Original Article

The Recognition of Persian Phonemes Using PPNet

Saber Malekzadeh[1,2], Mohammad Hossein Gholizadeh[1], Seyed Naser Razavi[3], Hossein Ghayoumi Zadeh[1]

[1]Department of Electrical Engineering, Vali-e-Asr University of Rafsanjan, Rafsanjan, [2]Khazar University, Baku, Azerbaijan, [3]Department of Computer Engineering, University of Tabriz, Tabriz, Iran

Abstract

Background: In this paper, a novel approach is proposed for the recognition of Persian phonemes in the Persian consonant-vowel combination (PCVC) speech dataset. Nowadays, deep neural networks (NNs) play a crucial role in classification tasks. However, the best results in speech recognition are not yet as perfect as human recognition rate. Deep learning techniques show outstanding performance over many other classification tasks, such as image classification and document classification. Furthermore, the performance is sometimes better than a human. The reason why automatic speech recognition systems are not as qualified as the human speech recognition system, mostly depends on features of data which are fed to deep NNs. **Methods:** In this research, first, the sound samples are cut for the exact extraction of phoneme sounds in 50 ms samples. Then, phonemes are divided into 30 groups, containing 23 consonants, 6 vowels, and a silence phoneme. **Results:** The short-time Fourier transform is conducted on them, and the results are given to PPNet (a new deep convolutional NN architecture) classifier and a total average of 75.87% accuracy is reached which is the best result ever compared to other algorithms on separated Persian phonemes (like in PCVC speech dataset). **Conclusion:** This method not only can be used for recognizing mono-phonemes but it can also be adopted as an input to the selection of the best words in speech transcription

Keywords: *Persian consonant-vowel combination, Persian, PPNet, speech recognition, short-time Fourier transform*

Submitted: 24-Jun-2019 **Revised:** 10-Aug-2019 **Accepted:** 06-Oct-2019 **Published:** 25-Apr-2020

Introduction

Speech is the most important means of exchanging information among people. In the speech production process, a message that the speaker is going to produce is first formulated in the speaker's mind. In the second step, the message is converted to language code. It includes converting the text into a phoneme set that produces sounds according to the labels of the duration, loudness, and pitch contour. After selecting the desired language code, the speaker executed a set of neuromuscular commands to vibrate the vocal cords at the appropriate time and activate the vocal tract at the appropriate locations, creating a speech signal, and sending it out of the mouth. The speech signal reaches the ear or any other recognition system as it passes through the transmission channel, which may be the air or the speaker. The phoneme is actually the smallest unit of the speech, which includes several tens in each language.[1] In the fourth industrial revolution, smart machines have become an important part of our modern life, and this has encouraged the expectations of friendly interactions with them. It is the impetus for the ever-expanding development of machines that receive the human speech as input and respond appropriately to the input.

The speech as a way for communication way has witnessed the successful development of quite several applications using automatic speech recognition (ASR), including command and control, dictation, dialog systems for people with impairments, and translation. Research on ASR is still a challenging issue for using the speech as input.[1] However, the actual challenge is to create inputs and to use speech to control applications and access information. ASR – the recognition of the information of a speech signal and its transcription to a set of characters – has been the focus of research for more than five decades, achieving notable results. It is expected that the advances in speech recognition make a speech in any language, as the best input

How to cite this article: Malekzadeh S, Gholizadeh MH, Razavi SN, Zadeh HG. The recognition of persian phonemes using PPNet. J Med Signals Sens 2020;10:86-93.

Address for correspondence:
Dr. Mohammad Hossein Gholizadeh,
Department of Electrical Engineering, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran.
E-mail: gholizadeh@vru.ac.ir

Access this article online

Website: www.jmssjournal.net

DOI: 10.4103/jmss.JMSS_32_19

Quick Response Code:

method when the recognizers reach error rates under 5%. While digit recognition has already reached a rate of 99.6%,[2] the phoneme recognition has not gone far more than 83%.[3]

In any large-vocabulary ASR (LVASR) systems, the performance depends on phoneme recognizer more than language model. Thus, the research groups still work on developing better phoneme recognizer systems. The phoneme recognition is, in fact, a recurrent problem for the speech recognition community. Phoneme recognition is found in many applications. In addition to some typical LVASR systems,[4] it is found in applications related to language and speaker recognition, music identification, and also translation. The challenge of building efficient acoustic models starts with applying useful training algorithms to a suitable set of data. The dataset contains sound units which can be trained on training algorithms and depends on the detailed annotation of those units. The most datasets are not labeled at the exact phoneme level.[5]

The database collected in this field, called PCVC, is distinctive because it is labeled at the phoneme level. Furthermore, unlike other phoneme-based datasets,[6] PCVC contains just two phonemes in every sample, which makes the training and recognition process more efficient. For the extraction of consonants, as they are just pronounced before vowels, it is possible to separate them approximately. However, in this paper, to show the usability of phoneme recognition on PCVC, the recognition algorithms are examined on PCVC to get the best recognition results.

Conventionally, the phonetic features, along with the classification algorithms, such as support vector machine (SVM), are employed for diagnosis. Over the past two decades, the neural network (NN) structures with one or two hidden layers, as well as the combination of NNs with hidden Markov models (HMMs), have been widely used in speech recognition. However, the lack of both high-speed hardware and deep-learning NN methods limits the performance of NN structures. Thus, the HMMs are employed more than NNs. Nowadays, the advances in deep learning methods and the provision of high-speed processing hardware make it possible to use deep NNs in speech recognition. Literature study shows that different techniques applied for mispronunciation detection, posterior probability-based methods, classier-based methods, and deep learning-based methods.[7,8]

In the study by Widrow *et al.*,[9] real-time speech processing capabilities are examined using an NN consisting of three types of neurons. These neurons are based on a hybrid model and are capable of detecting the auditory frequency patterns such as vowels. The words are known as the sequences of sounds. According to the studies, although some machine vision algorithms are proposed for detection and classification, but, the manually definition of the unique features to achieve the desired clustering is not extensible for the other similar tasks. Besides, the deep level of feature extraction may not be achieved using conventional feature extraction methods. Thus, a deep convolution NN has been introduced to address such limitations. Furthermore, the final deep features are robust to many of the parameters in the data that are considered as noise. The utilization of deep NNs in speech systems with various speech datasets demonstrates the superiority of these networks over the Markov models.[10] In most research on the deep NNs application in the field of phonetics, first, the output of a trained static network for a moving window of the input frames sequence is obtained. Then, the graphic models, such as HMM or conditional random field, are combined with modeling the linear chain dependencies of the output sequence. In fact, in these structures, deep NNs are employed as the audio models, and graphical models are considered as phonetic models, each being trained individually.[11,12] To optimize the deep NN structures for the better coordination with the speech structure, and reducing the number of training parameters and computational time, the convolutional network is one of the proposed approaches.[13] The use of deep convolutional NN in identifying the phonemes of the Arabic language is also discussed.[14] In the proposed procedure, the features from different layers of the convolutional network are employed to train the methods including the k-nearest-neighbor, SVM, and NN. To evaluate the performance of the system, 28 Arabic phonemes are compared and the highest accuracy of 92.2% is achieved. It is also concluded that the proposed deep learning method performed better than the previous conventional techniques.

Based on the above studies, we propose the application of an optimized convolutional deep NN in this paper. This paper is organized as follows: Section II discusses the PCVC speech dataset, which is used and presented in phoneme recognition task for the first time. Section III proposes the preprocessing level with sound signal processing algorithms such as short-time Fourier transform (STFT) and phoneme extraction from PCVC speech dataset. Section IV describes the deep artificial NN applied for the classification of samples. Section V is dedicated to the conclusion, and the last section is acknowledgment.

Persian Consonant-vowel Combination Speech Dataset

To create a comprehensive database of phonemes, it is necessary to record some words. The words should be selected so that they eventually include all the phonemes. To provide appropriate patterns for each phoneme, it is necessary to delineate the boundary between the phonemes of the recorded words. There are practically various automated methods for this purpose. However, the phoneme border separation is usually performed manually. In fact, the boundary of the phonemes is accurately determined by observing the recorded speech signals in the time domain; in addition, it is determined by listening to them, especially focusing on the distance between the phonemes. However,

the STFT method is applied to improve the accuracy of phoneme separation.

This dataset contains 23 Persian consonants and 6 vowels, which is listed in Table 1. Table 1 includes the vowels and consonants, just like the dataset. There are 13 speakers, including six males and six females and one child. All sound samples are possible combinations of vowels and consonants (132 samples for each speaker). The sample rate of all 2-s speech samples is 48,000, which means that there are 48,000 audio samples (values) in every second. Each sound sample is a 2-s speech sample, which on average, a 0.5-s interval of each sample is speech, and the rest is silence.

For the testing process, 15% of samples are selected, and 85% of those are used in the training process.

Speech Signal Preprocessing

Phoneme extraction

As demonstrated in Figure 1, each sound sample is a 2-s audio wave of speech which ends with silence for at least 0.25 s. Then, consonants and vowels are pronounced consecutively. The intensity of silence is almost (not exactly) zero. These values (intensity of silence and consonants) can be employed to detect the vowels which have a higher intensity than silence. Thus, the vowels are parts of the speech whose related intensity is more than 0.25 of the maximum intensity of the sound sample. The value 0.25 is a suitable benchmark for the detection of vowels from other elements on PCVC dataset. This part of speech is sufficient to detect vowels in a sound sample. Each vowel sample is cut in 50 ms samples. In this paper, the aim is to recognize phonemes; therefore, as said before, the vowels are pronounced just after consonants. Thus, the 50 ms of speech before the vowels are almost selected as the consonant speech sample. Actually, this splitting method is used in PCVC speech dataset. However, in the real world, there are sentences for which there is no rule for how phonemes are gathered together in them.

Table 1: Phoneme list in Persian consonant-vowel combination dataset

Persian form	English form	Persian example
آ	A	لآ
ای	I	لای
وا	ʊ	وا
اَ	Æ	لوا
اِ	E	امس
اُ	O	ودرا
پ	P	اپ
ب	B	اب
ت	T	ات
د	D	وراد
چ	tʃ	وقاچ
ج	dʒ	وراج
ک	K	یراک
گ	G	یراگ
ف	F	ممطاف
و	V	ممهاو
خ	Kh	مرطاخ
غ	Gh	زاغ
س	S	زاس
ز	Z	راز
ش	ʃ	راش
ژ	ʒ	تکاژ
م	M	تکام
ن	N	یدان
ه	H	یداه
ل	L	مبال
ر	R	مبهار
ق	Q	یراق
ی	j	یرای

Adaptive noise cancellation

A recorded voice is contaminated by noise. It is proven that one of the best statistical models for the noise is the Gaussian model. The noise affects the phoneme recognition and should be suppressed to have a reliable recognition. An adaptive noise canceller (ANC) is employed to reduce the noise effect to have more precise results. It includes two inputs, primary and reference. As depicted in Figure 2, the primary input receives the signals from the signal source that is corrupted by the noise n, uncorrelated with the signal. The reference input receives a noise n_0, uncorrelated with the signal, but, correlated in some way with the noise n. The noise n_0 passes through a filter to produce an output that is a close estimate of primary input noise. The estimated noise is subtracted from the corrupted signal to produce an estimate of the main signal, the ANC system output.[15] For more reliable results, the filter depicted in Figure 2 is considered as an adaptive filter that automatically adjusts its own impulse response. Adjustment is accomplished using an algorithm that responds to an error signal which is dependent on the output of the filter. The ANC is employed to have the best fit in the least squares sense to the main

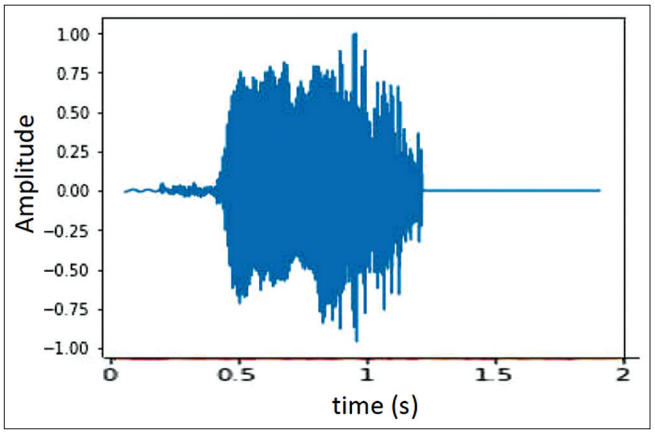

Figure 1: A two phonemes sample time-domain plot

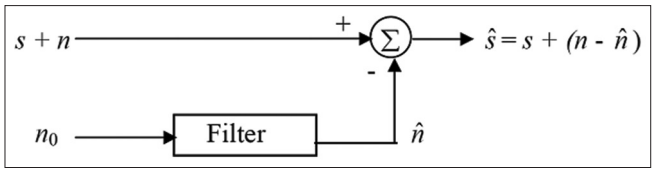

Figure 2: The structure of adaptive noise canceller

signal. It is performed by feeding the output of the system back to the adaptive filter and adjusting the filter through a least mean squares adaptive algorithm to minimize total system output power.[9]

Short-time Fourier transform

In this paper, the STFT algorithm is used to extract the features from sound samples. STFT is used as one of the best sound feature extraction algorithms for decades. STFT is a sound feature extraction algorithm that gives the ability to transform sound features from the temporal domain to the frequency domain and from the frequency domain to the time–frequency domain.[10] On time–frequency diagram, some features such as the shape of formants, distance, sound shocks, and also, the curvature of formants can be found that are the vital features for phoneme recognition. The mentioned features are the results of human local folds, lips, tongue, and teeth which are always creating patterns in the time–frequency domain in different shapes when phonemes are being pronounced.[11,12]

The STFT is commonly derived as:

1. Separating sound samples in fixed-size intervals
2. Applying Fourier transform on each sound interval
3. Assortment of frequencies in different frequency ranges
4. Initializing of each time–frequency domain point (rectangle) with suitable values based on their number of samples.[13]

In this paper, this algorithm is utilized to extract the spectral features of sound samples. For this purpose, the window length is 5 ms. Each window includes 150 frequency ranges. These parameters are identified as a suitable choice in tests. Note that the Mel-frequency cepstral coefficients (MFCC), STFT, and raw sound sample were tested in classification level on some of the samples, and the best results were achieved in STFT. Thus, the STFT is selected in this paper. STFT mathematically is written as:

$$STFT\{x(t)\} = X(\tau,\omega) = \int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-j\omega t}dt, \quad (1)$$

Where $w(t)$ is the window function, centered around by zero, and $x(t)$ is the signal to be transformed. As depicted in Figure 3, the function $X(\tau, \omega)$ is a complex function representing the phase and magnitude of the signal over time and frequency. The time axis is τ and the frequency axis is ω.[14]

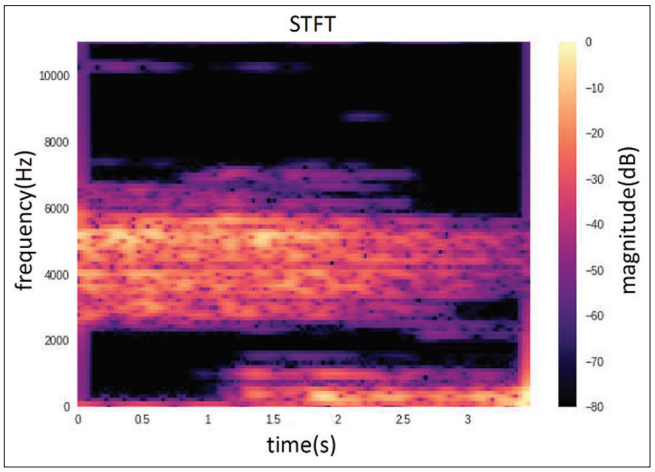

Figure 3: A phoneme sample time–frequency domain plot

Deep neural network

A deep NN is a kind of artificial NN including many layers to better perform the classification process. Generally, artificial NN s with more than three layers are called deep networks.[16] The convolutional NN is one of the most important deep learning methods in which multiple layers are trained in a reliable procedure. This method is completely efficient and is one of the common techniques in different applications of computer vision. An overview of convolutional NN architecture is shown in Figure 4.[17] Generally, a convolutional NN consists of three main layers: the convolutional layer, the pooling layer, and the fully connected layer. Different layers perform various tasks. There are two steps of feed-forward and back-propagation for training in a convolutional NN.

After preprocessing the data, it is given to the input of the first layer of convolution. Then, the data are convolved by the convolutional kernels of the first layer. Each convolution layer consists of three sublayers. In the first sublayer, the preprocessed data are convolved by the predefined convolutional kernels. The values and dimensions of the convolutional kernels are different depending on the type of the employed convolutional network. Furthermore, the output dimension can be larger, smaller, or the same as the input dimension. Multiple convolutional kernels can also be used.[17] Here, the convolution operator is an operator that is local and invariant with displacement. The convolution operator is mathematically shown as:

$$y_{i'j'k'} = \sum w_{ijk'} x_{i+i',i+j',k}, \quad (2)$$

In which W is the convolutional filter bank. In fact, the fourth-dimension is the filter number in the filter bank, and the filter itself is a three-dimensional (3D) weight mass. In other words, the first 3D convolutional filter slides over the 3D data, and in each position, the dot product is performed between the corresponding data, and the results of all of the multiplications are summed. Only one value of the

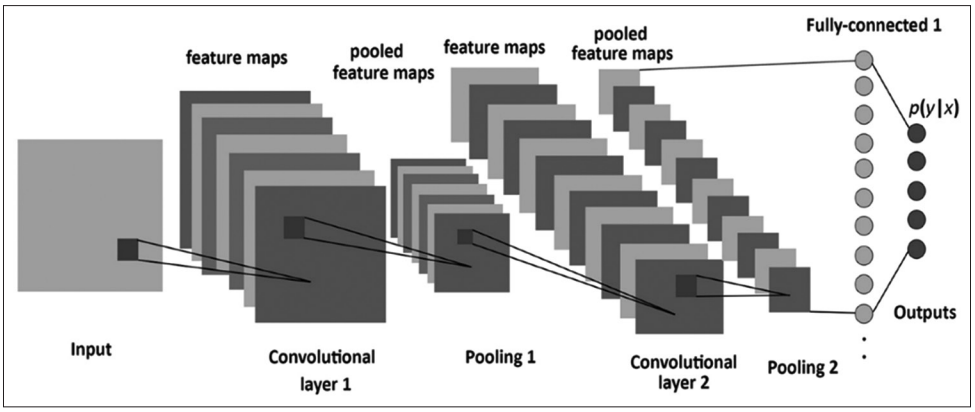

Figure 4: The structure of a convolutional neural network

pixel is obtained in each position. Thus, by sliding the filter number 1 over the whole data, a two-dimensional (2D) feature map is provided.[17] Likewise, the next filters are applied and the 2D feature maps are obtained again. By overlaying the feature maps in the third dimension, the final feature map is produced which is 3D. After applying the first convolutional sublayer, the output is given to the nonlinear sublayer. In this step, the nonlinear activator function is applied to the obtained values to reach the higher-level properties. The rectifier function is employed as the nonlinear function. In general, the deep convolution networks tend to use the rectifier function more than other nonlinear functions because the rectifier function is simply computed and does not involve a lot of computational resources. Furthermore, the utilizing of this function results in acceptable accuracy.[17] The rectifier function definition is as:

$$f(x)=\begin{cases} x & x>0 \\ 0 & else \end{cases}. \tag{3}$$

Finally, the output values are fed to the third layer for the merge operation. In the merge sublayer, the statistical summary of neighboring pixels around the central pixel replaces the central pixel value of the merge window. Merging makes the properties more stable and reduces the sensitivity to unwanted changes. The output data dimensions of this sublayer can be the same as the input data dimensions or have different dimensions. Therefore, if the dimension is reduced at the merge step, this sublayer results in holding the more valuable features and discarding the trivial features.[17] There are several types of merging, the most popular being max pooling, which is described as:

$$y_{ijk}=\max\left\{y_{i'j'k}:i\le i'<i+p, j\le j'<j+p\right\}. \tag{4}$$

Now, a convolutional layer is fully applied to the preprocessed image. Depending on the type of employed convolutional network, this step is repeated in a certain number. After the convolutional layers, one or more fully connected layers are used for the final feature mapping. A fully connected layer is exactly like a convolutional layer, except that the sparse interactions do not occur here, and just like traditional NNs, a complete connection is established between this layer and the layer before it. The output of the last layer is a 1D vector, and the number of members of this vector is equal to the number of the classification classes. This layer actually performs the classification problem. In all convolutional structures, the last fully connected layer is connected to a softmax layer. In fact, the softmax layer does the classifying on the convolutional networks. This layer contains a number of neurons equal to the number of clustering problem classes (there are 30 classes here) and is used for the final feature mapping and clustering operation.[17] Here, the output result is adopted to calculate the network error to adjust the network parameters and the network training. Thus, the network output is compared with the correct response by employing an error function, and then the error is calculated. The empirical error is obtained as:

$$L(w)=\frac{1}{n}\sum_{i=1}^{n} l\left(Z_i; f\left(X_i; w\right)\right), \tag{5}$$

In which $l(Z;\hat{Z})$ is a loss function that determines the amount of penalty when wrong predicting $\hat{Z}$ instead of Z. The next step starts based on the calculated error of the back-propagation step. At this step, the gradient of each parameter is calculated according to the chain rule, and all the parameters are changed according to the effect they have on the error generated in the network:

$$w^{t+1}=w^{t}-\rho_t\frac{\partial f}{\partial w}\left(w^{t}\right). \tag{6}$$

After updating the parameters, the next feed-forward step begins. After repeating a suitable number of these steps, the network training ends.

Results

Now, the evaluation criteria and the results are presented. The implementations are performed in the MatLab programming language, on the high-speed HPC processing system including several computing clusters that the

integration between them leads to task management in a focused manner. The hardware specifications are number of nodes 2, ×2 NVIDIA® Tesla K80 GPUs graphics card, 8 × 16 GB memory, and ×2 Intel®Xeon®E5-2695v3 at 2.30GHz processor.

Training process

To train phoneme speech samples, PPNet (a new architecture as a convolutional artificial NN) is used that the related structure is expressed in Table 2.

Six convolutional layers are used all with stride (1, 1), kernel size (3, 3), and Relu activation function. The first two convolution layers have 32 kernels. The second two ones have 64 kernels, and the third two ones have 128 kernels.

Convolution layer instructions are as follows:

- Accepts a volume of size W1 × H1 × D1 W1 × H1 × D1
- Requires four hyper-parameters:
  - Number of filters KK
  - Their spatial extent FF
  - The stride SS
  - The amount of zero padding PP.
- Produces a volume of size W2 × H2 × D2 W2 × H2 × D2 where:
  - W2 = (W1−F+2P)/S+1W2 = (W1−F+2P)/S+1
  - H2 = (H1−F+2P)/S+1H2 = (H1−F+2P)/S+1 (i.e., width and height are computed equally by symmetry)
  - D2 = KD2=K
- With parameter sharing, it introduces F·F·D1F·F·D1 weights per filter, for a total of (F·F·D1)·K(F·F·D1)·K weights and KK biases
- In the output volume, the dd-th depth slice (of size W2 × H2W2 × H2) is the result of performing a valid convolution of the dd-th filter over the input volume with a stride of SS and then is offset by dd-th bias.

A batch normalization layer is used after the first convolutional layer. Furthermore, six dropout layers are used to avoid over-fitting. In addition, three max-pooling layers are used to help the network learn data orientation and also general features. The batch size is 16, and the number of epochs is 50. Input shape is 100 × 150.[18]

Testing process

In the stochastic analysis, the F1 criterion or F1_score is the measure of accuracy in binary classification. It considers the precision and recall of the test to compute the measure. The precision is the number of the correct positive results divided by the number of all positive results returned by the classifier. Furthermore, the recall is the number of correct positive results divided by the number of all relevant samples (all samples that should be identified as positive):

- Precision = True positive/(true positive + false positive)
- Recall = True positive/(true positive + false negative).

Table 2: Convolution neural network model summary

Layer	Output shape	The Learnables
Conv2d_1 (conv2d)	(None, 100, 150, 32)	320
Batch_normalization_1	(None, 100, 150, 32)	128
Activation_1 (activation)	(None, 100, 150, 32)	0
Dropout_1 (Dropout)	(None, 100, 150, 32)	0
Conv2d_2 (conv2d)	(None, 100, 150, 32)	9248
Max_poolong2d_1	(None, 50, 75, 32)	0
Conv2d_3 (conv2d)	(None, 50, 75, 64)	18496
Dropout_2 (Dropout)	(None, 50, 75, 64)	0
Conv2d_4 (conv2d)	(None, 50, 75, 64)	36928
Max_poolong2d_2	(None, 25, 37, 64)	0
Conv2d_5 (conv2d)	(None, 25, 37, 128)	73856
Dropout_3 (Dropout)	(None, 25, 37, 128)	0
Conv2d_6 (conv2d)	(None, 25, 37, 128)	147584
Max_poolong2d_3	(None, 12, 18, 128)	0
Flatten_1 (Flatten)	(None, 27648)	0
Dropout_4 (Dropout)	(None, 27648)	0
Dense_1 (Dense)	(None, 1024)	28312576
Dropout_5 (Dropout)	(None, 1024)	0
Dense_2 (Dense)	(None, 128)	131200
Dropout_6 (Dropout)	(None, 128)	0
Dense_3 (Dense)	(None, 30)	3870

CNN – Convolution neural network

Table 3: Results table

	Precision	Recall	F1_score	Support
0	0.78	0.58	0.67	12
1	0.70	0.58	0.64	12
2	0.78	0.64	0.70	11
3	0.59	0.83	0.69	12
4	0.54	0.64	0.58	11
5	0.58	0.92	0.71	12
6	0.75	0.69	0.72	13
7	0.80	0.73	0.76	11
8	0.91	0.67	0.77	15
9	0.67	0.50	0.57	16
10	0.62	0.80	0.70	10
11	0.69	0.69	0.69	13
12	0.79	0.69	0.73	16
13	0.58	0.78	0.67	9
14	0.74	0.88	0.80	16
15	0.60	0.50	0.55	12
16	0.62	0.56	0.59	9
17	0.77	0.77	0.77	13
18	0.54	0.50	0.52	14
19	0.75	0.67	0.71	9
20	0.75	0.90	0.82	10
21	0.73	0.67	0.70	12
22	0.78	0.78	0.78	9
23	0.90	1.00	0.95	9
24	1.00	1.00	1.00	14
25	1.00	1.00	1.00	13
26	1.00	0.94	0.97	16
27	1.00	1.00	1.00	9
28	1.00	1.00	1.00	13
29	1.00	1.00	1.00	9

Table 4: The comparison between the conventional and proposed methods

	The conventional method	Recognition percentage of the conventional method	Recognition percentage of the proposed method
Vowel recognition	Reference[20]	94	98
Vowel recognition	Reference[21]	71	98
Phoneme recognition	Reference[22]	69	76

The F1 criterion is the harmonic mean of the precision and recall, where an F1 measure reaches the best value at 1 (perfect precision and recall) and the worst at 0. The F1 criterion is as follows:[19]

$$F = 2\frac{precision.recall}{precision + recall} \quad (7)$$

Test data are selected fully randomly from training data with a guaranty of selecting between 8 and 16 from each class of data for the test set. Results for the best-selected phonemes are shown in Table 3. The overall F1_score result is 75.78% of recognition accuracy for all phonemes. In Table 3, the first 23 phonemes are consonants with the order of Table 1, the rest six phonemes are vowels again like the order in Table 1, and the last phoneme is silence.

The arrangement of data when training the convolutional network has an impact on how the network is trained, just like other NNs. The basis for updating weights is based on the calculation of the error between the predicted value and the expected value. Thus, by growing the network training data, the model fits more into the training dataset. If the data given to the network sequentially do not all belong to a class, the network adapts itself during training so that it can better distinguish the data of those classes. Thus, before transmitting data to the convolutional network input, their ordering is disrupted, and all data related to one class are not entered sequentially.

The percentage of phoneme recognition, based on F1 criterion, is compared in Table 4. Note that the datasets employed in Table 4 have more vowels compared with the PCVC set.

In addition, the proposed method is compared with conventional algorithms. For this purpose, SVM algorithms and VGG16 deep artificial NN algorithms are performed on PCVC data samples, along with two preprocessing techniques STFT and MFCC. The results are depicted in Table 5 based on F1 criterion.

Table 5: The comparison between the conventional and proposed classifiers

Algorithm	The preprocessing technique	
	STFT	MFCC
SVM	53	45
VGG16	65	58

STFT – Short-time Fourier transform; SVM – Support vector machine; MFCC – Mel-frequency cepstral coefficients, VGG16 – Visual geometry group 16

Conclusion

A continuous speech recognition system should be designed and built so that it is capable of detecting natural speech efficiently. It must provide some conditions such as an identification that is independent of the speaker, or it should include all phonemes. In this paper, a new method is proposed for the recognition of phonemes in Persian language on PCVC. This method can be used not only for recognizing mono-phonemes, but also it can be adopted as an input to the selection of the best words in speech transcription. Based on the results shown in Table 3, the capability of the proposed vowels and consonants recognition system in predicting the phonemes is more efficient than the other phoneme recognition methods proposed in the conventional approaches.

Acknowledgment

So many thanks to those helped us to develop PCVC dataset especially speakers: Farideh Jabraili, Hedayat Malekzadeh, Hamed Afjuland, Mohammad Ataeizadeh, Tahereh Salari, Alireza Aghaei, Parisa Seyfpour, Sahel Soltani, and Mina Bayarash. Also special thanks to Prof. Beigi for their good information that helped us in this project.

Financial support and sponsorship

None.

Conflicts of interest

There are no conflicts of interest.

References

1. Li J. Soft Margin Estimation for Automatic Speech Recognition. The PHD Dissertation: Georgia Institute of Technology; 2008.
2. Janet MB, Deng L, Glass J, Khudanpur S, Lee CH. Developments and directions in speech recognition and understanding part 1 [dsp education]. IEEE Signal Process Mag 2009;26:75-80.
3. Mohamed AR, Dahl GE, Hinton G. Acoustic modeling using deep belief networks. IEEE Trans Audio Speech Lang Process 2011;20:14-22.
4. Morris J, Fosler-Lussier E. Conditional random fields for integrating local discriminative classifiers. IEEE Trans Audio Speech Lang 2008;16:617-28.
5. Carla L, Fernando P. Phoneme Recognition on the Timit Database. Speech Technologies; 2011.
6. Graves A, Mohamed AR, Hinton G, editors. Speech Recognition with Deep Recurrent Neural Networks. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE; 2013.

7. Gao Y, Xie Y, Cao W, Zhang J, editors. A Study on Robust Detection of Pronunciation Erroneous Tendency Based on Deep Neural Network. Sixteenth Annual Conference of the International Speech Communication Association; 2015.
8. Hu W, Qian Y, Soong FK, Wang Y. Improved mispronunciation detection with deep neural network trained acoustic models and transfer learning based logistic regression classifiers. Speech Commun 2015;67:154-66.
9. Widrow B, Glover JR, McCool JM, Kaunitz J, Williams CS, Hearn RH, *et al*. Adaptive noise cancelling: Principles and applications. Proc IEEE 1975;63:1692-716.
10. Ghoraani B, Krishnan S. Time – Frequency matrix feature extraction and classification of environmental audio signals. IEEE Trans Audio Speech Lang Process 2011;19:2197-9.
11. Maue-Dickson W, Dickson D. Anatomical and Physiological Bases of Speech. Butterworth: Heinemann; 1982.
12. Williams AL, McLeod S, McCauley RJ. Interventions for Speech Sound Disorders in Children. Education Resources Information Center: Brookes Publishing, 1st Ed.; 2010.
13. Beigi H. Fundamentals of Speaker Recognition: 1st Ed., Springer; 2011.
14. Jacobsen E, Lyons R. The sliding DFT. IEEE Signal Process Magaz 2003;20:74-80.
15. Kalamani M, Valarmathy S, Krishnamoorthi M. Adaptive noise reduction algorithm for speech enhancement. World Academy of Science, Engineering and Technology. Int J Comput Control Quantum Inf Eng 2014;8:1014-21.
16. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436-44.
17. Albelwi S, Mahmood A. A framework for designing the architectures of deep convolutional neural networks. Entropy 2017;19:242.
18. Karpathy A, Toderici G, Shetty S, Leung T, Sukthankar R, Fei-Fei L, editors. Large-Scale Video Classification with Convolutional Neural Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2014.
19. Olson DL, Delen D. Advanced Data Mining Techniques. Springer Science & Business Media; 2008.
20. Nazari M, Sayadiyan A, Valiollahzadeh SM, editors. Speaker-Independent Vowel Recognition in Persian Speech. 2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications. IEEE; 2008.
21. Sadeghi VS, Yaghmaie K. Vowel recognition using neural networks. Int J Comput Sci Netw Secur 2006;6:154-8.
22. Tavanaei A, Manzuri MT, Sameti H, editors. Mel-Scaled Discrete Wavelet Transform and Dynamic Features for the Persian Phoneme Recognition. 2011 International Symposium on Artificial Intelligence and Signal Processing (AISP). IEEE; 2011.

BIOGRAPHIES

Saber Malekzadeh has received his B.Sc. degree in Computer Science from University of Tabriz, Tabriz, Iran, in 2015 and his M.Sc. degree in Computer Science from Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran, in 2018. He is currently a lecturer at Khazar University, Baku, Azerbaijan. His research interest includes Artificial Intelligence, especially deep learning.

Email: smalekzadeh@khazar.org

Seyed Naser Razavi has received his B.Sc. degree in Computer Engineering from Petroleum University of Technology, Ahvaz, Iran, in 2001, his M.Sc. degree in Computer Engineering from Iran University of Science and Technology, Tehran, Iran, in 2003 and his PhD degree in Biomedical engineering from Iran University of Science and Technology, Tehran, Iran, in 2011. He is currently an assistant professor in University of Tabriz, Tabriz, Iran. His research interests include Artificial Intelligence, Deep learning and machine learning.

Email: n.razavi@tabrizu.ac.ir

Mohammad Hossein Gholizadeh has received the B.Sc. degree from Isfahan University of Technology, Isfahan, Iran, in 2007, the M.Sc. and Ph.D. degrees from Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran, in 2010 and 2015, all in electrical engineering. He is currently an Assistant professor with the Electrical Engineering Department, Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran. His research interests are image and voice processing. He is currently involved with system identification and wireless communications.

Email: gholizadeh@vru.ac.ir

Hossein Ghayoumi Zadeh has received his B.Sc. degree in Electronic Engineering from Shahid Rajaee Teacher Training University, Tehran, Iran, in 2008, his M.Sc. degree in Electronic Engineering from Hakim Sabzevari University, Sabzevar, Iran, in 2011 and his PhD degree in Biomedical engineering from Hakim Sabzevari University, Sabzevar, Iran, in 2016. He is currently an assistant professor in electrical Engineering Department of Vali-e-Asr University of Rafsanjan, Rafsanjan, Iran. His research interests include medical image analysis, Neural Network and Thermography.

Email: h.ghayoumizadeh@vru.ac.ir