Journal of Engineering and Applied Sciences 15 (7): 1664-1670, 2020
ISSN: 1816-949X

Cascade Deep Neural Networks Classifiers for Phonemes Recognition

Mohammad Smit and Abdel-Nasser Al-Assimi
Department of Communications Engineering, Higher Institute for Applied Science
And Technology, Damascus, Syria

Abstract: In the last few years, deep neural networks have taken the problem of automated voice recognition to a whole new level of accuracy. Where it provided the highest recognition rates whether on words or on phonemes. Voice recognition problem represents the first phase of automated speech recognition systems. In this research, we introduce the recognition of phonemes based on deep neural networks using the Convolutional Neural Network 'CNN'. We will discuss two approaches of recognition, the direct approach by recognizing the phonemes using a single classification phase by obtaining the correct phonemes directly through the input. The second proposed approach uses several phases of classification by taking into account the types of phonemes and their classes (vowels, semi-vowels, explosive, etc.). In both approaches, we rely on the mel spectrogram transform where the acoustic signal is converted into a two-dimensional matrix within the frequency domain, this matrix is then inserted as the input of the deep neural network. We tested the proposed classifier on TIMIT database, obtained 57% accuracy in the direct approach and a higher accuracy of 61% using our proposed approach.

Key words: Deep neural network, phonemes, TIMIT, accuracy, approach

INTRODUCTION

Speech is considered the most natural way of communication among human beings and it's what they use express their thoughts and feelings. Therefore, human speech has been under studying for many years which led to the emergence of many techniques to facilitate communication between humans and the machine. These techniques are used to process human voices or interact with humans (Pradeep and Rao, 2016).

The most important areas of research are speech synthesis, speech coding, speaker identification and authentication and Automatic Speech Recognition (ASR) (Zarrouk and Benayed, 2016). In this research, we focus on ASR which is used in many practical applications that aim to facilitate human-machine communication. One of the most popular applications nowadays for speech recognition systems is the voice-driven personal assistant such as Siri and Cortana.

The voice-controlled interface provides a more natural communication approach compared tot he old one which requires a keyboard or mouse, thus, allowing to make a call despite the preoccupation of hands and eyes, as in the case of driving which leads to more safety. The automatic speech recognition is also used in car navigation system to determine its destination.

Phonemes are the smallest units of intelligible sound and phonetic spelling is the sequence of phonemes that a word comprises. Phoneme classification is inherently complex for two reasons. First, the number of possible phonemes is at least 107, based on the International Phonetic Alphabet (IPA). Therefore, this problem is a many class classification task for time-series data. Second, phonemes suffer from variability in speakers, dialects, accents, noise in the environment and errors in automatic segmentation.

Phonemes recognition is an essential part of ASR (Zarrouk and Benayed, 2016). The development of the Phonemes recognition system and improving its performance leads to the development of ASR. The field of phonemes recognition is divided into four stages: initial processing, feature extraction, classification, linking sequenced phonemes and word conclusion. We will focus in this research on the first three stages.

Literature review: Most of the existing works on phonemes classification are based on the manually labelled dataset from linguistic data consortium named TIMIT. There is along chain of works on TIMIT dataset that use a variety of techniques and report classification performance on standard test set in the corpus.

One of the earliest technique used with TIMIT is HMM/GMM (Fauziya and Nijhawan, 2014). GMM efficiently processes the vectors of input features and estimates emission probabilities for each HMM state. HMM efficiently normalizes the temporal variability present in speech signal.

Corresponding Author: Mohammad Smit, Department of Communications Engineering, Higher Institute for Applied Science And Technology, Damascus, Syria

There are some other techniques used with TIMIT as Discrete Wavelet Transform 'DWT' (Hamooni *et al*., 2016) and Support Vector Machine 'SVM' (Yousafzai *et al*., 2010).

Artificial neural networks 'ANNs' can learn much better models of data laying on the boundary condition. One of the first phoneme recognition system based on neural network was time delay neural network. At the same time, the hybrid HMM/ANN architecture approach was developed, leading more scalable systems.

Deep neural networks 'DNN' as acoustic models tremendously improved the performance of ASR systems. DNNs have many hidden layers with a large number of nonlinear units and produce a very large number of outputs. Sreenivasa by Pradeep and Rao (2016) used HMM/DNN based on Mel Frequency Cepstrum Coefficients 'MFCC'. Recently, the CNN approach has been found to yield good performance in phoneme recognition, based on spectrogram of the phonemes (Malekzadeh *et al*., 2018).

Convolutional neural network CNN: Convolutional neural networks are a special kind of feed forward neural networks which is derived from biological processes in the visual lobe where it is considered a solution too many problems of computer vision and artificial intelligence (Krizhevsky *et al*., 2012).

In recent years, convolutional neural networks have gained considerable importance but its history dates back to the 1980's. Back then models of these networks were designed, specifically, for processing multidimensional matrices. By Hubel and Wiesel (1959) were examining the cortex of the cat when they discovered that its receptive field includes sub-regions that were overlapping on each other to cover the whole visual field, these layers act as filters that process input images which are then passed to subsequent layers.

In 1998, LeCun Yann and Joshua Bagnio attempted to depict neurons in the visual cortex of a cat as a form of artificial neural network that lead to the establishment of the first convolutional neuronal network. LENET one of the first convolutional neural networks that helped drive deep learning, this pioneering work was named LENET5 by LeCun Yann after many unsuccessful attempts. At that time, the LENET structure was primarily used for character recognition tasks such as reading postal codes, numbers and so on (Hubel and Wiesel, 1959).

By Krizhevsky *et al*. (2012) used CNN to win the image net classification challenge where he trained CNN on thousands of images and succeeded in passing the testing stage by classifying the images within reasonable error. In the meantime, CNN developed significantly and was later adopted to solve many computer vision related issues.

The main structure of CNN: The convolutional neural network is generally, made up of several different layers, each of which has its own function. According to LENET5, the main layers of any neural network can be classified in four stages: convolutional, pooling, flatten layer and the multilayer neural network.

Convolutional: Convolutional is the backbone of CNN and consists of several layers. The output of this process is the map feature which reflects the response of the filters to a specific pattern in the image through the weights of each filter. The output of each filter is a two-dimensional matrix filter that represents the filter's response. Filter weights are determined during network training.

The features map consists of several channels each one is the result of a filter. The dimensions of these channels are related to the dimensions of the input matrix and the dimensions of the filter as well as the following two factors:

Stride: Represents the number of elements for which the filter is displaced following each process.

Padding: We extend the matrix by either adding zeros to extend the boundary of the matrix or duplicating the values of the array ends. Thus, the entire matrix is utilized in the filtering process and no information is lost on the matrix ends.

The convolution function is summarized by extracting the attribute's beam. First, the network learns to discover simple features such as edges which in turn are used in the second layer to discover simple shapes, these shapes are then used to discover the higher-level attributes in the higher layers, so, the more layers of convolution the higher the level of attributes you learn.

Pooling: After the application of the activation functionon the features map, we reduce its dimensions in a way that retain information through the aggregation process. This is done in several ways, the most important of which is max pooling where each window is mapped (a group of adjacent elements) with a single element representing the highest value within this window.

The output of the aggregation process is a features map that has the same depth but varies in width and height. The aggregation process has several advantages:

- Reduce the dimensions of the features map and the number of variables and calculations in the network
- Makes the network resistant to a slight change or distortion in the input matrix

Flatten layer: After passing through the previous two layers (and for several phases), we then form the output of

the two stages a beam that fits the input of the neural networks to be entered into the last stage of the CNN algorithm.

Multilayer neural network phase: The features map does not necessarily have to be understood by humans but for a network it represents a code for a particular class.

After the extraction of the attributes, a classifier is used to classify those attributes using feed forward neural networks whose input is a beam of the features map after the aggregation phase and whose output is the row to which the features map belongs.

TIMIT speech dataset: An audio database created in collaboration with Texas Instrument (TI) and Massachusetts Institute of Technology (MIT), TIMIT contains 6,300 recorded audio sentences (16 kHz) of 630 speakers (men, women and children) (Lopes and Perdigao, 2011).

TIMIT consists of 61 phonemes but they are grouped and reduced to 39 phonemes. We chose the TIMIT database for two reasons. First, most audio studies and research rely on TIMIT to test its performance.

The second is that it has been cut to the level of phonemes manually (it contains the beginning and end of each phoneme) which gives more accuracy in the collection of phonemes.

Mel spectrogram: It represents an acoustic time-frequency representation of a sound by compute the power spectral density P(f, t).It is sampled into a number of points around equally spaced times t_i and frequencies f_j (on a mel frequency scale) (Xie *et al*., 2018). The mel frequency scale is defined as:

$$\text{Mel} = 2595 \times \log 10\left[\frac{1+f}{700}\right] \quad (1)$$

where, f is related to the common linear frequency in hertz. In this algorithm, the audio input is first buffered into frames of 'WindowLength' number of samples. The frames are overlapped by 'OverlapLength' number of samples. A periodic hamming window is applied to each frame and then the frame is converted to frequency-domain representation with 'FFTLength' number of points. The frequency-domain representation can be either magnitude or power, specified by 'SpectrumType'. Each frame of the frequency-domain representation passes through a mel filter bank. The spectral values output from the mel filter bank are summed and then the channels are concatenated so that each frame is transformed to a 'NumBands-element' column vector. Figure 1 illustrates the algorithm of compute mel spectrogram. Figure 2 and 3 show examples of mel spectrogram transform.

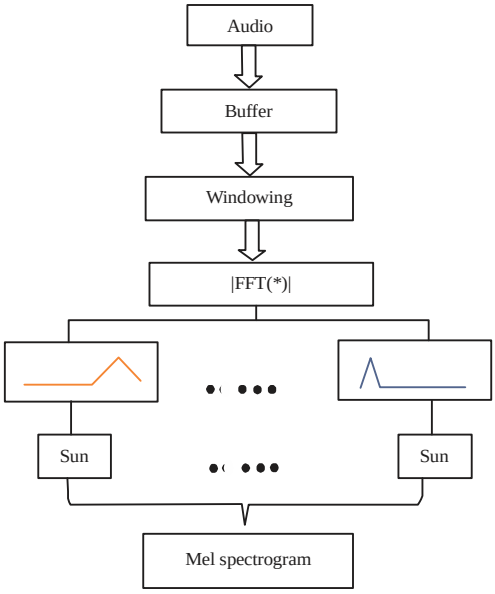

Fig. 1: Mel spectrogram

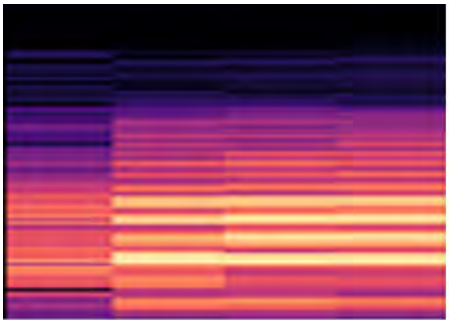

Fig. 2: Mel spectrogram transform for 'aa'

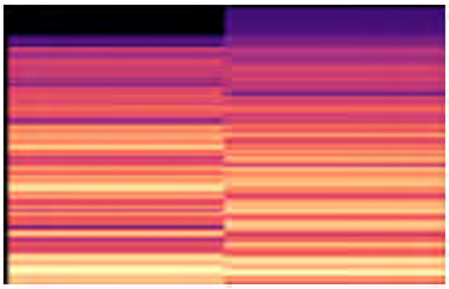

Fig. 3: Mel spectrogram transform for 'b'

Work algorithm: Figure 4 illustrates the work algorithm used in this study:

- Phonemesare initially collected and sorted from the TIMIT database (each set of segments relating to each phonemes is placed in a single folder)
- Perform a mel-spectrogram conversion on each Phonemes and save the results (training and test data)

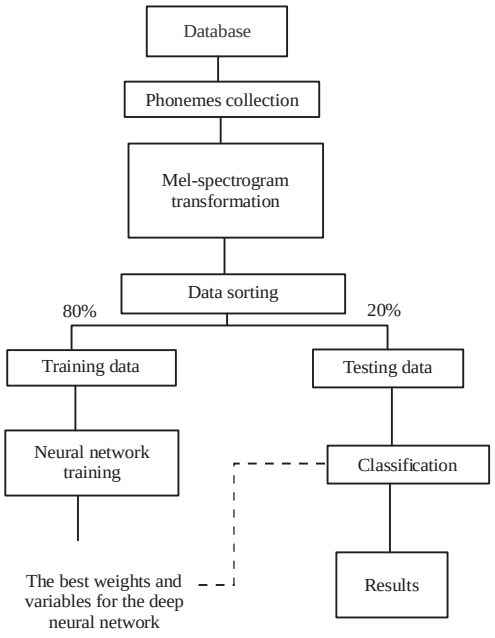

Fig. 4: Work algorithm

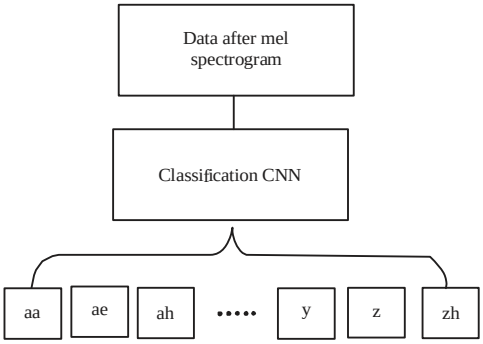

Fig. 5: Direct classification algorithm

- Train the deep neural network on training data
- Testing trained deep neural network on test data
- Collect the results and compare with previous studies

Direct approach: In this approach, the phonemes are classified in a single-phaseusing CNN where one deep neural network is used, the input of which is one of the sound segments related to a phoneme (after performing mel-spectrogram transformation on this segment) and its output is the corresponding phoneme for that audio segment (Malekzadeh *et al*., 2018). Figure 5 illustrates this approach.

MATERIALS AND METHODS

The proposed approach: In this approach, the phonemes are classified in several phases using CNN Fig. 6. The classification phases depend on the distribution of phonemes as signals. In the first phase, the classification is made among the vowel phonemes (periodic and semi-periodic signals) and constant phonemes (non-periodic signals). In the second phase, each of the previous two classes is classified into smaller secondary classes where the vowels phonemes are classified into three classes (vowels, semi-vowels, diphthongs) and constant phonemes are classified into three classes (fricatives, nasals, plosives).

In the third phase, each secondary class is aligned to the corresponding phonemes (this phase is the same as the direct approach but applied on secondary rows). Table 1 shows that each phonemes belongs to the corresponding secondary class.

Table 1: Distribution of phonemes on the classes (Glackin *et al*., 2018)

Secondary class	Phonemes
Plosives	b d g p t k jh ch
Fricatives	s sh z f th v dh hh
Nasals	m n ng
Semi-vowels	l r er w y
Vowels	iy ih eh ae aa ah uh uw
Diphthongs	ey aw ay oy ow

RESULTS AND DISCUSSION

We have used Python which is one of the most widespread programming languages with support for neural networks what distinguish this language is the constant and continuous support of developers (libraries and classes).

To deal with this language we chose a program called PyCharm, a development program that supports Python language. We relied on GPU for training which has a great ability to handle matrixes and repetitive operations (Yu *et al*., 2019).

In the direct approach case, we obtained a 57% accuracy in total. Figure 7 shows the confusion matrix of this approach and we can see in Table 2 the accuracy related to each phoneme. Using our proposed approach, we got the following results: in the first classification phase (between vowels and constant phonemes), the classification accuracy was 96%. Figure 8 shows the confusion matrix of first classification phase.

In the second classification phase (between types of vowels and types of constants phonemes), the classification accuracy was 84 and 88%, respectively. Table 3 shows the results of this classification phase. Figure 9 shows the confusion matrix of this classification phase.

From Fig. 9, we notice that we can separate the phonemes that belong to different classes more accurately than the phonemes that belong to the same class. In the third classification phase: each class (vowels, semi-vowels, diphthongs, plosives, nasals and fricatives)

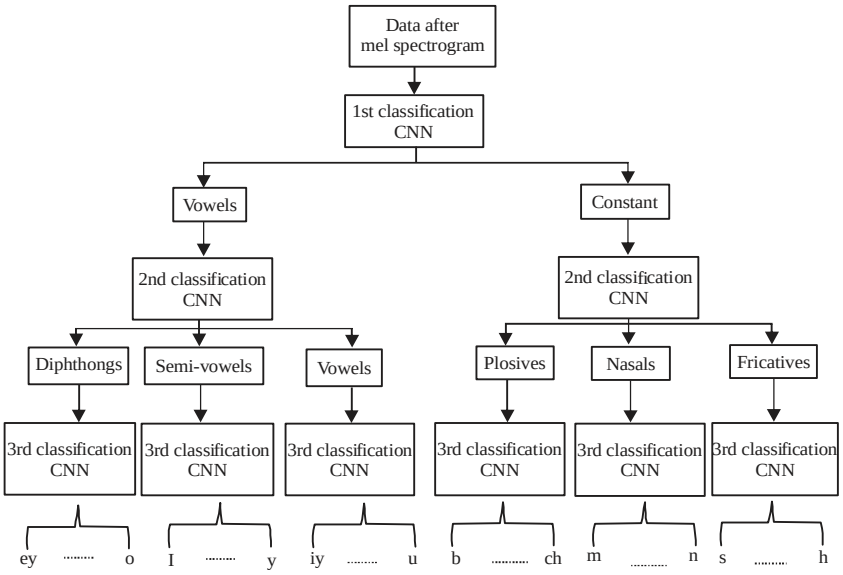

Fig. 6: Proposed classification algorithm

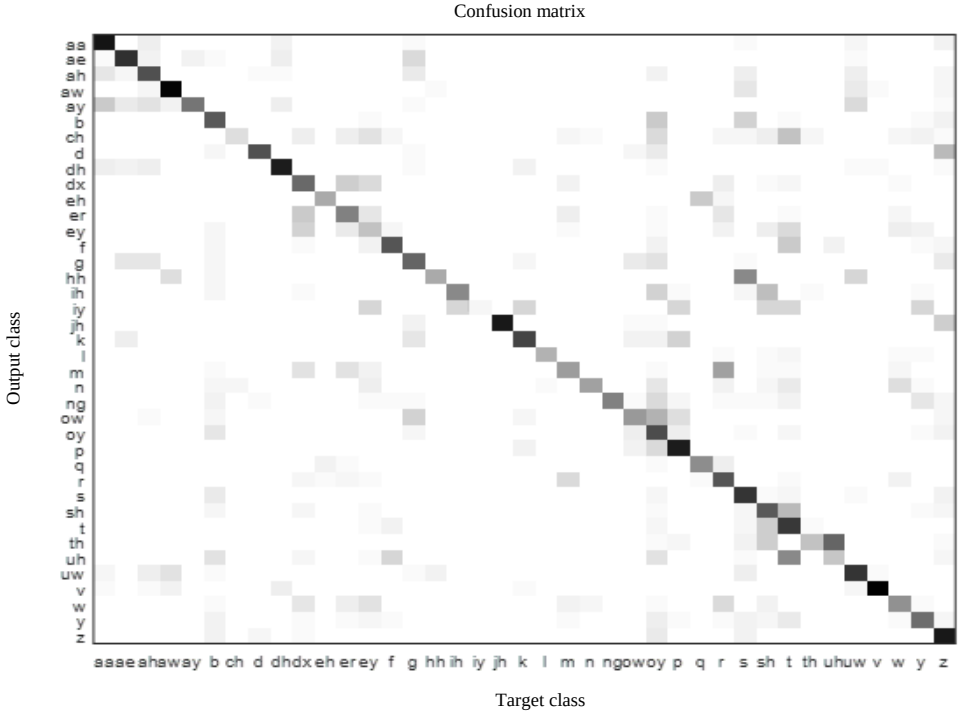

Fig. 7: Confusion matrix of direct approach

is aligned to the corresponding phonemes. Table 4 shows the results of this classification phase: we can find the overall accuracy using in Eq. 2:

$$\text{Accuracy} = \frac{\text{VCacc}\times[\text{Vacc}\times(\text{VVacc}+\text{VSacc}+\text{VDacc}\times(\text{Cpacc}+\text{CNacc}+\text{CFacc})]}{\text{No. of classes}} \quad (2)$$

Where:

VCacc : Classification accuracy between vowels and constants

Vacc : Classification accuracy between vowels types (vowels, semi-vowels and Diphthongs)

VVacc : Classification accuracy between phonemes belong to vowels class

VSacc : Classification accuracy between phonemes belong to semi-vowels class
Vdacc : Classification accuracy between phonemes belong to diphthongs class
Cacc : Classification accuracy between constants types (plosives, nasals and fricatives)
CPacc : Classification accuracy between phonemes belong to plosives class
CNacc : Classification accuracy between phonemes belong to nasals class
CFacc : Classification accuracy between phonemes belong to fricatives class

From Eq. 2, we obtained the overall 61.18%. We note from the results that our proposed approach gives better results compared to the first approach because in the direct approach, we rely on a single classification phase using CNN while in our proposed approach there are

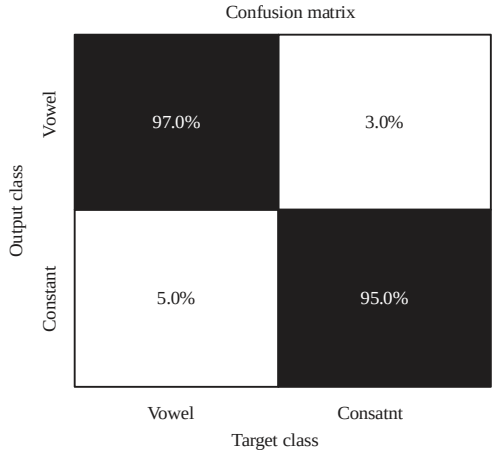

Fig. 8: Confusion matrix of first classification phase 'proposed approach'

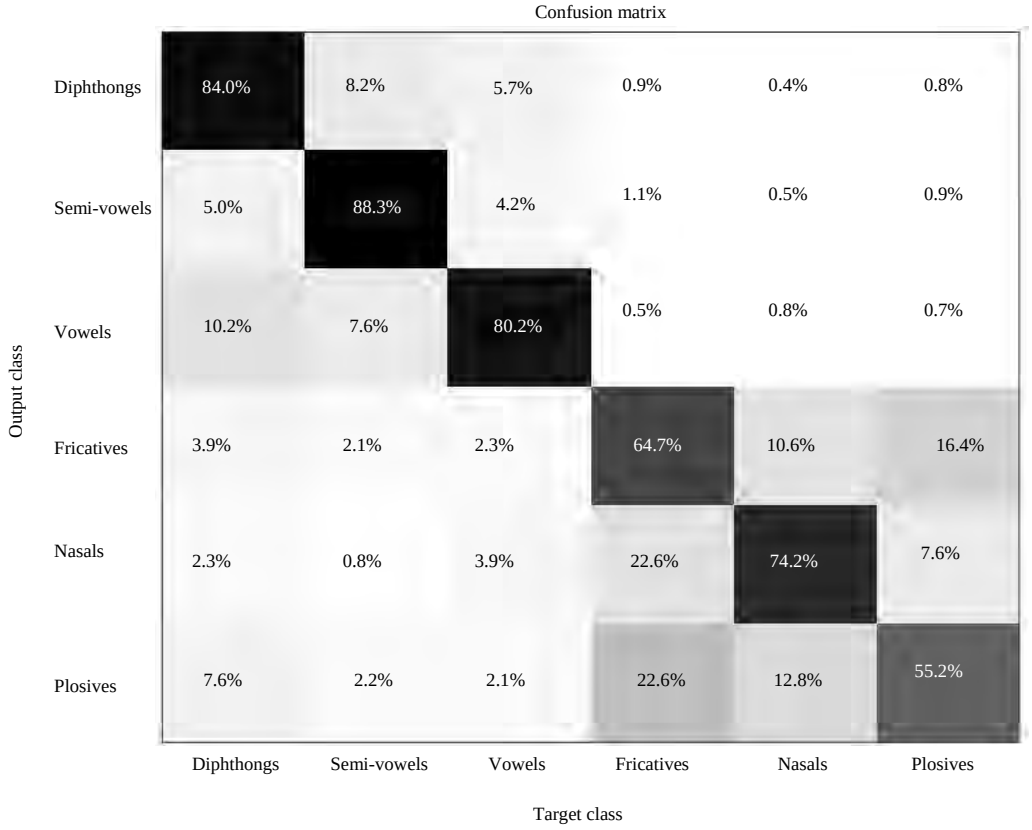

Fig. 9: Confusion matrix of second classification phase 'proposed approach'

Table 2: Accuracy of all phonemes 'Direct approach'

Phonemes	aa	ae	ah	aw	ay	b	ch	d	dh	dx
Accuracy	84.7	75.6	66.5	90	54.6	63	24	66.3	82.4	56.9
Phonemes	eh	er	ey	f	g	h	ih	iy	jh	k
Accuracy	39.2	47.1	31	64.9	59.4	38.1	49.2	5	82.9	70.4
Phonemes	l	m	n	ng	ow	oy	p	q	r	s
Accuracy	35.8	41.9	41.6	41.5	43.6	67.2	82	47.9	64.2	75.3
Phonemes	sh	t	th	uh	uw	v	w	y	z	
Accuracy	63.5	73.5	31	29	74.4	91.5	45.7	57.8	83.4	

Table 3: Results of phonemes accuracy in second classification phase 'Proposed approach'

Classes	Accuracy (%)
Vowel phonemes (Vowels, semi-vowels, Diphthongs)	84
Constant phonemes (Plosives, nasals, fricatives)	88

Table 4: Classification results of each class to the corresponding phonemes in third classification phase 'proposed approach'

Phonemes classes	Accuracy (%)
Plosives	55
Nasals	74
Fricatives	65
Vowels	76
Semi-vowels	86
Diphthongs	90

several phases of classification, the network is trained to perform a partial classification in each classification phase which makes the weights of the network related to each partial classification phase.

CONCLUSION

In this study, we proposed a phoneme recognition algorithm using convolutional neural network. We compared two methods: the direct approach by recognizing the phonemesusing a single classification phase and we obtained 57% accuracy. The second proposed approach uses several phases of classification by taking into account the types of phonemes and their classes (vowels, semi-vowels, explosive, etc.) and obtained 61% accuracy. This improvement occurred as a result of that the network is trained to perform a partial classification in each classification phase which makes the weights of the network related to each partial classification phase.

REFERENCES

Fauziya, F. and G. Nijhawan, 2014. A comparative study of phoneme recognition using GMM-HMM and ANN based acoustic modeling. Int. J. Comput. Appl., 98: 12-16.

Glackin, C., J.A. Wall, G. Chollet, N. Dugan and N. Cannings, 2018. Convolutional neural networks for phoneme recognition. Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods ICPRAM, Vo. 1, January 2018, SciTePress, Setúbal, Portugal, pp: 190-195.

Hamooni, H., A. Mueen and A. Neel, 2016. Phoneme sequence recognition via DTW-based classification. Knowl. Inf. Syst., 48: 253-275.

Hubel, D.H. and T.N. Wiesel, 1959. Receptive fields of single neurones in the cat's striate cortex. J. Physiol., 148: 574-591.

Krizhevsky, A., I. Sutskever and G.E. Hinton, 2012. Imagenet Classification with Deep Convolutional Neural Networks. In: Advances in Neural Information Processing Systems, Leen, T.K., G.D. Thomas and T. Volker (Eds.). MIT Press, Cambridge, Massachusetts, USA., ISBN:0-262-12241-3, pp: 1097-1105.

Lopes, C. and F. Perdigao, 2011. Phone recognition on the TIMIT database. Speech Technol./Book, 1: 285-302.

Malekzadeh, S., M.H. Gholizadeh and S.N. Razavi, 2018. Persian phonemes recognition using PPNet. Audio Speech Proc., Vol. 1, 10.13140/RG.2.2.12187.72486

Pradeep, R. and K.S. Rao, 2016. Deep neural networks for Kannada phoneme recognition. Proceedings of the 2016 Ninth International Conference on Contemporary Computing (IC3), August 11-13, 2016, IEEE, Noida, India, pp: 1-6.

Xie, Y., L. Le, Y. Zhou and V.V. Raghavan, 2018. Deep Learning for Natural Language Processing. In: Handbook of Statistics Vol. 38, Brownlee, J. (Ed.). Elsevier, Amsterdam, Netherlands, pp: 317-328.

Yousafzai, J., P. Sollich, Z. Cvetkovic and B. Yu, 2010. Combined features and kernel design for noise robust phoneme classification using support vector machines. IEEE. Trans. Audio Speech Lang. Proc., 19: 1396-1407.

Yu, C., Y. Chen, Y Li, M. Kang, S. Xu and X. Liu, 2019. Cross-language end-to-end speech recognition research based on transfer learning for the low-resource Tujia language. Symmetry, Vol. 11, No. 2. 10.3390/sym11020179

Zarrouk, E. and Y. Benayed, 2016. Hybrid SVM/HMM model for the Arab phonemes recognition. Int. Arab J. Inf. Technol. (IAJIT), 13: 574-582.