)
input_5 | InputLayer | input: [(None, 44, 1)] | output: [(None, 44, 1)]
input_6 | InputLayer | input: [(None, 44, 44, 1)] | output: [(None, 44, 44, 1)]
conv1d_4 | Conv1D | input: (None, 44, 1) | output: (None, 42, 64)
conv2d_2 | Conv2D | input: (None, 44, 44, 1) | output: (None, 42, 42, 64)
max_pooling1d_4 | MaxPooling1D | input: (None, 42, 64) | output: (None, 21, 64)
max_pooling2d_2 | MaxPooling2D | input: (None, 42, 42, 64) | output: (None, 21, 21, 64)
conv1d_5 | Conv1D | input: (None, 21, 64) | output: (None, 19, 32)
conv2d_3 | Conv2D | input: (None, 21, 21, 64) | output: (None, 19, 19, 32)
max_pooling1d_5 | MaxPooling1D | input: (None, 19, 32) | output: (None, 9, 32)
max_pooling2d_3 | MaxPooling2D | input: (None, 19, 19, 32) | output: (None, 9, 9, 32)
flatten_3 | Flatten | input: (None, 9, 32) | output: (None, 288)
flatten_4 | Flatten | input: (None, 9, 9, 32) | output: (None, 2592)
concatenate | Concatenate | input: [(None, 288), (None, 2592)] | output: (None, 2880)
dense | Dense | input: (None, 2880) | output: (None, 128)
dense_1 | Dense | input: (None, 128) | output: (None, 64)
dense_2 | Dense | input: (None, 64) | output: (None, 10)

(a) (b) (c)

Fig.2. CNN architecture with STE (a) and MFCC(b) as input, A Parallel CNN Architecture with both STE and MFCC

5. Results

Three experiments are conducted to obtain the results on the existing dataset of sound samples i.e., Speech Command V1 (SCV1) is used for the construction of sample patterns for training & testing with different feature extraction methods. In our first and second experiments, we considered temporal and frequency features sequentially whereas in third experiment we have used parallel CNNs for temporal and frequency features. We considered 1D-CNN with temporal features as input for the training & testing and 2D-CNN with spectral features as input for the training & testing. We have trained the CNN model using Adam stochastic gradient learning and sparse categorical cross entropy method is used to minimize the error. We have considered 50 epochs for the training. The simulation results are presenting in Table 2.

Table 2: Simulated results on the CNN model

Experiments	Features	Epochs	Parameters	Accuracy
Exp. 1	Short Time Energy	50	52330	79.45
Exp. 2	MFCC	8	359914	99.47
Exp. 3	Both (STE & MFCC)	18	403210	99.96

The comparative analysis of different experiments is shown in Figure 3.

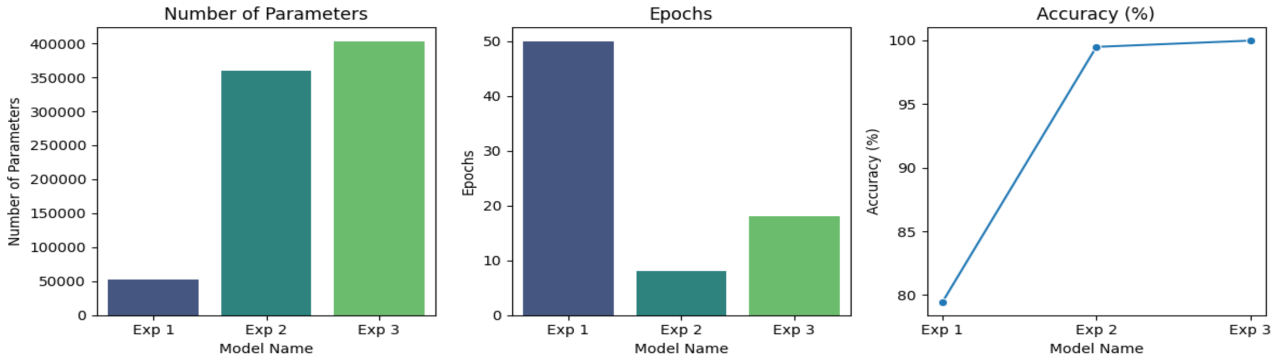


Fig.3. Comparative analysis of all three experiments

The confusion matrix of sequential and parallel CNN experiments for testing data is presented in Figure 4.

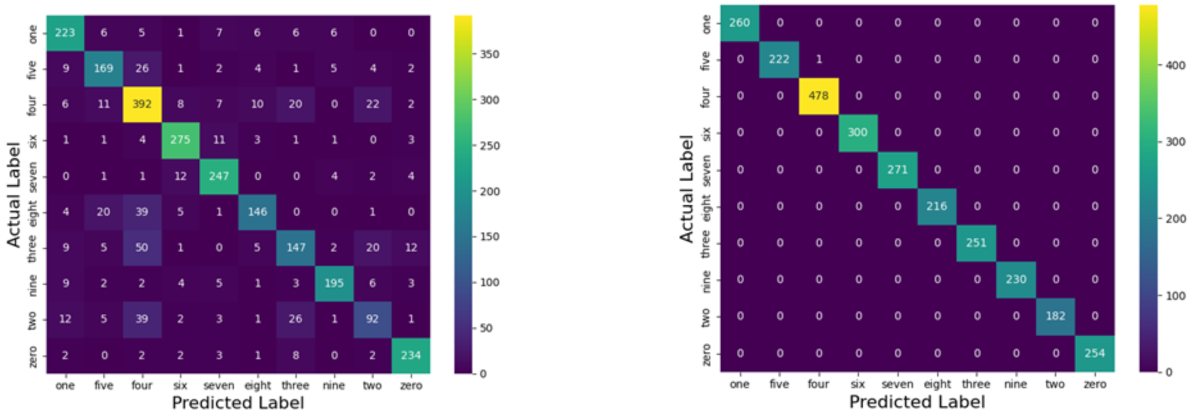


Fig.4. Confusion Matrix of sequential (Exp 1) and parallel CNNs (Exp 3)

6. Conclusion

Three different experiments are considered on convolutional neural network to analyse the performances of classification for spoken English digits. Different experiments are conducted to obtain the improved model with effective classification accuracy. Parallel convolutional neural network (exp. 3) with temporal and spectral features is identified as the most optimized architecture with 99.96% classification accuracy for the sound samples. Thus, the experimental results exhibit the effectiveness of the parallel CNN approach in sound recognition tasks. It is also observed from the simulated results that the rate of misclassification is less in comparison to other models. This approach can be further enhanced by exploring other feature extraction techniques and more complex neural network architectures.

7. Acknowledgements

The authors thankfully acknowledge the financial support of the Dr. Bhimrao Ambedkar University, Agra India in the form of a minor research project under J C Bose Pariyojana.

8. References

[1] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, & A. Lopez, (2020). A comprehensive survey on support vector machine classification: Applications, challenges and trends. Neurocomputing, 408, 189-215.

[2] S. Mao, D. Tao, G. Zhang, P. C. Ching, & T. Lee, (2019). Revisiting hidden Markov models for speech emotion recognition. In ICASSP 2019-2019 IEEE (ICASSP) 6715-6719. IEEE.

[3] B. Zada, & R. Ullah, (2020). Pashto isolated digits recognition using deep convolutional neural network. Heliyon, 6(2), e03372.

[4] K. M. Maddimsetti Srinivas and G. L. P. Ashok, (2019), "Spoken English digit classification using supervised learning," International Journal of Research in Signal Processing, Com & Com System Design, vol. 5, pp. 49–53.

[5] Y. Zhang, M. Pezeshki, P. Brakel, S. Zhang, C. L. Y. Bengio, & A. Courville, (2017). Towards end-to-end speech recognition with deep convolutional neural networks. arXiv preprint arXiv:1701.02720.

[6] I. Rida, (2018). Feature extraction for temporal signal recognition: An overview. *arXiv preprint arXiv:1812.01780*.

[7] J. Oruh, J. and Viriri, S., (2022). Deep Learning-Based Classification of Spoken English Digits. Computational Intelligence and Neuroscience, 2022(1), p.3364141.

[8] P. Warden, Speech Commands: A public dataset for single-word speech recognition, 2017.

[9] X.Yang, B.Tan, JDing, J.Zhang, J.Gong, "Comparative Study on Voice Activity Detection Algorithm," International Conference on Electrical and Control Engineering, ICECE, 2010

[10] Y.K.Lau, C.K.Chan. "Speech Recognition Based on Zero Crossing Rate and Energy," IEEE Transactions on Acoustic, Speech and Signal Processing, Feb.198.

[11] S. V. Chapaneri. Spoken digits recognition using weighted MFCC and improved features for dynamic time warping. Int J Comput Appl 2012;40:6–12.