Multimedia Systems
https://doi.org/10.1007/s00530-021-00822-5

SPECIAL ISSUE PAPER

A computer-aided speech analytics approach for pronunciation feedback using deep feature clustering

Faria Nazir[1] · Muhammad Nadeem Majeed[2] · Mustansar Ali Ghazanfar[3] · Muazzam Maqsood[4]

Received: 22 June 2020 / Accepted: 9 June 2021

Abstract
Nowadays, the demand for language learning is increasing because people need to communicate with other people belonging to different regions for their business deals, study, etc. During language learning, a lot of pronunciation mistakes occur due to unfamiliarity with a new language and differences in accent. In this paper, we perform speech mistakes analysis using deep feature-based clustering. We proposed two novel methods for speech analysis, one to deal with phonemic errors (confusing phonemes) and the other to deal with the prosodic errors (partially changed pronunciation variation of phones). For accurate and efficient language learning, it is important to learn both phonemic as well as prosodic error corrections. In our first method, we perform speech analysis by combining deep CNN features and clustering algorithm to detect the phonemic errors. We classify the phonemes using K-nearest neighbor, Naïve Bayes, and support vector machine (SVM). We perform experiments on the six most frequently mispronounced confusing pairs of Arabic to handle phonemic errors and achieve an accuracy of 94%. In our second method, we proposed the unsupervised phone variation model (PVM) to detect prosodic errors. In PVM, each phone is extended to represent the different types of pronunciation variation of that phone with different proficiency levels. We use an Arabic dataset of 28 individual phones for speech analysis and provide feedback based on the variation of each phone and achieves an accuracy of 97%.

Keywords Speech analytics · Deep convolutional neural network · Multimedia tools · Deep clustering · Phone variation model

1 Introduction

With the developing population of second language learners, there is a necessity for additional language learning assets. Viable language learning devices, and especially pronunciation training, necessities to furnish students with definite remedial criticism.

Computer-based language guidance frameworks with text to speech [1, 2] possibly can offer more advantages over traditional strategies, particularly in territories, for example, pronunciation training, which frequently requires full consideration of the educator to a solitary student. If the computer-based system replaces the teacher's responsibilities, e.g., giving corrective feedback to students about pronunciation, then it is much easier for students to learn a second language. They can access the system anytime, at anyplace. The latest advances in research on programmed pronunciation scoring [3, 4] enable us to acquire articulation quality evaluations for sentences or a combination of

Communicated by Muazzam Maqsood.

✉ Muazzam Maqsood
muazzam.maqsood@cuiatk.edu.pk

Faria Nazir
faria.nazir@uettaxila.edu.pk

Muhammad Nadeem Majeed
nadeem.majeed@pucit.edu.pk

Mustansar Ali Ghazanfar
mghazanfar@uel.ac.uk

[1] Department of Software Engineering, University of Engineering and Technology Taxila, Taxila, Pakistan

[2] Department of Data Science, University of the Punjab, Lahore, Pakistan

[3] School of Architecture, Computing and Engineering, University of East London, London, UK

[4] Department of Computer Science, COMSATS University Islamabad, Attock Campus, Pakistan

Published online: 19 July 2021

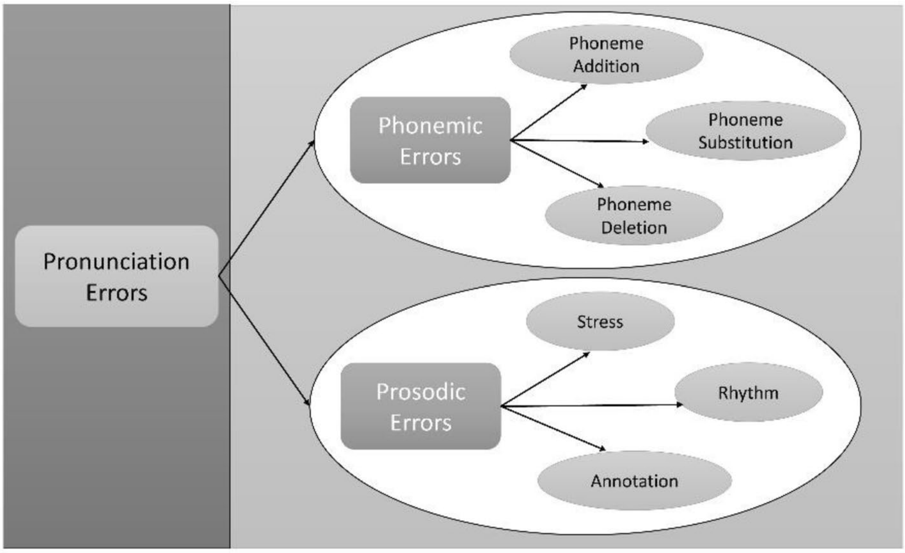


Fig. 1 Types of pronunciation errors

sentences, with discretionary content, with evaluating consistency like that of a specialist educator.

The errors in pronunciation are broadly categorized into two types, phonemic errors and prosodic errors [5] as shown in Fig. 1.

To make a complete and accurate mispronunciation detection system, it is important to deal with both phonemic as well as prosodic errors. Phonemic errors are caused by the addition, deletion, and substitution of one phone with another. On the phonemic side, there are the ’extreme’ mistakes where phonemes may be replaced with another phoneme, removed, or on the other hand embedded.

Such types of errors are also called complete pronunciation mistakes. Prosodic errors are variation due to stress [6], annotation, and rhythm. All such errors are firmly connected which is demonstrated in Fig. 1. This reality makes pronunciation a multi-dimensional issue that is hard to nail down with a solitary methodology. Or maybe, an effective framework will require a mixture of a wide range of strategies.

Traditionally, phonemic errors are handled as classification problems but, there are some limitations to the supervised approach (1) Labeled data is needed that requires a lot of time. (2) Acoustic phonetic features are used that hard to decide which feature to consider and which to remove. To deal with all such limitations we propose a deep clustering technique to detect phonemic mistakes. To deal with pronunciation variation mistakes, traditionally two techniques are used (1) Multi-training method but its limitation is that this method ingests few variations. (2) To train the model with only correct pronunciation but false alarm rate increases. To deal with all such limitations we propose a phone variation model to detect variation mistakes. In this paper, we proposed two techniques for speech analysis. The first technique is used to analyze the speech for complete pronunciation errors or phonemic errors while the second technique is used to analyze the partial pronunciation mistakes or prosodic errors. A complete mispronunciation detection system cannot work properly unless both phonemic and prosodic errors are handled. As the nature of errors are different, two different methods are proposed for an efficient mispronunciation detection system.

In the phonemic error detection method, we first find which phonemes are confusing and group them based on similarity in pronunciation. After that, we convert the phonemes audio files to spectrograms and pass them to a convolutional neural network for feature extraction. We extract the features from different layers of CNN and choose the features of that layer that gives optimal results. We pass that deep features to our clustering algorithm and obtain the pseudo labels. In the end, we apply a classification algorithm to analyze the speech to detect mistakes. In the partial pronunciation error detection method, we detect the pronunciation variation of each phone. First of all, we collect all samples of a specific phone and extract the features of that phone using a convolutional neural network. We use a Gaussian mixture model to detect the variation of a phone. In the end, we apply the support vector machine algorithm to detect pronunciation mistakes due to phone variations.

The main contributions of this paper are as follows.

- We automate the process of data labeling that is closer to human experts labeling for Arabic phonemes using the unsupervised method.
- We proposed an efficient algorithm to deal with phonemic errors / complete pronunciation errors using a deep clustering method.

- We proposed an algorithm to detect partial pronunciation errors/pronunciation variations of a single phone using a phone variation model (PVM).

The rest of the paper is organized as follows. Section 2 describes the previous research related to pronunciation detection, Sect. 3 presents the methodologies of our proposed methods, and Sect. 4 consists of experimental results and comparison with traditional approaches, and Sect. 5 presents the conclusion.

2 Literature review

This section presents the literature review of the techniques used for speech analysis and feedback. Over the previous period, many speech analysis techniques are developed; we categorized the techniques based on the methods used.

2.1 Posterior probability-based methods

Franco et al. [7] proposed two techniques; in the first technique, for each phone, scores based on posterior probability are figured. These probabilities rely upon acoustic models of local speech. The posterior probability of phone q with observation m is calculated as

$$Pb(q|m) = \frac{1}{T} \sum_{t=t_0}^{t=t_{0+T-1}} \frac{Pb\left(m_t|q\right) Pb(q)}{\sum_{a=1}^{Q} Pb\left(m_t|q_a\right) Pb\left(q_a\right)}, \tag{1}$$

where t_0 is the starting edge file of x, T is the time duration of x and mt is the acoustic perception for tth frame.

S.M. Witt et al. [8] presented a posterior probability based on goodness of pronunciation (GOP) scoring and proposed different estimations that can be used to contrast execution and human judges. The blend of the standard GOP technique with a couple of refinements gives enhancements in scoring execution. The GOP of phone q with observation m is calculated as

$$GOP(q,m) = (\log Pb(m|q))/T$$
$$\left|\log\left(\frac{Pb(m|q)\ \ Pb(q)}{\sum_{a=1}^{Q} Pb\left(m_t|q_a\right)Pb\left(q_a\right)}\right)\right| \Big/ T \tag{2}$$

where Q represents total phone labels and T represents the duration of a frame of m observation. Zhang et al. [9] displayed the strategies to upgrade the execution of error detection at syllable level for Mandarin from two perspectives: proposing scaled log posterior probability (SLPP) and weighted phone SLPP to improve the measure of pronunciation quality.

The above strategies would all be able to be seen as various techniques to approximate the posterior probability. Franco et al. [7] utilize a frame-level estimate to $P\ (m)$, while Witt and, furthermore, Young [10] utilize free acknowledgment results from a free-running phone loop to acquire the estimation to $P\ (m)$. The key inquiry here is whether the guess to $P\ (m)$ is sufficiently exact. For speech analysis, utilizing only the acoustic model arrangement of the objective language to measure $P\ (m)$ cannot state all errors affected by the student's primary language.

2.2 Likelihood ratio test-based methods

Franco et al. [7] use the LRT-(likelihood ratio test) based method to detect pronunciation errors. The methodology uses phonetically named nonnative speech databases to prepare two distinctive acoustic models for each phone. Execution of the systems was evaluated in the database of 130,000 phones communicated in constant speech sentences by 206 nonnative speakers. The log-likelihood score of phone q is calculated as

$$\mathrm{LRT}(m,q) = \frac{1}{T} \sum_{t=t_0}^{t=t_{0+T-1}} \left[\begin{array}{c} \log P\left(m_t|q, \lambda_{nc}\right) - \\ \log P\left(m_t|q, \lambda_c\right) \end{array} \right], \tag{3}$$

where T, t0, and mt are the same as in Eq. (1). nc represents mispronounced and C represents correct pronunciations. Ito et al. [11] likewise propose a strategy like LRT for error discovery. In their work, given a word or pronunciation alongside its comparing acoustic perception m, the right pronunciation model is made by interfacing the Hidden Markov Models (HMM) of each phone as per the phone succession of the word or expression. Next, some error rules are applied to create error models $a1$, $a2$, … ak every one of which speaks to the conceivable error of the phones in the phone arrangement. Here, the error is executed by supplanting the ith phone with an HMM got from the speaker's local language. At that point, the log-probability distinction is utilized as the estimation of error as follows

$$D\left(m|\alpha, \alpha_i\right) = L(m|\alpha) - L\left(m|\alpha_i\right). \tag{4}$$

As a rule, all the above techniques endeavor to assemble or select an error model all around supported by the acoustics. LRT depends on the examination of H0 furthermore, H1 model for a check, which is the correlation of local and non-local models in the past technique. On the off chance that the hidden error models can speak to the genuine circulation of error, LRT can ensure that the outcome is ideal. Lamentably, the conveyance of errors shifts seriously with speakers and writings. In this manner, it is hard to construct genuine error models, particularly with a restricted error corpus marked by human annotators.

2.3 Confidence measure

The error discovery issue is like the confidence measure CM of automatic speech recognition ASR which checks whether the recognition result is right or not. Jiang [12] gives an intensive and point by point review of CM. Jiang [12] categorize all techniques for CM into three classifications: (I) LRT-based strategies [13]; (ii) posterior probability-based strategies [14]; and (iii) the predictive features-based techniques [15]. As referenced beforehand, the LRT-based and the posterior probability-based error discovery techniques have been explored by Franco et al. [7] Witt [8], and Young [10], while the predictive feature-based technique has not yet been examined. The error identification strategy proposed by this paper depends on that thought.

2.4 Pronunciation modeling

Pronunciation mistakes are a unique instance of pronunciation variation. Much work has been done to address pronunciation diversity in ASR because pronunciation variation is one of the significant reasons for ASR misclassifications [16]. Saraclar [17] acquaints a strategy with share Gaussians crosswise over phone models to show incomplete changes of pronunciation. Liu and Fung [16] propose Partial Change phone Models(PCPMs) to speak to fractional changes and at that point consolidate PCPMs with unique acoustic models to improve the model goals for pronunciation variation. Roused by pronunciation demonstrating, we additionally propose to construct various acoustic models for each phone in the accompanying.

3 Speech analysis methods

3.1 Problem statement

As discussed in the previous section, speech analysis and their feedback can be formulated as a classification problem, but the existing approach has some shortcomings. (1) The existing approaches used acoustic–phonetic features (MFCC, pitch, ZCR, and so on) for speech analysis and it is hard to analyze which features should be included. (2) The labeling of data from human experts is very time consuming and difficult. In line with the above-mentioned problems, we proposed a method that resolves these problems. We also formulated pronunciation error detection as a classification problem, but we use Convolutional neural network features (deep features) that are automatically extracted from CNN. Secondly, we label the data using unsupervised methods *K*-means. The labeling of data is close to human labeling due to the resemblance of CNN with the human visual system.

The advantages of dealing the speech analysis as a classification problem include (1) several classification algorithms can be applied instantaneously to detect pronunciation errors and (2) various features can be combined for classification instead of a solitary feature used in conventional methods.

3.2 Speech analysis based on deep clustering to detect phonemic errors

In this method, we use the concept of deep clustering (deep features + simple clustering) to detect the pronunciation

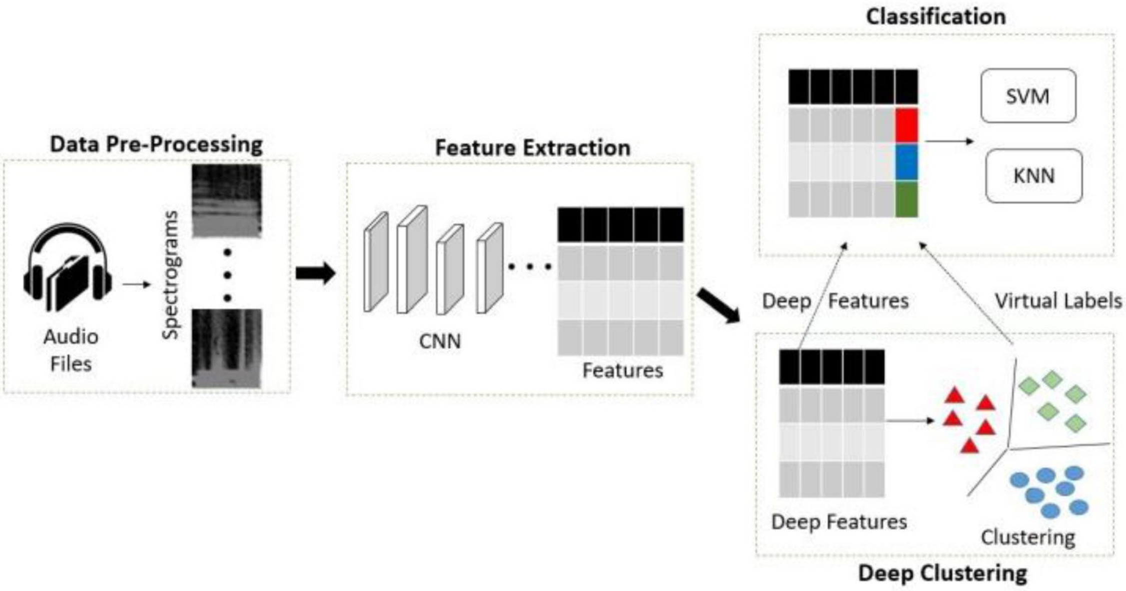


Fig. 2 Speech analysis using a deep clustering method

mistakes (phonemic errors) as shown in Fig. 2. The phonemic errors are caused by confusing phonemes. The confusing phonemes are separate to deal with as compared to other pronunciation mistakes because they have a similar sound and hard to differentiate a second language learner. For example, the phonemes ط,ت have similar t-like sounds, while the other pronunciation mistakes may be based on variation and deal differently. The first step is to preprocess the audio files and convert them to spectrograms. Second, we extract the features of spectrograms with help of a convolutional neural network and pass these deep features to our clustering algorithm to split the data into multiple clusters. We assign the pseudo labels to our data based on clustering. Lastly, we apply the classification algorithms to train the model for pronunciation error detection. This method detects pronunciation errors due to confusing phonemes.

(1) Data pre-processing

The first step in this method is to preprocess the data. In data pre-processing, first we analyze the audio files and eliminate those files that contain noise. After that, we convert the audio files to spectrograms and convert all stereotype signals to monotype signals by taking the average of signals. Lastly, we resize all the spectrograms to 227 × 227 size and contain RGB channels, so they are acceptable for input to CNN for feature extraction.

(2) Features extraction using CNN

In this paper, we use the features of the well-established pre-trained convolution neural network (CNN) AlexNet for classification. We pass the input image (spectrogram) f to pre-trained CNN AlexNet. In the feature extraction process, we take a kernel or filter and pass it over the input image and transform it according to the values of the filter. The feature map values are calculated as in Eq. (5)

$$G[m,n] = (f * h)[m,n] \\ = \sum_j \sum_k h[j,k] f[m-j, n-k] \tag{5}$$

The reasons for choosing these CNN features include (1) convolutional neural network automatically extract the discriminative features, and we do not need to manually extract the acoustic–phonetic features. (2) CNN features resemble the human visual system, and they encode the image in the same way as a human, so they are particularly useful for the classification of images.

We could further extend the features by adding acoustic–phonetic features like MFCC, pitch zero-crossing rate and so on. But this paper uses only the CNN features for classification.

3.2.1 CNN architecture

We used the AlexNet to extract the deep features. AlexNet architecture consists of five convolutional layers, max-pooling layers, and three fully connected layers as shown in Fig. 3. Convolutional layers contain many kernels that extract interesting information from images. The first convolutional layer contains 96 kernels of 11 × 11 size with depth 3. The depth is usually the same as the number of channels of spectrograms. The details of the number of kernels and the size of each layer are explained in Table 1. The first two convolutional layers are followed by the max-pooling layer. The max-pooling layers aim to downsample the width and height of kernels while keeping the depth the same. The convolutional layers 3, 4, and 5 are directly connected and C5 (Convolutional layer 5) is followed by max pooling. The

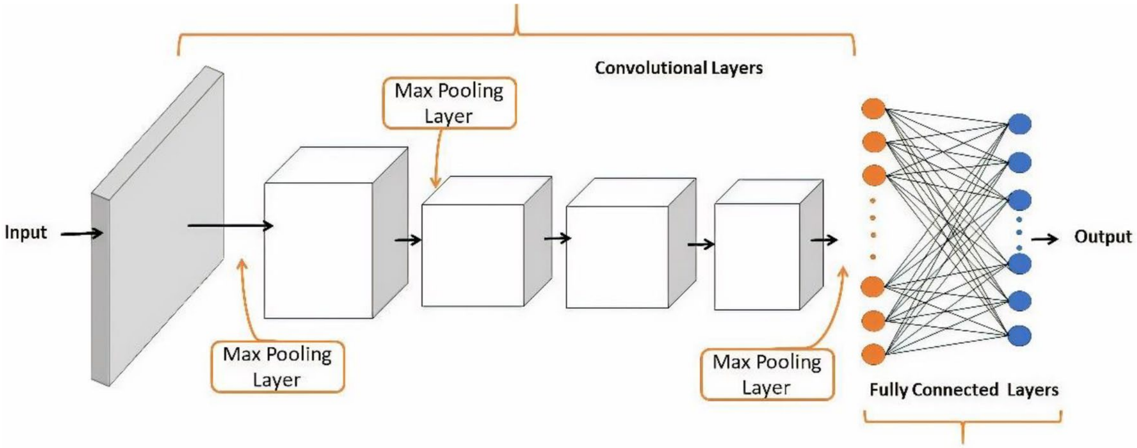


Fig. 3 Alexnet convolutional neural network architecture

Table 1 Alexnet architecture layers with kernel size

Layer	Input	Output	Kernel size
C1	227×227×3	55×55*96	11×11
Max Pool L1	55×55×96	27×27×96	3×3
C2	27×27×96	27×27×256	5×5
Max Pool L2	27×27×256	13×13×256	3×3
C3	13×13×256	13×13×384	3×3
C4	13×13×384	13×13×384	3×3
C5	13×13×384	13×13×256	3×3
Max Pool L5	13×13×256	6×6×256	3×3
FC-6			1×1
FC-7			1×1
FC-8			1×1

output of convolutional goes to fully connected layers. The detail of AlexNet architecture is given below.

3.2.2 Convolutional layer

This is the most critical layer in the convolutional neural network framework that creates the Feature maps which are exposed to fully connected layers for classification purposes. It comprises a kernel that slides over the input data and produces the output known as a feature map [18]. The feature map is defined in the equation.

$$F_x^r = \frac{F_x^{r+1} - K_x^r}{P_x^r} + 1, \tag{6}$$

$$F_y^r = \frac{F_y^{r+1} - K_y^r}{P_y^r} + 1, \tag{7}$$

where $(F_x,\ F_y)$ represents the width and height of the feature map, while $(k_x,\ k_y)$ shows the kernel size with skipped pixels horizontally and vertically $(P_x,\ P_y)$ that are not important for training. As we have five convolutional layers in Alexnet architecture, so r represents the number of layers, i.e., $r=1$. The convolutional layers convolved the input feature map I_f with a kernel, K to obtain the output feature map O_f that is formulated as:

$$O_f(m,n) = \left(I_f * K\right)(m,n)\ , \tag{8}$$

where $(m,\ n)$ represents the dimensions of the feature map. The symbol * represents the convolution between the input feature map using the K kernel of size $((k_x,\ k_y))$. The convolutional function is defined as

$$O_f(m,n) = \sum_{a=-\frac{k_x}{2}}^{a=\frac{k_x}{2}} \sum_{b=-\frac{k_y}{2}}^{b=\frac{k_y}{2}} I_f(m-a,\ n-b)K(a,b). \tag{9}$$

3.2.3 Max-pooling layer

A pooling layer is present in Alexnet architecture after the first and second convolution layers and at that point after the fifth convolution layer to diminish the spatial size of each casing to lessen the computational cost of the proposed deep learning system. The pooling activity typically midpoints or picks the most extreme incentive for each cut of the picture layer.

3.2.4 Fully connected layers:

The fully connected layer neurons are related to neurons of neighboring layers. In the AlexNet, there are three completely associated layers. These layers blend the features of two channels to get a 4096-dimensional feature vector. The ReLU, as shown in Eq. (10), is a half-wave rectifier work, which can altogether quicken the training stage and reduce overfitting.

$$f(a) = \max(a,0). \tag{10}$$

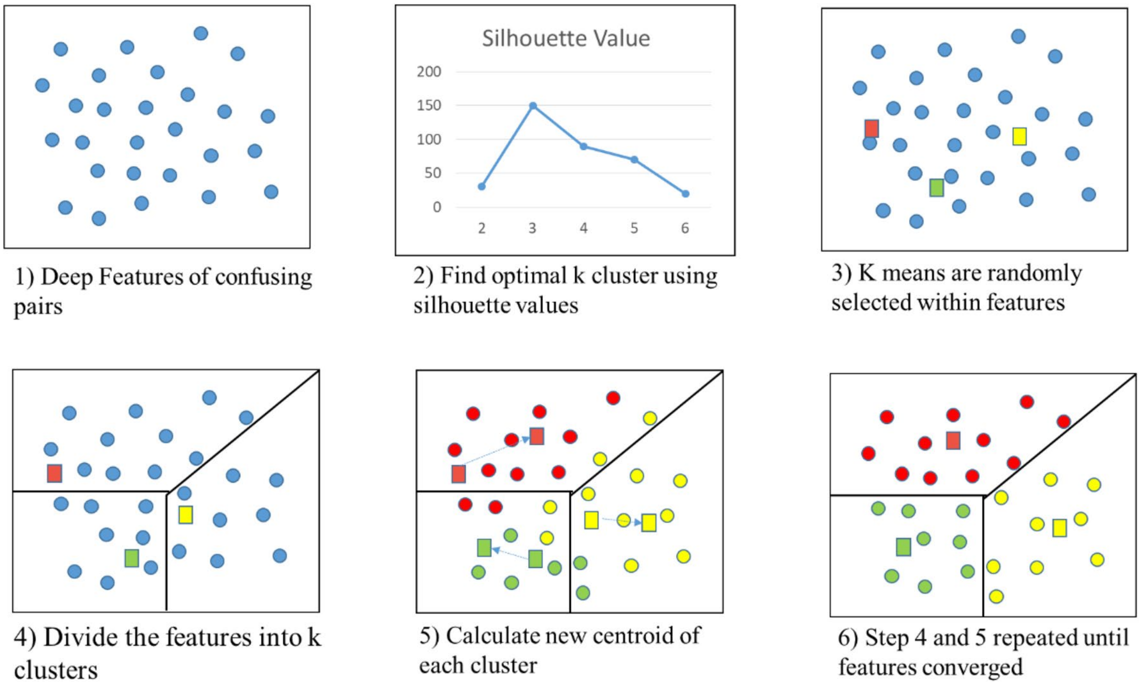
The dropout system can be viewed as a sort of regularization by stochastically setting some of the information neurons or concealed neurons to be zero to diminish the co-adjustments of the neurons, which is typically used in the fully connected layers in the AlexNet framework.

We extract the features from different layers of Alexnet (C1, C2, C3, C4, C5, FC6, and FC7). We aim to find the most discriminating features so that confusing phones can be distinguished easily. To obtain this aim, we compare the features extracted from each layer and choose the features of that layer that give us optimized results.

3.2.5 Deep clustering

Traditionally the speech corpus is collected and labeled as either correct pronunciation or incorrect by human experts. It is hard to find the language experts and label a huge amount of data so the proposed method first labels the confusing phones using some simple clustering techniques and then used that pseudo labels for classification. The deep clustering mechanism to obtain pseudo labels is as follows and also shown in Fig. 4.

Step 1: Collect all samples confusing to phone q from a dataset and mark them as Oq. Where $Oq = \{Oq1,\ Oq2\ \ldots.$



1) Deep Features of confusing pairs

2) Find optimal k cluster using silhouette values

3) K means are randomly selected within features

4) Divide the features into k clusters

5) Calculate new centroid of each cluster

6) Step 4 and 5 repeated until features converged

Fig. 4 Step-by-step algorithm of deep clustering

Oqn}, where n represents the number of samples. We have 400 samples of each phone.

Step 2: Calculate the feature vector of each sample using CNN as described in 3.2.1.

Step 3: Split the data *Oq* into several *K* groups based on CNN features using K-means clustering. *K* represents the number of confusing phonemes. Some confusing phonemes are 2 and some are 3.

Step 4: The value of *K* is determined by Silhouette and Calinski–Harabasz analysis. *K* cannot be fixed because confused phone groups are different in number.

The deep clustering (deep features + simple clustering) does not give us exactly true labels, but we ensure that these labels are as close as human labels. We also compare pseudo labels from deep clustering with human expert labels to check how accurately our algorithm split the confusing phonemes.

3.2.6 Classification to detect pronunciation errors

In this paper, we used many classifiers for classification such as K-nearest neighbor, Naïve Bayes, and support vector machine. The SVM classifier is best suited due to its generalizability and suitability for classification. We assume that we have *R* confusing phonemes {$q1$, $q2$… and have {$l1$, $l2$, $l3$……lR} pseudo labels. The feature vector for each phone q is $f=\{f1, f2 \dots fn\}$, where n represents the numbers of features for each phone. By combining the feature vector with labels, we built the dataset for classification as in Eq. (11)

$$(f_1^n,\ l1),\ (f_1^n,\ l2)\dots(f_1^n,\ lR) \tag{11}$$

3.2.7 KNN

K-nearest neighbor (KNN) classifies the new data based on similarity measure/ distance function. The similarity measure can be Euclidean, Manhattan, and Minkowski. We use Euclidean distance as a similarity measure. To detect pronunciation mistakes of two confusing phones $q1$ and $q2$, the Euclidean distance can be measured as

$$\text{Euclidean} = \sqrt{\sum_{i=1}^{n}\left(q_i - q2_i\right)^2} \tag{12}$$

We use 10 k nearest neighbors and n represents the number of features of the phone $q1$ and $q2$.

3.2.8 Naïve Bayes

Naïve Bayes used the probabilistic approach to classify the data. Naive Bayes uses the concept of Bayesian theorem and is best suited for high-dimensional data. This theorem assumes that the features are independent. Any probabilistic classifier aims to measure the probability of each feature xi where $i=0, 1, 2 \dots n$ occurring in each class C, where $C=0$, 1, 2…k, and then return the most probable class. For this purpose, we must calculate the $Pb\ (Ci\ |\times 0, \times 1, \times 2 \dots xn)$

$$Pb\left(C_i|x0,x1,x2\ldots..xn\right)\ \alpha\ Pb\left(x0,x1,x2..xn|C_i\right)\times Pb\left(C_i\right) \tag{13}$$

We assume that the features are independent of each other, so the equation becomes.

$$Pb\left(x0,x1,x2..xn|C_i\right)=Pb\left(x0|C_i\right)^{*}Pb\left(x1|C_i\right)\ldots\ldots Pb\left(xn|C_i\right). \tag{14}$$

So, the final representation of the equation is as follows.

$$Pb\left(C_i|x0,x1,x2\ldots..xn\right)\propto Pb\left(C_i\right)\prod_{m=1}^{n}Pb\left(x_m|C_i\right), \tag{15}$$

where x represents features and C represents the classes.

3.2.9 SVM

The support vector algorithm outputs an optimal hyperplane that categorizes the data by labeled classes. SVM is best for binary classification, but it is optimized to deal with multiclass problems. In two-dimensional space, the SVM classifier makes a direct hyperplane that isolates the two classes. If the data are not linear, we have to tune the SVM using some parameters like the kernel trick. Kernel function transformed the nonlinear data to linear in high-dimensional space. We apply SVM on a multiclass dataset that contains confusing phonemes.

We use 10 cross-fold validation instead of fixed training and testing dataset for classifier training. We choose polynomial kernel with degree 3 for classification problem with discriminative function as follows.

$$d(f)=\left(w^T+c\right)^d, \tag{16}$$

where $d(f)$ is the discriminative function and d represents the degree of the polynomial kernel.

3.3 Speech analysis based on PVM

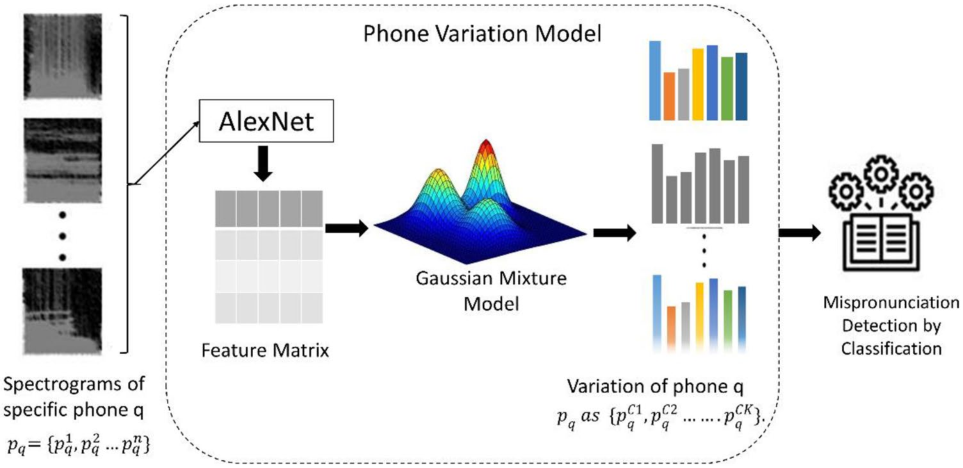
Speech analysis of confusing phonemes can be used to detect only completely changed pronunciation mistakes, e.g., phone substituted as another similar/confusing phone. But to detect pronunciation variations, this type of model is not as effective. Pronunciation variations are mostly produced due to accent and mother tongue that is changed from the native language. These types of pronunciation mistakes are most common and some partially changed pronunciation errors are bearable to human experts and they label them as correct and some are unbearable, and they label them incorrect. Such pronunciation errors are important to handle for an accurate mispronunciation detection system. To handle partially changed pronunciation errors, we perform speech analysis due to pronunciation variation. We proposed a model based on the Phone variation model because they cannot be handled with the same technique used for phonemic errors.

In this model, we detect the pronunciation mistakes caused due to pronunciation variation as shown in Fig. 5. To detect partial pronunciation errors, first, we take the spectrograms of the specific phone q and all its variations as input to the convolutional neural network for feature extraction. Second, we extract the features from Alexnet and pass them to the Gaussian mixture model to detect the number of variations of the phone q. After the detection of the phone variations, we train the classifier to detect pronunciation errors due to phone variations.

3.3.1 Why phone variation model

There are two distinct ways to deal with pronunciation mistakes due to pronunciation variations. One method is the multi-training technique that utilizes correct and incorrect pronunciation data to train the model. The false alarm rate decreases because this technique ingests a few pronunciation variations. Multi-training strategy deals with all

Fig. 5 Detection of pronunciation variation using phone variation model


Phone Variation Model
AlexNet
Feature Matrix
Gaussian Mixture Model
Spectrograms of specific phone q
$p_q=\{p_q^1,p_q^2\ldots p_q^n\}$
Variation of phone q
p_q as $\{p_q^{C1},p_q^{C2}\ldots\ldots.p_q^{CK}\}$.
Mispronunciation Detection by Classification

pronunciation mistakes similarly as the true pronunciations, it will lose the capacity of recognizing errors from true pronunciations. The second methodology in this manner is to construct models just utilizing true pronunciation. This sort of model holds the capacity to recognize true pronunciations from pronunciation variations, but the false alarm rate increases due to ignoring variation of partially changed pronunciations. This implies the model will be delicate to partially changed pronunciation errors. To manage all types of pronunciation variations we proposed reasonable acoustic models ought to be worked to a wide range of pronunciation variation.

To deal with the above-stated problem, we try to build a model that deals with all types of pronunciation variations with diverse proficiency levels known as the phone variation model. The phone variation model is important because it not only deals with correct vs incorrect or native vs nonnative pronunciation, but this model also focuses on the variation of the pronunciations that we are unable to cater to with simple acoustic models. The performance of the training system increases due to PVM because previous methods mostly use posterior probability for pronunciation evaluation while this method uses GMM to calculate the probabilities which makes them more precise as compared to traditional acoustic models We gathered the data from Asian speakers of different age and belonging to different areas so there are variations in pronunciation we proposed a model by extending the concept of [19], we used GMM instead of posterior probability values to detect all the variations of pronunciation for a given phone using Phone variation model. In the phone variation model, we extended the specific phone q to multiple variations $\{q1, q2\ldots.\ qk\}$, where k represents the number of variations of that phone q. These variations range from true pronunciation to unbearable/ incorrect pronunciation and some intermediate pronunciations are bearable.

(1) PVM construction based on Gaussian mixture model

The proposed PVM method deals with all types of partially changed pronunciation errors (pronunciation variations) of specific phone q. If we have enough data label by language expert as correct or incorrect pronunciation, then we directly train the models using feature vector. Unfortunately, we do not have enough data labeled by a language expert and it is hard to label the huge amount of data as correct or incorrect pronunciation and more difficult to label the variations of pronunciation belonging to diverse proficiency levels by a human expert. To deal with all these problems, it is hard to implement supervised learning, so we introduced an unsupervised Phone variation model (PVM) with Gaussian mixture model which proceed as follow.

Step1: Collect all the samples of phone q from the dataset and represent them with $p_q = \left\{p_q^1, p_q^2 \ldots \ldots .p_q^n\right\}$, where n represents the number of samples of phone q. Then, the convolutional neural network is used to calculate the feature vector of each sample.

Step 2: We split the data (feature vectors) of n samples into K variations using the Gaussian mixture model. We represent $p_q\ as\ \left\{p_q^{C1}, p_q^{C2} \ldots \ldots .p_q^{CK}\right\}$. We can use different k values for different phones but for sake of simplicity, we fixed the k value to 3 for all phones. K value equal to 3 means true, intermediate, and incorrect pronunciation variation. To avoid the problem of imbalanced data, we make sure that our groups are of the same size.

This unsupervised method does not perfectly group the data into correct and incorrect pronunciations, but this process gives us a basic idea about partially changed pronunciation errors.

3.3.2 Gaussian mixture model

We use the concept of the Gaussian mixture model to represent the multiple variations of the specific phone q. We have used the Gaussian mixture model to group the variation of the phone because it can be viewed as an enhancement of k-mean and it is a generative model and commonly used in the industry. It attempts to discover an intermingled depiction of the likelihood appropriation of the multi-dimensional Gaussian model, along these lines fitting an information dispersion of arbitrary shape. Gaussian distribution means the bell-shaped curve and all the data points are distributed along with the mean value. Multiple bell-shaped curves represent the variation of that specific phone q as shown in Fig. 6.

The probability density function is given by equation

$$f\left(x|\mu, \sum\right) = \frac{1}{\sqrt{2\pi\left|\sum\right|}} \exp -\frac{1}{2}(x-\mu)^t \sum^{-1} (x-\mu), \qquad (17)$$

where x represents the input, μ is the mean and $\sum$ is the covariance. We have a phone q with d features with K Gaussian distributions. K represents the variation of the phone q. For the sake of simplicity, we fixed the value of k as 3. We have three variations of each phone, correct, intermediate, and incorrect pronunciations. The value of the mean and variance of each Gaussian is assigned by Expectation maximization [20].

3.3.3 E-Step

We have to assign each data point xi to a specific Gaussian distribution $c1, c2, c3 \ldots cn$.

For this purpose, we calculate the probability that this sample belongs to the distribution.

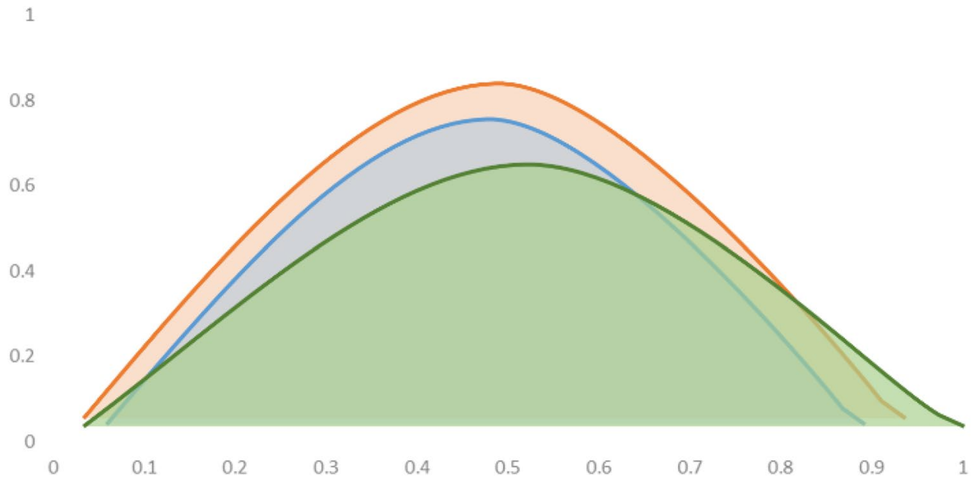


Fig. 6 Multiple Gaussian curves with different mean and variance represent the multiple pronunciation variations of a single phone

$$r_{ic} = \frac{\text{probability } Xi \text{ belongs to } c}{\text{Sum of probability } Xi \text{ belongs to } C_1, C_2, \dots C_k}. \tag{18}$$

3.3.4 M-Step

In the maximization step, we update the $\sum$, μ and $\prod$. The new density is characterized by the proportion of the number of points in the cluster and the all-out number of points

$$\prod = \frac{\text{number of points assigned to cluster}}{\text{total number of points}}. \tag{19}$$

The mean and the covariance lattice are refreshed dependent on the values relegated to the distribution, in extent with the likelihood esteems for the data point. Subsequently, a data point that has a higher likelihood of being a part of that distribution will contribute a bigger segment:

$$\mu = \frac{1}{\text{Number of points assigned to cluster}} \sum_i r_{ic} X_i, \tag{20}$$

$$\sum_c = \frac{1}{\text{Number of points to cluster}} \sum_i r_{ic} (X_i - \mu_c)^T (X_i - \mu_c). \tag{21}$$

Considering the updated values created from this progression, we ascertain the new probabilities for every data point and update the values iteratively.

(2) Classification with PVM

We use KNN, Naïve Bayes, and SVM with PVM to detect pronunciation variations. The classification algorithms are explained in the Sect. 3.2.4. First, the feature vector is calculated using PVM and then split the data to testing and training and fed to classifiers for classification similar to one presented in [21–23]. The dimensions of the feature vector are expanded from q to qk according to the number of variations in PVM.

3.3.5 Pronunciation errors detection systems

In this section, we describe the systems used in this experiment to detect pronunciation mistakes. One baseline method and two improved systems proposed in previous sections. The construction of the baseline method includes the following steps. First, we extract the features from the audio dataset. The features extracted include MFCC [24], Chroma, Mel spectrogram, spectral contrast, Tonnetz, pitch, root mean square energy, zero-crossing rate [25], and some statistical features.

Second, we preprocess the data, e.g., data cleaning and handle data sparsity. Third, we select the discriminative features using an information gain algorithm that uses the entropy concept for feature selection. Fourth, we pass the selected discriminative features to the classification algorithm to detect pronunciation mistakes.

The speech analysis based on the deep clustering method is implemented by automatically extracting the feature vector from the convolutional neural network instead of manually extracting the features as in the baseline method. We apply clustering on deep feature extracting from CNN and obtain the pseudo labels. We pass that pseudo labels to the classification algorithm SVM to detect pronunciation errors of confusing phonemes.

The speech analysis based on the PVM method is implemented similar to speech analysis of confusing phones except that the specific phone q is extended to K pronunciation variations and used the Gaussian mixture model to split the data.

In this paper, speech analysis is performed on six frequently mistaken confusing phenomes, and partially changed pronunciation mistakes are detected of all 28 phonemes of Arabic.

4 Experimental results

We performed the experiments on the Arabic dataset collected from Asian (Non-native) speakers of Arabic. In this section, we first discuss the dataset used and performance measures to evaluate our results. Second, we present the results to detect completely changed pronunciation mistakes using a deep clustering method and partially changed pronunciation mistakes using the PVM method. Finally, we compare the results with the baseline method.

4.1 Dataset

In this paper, we used an audio dataset of Arabic consisting of 28 phonemes. We collected the dataset from Asian speakers belonging to different regions and of different age groups. The dataset consists of 28 phonemes of Arabic collected from 400 Asian speakers with different proficiency levels (range from language expert's to starting stage of learning). Each speaker is asked to read the 28 phonemes of Arabic and save each phone to a separate audio file. For speech analysis of confusing phonemes (phonemes with similar sounds), we only need the correct pronunciation of each phone. We used the deep clustering method to distinguish the phone from other similar sound phones. In this paper, we used the most frequently mistaken six confusing pairs of Arabic (ت, ط), (س,ص), (د,ض), (ذ,ظ), (ح,خ ,ه) and (ك ق) that constitutes the majority of errors in Arabic language learning as shown in Table 2. All other distinctive phonemes are placed in a different group.

To detect the pronunciation mistakes due to pronunciation variation, we need the correct pronunciations and as well incorrect pronunciations of different proficiency levels. We experimented on all 28 phonemes [26] to detect the partially changed pronunciation.

Table 2 Mispronounced phoneme pairs addressed in this research

Phone with IPA	Mispronounced Phone with IPA
ت , /t/	ط ,/ţ/
س, /s/	ص ,/ ş/
د , /d̪/	ض, / ď/
ذ , / ð/	ظ, / đ/
ح , /h/	خ, /x/, ه, / ħ /
ك , /k/	ق, /q/

4.2 Performance measures

This paper uses accuracy as performance measures to evaluate the classification results and normalized mutual index NMI scores to evaluate the clustering results. These measures are defining as

$$\text{Accuracy} = \frac{M_C}{M_n} \tag{22}$$

$$\text{NMI Score}\ (Z, C) \ = \frac{2 * I(Z;C)}{[H(Z) + H(C)]} \tag{23}$$

where M_c represents the number of pronunciation mistakes detected by the machine by the total number of samples M_n. H represents entropy, I represents the mutual information of class (Z) and cluster labels (C).

4.3 Results of the deep clustering-based method

The speech analysis using deep clustering methods consists of multiple steps. We discuss the results of each step one by one. First, we need to investigate that which layer of the convolutional neural network gives optimal results. For this purpose, we extract the features from convolutional layers (4 and 5) and fully connected layers (FC6 and FC7). To find the layer that gives the best discriminative features, we apply a classification algorithm to these features. We experimented on six confusing pairs and all other phones with distinctive sounds are placed in one group. Table 3 represents the accuracies obtained on features of (C4, C5, F6, and F7) layers of the convolutional neural network. The features obtained from fully connected layers 6 and 7 are 4096, from convolutional layer 4 are 64,894, and from layer 5 are 43,296,

Table 3 Accuracies of phoneme pairs on features of different layers of convolutional neural network

Phones	IPA Symbols	Accuracy			
		Conv4	Conv5	FC6	FC7
ط, ت	/t/,/ţ/	90	90.08	81.89	80.57
ص,س	/s/,/ ş/	99.40	99.30	99.12	98.68
ض,د	/d̪/,/ ď/	96.51	95.66	94.41	90.89
ظ,ذ	/ ð/,/ đ/	95.57	94.71	92.72	88.21
ح,خ ,ه	/h/./x/,/ ħ/	76.68	74.21	73.85	70.34
ق ك	/q/, /k/	81.62	85.84	83.44	75.30
Other Phones (15)	-	90	90	85.62	81.09

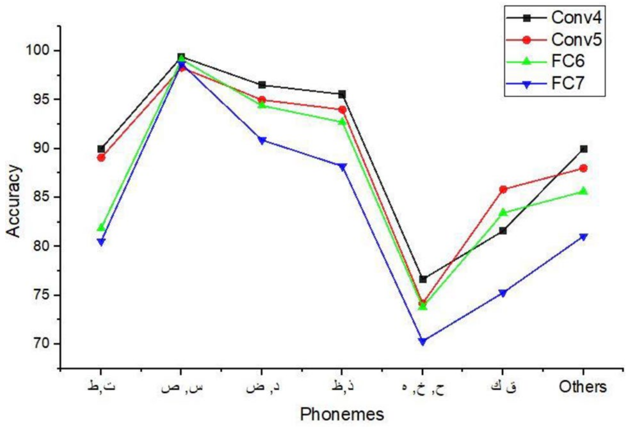


Fig. 7 Comparison of accuracies obtain from different layers of CNN

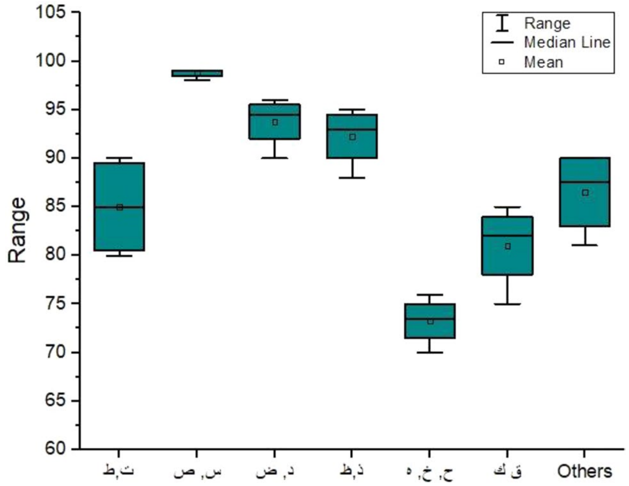


Fig. 8 Variations of accuracies for each phoneme pair from different CNN layers

respectively, the classification accuracies obtained on each phoneme pair against features of each layer (Fig. 7).

We compare the accuracies achieved by each layer (Conv4, Conv5, FC6, and FC7) for six confusing phonemes and distinctive phonemes group. The result comparison shows that convolutional layer 4 extracts most discriminative features and obtain highest accuracies of 90% for (ت, ط), , 99 for (ص,س), , 96% for (ض,د), , 95% for (ظ,ذ),, 76% for (خ,ه,ح), , 81% for (ق ك), , and 90% distinctive group as shown in Fig. 8

The comparison analysis shows that the convolutional layer 4 achieves the highest accuracies, while the FC7 layer achieves the lowest accuracies. We show the variation of accuracy from different layers for each phoneme pair in Fig. 8. The Box plot for phoneme pair (ت, ط) shows the FC7 achieves 80% accuracy.

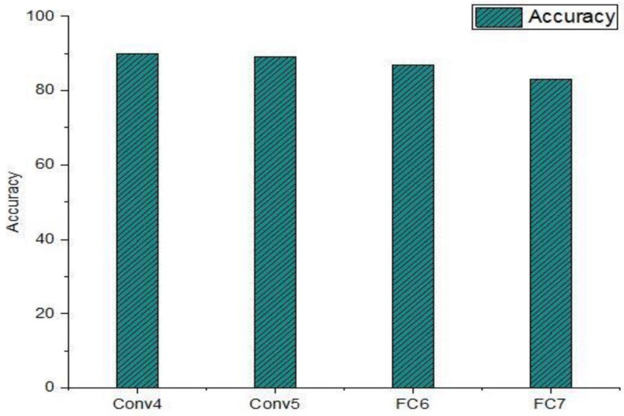


Fig. 9 Comparison of average accuracies on features of different layers of CNN

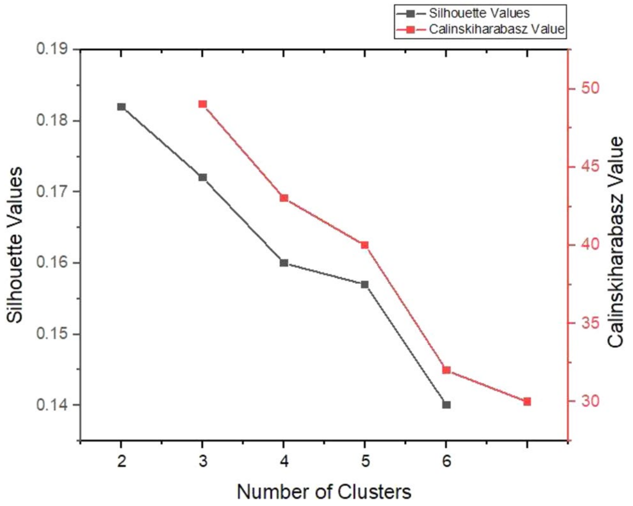


Fig. 10 Silhouette and Calinski–Harabasz value for phoneme pair ت, ط to select the optimal number of clusters in data

While Conv4 achieves an accuracy of 90%. This shows the improvement of results while extracting features from different layers.

Figure 9 represents the classification accuracies obtained from features of convolutional layers 4 and 5 and fully connected layers 6 and 7. The average accuracy obtained on C4 features is 90%, on C5 features is 88%, F6 features are 85% and F7 is 83%. So, we conclude from the experimental results that the best discriminative features are obtained by convolutional layer 4 and we choose the C4 deep features for deep clustering.

After choosing the optimal layer for feature extraction, the next step to detect pronunciation mistakes using the deep clustering method is to find the optimal number of clusters.

Table 4 NMI score for each phoneme pair

Phones	IPA Symbols	Sound Similarity	NMI score
ط, ت	/t/,/ţ/	t-like sound	0.867
ص,س	/s/,/ ş/	s-like sound	0.92
ض,د	/d̪/,/ ď/	d-like sound	0.88
ظ,ذ	/ ð/,/ đ/	th-like sound	0.82
ه, خ,ح	/h/./x/,/ ħ/	h-like sounds	0.89
ق ك	/q/, /k/	k-like sounds	0.95
Other Phones (15)	-	Different sounds	0.97

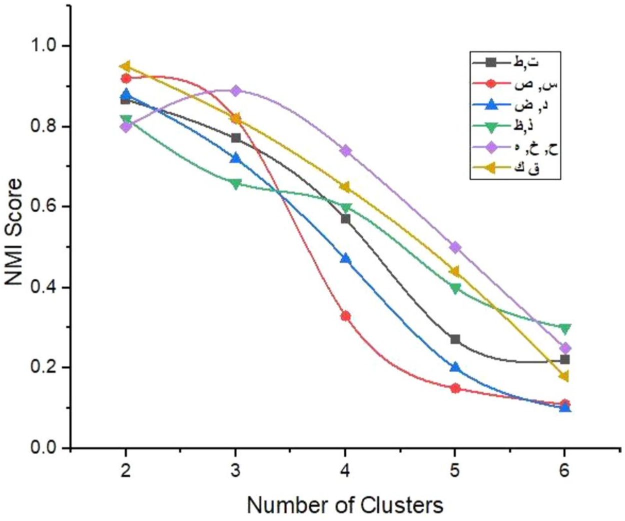


Fig. 11 Comparison of NMI value for each phoneme pair

Table 5 Speech analysis of confusing phonemes using the deep clustering method

Phones	IPA Symbols	Sound Similarity	SVM	KNN	Naïve Bayes
ط, ت	/t/,/ţ/	t-like sound	97.4	85.2	87.34
ص,س	/s/,/ ş/	s-like sound	100	91.3	93.44
ض,د	/d̪/,/ ď/	d-like sound	99.3	87.4	89.22
ظ,ذ	/ ð/,/ đ/	th-like sound	99.7	85.5	86.2
ه, خ,ح	/h/./x/,/ ħ/	h-like sounds	85.2	76.7	78.7
ق ك	/q/, /k/	k-like sounds	90.3	79.3	84.27
Other Phones	-	Different sounds	89.5	77.2	81.55

For this purpose, we evaluate the cluster based on Silhouette and Calinski–Harabasz values. We obtain the Silhouette and Calinski–Harabasz values for clusters 2, 3, 4, 5, and 6 as shown in Fig. 10. The Silhouette value is 0.18, 0.172, 0.17, 0.16, 0.14 and Calinski–Harabasz values are 49,43,40,32 and 30 for phoneme pair ط, ت for $k=2$, 3, 4, 5, 6. Both values show that an optimal number of clusters is two that means two confusing pairs are present in data. For each phoneme pair, we apply the same technique and select the optimal value of clusters in data we obtain 2 clusters for. (ط, ت), (ص,س), (ض,د), (ظ,ذ) and (ق ك) pairs, $k=3$ for (ه, خ,ح) phoneme pair and $k=15$ for distinctive phonemes group.

After the selection of optimal cluster value and feature layer, we pass the convolutional layer 4 features to the clustering algorithm with an optimal number of clusters, and the clustering algorithm provides pseudo labels to each data sample. We use the deep feature with a simple clustering algorithm to label the data automatically.

To check whether these pseudo labels are close to human labeling, we use NMI (normalized mutual index) measure. The value to NMI = 1 means that predictive labels and ground truth values are the same. The zero value of NMI shows that no predictive label is the same as ground truth values. The NMI score closer to 1 means the best prediction of pseudo labels. Table 4 shows the NMI score of each phoneme pair and our NMI values are close to 1 so they are close to ground truth values.

We choose different cluster values $k=2$–6 and observe the NMI value for each phoneme pair against those cluster values as shown in Fig. 11. We observe that NMI value is 0.86 for (ط, ت),, 0.92 for (ص,س),, 0.88 for (ض,د),, 0.82 for (ظ,ذ) and 0.95 for (ق ك) pairs when clustering $k=2$. While for phoneme pair (ه, خ,ح) , NMI value is 0.89 closer to 1 when $k=3$. We can see that for other cluster values the NMI score decreases.

After automatic labeling of data using the clustering algorithm, we apply the classification algorithms KNN, Naïve Bayes, and SVM to detect pronunciation mistakes.

Table 5 shows the accuracies achieved by SVM, KNN, and Naïve Bayes classifier on phonemes pairs. SVM achieves an accuracy of 97%, KNN achieves 85% and naïve Bayes achieves an accuracy of 87% for phoneme pair (ط, ت). .

A comparison of experimental results shows that SVM outperforms the KNN and naïve Bayes classifier as shown in Fig. 12. SVM achieves the accuracy of 97%, 100%, 99%, 99% 85%, 90% and 89% for Arabic phonemes (ط, ت), (ص,س), (ض,د), (ظ,ذ), (ه, خ,ح), (ق ك) and distinctive group, respectively. Naïve Bayes achieves the least accuracy. SVM achieves 94%, while KNN achieves 83.22% and Naïve Bayes achieves 85.8% average accuracy for all confusing phonemes.

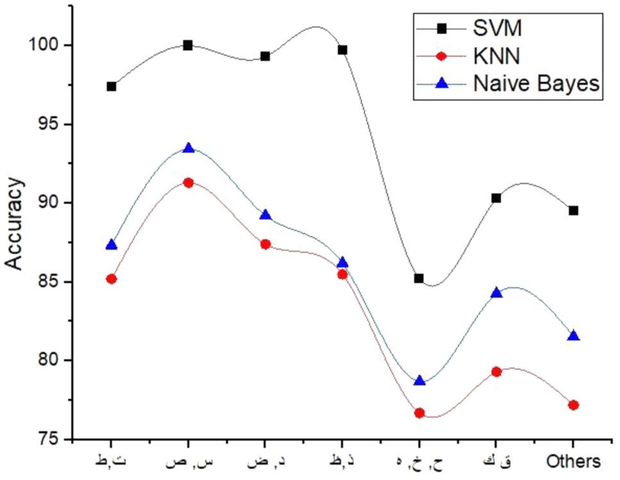


Fig. 12 Comparison of SVM, Naïve Bayes and KNN for confusing phoneme pairs

Table 6 Performance of all phonemes using phone variation model

Phonemes	IPA Symbols	Accuracy	Precision	Recall
ا	/ ʔ/	89.8	0.876	0.89
ب	/b/	94.3	0.926	0.94
ت	/t/	92.5	0.895	0.90
ث	/θ/	93.3	0.92	0.93
ج	/g/	95.0	0.94	0.943
ح	/h/	97.9	0.97	0.97
خ	/x/	90.1	0.89	0.896
د	/d̪/	96.3	0.95	0.95
ذ	/ ð/	96.5	0.956	0.95
ر	/r/	94.2	0.92	0.93
ز	/z/	95.6	0.94	0.93
س	/s/	96.5	0.956	0.95
ش	/ š/	97.3	0.96	0.953
ص	/ ş/	92.4	0.92	0.92
ض	/ ď/	94.5	0.93	0.93
ط	/ţ/	95.0	0.94	0.96
ظ	/ đ/	96.5	0.95	0.96
ع	/ ʕ/	94.2	0.93	0.92
غ	/ ɣ/	93.6	0.93	0.92
ف	/f/	98.1	0.97	0.97
ق	/q/	95.7	0.95	0.953
ك	/k/	96.2	0.953	0.95
ل	/l/	96.9	0.93	0.95
م	/m/	94.0	0.89	0.90
ن	/n/	97.2	0.94	0.95
ه	/ ħ /	96.8	0.943	0.94
و	/w/	96.7	0.95	0.95
ي	/y/	95.4	0.92	0.93

4.4 Results of phone variation-based model

The speech analysis results using the Phone variation model deal with a variation of a single phone as shown in Table 6. We detect the partially changed pronunciation mistakes of all 28 phonemes. The accuracy, precision, and recall obtained by each phone variations are presented. The average accuracy obtained on 28 phonemes is 97%. We observe from the result that the accuracy of 89% is achieved for phone ا while precision and recall of that phone are 0.87 and 0.89, respectively. Highest accuracy of 97% achieved by phones /h/ and / š/ using Phone variation model. The variation of phones /h/ and / š/ is highly distinguishable. The phone /ʔ/ obtained the least accuracy of 89% that means variation of that phones is hard to distinguish or we can say that variation of that phones is very close to each other.

We compare the results of the deep clustering-based method with the baseline method described in the previous section. The result shows that our proposed methods to detect pronunciation errors due to confusing phonemes outperform the baseline method in terms of accuracy. Figure 13 shows that the /t/, /ţ/ phoneme pair achieves an accuracy of 86% using baseline method while 97.4% using deep clustering method; similarly, we see that at each phoneme pair, deep clustering method outperforms the baseline method.

4.5 Comparison with state-of-the-art methods:

To check the effectiveness of our proposed system, we compare our proposed methods with state-of-the-art techniques as shown in Table 7. Most of the techniques deal with phonemic errors or completed changed pronunciation error detection.

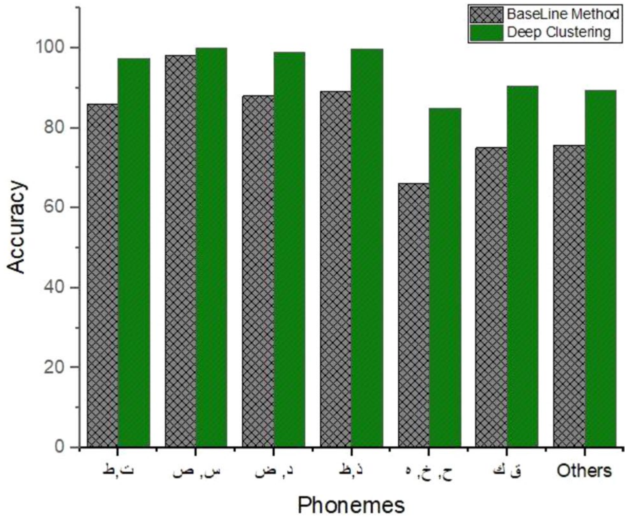


Fig. 13 Performance of baseline method and deep clustering-based method for mispronunciation detection of confusing phonemes

Table 7 Comparison of the proposed techniques with state-of-the-art methods

Sr.#	Technique	Dataset/ # of datasets	Error type		Techniques	Language		Accuracy
			Phonemic	Prosodic		Arabic	Other	
1	Georgouls [27]	pseudo word /asa/ collected from 36 children	✓		SVM with wavelets		✓	0.778
2	Abdou [28]	manually constructed database of Holy Qur`an recitation	✓		Confidence scoring	✓		0.62
3	Kun Li. [29]	English speech recordings from Mandarin and Cantonese speakers		✓	DBN		✓	0.80
4	Al Hindi [30]	Five Arabic phonemes data from (KSU) Arabic Speech Database	✓		GOP	✓ ✓		0.87–0.1
5	Kun Li [31]	TIMIT, CU-CHLOE (2)	✓		multi-distribution DNN			0.833
6	Muazzam et al. [32]	Arabic phoneme dataset	✓		ANN	✓		0.827
7	Muazzam et al. [33]	Two confusing Arabic phoneme pairs dataset	✓		Classifier + APF	✓		0.954
8	Muazzam et al.[34]	five Arabic phonemes datasets	✓		SVM	✓		0.97
9	Nazir et al [26]	Manually constructed Arabic phoneme dataset	✓		Transfer learning	✓		0.92
10	Proposed Method 1	Arabic phoneme dataset	✓		deep clustering + SVM	✓		0.94
11	Proposed method 2	Arabic phoneme dataset		✓	PVM + SVM	✓		0.97

We compare our proposed method to handle phonemic errors with [27, 28, 30, 31, 33, 34] state-of-the-art methods. The dataset used by Al Hindi, Abdou, Muazzam and Nazir is of Arabic, while rest of the state of the art work was evaluated on different language dataset. Our method achieves an accuracy of 94% that is higher than all the methods except Muazzam work [33] and [34]. The reason for the high accuracy of Muazzam's work is that in both methods, he used 2 and 5 confusing phonemes while we use all 28 phonemes of Arabic. Muazzam et al. work achieves an accuracy of 82% on all 28 phonemes which is less than our proposed method. Our proposed method also outperformed Nazir et al. work and enhance the accuracy by 2%. The proposed method to handle prosodic errors achieved the accuracy of 97% and compare the result with Kun Li. [29] work which also deal prosodic errors but he worked on English speech while we use Arabic dataset..

Pronunciation mistakes are broadly categorized into phonemic and prosodic errors. Phonemic and prosodic awareness are both phonological cycles that work at various levels: the former at the degree of the individual sound section and the latter at the supra-segmental level across syllables. For correct pronunciation of each Arabic phoneme, both phonemic and prosodic errors should be corrected. We proposed two methods, one for phonemic errors and the other for prosodic errors. As the nature of errors are different, only one method is not enough for learning all pronunciation mistakes efficiently. The purpose of our research work is to cover the pronunciation mistakes of all Arabic phonemes (28) instead of considering only confusing phonemes. Phonemic error correction and prosodic error correction can be benefited from each other in such a way that once a second language learner can differentiate and know the correct pronunciation of the confusing phonemes (6 pairs), then he easily learns the variation of other phonemes. Both methods in integrated form help the second language learner to learn the language efficiently.

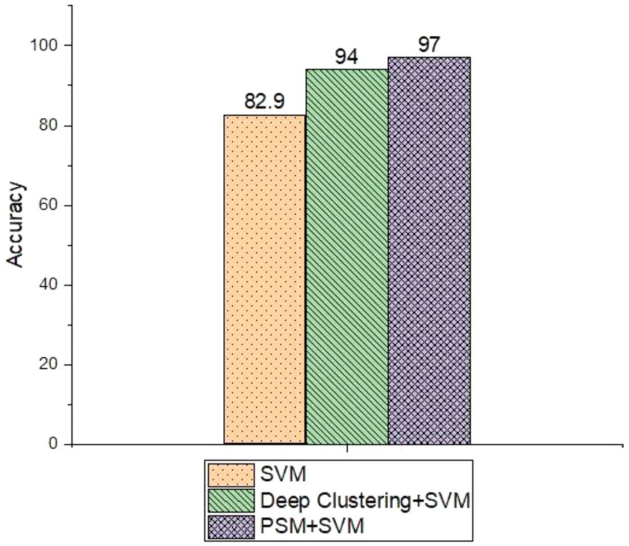


Fig. 14 Comparison of baseline, deep clustering, and PVM-based methods for speech analysis and feedback

5 Conclusion

In this paper, we proposed two techniques to detect pronunciation mistakes of Arabic phonemes that used unsupervised methods instead of supervised methods**.** We compare the result of our two techniques with the baseline method as shown in Fig. 14. The baseline method achieves an accuracy of 82.90%; speech analysis to detect phonemic errors using deep learning achieves an accuracy of 94% and outperforms the baseline method. Speech analysis to detect prosodic errors using the PVM method achieves an accuracy of 97% and outperforms the baseline as well as the deep clustering method.

In our first technique, we extract the features of 6 confusing pairs using a deep convolutional neural network and apply a simple clustering algorithm to obtain the pseudo labels. We apply KNN, Naïve Bayes, and SVM for classification. We achieve an accuracy of 83% by KNN, 86% by Naïve Bayes, and 94% using SVM. So, SVM outperforms the KNN and Naïve Bayes in terms of accuracy.

In our second technique, we detect pronunciation mistakes using the concept of the Phone variation model. We use the support vector machine as a classification algorithm and achieve an average accuracy of 97% on all 28 phonemes. Experimental analysis shows that both techniques outperform the baseline method and speech analysis using the Phone variation model outperform the speech analysis using deep clustering. Our techniques introduced the concept of unsupervised learning and detect pronunciation mistakes due to pronunciation variation.

Funding Not applicable.

Availability of data and material The data can be provided on request.

Code availability Not applicable.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

1. Precoda, K., Halverson, C.A., Franco, H.: Effects of speech recognition-based pronunciation feedback on second-language pronunciation ability. Proc. InSTILL **2000**, 102–105 (2000)
2. Panda, S.P., Nayak, A.K.: An efficient model for text-to-speech synthesis in Indian languages. Int. J. Speech Technol. **18**(3), 305–315 (2015)
3. Franco, H., Neumeyer, L., Kim, Y., Ronen, O.: Automatic pronunciation scoring for language instruction. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, IEEE, pp. 1471–1474 (1997)
4. Neumeyer, L., Franco, H., Weintraub, M., Price, P.: Automatic text-independent pronunciation scoring of foreign language student speech. In: Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96, IEEE, pp. 1457–1460 (1996)
5. Witt, S.M.: Automatic error detection in pronunciation training: Where we are and where we need to go. In: International Symposium on Automatic Detection of Errors in Pronunciation Training, Stockholm, Sweden (2012)
6. Hafen, R.P., Henry, M.J.: Speech information retrieval: a review. Multimed. Syst. **18**(6), 499–518 (2012)
7. Franco, H., Neumeyer, L., Ramos, M., Bratt, H.: Automatic detection of phone-level mispronunciation for language learning. In: Sixth European Conference on Speech Communication and Technology (1999)
8. Witt, S.M., Young, S.J.: Phone-level pronunciation scoring and assessment for interactive language learning. Speech Commun. **30**(2–3), 95–108 (2000)
9. Zhang, F., Huang, C., Soong, F.K., Chu, M., Wang, R.: Automatic mispronunciation detection for Mandarin. In: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, IEEE, pp. 5077–5080 (2008)
10. Young S., Kershaw, S., Odell, J., Ollason, D., Valtchev, V., Woodland, P.: The HTK Book (for HTK Version 3.0) (2000)
11. Ito, A., Lim, Y.-L., Suzuki, M., Makino, S.: Pronunciation error detection method based on error rule clustering using a decision tree. In: Ninth European Conference on Speech Communication and Technology (2005)
12. Jiang, H.: Confidence measures for speech recognition: A survey. Speech Commun. **45**(4), 455–470 (2005)
13. Rose, R.C., Juang, B.-H., Lee, C.-H.: A training procedure for verifying string hypotheses in continuous speech recognition. In: International Conference on Acoustics, Speech, and Signal Processing, IEEE, pp. 281–284 (1995)
14. Wessel, F., Schluter, R., Macherey, K., Ney, H.: Confidence measures for large vocabulary continuous speech recognition. IEEE Trans. Speech Audio Process. **9**(3), 288–298 (2001)
15. Zhang, R., Rudnicky, A.I.: Word level confidence annotation using combinations of features. In: Seventh European Conference on Speech Communication and Technology (2001)
16. Liu, Y., Fung, P.: Modeling partial pronunciation variations for spontaneous Mandarin speech recognition. Comput. Speech Lang. **17**(4), 357–379 (2003)
17. Riley, M., Byrne, W., Finke, M., Khudanpur, S., Ljolje, A., McDonough, J., Nock, H., Saraclar, M., Wooters, C. and Zavaliagkos, G. (1999)Stochastic pronunciation modelling from hand-labelled phonetic corpora. Speech Communication, 29(2-4), pp.209–224
18. Minhas, R.A., Javed, A., Irtaza, A., Mahmood, M.T., Joo, Y.B.: Shot classification of field sports videos using alexnet convolutional neural network. Appl. Sci. **9**(3), 483 (2019)
19. Wei, S., Hu, G., Hu, Y., Wang, R.-H.: A new method for mispronunciation detection using support vector machine based on pronunciation space models. Speech Commun. **51**(10), 896–905 (2009)
20. Lu, L., Zhang, H.-J.: Unsupervised speaker segmentation and tracking in real-time audio content analysis. Multimed. Syst. **10**(4), 332–343 (2005)
21. Lu, L., Jiang, H., Zhang, H.: A robust audio classification and segmentation method. In: Proceedings of the ninth ACM international conference on Multimedia, pp. 203–211 (2001)
22. Lu, L., Li, S.Z., Zhang, H.-J.: Content-based audio segmentation using support vector machines. In: IEEE International Conference on Multimedia and Expo, 2001. ICME 2001, IEEE, pp. 749–752 (2001)

23. Li, D., Sethi, I.K., Dimitrova, N., McGee, T.: Classification of general audio data for content-based retrieval. Pattern Recogn. Lett. **22**(5), 533–544 (2001)
24. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted Gaussian mixture models. Dig. Signal Process. **10**(1–3), 19–41 (2000)
25. Khan, M.K.S., Al-Khatib, W.G.: Machine-learning based classification of speech and music. Multimed. Syst. **12**(1), 55–67 (2006)
26. Nazir, F., Majeed, M.N., Ghazanfar, M.A., Maqsood, M.: Mispronunciation detection using deep convolutional neural network features and transfer learning-based model for arabic phonemes. IEEE Access **7**, 52589–52608 (2019)
27. Georgoulas, G., Georgopoulos, V.C., Stylios, C.D.: Speech sound classification and detection of articulation disorders with support vector machines and wavelets. In: Engineering in Medicine and Biology Society, 2006. EMBS'06. 28th Annual International Conference of the IEEE, IEEE, pp. 2199–2202 (2006)
28. Abdou, S.M., Hamid, S.E., Rashwan, M., Samir, A., Abdel-Hamid, O., Shahin, M., Nazih, W.: Computer aided pronunciation learning system using speech recognition techniques. In: Ninth International Conference on Spoken Language Processing (2006)
29. Li, K., Qian, X., Kang, S., Meng, H.: Lexical stress detection for L2 English speech using deep belief networks. In: Interspeech, pp 1811–1815 (2013)
30. Al Hindi, A., Alsulaiman, M., Muhammad, G., Al-Kahtani, S.: Automatic pronunciation error detection of nonnative Arabic Speech. In: Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on, 2014. IEEE, pp. 190–197 (2014)
31. Li, K., Qian, X., Meng, H.: Mispronunciation detection and diagnosis in l2 english speech using multidistribution deep neural networks. IEEE/ACM Trans. Audio Speech Lang. Process. **25**(1), 193–207 (2017)
32. Maqsood, M., Habib, H.A., Nawaz, T.: An efficientmis pronunciation detection system using discriminative acoustic phonetic features for arabic consonants. Int. Arab. J. Inf. Technol. **16**(2), 242–250 (2019)
33. Maqsood, M., Habib, H., Anwar, S., Ghazanfar, M., Nawaz, T.: A comparative study of classifier based mispronunciation detection system for confusing arabic phoneme pairs. Nucleus **54**(2), 114–120 (2017)
34. Maqsood, M., Habib, H.A., Nawaz, T., Haider, K.Z.: A complete mispronunciation detection system for Arabic phonemes using SVM. Int. J. Comput. Sci. Netw. Sec. (IJCSNS) **16**(3), 30 (2016)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.