Scientia Iranica D (2023) 30(1), 116–123

Sharif University of Technology
Scientia Iranica
Transactions D: Computer Science & Engineering and Electrical Engineering
http://scientiairanica.sharif.edu

Research Note

Language recognition by convolutional neural networks

L. Khosravani Pour and A. Farrokhi*

Department of Electrical Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran.

Received 26 September 2021; received in revised form 24 May 2022; accepted 15 August 2022

KEYWORDS
Speech recognition;
Speech segmentation;
Convolutional neural networks;
Persian language;
Artificial intelligence.

Abstract. Speech recognition representing a communication between computers and human as a sub field of computational linguistics or natural language processing has a long history. Automatic Speech Recognition (ASR), Text To Speech (TTS), speech to text, Continuous Speech Recognition (CSR), and interactive voice response systems are different approaches to solving problems in this area. The performance improvement is partially attributed to the ability of the Deep Neural Network (DNN) to model complex correlations in speech features. In this paper, unlike the use of conventional model for sequential data like voice that employs Recurrent Neural Networks (RNNs) with the emergence of different architectures in deep networks and good performance of Conventional Neural Networks (CNNs) in image processing and feature extraction, the application of CNNs was developed in other domains. It was shown that prosodic features for Persian language could be extracted via CNNs for segmentation and labeling speech for short texts. By using 128 and 200 filters for CNN and special architectures, 19.46 error in detection rate and better time consumption than RNNs were obtained. In addition, CNN simplifies the learning procedure. Experimental results show that CNN networks can be a good feature extractor for speech recognition in various languages.

1. Introduction

ASR systems have been developed and changed in the past decade and a plenty of methods have been applied to solve different problems in the domain. These developments started in the 1950s in AT and T Bell. First systems focused on phonemes and isolated words but after these early systems, a major trend was recognition connected words and continuous speech recognition. In the 80s, Human Markov Model (HMM) and neural networks were employed and researchers used pattern recognition and clustering methods [1]. In spite of enormous effort on speech recognition, a robust continuous speech recognition and spoken language understanding represented an unresolved problem. One popular topic in the last decade was the implementation of a conversational speech recognition system.

A simple TTS system consists of natural language processing for extracting prosody data and phonemes. In the following, digital signal processing is employed to produce voice or acoustic data. Supervised and unsupervised systems in machine learning techniques and Artificial Intelligence (AI) are employed by ASR systems to solve dilemmas. Nowadays, speech recognition systems remain a difficulty and much effort has been made for these types of systems.

Figure 1 shows a simple speech generator system block diagram that is specifically used to generate voice from noise and periodic pulse with a bank of filters generates different frequencies based on human voice-generating organs.

In recent years, several successful ASR sys-

*. *Corresponding author.*
E-mail addresses: St_l_khosravanipoor@azad.ac.ir (L. Khosravani Pour); ali_farrokhi@azad.ac.ir (A. Farrokhi)

doi: 10.24200/sci.2022.59110.6064

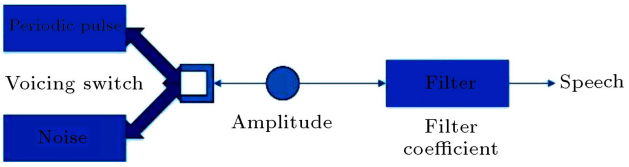

Figure 1. Speech generator system block diagram.

tems have been introduced (a review could be found in [2]). Among these systems, pattern recognition, AI techniques, and statistical methods are dominant approaches in the most successful systems [2]. Like other machine learning problems, the most important block in this system is feature extraction. The most common feature used for ASR is Mel-frequency cepstral coefficient [3–7]. As mentioned before, modeling is the next block after extracting features. Modeling is usually performed by ANN or HMM methods. Linguistic, statistical, and grammatical information is also used for an ASR system. Language model is also built based on language phonetic, structure, and grammar.

Automatic speech recognition relies on contextual constraints (i.e., language modeling) to guide the search algorithm. The task of detecting phonemes is much more limited than decoding the word; thus, the error rate (even if measured in terms of phonemic error to decode the word) is significantly higher. For this reason, phonological decoding is very slow and, even, decoding incorrect words can be useful for things like pronunciation modeling and sound conversion. The acoustic model identifies the phonemes. The acoustic model is created from machine learning algorithms. Machine learning is divided into two categories: training and testing. At the training phase, the acoustic model is computed and, then, used in the decoding step to transfer the acoustic expression to the text.

Deep learning in recent years has good results in different areas. A TTS system integrates many modules such as text analyzer, F_0 generator, spectrum generator, a pause estimator, and a speaker and creates a waveform. However, deep learning can unite these blocks into a single model [8]. Convolutional neural networks show a good outcome in research. Due to simple training step and network architecture, the frequency of using CNN in other domains like speech recognition has increased. Segmentation or labeling is a challenging problem in ASR system to relate each event in speech signal to a specific phoneme and after that specific letter.

In the reminder of this paper, Section 2 reviews the related works. An overview of Persian language (Farsi) and implemented ASR systems is given in this section. Section 3 provides a review of the proposed method for Persian language segmentation. The next section examines the experiments in detail and presents the results. Section 5 presents the summary and conclusions of the paper.

2. Related works

Some recent studies have achieved surprising results on deep learning-based TTS systems. The purpose of some of them is to reduce the dependency on manual internal modules [9]. Most methods use RNN, which is a technique for predicting time series [10]. Phoneme recognition has also been the target of several studies [11–13]. Plenty of methods and techniques have been introduced for different languages but English mostly. This research, focusing on the Persian language and after a brief description of the structure of the language, presented an approach for independent phoneme recognition by the Persian speaker. Further, the efficiency of the proposed method was enhanced by neural network in comparison to the reviewed cases existing in the literature [14,15].

Persian is an Iranian language in the Indian and Iranian branches of the Indo-European languages [1]. There is a special ability in this language to be free-word-order, especially in appendices and supplements. For example, adverbs can be placed at the beginning, end, or middle of a sentence without changing the meaning of the sentences. This flexibility in word ordering makes it difficult to extract Persian grammar [1]. Farsi has 6 vowels and 23 consonants (Figure 2). Three vowels are long ($/i/$, $/u/$, $/\partial/$) and the other three are short or themes ($/e/$, $/o/$, $/a/$). They are often called short and long vowels. The point is to distinguish the three long vowels from their short counterparts in terms of position rather than length. Farsi is characterized as a syllable-timed language. This means that the length of time a sentence takes depends on the number of syllables which, in turn, depends on the number of stress syllables, unlike languages that have stress times such as English and German. In addition, Persian syllables are always one of the patterns (V, CV, CVCC, CVC, VC, VCC). C and V represent consonants and vowels, respectively [16].

3. Method

In general, a DNN refers to a feedforward neural network with more than one hidden layer. Each hidden layer has a number of units (or neurons), each of which takes all outputs of the lower layer as input, multiplies them by a weight vector, sums the result, and passes it through a nonlinear activation function such as sigmoid. The CNN can be regarded as a variant of the standard neural network. Instead of using fully connected hidden layers, the CNN introduces a special network structure that consists of so-called alternating convolution and pooling layers.

3.1. ASR, TTS systems based on CNN

In recent years, convolutional neural networks have been widely used in different systems. Hideyuki et al.

p	/por/	'full'	d	/dir/	'late'	k	/kæm/	'little'
b	/bɒm/	'roof'	n	/nɒn/	'bread'	g	/guni/	'sack'
m	/mæn/	'I'	s	/særd/	'cold'	χ	/χæm/	'sorrow'
f	/nɒf/	'navel'	z	/zir/	'below'	ɣ	/ɣuri/	'pot'
v	/gɒv/	'cow'	ʃ	/ʃen/	'sand'	ʔ	/tæʔsir/	'impression'
r	/roz/	'day'	ʒ	/ʒærfa/	'depth'	h	/kɒh/	'hay'
l	/sili/	'slap'	tʃ	/tʃin/	'crease'	j	/jek/	'one'
t	/tir/	'arrow'	dʒ	/dʒɒn/	'soul'			

Figure 2. Farsi consonant (left side: example words with English meanings) and Farsi vowels, international phonetic alphabet.

introduced a Deep TTS (DCTTS), a useful and novel neural TTS, which is confidential [8], and Osama et al. used CNN to detect speech [3]. By using a novel acoustic model, a hybrid composed of a pre-trained, deep neural network, and a context-dependent hidden Markov model was proposed [10].

Speech recognition has been applied over windows of acoustic frames that overlap in time to learn classes such as phone, speaker, and gender [15]. Dimitri introduced phoneme sequence recognition system by convolutional neural network [15]. Yao investigated various training aspects of DNN as a generation model for TTS synthesis [16]. In different speech domain applications, emotion recognition was also implemented by deep neural networks [10]. Size of spectrogram axis and fine-grained localized features were discussed in [17]. Finally, an end-to-end deep neural network for automatic speech recognition was introduced by the authors in [6]. The mentioned work was conducted based on two main components, frame wise classification and phone decoding. This paper deals with the acoustic model, recognizes phonemes in Persian language by training a CNN as the first step, and tests it to use the obtained acoustic model to label each part of speech with a phone.

Much effort in Persian language has been made to build a language model [18,19] and implement ASR or TTS system [20–23]. Some projects like the one in [20] were based on the interactive voice response model that consisted of different parts as commercial projects. Although all the efforts have achieved attentional result, the application of CNN to speech recognition like other languages is a new branch. This section presents a brief description of the basic CNN layers first and then, explains how audio signal can be fed for phone recognition. Network architecture and how it solves the segmentation problem are elaborated in the next section. Different network structures have been analyzed and tested in some research studies [24–26]. In this research, the experiments are used to find the best network structure based on different architecture.

3.2. Convolution layers

A convolution layer contains a set of filters whose parameters need to be learned. It differs from a standard, fully connected hidden layer in two important aspects, however. First, each convolutional unit receives input only from a local area of the input to compute an activation map. This means that each unit represents some features of a local region of the input. Second, the units of the convolution layer can themselves be organized into a number of feature maps, where all units in the same feature map share the same weights, but receive input from different locations of the lower layer.

3.3. Pooling layers

A pooling operation is applied to the convolution layers to generate its corresponding pooling layer. The pooling layer is also organized into feature maps, and it has the same number of feature maps as the number of feature maps in its convolution layer, but each map is smaller. The purpose of the pooling layer is to reduce the resolution of feature maps. This means that the units of this layer will serve as generalizations over the features of the lower convolution layer, and because these generalizations will again be spatially localized in frequency, they will also be invariant to small variations in location. This reduction is achieved via a pooling function to several units in a local region of a size determined by a parameter called pooling size. It is usually a simple function such as maximization or averaging. The pooling function is applied to each convolution layer. Eq. (1) illustrates the convolution function in a convolutional neural network:

$$Q_j = \sum_{i=1}^{I} (Q_i * W_{i,j}), \quad j = 1, ..., J, \tag{1}$$

where Q_i represents the ith input feature map and $W_{i,j}$ represents each local weight matrix, flipped to adhere to the convolution operation definition. Both O_i and $W_{i,j}$ are vectors if one-dimensional feature maps are used, and they are matrices if two-dimensional feature

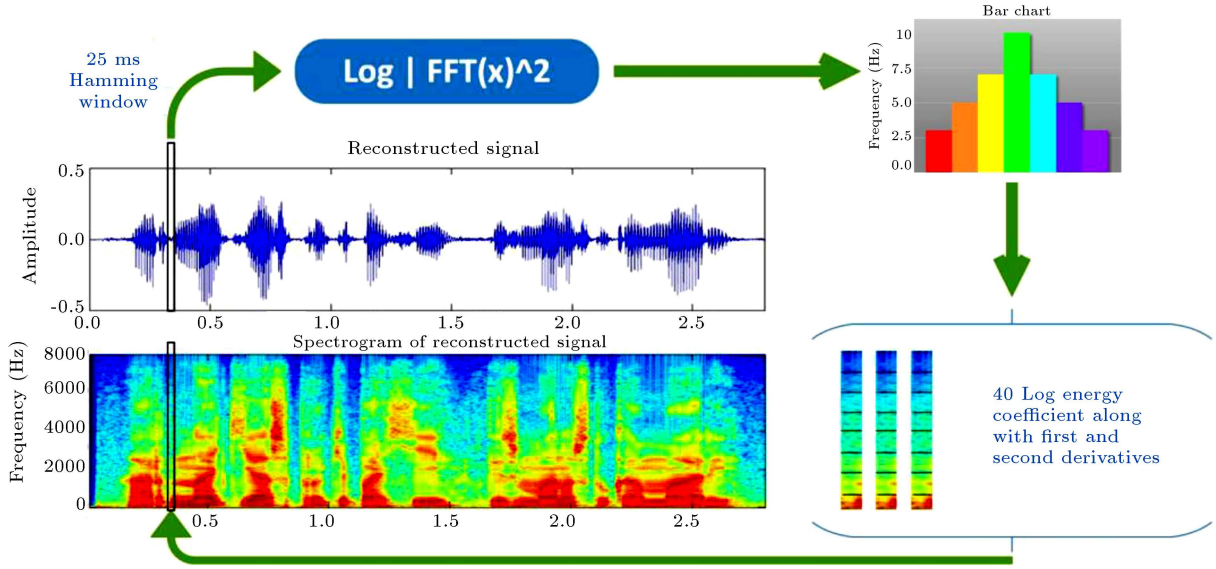

Figure 3. Conversion of speech signal into spectrogram.

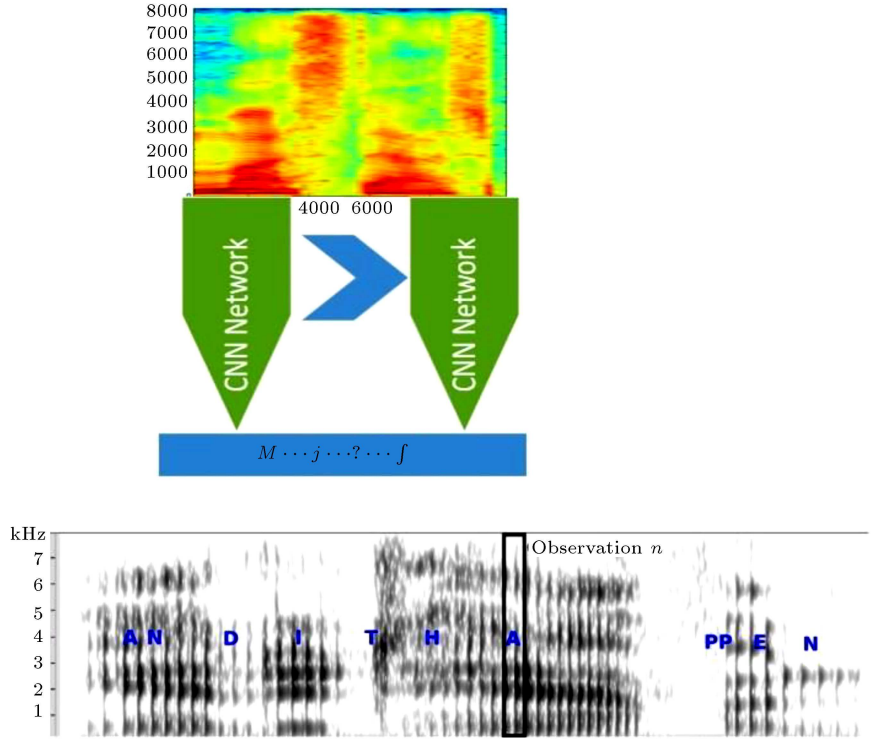

Figure 4. Visualization of the CNN architecture.

maps are used (where 2-D convolution is applied to the above equation), as presented in Figures 3 and 4.

3.4. How to feed speech to a CNN

To use CNN for pattern recognition, the input data should be organized as a number of feature maps to organize CNN (see Figure 5). Like visual image processing applications, the input is organized as a two-dimensional array (2-D) because the pixel values are in the horizontal and vertical coordinate indices [3]. CNNs run a small window of the input image both during training and testing phases so that the weight of the network looking through this window can learn different characteristics of the input data. This section examines how speech feature vectors are organized in maps with properties suitable for CNN processing. The input spectrometer, with static features, is delta and delta-delta (i.e., first and second time derivatives) [3,27].

Acoustic model should not be dependent on some

vocal parameters like different pitches resulting from speaker's gender and age, by using CNN due to its structure, only frequencies remain as final feature. In this paper, the Mel frequency spectral coefficients are used to calculate the energy input to the system [28]. The delta is employed to illustrate the shape of the acoustic energy distribution at each frequency band [3].

3.5. Network architecture

The proposed structured phoneme recognition model like many architectures seen in recent years contains several convolutional layers presented as C, max-pooling layers presented by M, and finally fully connected layers shown by F. Figure 5 depicts the architectures. The detailed model configuration is:

$$\begin{aligned} &Input(40 \times 25 \times 3) - C(38 \times 23 \times 128) - M(13 \times 23 \\ &\times 128) - C(11 \times 19 \times 256) - M(3 \times 19 \\ &\times 256) - C(3 \times 15 \times 384) - C(3 \times 11 \times 384) \\ &- F(1024) - F(256) - F(32). \end{aligned}$$

The sizes of C and M layers are defined as *width* $\times$ *height* $\times$ *depth*, where *width* $\times$ *height* determines the dimensionality of each feature map and *depth* represents the number of feature maps. Since the input signal size is relatively small, the filter size is chosen as 3×5 for the first convolutional layer and 3×1 for the other. The max-pooling layer uses a window of size 3×1 with a stride of 1, which has been widely adopted in current object detection algorithms, and it gives an encouraging performance. The input image size depends on cell scales, and a 40×25 patch is large enough to cover a phone. Typing speed is usually measured by the number of words typed per minute; however, the unit of measurement is usually syllables per second or phonemes per second. According to studies, most speakers can produce speech at a rate of five or six syllables per second or ten to twelve phonemes per second. Given the small size of the input signal patch, it is sufficient to assemble two pairs of C-M layers to calculate the feature. Meanwhile, several F layers are designed to extract different structures and syllables with different sequences of telephone, which can benefit the end result. Sigmoid is used for the activation function and also, a ReLu function is employed for all C layers.

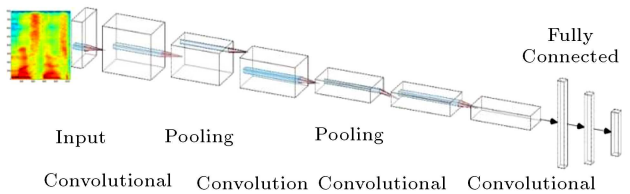

Figure 5. Working structure (CNN slides all over the input signal and detects phoneme in each window).

3.6. Training

The CNN audio model is trained using cross entropy. The CNN model has a six-ring layer and three fully connected layers. The torsional layer consists of 128 convolutional filters with 40 bands. By using a few initial iterations at a constant learning rate of 0.08, the networks are trained to discriminate, thereby reducing the rate of learning based on reduced validation error. Training stops and there is no significant reduction in validation error or increased cross-validation error. Backpropagation was performed using a random batch descent with a mini-batch of 128 training samples. In the training process, the mini-batch random slope for Stochastic Gradient Descent (SGD) landing algorithm is the main problem used to train DNNs. SGD is inherently a trail, and the machines are difficult to pull along. So far, SGD with the Graphics Processing Unit (GPU) has been the best training strategy for CNNs because the GPU exploits parallelism in the layered DNN structure. Twenty-nine target class labels (28 Persian letters and 1 pause) were used after decoding; different syllables and structures by fully connected layers and 256 classes were mapped to a set of 29 classes of Persian phonemes. For neural-network training, (a) learning rate annealing, in which the learning rate steadily decreases over successive iterations, and (b) early stopping strategies, in which a held-out development set is used to determine when overfitting starts, were utilized.

4. Results

In this section, the experimental results are presented and the input materials are considered as follows:

1. **Dataset**: A local dataset was used for training and testing. The proposed CNN consisted of 1000 Persian sentences that were segmented at phoneme and syllable and word levels [29]. Sentences uttered by a native Persian male speaker involved a set of every day conversational speech. It was sampled at 10 kHZ with a 16-bit resolution and average speed of 4 syllable/sec. The dataset consisted of pitch contour, energy contour, and duration, length of syllable, etc. which can be found in detail in [30];
2. **Test System**: implementation and test were done in a system with the following configuration. Obviously, the values can be changed in other systems with different performance. Hardware specification and implementation platform are as follows:

   (a) Hardware: The models were trained on a gaming PC equipped with two GPUs. The main memory of the machine was 32GB, which is

much larger than the audio dataset. Both GPUs featured NVIDIA GeForce GTX 980 Ti with a memory capacity of 6 GB;

(b) DNN platform: The proposed neural networks using Keras with Tensorflow backend were implemented in Python environment. TensorFlow is a very powerful and open-source library for implementing and deploying large-scale machine learning models and Keras is a high-level API library for implementation of machine learning algorithms.

4.1. Recognition performance

This section examines the recognition performance of the proposed method for the mentioned dataset. 50 sentences in the dataset were used (780 syllable) as test data to evaluate the performance of the network. As described in [28] and based on the dataset used, the performance in six groups (Table 1) of consonants classified by their vocal features was evaluated. Another parts correspond to letters that have a sound (vowel) and differ in sound characteristics from all groups that do not have a sound (consonant). All evaluations performed by PER (Phone Error Rate) that is equal to true detected phone per all entered phone into the network. In the next evaluation, the feature maps change to 200 for the first two convolutional layers. Increasing the number of feature maps helps find complicated features and introduce an abstraction level in feature engineering. The results are shown in Table 1. The best PER is 20.33% for consonants, while the worst one is 23.5% for explosive phonemes. Explosive sounds have a shorter duration in all languages and are harder to be distinguished. The final average error in the first architecture without vowels is 21.5%, while considering the vowels, the average error for both architectures is 19.46 and 19.1, respectively.

Figure 6 shows the epochs at the training phase. As mentioned before in the training section, the training required to reach a minimum error continues and following about 25 epochs, the network reaches the target performance. In this research, the data set of a speaker is used. In order to better evaluate the performance of the network, as a future task, the input data set can be increased by using different speakers. The performance of the network was found to be much better due to training data, but it intentionally prevents overfitting due to the limited amount of training data.

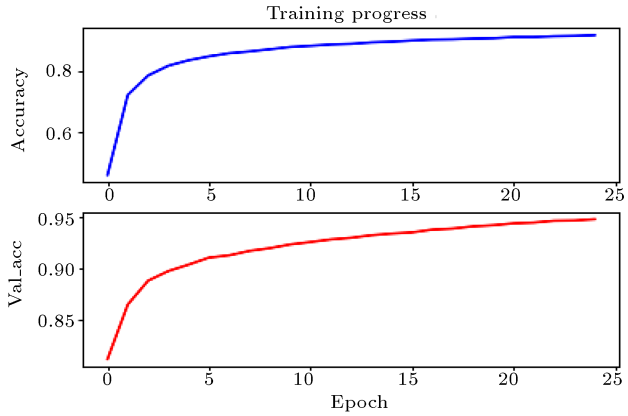

Figure 6. Error rate at the training phase.

Table 2 describes the results of the proposed method for different phone placement in the syllable.

Table 2. Error rate per syllables.

Persian syllable structures	Error rate for target class (%)	
	Vowel	Consonant
V	17.4	–
VC	17.6	21.3
CV	17.4	22.1
CVC	17.4	20.4
CVCC	17.5	21.1
Average	17.4	21.1

Table 1. Phone error rates.

Consonant group	PER (Phone Error Rate)%		
	Phonetics	Error (128 feature map)	Error (200 feature map)
C1	p,t,k	23.5	22.9
C2	b,d,g	21.85	21.73
C3	m,n,l,r,y	22.03	21.88
C4	s,sh,f	20.78	20.62
C5	z,j,zh,ch,v	20.56	20.44
C6	h,x,e	20.33	19.97
Vowels	A,e,o,∂,u,i	17.43	17.08
Average	–	19.46	19.1

As mentioned before, Persian language has certain syllables that are shown in the table. Recognition of each vowel and consonant in each position should produce different acoustic information. The network in the first layer of fully connected layers determines this level of feature to get the best regression for mapping each phone to a proper syllable and, finally, detects them. Detailed information could be seen in the bellow table. Due to the lack of significant change in error rates across the two CNN structures, the first architecture was tested for vowels and consonants at each position in the syllable. Table 2 shows the average error rate for vowels and consonants.

5. Conclusion

In this paper, the CNN network was proposed for phone recognition in Persian language. An acceptable performance compared to standard algorithms was achieved based on convolutional neural networks. In contrast to the more equivocal results of synthesizer method and unsupervised techniques, feature extraction by CNN helped reach a better acoustic model independent of speaker by removing prosody information from speech signal. Two architectures were assessed to reach better results in the subject. Reducing the feature map size and convolutional layers helps increase detection speed, but it reduces the PER (Phone Error Rate). Persian language consonant groups were discussed for each error rate and it was found that the average performance was comparable to the state-of-the-art model in other languages.

References

1. Sameti, H., Veisi, H., Bahrani, M., et al. "A large vocabulary continuous speech recognition system for Persian language", *EURASIP Journal on Audio, Speech, and Music Processing*, **2011**(1), pp. 16–28 (2021).
2. Kurzekar, P.K., Deshmukh, R.R., Waghmare, V.B., et al. "Continuous speech recognition system: A review", *Asian Journal of Computer Science and Information Technology*, **4**(6), pp. 62–66 (2014).
3. Ossama, A.-H., Abdel-rahman, M., Hui, J., et al. "Convolutional neural networks for speech recognition", *Audio, Speech, and Language Processing*, IEEE/ACM, **22**, pp. 1533–1545 (2014). 10.1109/TASLP.2014.2339736
4. Hayani, S., Benaddy, M., El Meslouhi, O., et al. "Arab sign language recognition with convolutional neural networks", In *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*, UAE, IEEE (2019).
5. Vildan, İ.K. and Kavak, E. "Variation sets in child-directed and child speech: A case study in Turkish", *Eurasian Journal of Applied Linguistics*, **7**(1), pp. 1–10 (2021).
6. Boukdir, A., Benaddy, M., Ellahyani, A., et al. "Isolated video-based Arabic sign language recognition using convolutional and recursive neural networks", *Arabian Journal for Science and Engineering*, **47**(2), pp. 2187–2199 (2022).
7. Anil, M.A., Rebello, R.M., and Bhat, J.S. "Speech-language profile of a child with fahr's disease: Case report of a rare neurodegenerative disorder", *Journal of Natural Science, Biology and Medicine*, **11**(2), pp. 206–211 (2020).
8. Hideyuki, T., Uenoyama, K., and Aihara, S. "Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention", *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4784–4788 (2017).
9. Mitra, V. and Franco, H. "Time-frequency convolutional networks for robust speech recognition", *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Singapour, IEEE (2015).
10. Dahl, G.E., Yu, D., Deng, L., et al. "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition", *IEEE Transactions on Audio, Speech, and Language Processing*, **20**(1), pp. 30–42 (2021).
11. Almutairi, M., Nouf, A.L., and Zitouni, M. "The uses and functions of barack Obama's hedging language in selected speeches", *Eurasian Journal of Applied Linguistics*, **8**(1), pp. 73–84 (2022).
12. Cheng, K.L., Yang, Z., Chen, Q., et al. "Fully convolutional networks for continuous sign language recognition", In *European Conference on Computer Vision*, Springer, Cham, Germany, pp. 697–714 (2020).
13. Tóth, L. "Combining time- and frequency-domain convolution in convolutional neural network-based phone recognition", *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Berlin, Germany, IEEE (2014).
14. Qazani, M.R.C., Asadi, H., Lim, C.P., et al. "Prediction of motion simulator signals using time-series neural networks", *IEEE Transactions on Aerospace and Electronic Systems*, **57**(5), pp. 3383–3392 (2021).
15. Qazani, M.R.C., Asadi, H., Mohamed, S., et al. "An optimal washout filter for motion platform using neural network and fuzzy logic", *Engineering Applications of Artificial Intelligence*, **108**, p. 104564 (2022).
16. Elaraby, A. and Moratal, D. "A generalized entropy-based two-phase threshold algorithm for noisy medical image edge detection", *Scientia Iranica*, **24**(6), pp. 3247–3256 (2017).
17. Gerazov, B., Bailly, G., Mohammed, O., et al. "A variational prosody model for mapping the context-sensitive variants of functional prosodic prototypes", ArXiv Preprint arXiv: 180608685 (2018).
18. Qian, Y., Fan, Y., Hu, W., et al. "On the training aspects of Deep Neural Network (DNN) for parametric TTS synthesis", *IEEE International Conference on*

Acoustics, Speech and Signal Processing (ICASSP); Berlin, Germany, IEEE (2014).

19. Topaloglu, I. "Deep learning based convolutional neural network structured new image classification approach for eye disease identification", *Scientia Iranica* (2022) (In Press). DOI: 10.24200/sci.2022.58049.5537
20. Awwad, S., Tartory, R., Johar, M.G.M., et al. "Use of rhetoric and metaphorical expressions in Jordanian political discourse (speeches): an exploratory study", *Eurasian Journal of Applied Linguistics*, **7**(2), pp. 162–170 (2021).
21. Shamrat, F.J.M., Chakraborty, S., Billah, M.M., et al. "Bangla numerical sign language recognition using convolutional neural networks", *Indonesian Journal of Electrical Engineering and Computer Science*, **23**(1), pp. 405–413 (2021).
22. Malekmohammadi, A., Mohammadzade, H., Chamanzar, A., et al. "An efficient hardware implementation for a motor imagery brain computer interface system", *Scientia Iranica*, **26(1)** (Special Issue on: Socio-Cognitive Engineering), pp. 72-94 (2019).
23. Çevik, M. and Tabaru-Örnek, G. "Comparison of MATLAB and SPSS software in the prediction of academic achievement with artificial neural networks: Modeling for elementary school students", *International Online Journal of Education and Teaching*, **7**(4), pp. 1689–1707 (2020).
24. Uni, K. "Benefits of Arabic vocabulary for teaching Malay to persian-speaking university students", *Eurasian Journal of Applied Linguistics*, **8**(1), pp. 133–142 (2022).
25. Jafari, H.S. and Homayoonpoor, M.M. "Persian speech sentence segmentation without speech recognition", *Iranian Conference on Intelligent System* (ICIS); Tehran, Iran (2014).
26. Sharma, S. and Kumar, K. "ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks", *Multimedia Tools and Applications*, **80**(17), pp. 26319–26331 (2021).
27. Adane, K. and Beyene, B. "Machine learning and deep learning based phishing websites detection: The current gaps and next directions", *Review of Computer Engineering Research*, **9**(1), pp. 13–29 (2022). https://doi.org/10.18488/76.v9i1.2983
28. Rustamovich, S., Ilgiz, I., and Larisa, Y. "Development of an application for creation and learning of neural networks to utilize in environmental sciences", *Caspian Journal of Environmental Sciences*, **18**(5), pp. 595–601 (2020).
29. Al-masaeed, S., Alshareef, H.N., Johar, M.G.M., et al. "A study on educational research of artificial neural networks in the Jordanian perspective abstract", *Eurasian Journal of Educational Research*, **96**(96), pp. 281–301 (2021).
30. Gao, R., Du, L., and Yuen, K.F. "Robust empirical wavelet fuzzy cognitive map for time series forecasting", *Engineering Applications of Artificial Intelligence*, **1**(96), p. 103978 (2020).

Biographies

Ladan Khosravani Pour was born in Tehran, Iran in 1986. He received the BS degree in 2013, the MS degree in 2014, and the PhD degree in 2022, all in Electronic Engineering. She is currently an Associate Professor at the Electrical Engineering Department, Islamic Azad University-South Tehran Branch.

Ali Farrokhi was born in Tehran, Iran in 1964. He received the BS degree in 1990, the MS degree in 1993, and the PhD degree in 2003, all in Electronic Engineering. He is currently an Associate Professor at the Electrical Engineering Department of Islamic Azad University-South Tehran Branch. Dr. Farrokhi has published more than 50 journal and conference papers. He has published one book in Farsi and his research interests include low power circuit, analog and digital circuit, speech signal processing, and neural networks.