Paper

Phonetic Segmentation using a Wavelet-based Speech Cepstral Features and Sparse Representation Classifier

Ihsan Al-Hassani, Oumayma Al-Dakkak, and Abdlnaser Assami

Higher Institute for Applied Science and Technology HIAST, Damascus, Syria

https://doi.org/10.26636/jtit.2021.153321

Abstract—Speech segmentation is the process of dividing speech signal into distinct acoustic blocks that could be words, syllables or phonemes. Phonetic segmentation is about finding the exact boundaries for the different phonemes that composes a specific speech signal. This problem is crucial for many applications, i.e. automatic speech recognition (ASR). In this paper we propose a new model-based text independent phonetic segmentation method based on wavelet packet speech parametrization features and using the sparse representation classifier (SRC). Experiments were performed on two datasets, the first is an English one derived from TIMIT corpus, while the second is an Arabic one derived from the Arabic speech corpus. Results showed that the proposed wavelet packet decomposition features outperform the MFCC features in speech segmentation task, in terms of both F1-score and R-measure on both datasets. Results also indicate that the SRC gives higher hit rate than the famous k-Nearest Neighbors (k-NN) classifier on TIMIT dataset.

Keywords*—*Arabic speech corpus, ASR, F1-score, phonetic segmentation, sparse representation classifier, TTS, wavelet packet.

1. Introduction

The phonetic segmentation technique aims for identifying the starting and ending boundaries of each phoneme segment in continuous speech. This segmentation is crucial for creating phoneme databases used in text-to-speech (TTS) systems [1]–[3], and for transcribing speech corpus used in training hidden Markov models (HMMs) in ASR systems. Phonetic segmentation is also used in building a query-by-example (QbyE) spoken term detection (STD) application which is relatively a new application drawing increasing attention in recent years [4]. Knowledge of phoneme boundaries is also necessary in some cases of health-related research on human speech processing [4], such as diagnostic marker for Childhood Apraxia of Speech (CAS) [5]. Phonetic segmentation and annotation can be done either automatically or manually by expert phoneticians [6]. The main difficulty of this task is its subjectivity, because of the lack of distinct physiological or acoustic events that signal a phoneme boundary in some cases. In continuous speech, phoneme boundaries are sometimes difficult to locate due to glottalization, extremely reduced vowels or gradual decrease in energy before a pause [5]. As a result, there is no “correct” answer to the phoneme segmentation problem. Instead, a measure of the agreement between two alignments takes place, such as the agreement between two humans, or the agreement between human and machine [5]. Though manual segmentation is the most adequate [7] way for phonetic transcription. It suffers from being very tedious and time consuming, especially in the case of large speech corpora and spontaneous speech. In addition, manual segmentation suffers from labeler subjectivity and may not be able to maintain labeling consistency [8]. These difficulties stimulate the development of automatic phonetic segmentation techniques for continuous speech waveforms. These segmentation techniques are divided into two major categories: text-dependent (TD) and text-independent (TI) [9], [10]. In TD techniques, the phonetic annotation of the speech signal is already known and we need only to find the boundaries of each phoneme segment. Most text dependent segmentation techniques (also called explicit) are based on HMM with forced alignment Viterbi algorithm [9], [11]. On the other hand, TI segmentation methods (also called implicit or unsupervised) do not need any phonetic annotation for the speech signal to be segmented. Instead, they are generally based on sets of rules derived from encoding human knowledge to segment speech [12], like acoustic rate of change or other spectral variation metrics [13]–[15]. Such methods are called blind or model-free because they do not use modeling stage. Recently several studies proposed using different supervised and unsupervised machine learning techniques like ANN for phoneme segmentation [16], [17].

Sparse representation classifier [18]–[20] is relatively new machine learning technique that has demonstrated excellent performance in face recognition applications [18] and other applications [19], [20]. This classifier is based on extracting sparse code as discriminative features. Sainath *et al.* used Sparse coding for phoneme classification [21], [22] from test samples on a dictionary composed of phoneme exemplars as discriminative features, and fed these sparse codes to the sparse representation classifier. Sivaram *et al.* in [23] proposed employing sparse coding for phoneme recognition. They used the sparse code as a new speech feature to train multi-layer perceptron (MLP) network to get the posterior probabilities that will be used as emission likelihood of the HMM states. Every phoneme is modeled as a 3 state HMM and Viterbi decoder is used for phoneme recognition.

In this paper, we propose to use SRC for phoneme border detection and a speech parametrization algorithm based on the equivalent rectangular bandwidth (ERB) [24], [25] like wavelet packet decomposition entitled WP-ERB. The performance of the proposed classifier is compared to the k-NN classifier, and the performance of the proposed WP-ERB features are compared to the MFCC features in phoneme segmentation. In Section 2 we present related works for phoneme segmentation. The proposed phonetic segmentation system is described in Section 3. In Section 4 we present the conducted experiments and results. In Section 5 we summarize and conclude the paper.

2. Related Works

Significant work has been done on the problem of text independent speech segmentation. Some works used a set of rules derived from encoding human knowledge to segment speech [12], like acoustic rate of change, or other spectral variation metrics [13], [14]. Such methods are called model-free phonetic segmentation methods (also called metric-based or blind methods) because they do not incorporate any modeling strategy. Instead they rely on distance measures of the spectral changes among consecutive speech frames. These methods use the signal characteristics extracted in a signal analysis stage and a collection of thresholds to segment the signal [26]. The main issue with this approach is the difficulty to determine the optimal thresholds.

Javed *et al.* [27] proposed a strategy driven by cosine distance similarity scores for identifying phoneme boundaries. The proposed strategy helped in the selection of appropriate feature extraction technique for speech segmentation applications. Dusan in [28] investigated the use of spectral transition in segmentation, as he found high correlation between the maximum of the spectral transition and phoneme boundaries. The proposed method detects phoneme boundaries by looking for peaks in a spectral transition metric. Results showed an accuracy of 84.6% for frames of 20 ms TIMIT dataset, while no other performance metric was reported. Ramteke *et al.* [29] noted that in a well-spoken word, phonemes can be characterized by the changes observed in speech waveform. To get phoneme boundaries, Ramteke studied the signal level properties of speech waveform i.e. changes in the waveform during transformation from one phoneme to another. He addressed the problem of phoneme level segmentation from two aspects: segmentation of phonemes between voiced and unvoiced portions, and segmentation of phonemes within voiced and unvoiced regions. He used pitch and zero-frequency filter to get the region of change from voiced to unvoiced and vice versa. The segmentation of phoneme boundaries within voiced and unvoiced regions are approximated using the properties of power spectrum of correlation of adjacent frames of the signal. Finally, he proposed a finite set of rules on the variations observed in the power spectrum during phoneme transitions. The segmentation results of both approaches are combined to get the final phoneme boundaries. Three databases were used to test the proposed approach. An accuracy of 95.40%, 96.87% and 96.12% is achieved within the tolerance range of 10 ms respectively.

Recently several studies proposed using different supervised and unsupervised machine learning techniques to build a discriminative model that can be used in the phoneme segmentation task [4], [16], [17]. These methods are called model-based methods. Modeling stage is performed using either supervised or unsupervised approaches. Recently self-supervised learning algorithm was used for phoneme segmentation task [17].

In literature, research studies proposed various types of modeling approaches, like generalized gamma distribution model [30], graphical models [31], microcanonical multiscale formalism (MMF) [10], and acoustic segment modeling (ASM) [4]. Supervised and unsupervised machine learning techniques like ANN [16], [17], and genetic algorithm (GA) [32] were also used to learn the discriminative acoustic models.

Inspired by the success of using neural networks in speech recognition, different studies [16], [17] considered applying them to phoneme segmentation task. Different types of ANN were investigated. Dinler *et al.* [16] and Wang [33] suggested using gated recurrent unit (GRU) recurrent neural networks, while Kreuk [34] and Franke [35] proposed using bidirectional long-short term memory (LSTM) network. Lu [36] investigated the use of segmental recurrent neural network (RNN) for feature extraction. Lee [37] proposed using the cross-entropy loss with connectionist temporal classification loss in deep speech architecture for phoneme segmentation in view of performing speech synthesis. Wang [33] observed through experiments on the TIMIT corpus that GRU forget gate activations in trained recurrent acoustic neural networks correlate very well with phoneme which makes them preferable architecture for the task of boundary detection task. The advantage of both GRU and LSTM over standard RNNs lies in their ability to incorporate long temporal context information, and thus they give higher performance [16]. The GRU ensures the control of the information flow, similar to the LSTM unit, but without a need to utilize a memory unit [16]. The GRU has a simpler structure compared to standard LSTM models, and its popularity is gradually increasing [16].

Kreuk *et al.* [17] proposed a self-supervised representation learning (SSL) model for phoneme boundary detection. They proposed learning a feature representation from the raw waveform to identify spectral changes that match phoneme boundaries accurately. For this task, they designed a convolutional neural network (CNN) to distinguish between pairs of adjacent frames and pairs of random distractor pairs. At test time, a peak detection algorithm is applied over the model outputs to produce the final boundaries [17]. Results show that the proposed SSL technique surpasses other unsupervised segmentation techniques.

All previous works used MFCC as acoustic features, though wavelet-based features has been shown to outperform MFCC in phoneme recognition application [25]. Also, no study has considered the application of the sparsity model in speech segmentation though it has achieved good success in many applications like noise robust ASR application [39], speech enhancement [40], [41], and speaker verification and identification [42]. In this work we propose using wavelet packet based acoustic features, as well as we examine the usefulness of sparse representation classifier SRC in phonetic segmentation task.

3. Proposed Method

The proposed segmentation system contains four stages: signal pre-processing, features extraction, dictionary creation, and phoneme segmentation. At the pre-processing stage silence from speech segments are removed, and pre-emphasis filter is applied to compensate for lips effects. Speech segments are then divided into overlapped frames of length 16 ms with 4 ms overlapping. After speech framing, acoustic features are being extracted: mel frequency cepstral coefficients (MFCC), and the proposed wavelet packet-ERP features [24], [25]. The block diagram of the proposed system is depicted in Fig. 1.

3.1. Wavelet Based Feature Extraction

The proposed wavelet packet feature extraction is based on the equivalent rectangular bandwidth (ERB) like wavelet packet decomposition proposed by Sahu in [25]. The whole frequency band is decomposed into 24 sub-bands according to the wavelet packet tree shown in Fig. 2 [25]. Once the WP decomposition is performed, energy in each frequency band is calculated, and the log of weighted energy is applied resulting in 24 cepstral coefficients. Discrete cosine transform (DCT) is then applied to decorrelate the 24 coefficients of filter bank energies, and variance feature (VF) of the 24 coefficients is also calculated. Finally, a total of 25 features are obtained for each frame.

Figure 3 illustrates the block diagram of the proposed wavelet packet-based feature extraction WP-ERP algorithm. We examined different types of wavelet filters with different degrees, such as Daubechies, coiflets and symlets filters. Experiments showed that the Coif5 filter gives the best performance in terms of segmentation accuracy.

3.2. Exemplar Dictionary Creation

At the dictionary creation stage, the feature vectors of the training phoneme/borders samples are warped together to form one matrix – the exemplar dictionary. The frame is labeled as phoneme (not a border, class 1) if either it does not contain a border (a border is a transition between two phonemes according to the manual annotation), or if most of the frame belongs to one phoneme, i.e. the border is not at the very start or the very end of the frame. On the other hand, the frame is labeled as border (class 2) if it contains a border between two phonemes and if the frame contains good percentage of both phonemes. The interval

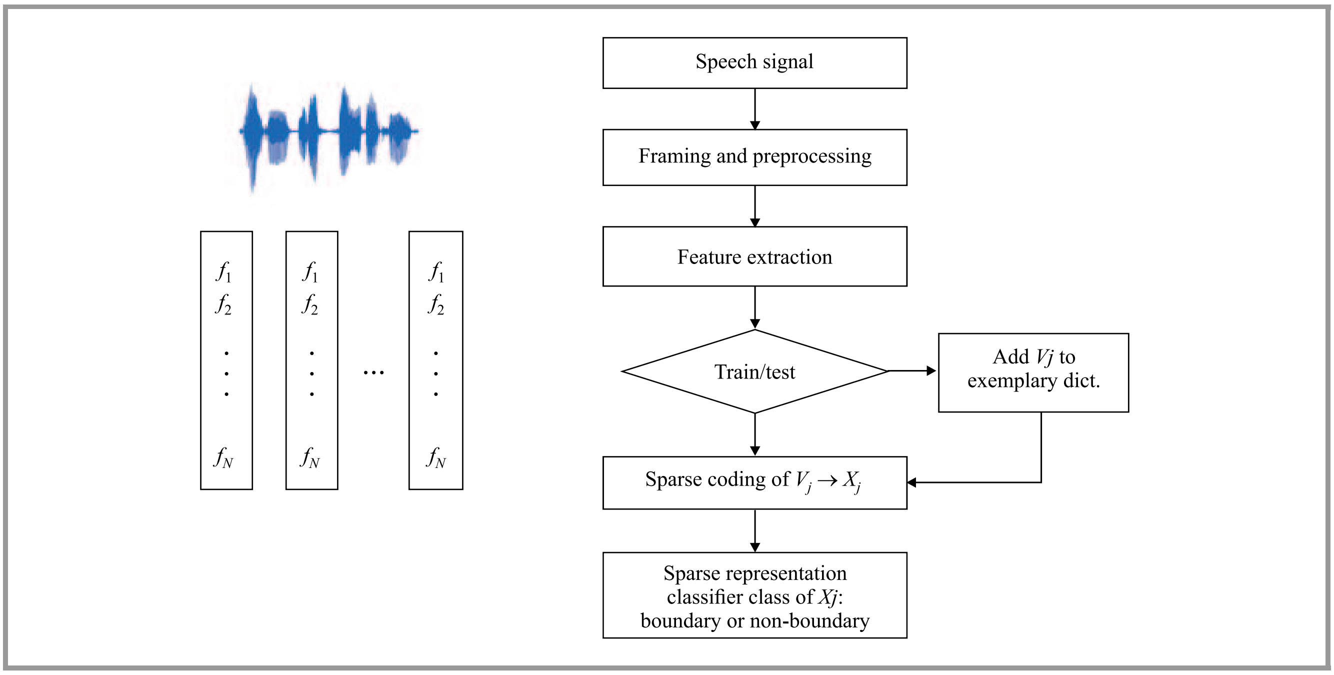
f_1
f_2
f_N
f_1
f_2
f_N
...
f_1
f_2
f_N
Speech signal
Framing and preprocessing
Feature extraction
Train/test
Add Vj to exemplary dict.
Sparse coding of $V_j \rightarrow X_j$
Sparse representation classifier class of Xj: boundary or non-boundary

Fig. 1. Proposed phonetic segmentation system using SRC.

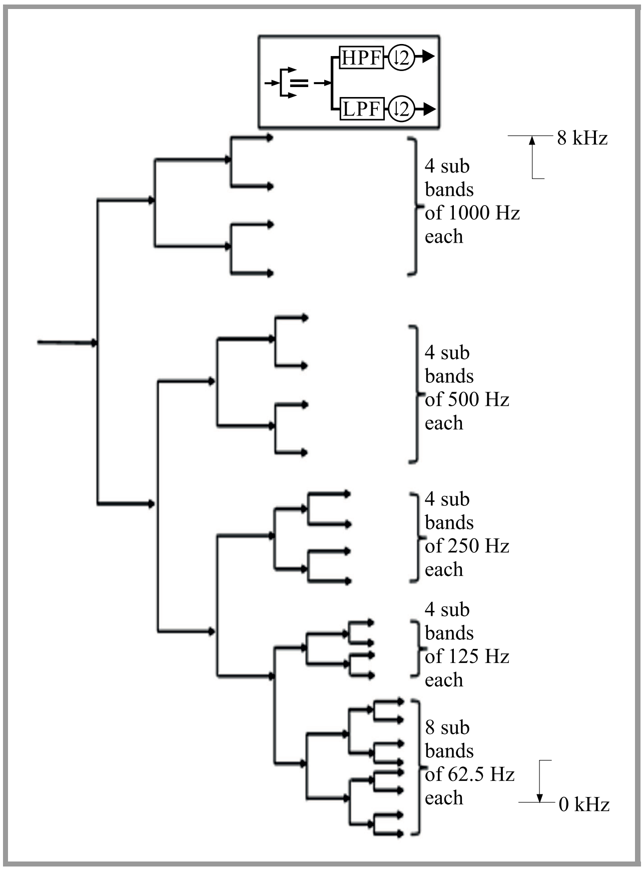
HPF
↓2
LPF
↓2
8 kHz
4 sub bands of 1000 Hz each
4 sub bands of 500 Hz each
4 sub bands of 250 Hz each
4 sub bands of 125 Hz each
8 sub bands of 62.5 Hz each
0 kHz

Fig. 2. 24 sub-band wavelet packet tree based on ERB scale.

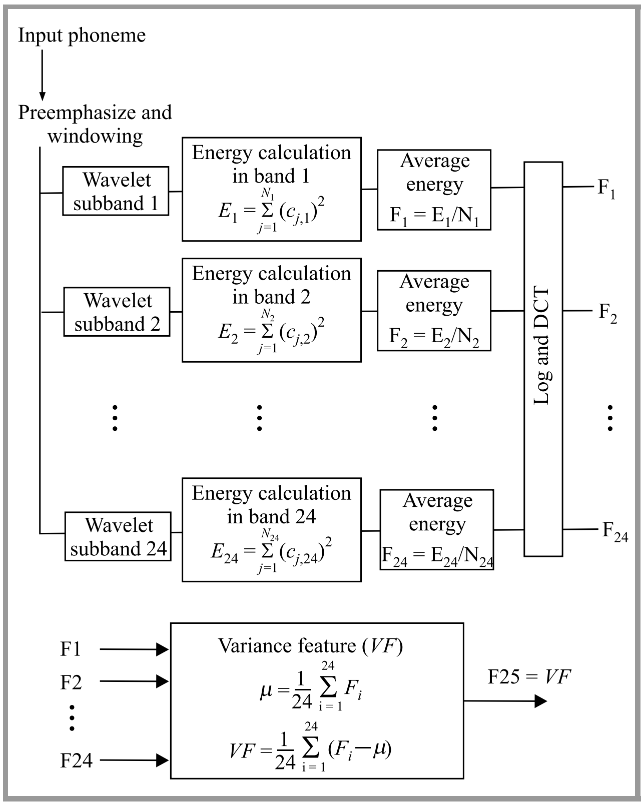
Input phoneme
Preemphasize and windowing
Wavelet subband 1
Energy calculation in band 1
$E_1 = \sum_{j=1}^{N_1} (c_{j,1})^2$
Average energy
$F_1 = E_1/N_1$
Wavelet subband 2
Energy calculation in band 2
$E_2 = \sum_{j=1}^{N_2} (c_{j,2})^2$
Average energy
$F_2 = E_2/N_2$
Wavelet subband 24
Energy calculation in band 24
$E_{24} = \sum_{j=1}^{N_{24}} (c_{j,24})^2$
Average energy
$F_{24} = E_{24}/N_{24}$
Log and DCT
F_1
F_2
F_{24}
F1
F2
F24
Variance feature (*VF*)
$\mu = \frac{1}{24}\sum_{i=1}^{24} F_i$
$VF = \frac{1}{24}\sum_{i=1}^{24} (F_i - \mu)$
F25 = *VF*

Fig. 3. Block diagram of the proposed wavelet-based WP-ERP feature extraction.

that defines the very start and the very end of the frame is taken equal to 3 ms, so that the tolerance interval for boundary detection is within $16 - 2 \times 3 = 10$ ms. Figures 4 and 5 depict the labeling strategy for both classes (border and not-border).

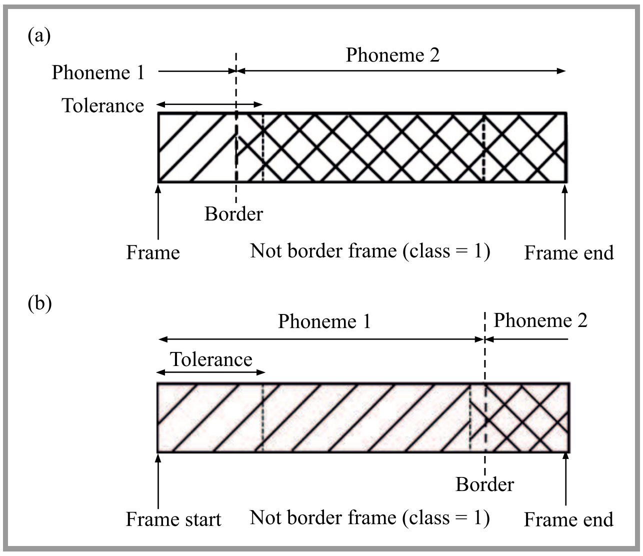

Fig. 4. Frame labeled as class 1 (not-border): the border is at the very start of the frame (a), the border is at the very end of the frame (b).

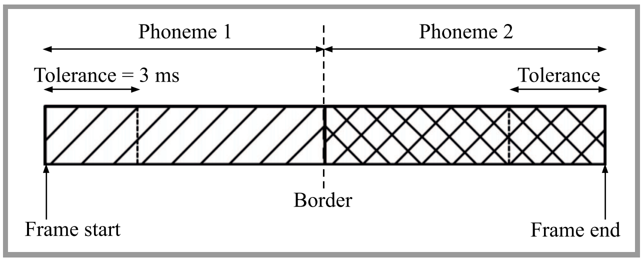

Fig. 5. Frame labeled as class 2 (border).

3.3. The Sparse Representation Classifier

The SRC is based on the sparse code, which can be defined as follows. Let $x \in \mathbb{R}^M$ be the signal that we want to encode sparsely. We suppose that there is a matrix (dictionary) $D \in \mathbb{R}^{M \times N}$, where $M \leq N$ such that y can be written as a linear combination of at most k columns of D, and $k \ll N$ is called the sparsity degree. The "sparse coding" problem [42] is to find the vector $a \in \mathbb{R}^N$ that contains only k non-zero elements such that:

$$(P0)\ \underset{x \in \mathbb{R}^N}{\min} ||a||_{\ell} l_0 \quad \text{s.t.} \quad x = Da \ , \tag{1}$$

where $||a||_0$ is the l_0 pseudo-norm which represents the number of non-zero elements in a. The vector a which contains the sparse (few out of many) coefficients of the linear combination of the elements (called atoms) of the dictionary D, that represents the signal x is called the sparse code. As D is over-complete (number of rows is less than the number of columns), this is an ill-posed inverse problem.

As the signal x might be corrupted with noise, the previous problem has been reformulated as:

$$(P_{0,\varepsilon}) \quad \min_{a\in\mathbb{R}^N} ||a||_0 \text{ s.t. } ||x-Da||_2 \leq \varepsilon \ . \qquad (2)$$

This is a non-convex optimization problem that has been proved to be an NP-hard [38]. Replacing the l_0 norm by the l_1 norm "convexify" the problem and gives an equivalent problem called the basis pursuit denoising problem BPDN [39]:

$$(BPDN) \min_{a\in\mathbb{R}^N} ||a||_1 \text{ s.t. } ||x-Da||_2 \leq \varepsilon \ , \qquad (3)$$

where $||a||_1 = \Sigma_1^N |a_i|$, and ε is some noise level energy. We can take the dual of the previous problem [42]:

$$(LASSO) \min_{a\in\mathbb{R}^N} ||x-Da||_2 \text{ s.t. } ||a||_1 \leq \tau \ , \qquad (4)$$

where $\tau > 0$ is a regularization parameter, through which we control the sparsity degree (number of non-zero elements) of the sparse code a. This is called the "absolute shrinkage and selection operator" (LASSO) problem.

The SRC [18] works as follows. Having a training dataset that belongs to n class, for each class i the sub-dictionary D_i is formed by concatenating the corresponding training samples. These sub-dictionaries are wrapped together to form one dictionary D. The sparse code a of the feature vector for the test sample x is calculated using one of the sparse solvers available in the literature. For each class i, the selection operator $\delta_i(a)$ is applied on a, so that the elements of sparse code a corresponding to the sub-dictionary D_i are preserved, while all others are set to zero. Afterward, the linear approximation $D \times \delta_i(a)$ is calculated. SRC returns the class c that gives the closer approximation to the test sample x using the minimum Euclidean distance $||x - D \times \delta_i(a)||_2^2$.

Here, the SRC is calculated by:

1. Find the sparse code for the feature vector y, by solving Lasso – Eq. (4).
2. The class of x, is the index of sub-dictionary whose corresponding sub-sparse code energy is the highest:

   $$c = \arg\min_i ||x - D \times \delta_i(a)||_2^2 \ ,$$

   where $\delta_i(a)$ is a selector operator that selects the elements of sparse code a corresponding to the sub-dictionary D_i, and sets all others to zero.

In the literature [45] there are many algorithms that were developed for solving the previous sparse coding problems – Eqs. (2)–(3) and Eq. (4) – like: greedy orthogonal matching pursuit OMP, L1-minimization algorithms: GPRS, SPGL1, DALM, homotopy, L1LS. The Matlab implementations for these methods are available at [46], [47]. As many of the previous solvers include matrix inversion step, and due to the large size of the dictionary we used in our experiments, we could only use SPGL1 and OMP.

4. Results and Discussion

Experiments were conducted on two different datasets, the first one is an American English corpus derived from the TIMIT [49], and the second one is an Arabic one derived from the Arabic speech corpus [50].

TIMIT is one of the standards and phonetically balanced read speech English corpus, used in three domains: phoneme segmentation, phoneme classification and phoneme recognition systems to develop and evaluate the performance of these systems. This corpus consists of 6300 sentences recorded at 16 kHz rate with 16-bit sample, for the eight major dialects of American English spoken by 630 different speakers (438 males and 192 females), ten sentences for each [51]. These sentences are distributed in two sets, the training set with 4620 utterances from 462 speakers and the test set with 1344 sentences from 168 speakers. All sentences were segmented and labeled manually at the phoneme level.

TIMIT original transcriptions are based on 61 phonemes. Table 1 shows the TIMIT phoneme set, classified into voiced phonemes and unvoiced phonemes.

For experiments, a subset of 380 utterances from the complete set was used for training. Another subset of 100 utterances from complete test set was used for testing. We have excluded the "dialect" sentences (SA sentences) for both training and testing. Boundaries between two pauses, including stop closures, were also excluded from evaluation.

Arabic speech corpus [50] is a modern standard Arabic (MSA) speech corpus for speech synthesis. It contains phonetic and orthographic transcriptions of more than 3.7 hours of MSA speech aligned with recorded speech on the phoneme level. The annotations include word stress marks on the individual phonemes [52]. The corpus includes 1813 utterances recorded by a single speaker, with a 16-bit, 48 kHz speech waveform file for each utterance,

Table 1
TIMIT phoneme set (61 phonemes)

Voiced/ unvoiced	Type	Phonemes
Voiced	Vowels	iy, ih, eh, ey, ae, aa, aw, ay, ah, ao, oy, ow, uh, uw, ux, er, ax, ix, axr, ax-h
	Glides/ semi-vowels	l, r, w, y, hh, hv, el
	Stops	b, d, g
	Fricative	s, sh, f, th
	Nasal	m, n, ng, em, en, eng, nx
Unvoiced	Stops	p, t, k, dx, q
	Affricative	jh, ch
	Fricative	z, zh, v, dh
Pause and stop closures		pau, epi, h#, dcl, kcl, gcl, tcl, pcl, bcl

and a corresponding Praat [53] textgrid file for annotation. The annotation is based on a set of 82 Arabic phoneme. In out experiments, we used 200 utterances for training and another 100 utterances for testing. The wave files are down sampled to 16 kHz.

4.1. Performance Metrics

We used 6 common metrics (described below) that are generally used to assess phonetic segmentation algorithm. In the case of text independent segmentation (TI) techniques, the number of discovered segments might differ from the number of segments produced by manual segmentation. TI segmentation can be viewed as a boundary detection problem. The assessment of any detection algorithm is done by measuring: how much it truly detects, what should be detected, how much it truly rejects and what should be rejected:

Hit rate is a metric that measures how good an algorithm truly detects the goal. When a detected boundary matches corresponding boundary in the reference signal, this is called a hit rate (also called recall, RCL). It can be calculated as:

$$Hit\ rate = RCL = \frac{CDB}{ATB}\ 100\%\ , \tag{5}$$

where CDB is the number of correctly detected boundaries, and ATB is the number of all true boundaries. A hit rate of 100% implies that the algorithm is perfect in detecting boundaries, but it might detect non-boundaries frames and misclassify them as boundaries. For this issue, we use the precision measure.

Precision (PRC) is a metric that measures how precise the detection is, i.e. how good the algorithm is in detecting only what should be detected. It can be calculated as:

$$PRC = \frac{CDB}{CDB+IB} = \frac{CDB}{ADB}\ 100\%\ , \tag{6}$$

where ADB is the number of all detected boundaries (true and false) by the algorithm, and IB is the number of inserted boundaries (false detection). A precision of 100% means that the algorithm does not fire a false alarm which means detecting false boundaries. This is called over-segmentation error.

Specificity is a metric that measures how good the algorithm is in rejecting what should be rejected, and this is calculated as follows:

$$Specficity = \frac{AP-CDB}{AP-ATB}100\%\ , \tag{7}$$

where AP is the number of all points (frames in our case). We can see that a higher hit rate might come at the expense of lower specificity, and lower precision. Thus, hit rate and precision are not good metrics for assessing the overall performance of segmentation algorithm, as the increase in one of them might cause a decrease in the other. The overall objective effectiveness of the segmentation algorithm can be evaluated by three different measures: accuracy, the F1-score, and the R-measure [43].

Accuracy measures how accurate the algorithm is in both detection and rejection, and it is calculated by the formula:

$$Accuracy = \frac{CDB+CDNB}{AP}100\%\ , \tag{8}$$

where CDNB is the number of all true points detected as non-boundaries.

F1-score is the harmonic mean of recall and precision, which is used for assessing classification and prediction algorithms. It is calculated according to:

$$F1 = \frac{2\ PRC \times RCL}{PRC+RCL}\ . \tag{9}$$

F1-score takes its value in the unit interval between $0,\dots,1$, where the score closer to 1 is better. A system with high recalls but low precision returns many results, but most of its predicted labels are incorrect. A system with high precision but low recall is just the opposite, returning very few results, but most of its predicted labels are correct. An ideal system with high precision and high recall will return many results, with all results labeled correctly [35].

Optimizing the operation of a speech segmentation algorithm is often a tradeoff between hit-rate and over-segmentation (or inversely, false-alarm rate and miss-rate) [48]. F1-score is one possible way to describe overall performance of an algorithm with a single value. However, F1-score is prone to stochastic hit-rate increases due to the over-segmentation issue [48].

R-value is a new distance measure proposed to describe performance using a single value that properly penalizes over-segmentation [48]. The optimal goal of segmentation is to achieve a hit-rate of 100% and an over-segmentation of 0%. This is called the target point (TP). The basis of the new metric is the algorithm's distance from TP and not the (hit-rate) gain achieved by over-segmentation.

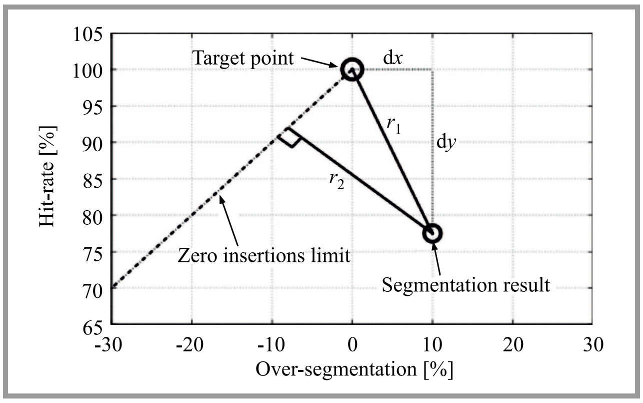

Fig. 6. Calculating R-measure [48].

On the segmentation performance plane illustrated in Fig. 6, a distance r_1 is derived (Eq. (10)) and a distance r_2 is measured (Eq. (11)), to appreciate the value of undersegmentation compared to over-segmentation in the algorithm (i.e. less false positives).

$$r_1 = \sqrt{(100 - HR)^2 + OS^2}\ , \qquad (10)$$

$$r_2 = \frac{1}{\sqrt{2}}\ (-OS + HR - 100)\ . \qquad (11)$$

The distances r_1 and r_2 are then added together and normalized to have a maximum value of 1 at the target-point (Eq. (12)). This new distance measure is referred to as the R-value:

$$R = 1 - \frac{\text{abs}(r_1) + \text{abs}(r_2)}{200}\ . \qquad (12)$$

R-value decreases as the distance to the target grows, similarly as F1-score does, but it makes more emphasis on over-segmentation by arguing that better hit rates might be achieved by simply adding random boundaries without any algorithmic improvement. This measure evaluates how close one is to the ideal segmentation R= 1.

4.2. Results

The performance of the proposed WP-ERB features was compared, against the performance of the well-known MFCC features. We have calculated the MFCC features using 24 mel coefficients[1], so that their dimensionality is close to that of the WP-ERB. All reported results are for boundary detection within a tolerance of 10 ms, and averaged over 10 random runs. Experiments were performed using a training set of 40,000 frames (20,000 frames for each class) and a test set of 6000 frames (3000 for each class). We tried different values for the sparsity regularization parameter τ in Eq. (4), and found that $\tau = 0.98$ gives that highest performance.

Table 2 shows the performance of the proposed segmentation system using the MFCC features and the proposed WP-ERB features, for both TIMIT and Arabic speech corpus. For TIMIT dataset, though MFCC achieves higher hit rate, but this at the expense of considerably lower specificity and lower precision. The overall performance of the proposed WP-ERB features gives the highest performance in terms of accuracy, F1-score, R-value, specificity and precision. The gain in terms of accuracy is about 2%, the gain in terms of precision and R-value is about 4%, and the gain in terms of specificity is about 12%, while the values of F1-score for both features are very close.

On the Arabic speech corpus, WP-ERB achieves a gain of 5% in terms of hit rate and 1.5% in terms of F1-score, over MFCC features, while the values of accuracy and R-value for both features are very close.

To assess the performance of SRC, we conducted the same segmentation scenarios using SRC, k-NN and SVM classifiers. We have not reported the results using SVM as it gave very bad performance, which can be explained by the fact that the two classes are mingled and cannot be separated by hyperplane. For k-NN tuning, we tried different values and found that $k = 80$ gives the best segmentation performance. Results are reported in Table 3.

On TIMIT dataset, we can see that the performance of the two classifiers are very close, which hints that the non-zero atoms that has the highest energies in the calculated sparse code are within the k-nearest points. SRC achieves higher hit rate at the expense of lower precision and specificity, but the overall performance in terms of F1-score is very close, though k-NN achieves a higher R-value and higher accuracy with a gain of about 2%. On the Arabic speech corpus k-NN achieves a better performance over SRC in terms of all performance parameters, with a mean gain of 1%.

Concerning the Arabic speech corpus, results show lower performance of about 2% than those on TIMIT in terms F1 score, and 6% lower in terms of hit rate. The modest results we obtained on the Arabic corpus are due to the consideration of 82 different phonemes. Some of these phonemes are so similar that they cannot be separated neither in time domain nor in the frequency domain, such as

[1] We tried different numbers for the mel coefficients of MFCC and all gave inferior performance.

Table 2
Segmentation performance: MFCC vs. WP-ERPC

Dataset	Feature	Accuracy	Hit rate	Specificity	Precision	F1-score	R-value
TIMIT	WP-ERB	**66.84**	81.60	**52.08**	**63.01**	71.11	**72.77**
	MFCC	64.60	**88.95**	40.25	59.82	**71.53**	68.86
Arabic speech corpus	WP-ERB	**65.17**	**75.55**	54.78	62.56	**68.45**	74.05
	MFCC	65.09	70.50	**59.67**	**63.62**	66.88	**74.15**

Table 3
Segmentation performance: SRC vs. k-NN classifier

Dataset	Classifier	Accuracy	Hit rate	Specificity	Precision	F1-score	R-value
TIMIT	SRC	64.67	**83.38**	45.97	60.68	70.24	70.63
	k-NN	**66.84**	81.60	**52.08**	**63.01**	**71.11**	**72.77**
Arabic	SRC	64.33	74.10	54.57	61.99	67.51	72.72
	k-NN	**65.17**	**75.55**	**54.78**	**62.56**	**68.45**	**74.05**

a phoneme and the geminated version of it. Also, in some vowels, as it is not possible to separate between the short version and the long version of the same phoneme (e.g. phoneme "a" and the phoneme "aa") if they are adjacent to each other.

We have examined different wavelet filter for extracting the proposed WP-ERB features. Results for TIMIT dataset are reported in Table 4, which shows that Coif5 gives the highest performance in terms of R-value and accuracy. The Sym8 filter gives the highest performance in terms of hit rate, with F1-score very close to this of Coif5, while a smaller R-value. Though the Haar wavelet achieves the highest specificity and precision but it has a low hit rate and thus low F1-score. In all, we can see the Coif5 has the best segmentation performance. Coiflets are the only wavelet basis that has vanishing moments of the scaling function φ [54], which is related to the "goodness" of the approximation of high-resolution scaling coefficients [54].

To assess the segmentation performance of the proposed algorithm depending on the type of phoneme boundary, we have calculated the hit rate for 3 different boundary types:

- V-V – boundaries that separate two voiced phonemes,
- U-V – boundaries that separate an unvoiced phoneme and voiced phoneme,
- U-U – boundaries that separate two unvoiced phonemes.

Results on TIMIT dataset and using SRC classifier are reported in Table 5. We can see that the boundary that separates two unvoiced phonemes are the hardest to detect achieving the lowest hit rate, while the boundary that separates two voiced phonemes are the easiest achieving the highest hit rate.

To study the effect of the size of the training set on the segmentation performance, we conducted three experiments using three training sets of different sizes: 10,000, 20,000, and 30,000 frames for each class, results are reported in Table 6. We can see that a training set of size 20,000 frames gives the best performance and increasing the size to 30,000 does not improve the performance. This can be explained by the fact that increasing the size of the training set might result in overfitting.

Table 5
Segmentation performance of WP features for different boundary types

Boundary type	Hit rate
V-V	84.34
U-V	81.69
U-U	67.73

To study the effect of the sparse coding solver we used 2 different sparse solvers: the simple greedy orthogonal matching pursuit OMP solver and SPGL-LASSO solver. Results are reported in Table 7 using the proposed WP-ERP features and a training set of 20,000 frames. We can see that though both solvers give very close hit rates, but SPGL-LASSO has a considerable increase over OMP in all other performance metrics.

Finally, in Table 8 we compared the performance of the proposed algorithm against two state of the art (SOTA) supervised phoneme segmentation on TIMIT dataset: Kreuk *et al.* [34] and Frank *et al.* [35]. Though results suggest that the proposed algorithm is inferior to the SOTA models over all metrics, but this is due to the classifier performance, as the two studies uses neural networks as classifier. The key result of this research is to show that the proposed wavelet based acoustic features outperform MFCC in the task of speech segmentation which was verified using the famous classifier k-NN and the proposed SRC.

Table 4
Segmentation performance of WP features using different wavelet filters on TIMIT dataset

Wavelet filter	Classifier	Accuracy	Hit rate	Specificity	Precision	F1-score	R-value
Sym8	SRC	64.99	**85.36**	44.62	60.65	70.91	70.27
	k-NN	66.45	82.90	50.01	62.38	**71.19**	72.08
Sym6	SRC	64.59	83.74	45.43	60.55	70.28	70.46
	k-NN	64.75	80.77	48.73	61.17	69.62	71.41
Haar	SRC	61.19	62.95	59.42	60.81	61.86	72.27
	k-NN	62.70	59.56	**65.83**	**63.55**	61.49	71.37
DB12	SRC	64.64	83.47	45.80	60.63	70.24	70.58
	k-NN	66.51	80.89	52.14	62.83	70.72	72.72
DB8	SRC	64.38	82.48	46.29	60.56	69.84	70.67
	k-NN	66.24	80.19	52.29	62.70	70.38	72.70
Coif5	SRC	64.67	83.38	45.97	60.68	70.24	70.63
	k-NN	**66.70**	81.24	52.15	**62.94**	70.93	**72.76**

Table 6
Segmentation performance using training sets for different size on TIMIT dataset

Size of training set	Features	Accuracy	Hit rate	Specificity	Precision	F1-score	R-value
10,000	WP-ERB	50.35	49.77	50.93	50.35	50.06	64.48
	MFCC	56.53	66.50	46.57	46.57	55.47	68.22
20,000	WP-ERB	**66.84**	81.60	**52.08**	**63.01**	71.11	**72.77**
	MFCC	64.60	**88.95**	40.25	59.82	**71.53**	68.86
30,000	WP-ERB	64.55	79.40	49.71	61.22	69.13	71.62
	MFCC	60.63	79.76	41.50	57.69	66.95	68.74

Table 7
Segmentation performance of WP features for different boundary types

Solver	Accuracy	Hit rate	Specificity	Precision	F1-score	R-value
OMP	59.46	77.61	41.30	56.93	65.68	68.42
SPGL-LASSO	**66.70**	**81.24**	**52.15**	**62.94**	**70.93**	**72.76**

Table 8
Comparison of phoneme segmentation models using TIMIT dataset

Model	Hit rate	Precision	F1-score	R-value	Tolerance
Kreuk *et al.* [34]	90.46	94.03	92.22	92.79	20 ms
Frank *et al.* [35]	88.10	91.10	89.60	90.80	20 ms
Proposed	66.70	62.94	70.93	72.76	10 ms

5. Conclusion

In this paper we proposed a new phonetic segmentation method based on speech parametrization technique entitled WP-ERB and sparse representation classifier. Results show that the proposed wavelet packet-based features outperform the classical MFCC features in speech segmentation task in terms of segmentation accuracy, precision, F1-score, and R-measure. The proposed WP-ERB features achieve a gain of about 4% in R-value and 2% in accuracy over MFCC on TIMIT dataset. On Arabic speech corpus the proposed WP-ERB features achieves a gain of 1.5% in terms of F1-score and 5% in terms of hit rate. We have also shown that using the SRC in phonetic segmentation achieves a higher hit rate over k-NN classifier on TIMIT dataset at the expense of lower precision and specificity, while no gain is achieved in terms of F1-score and R-value.

We think the moderate results with the Arabic corpus is due to the large number of considered phonemes (about twice the number of real phonemes). In later work, we will work on the Arabic dataset and merge the phonemes that cannot be separated and treat them as one phoneme (like geminated phonemes, short and long vowels of the same nature). As better results are obtained with TIMIT after phonemes merging, we expect the same for the Arabic corpus. This work is to be continued to see the effect of different dialects of the same language. TIMIT already contains many dialects, a comparative study will be undertaken to see the segmentation and classification performance on different dialects. On Arabic we try to collect data from other dialects, we expect some dialects far from Standard Arabic to be difficult to segment.

Speech style will also be an important point to study. As humans find sometimes difficulties in understanding some speech styles like fast speech or speech mixed with strong emotions. It will be interesting to see how far will differ the results with different speech styles.

The proposed phoneme segmentation system can further be improved by finding correlates between phonemes borders and prosodic features. Using those features together with acoustic knowledge of the phonemes, can be incorporated in a rule based to help increasing the system robustness.

References

[1] J. Glass, “A probabilistic framework for segment-based speech recognition”, *Computer Speech & Language*, vol. 17, no. 2–3, pp. 137–152, 2003 (DOI: 10.1016/S0885-2308(03)00006-8).

[2] D. T. Chappell and J. Hansen, “A comparison of spectral smoothing methods for segment concatenation based speech synthesis”, *Speech Commun.*, vol. 36, no. 3–4, pp. 343–373, 2002 (DOI: 10.1016/S0167-6393(01)00008-5).

[3] J. Adell and A. Bonafonte, “Towards phone segmentation for concatenative speech synthesis”, in *Proc. of the 5th ISCA Speech Synthesis Workshop (SSW5)*, Pittsburgh, PA, USA, 2004, pp. 139–144 [Online]. Available: https://nlp.lsi.upc.edu/papers/adell04b.pdf

[4] H. Wang, T. Lee, C. Leung, B. Ma, and H. Li, “Acoustic Segment Modeling with Spectral Clustering Methods”, in *IEEE/ACM Transac. on Audio, Speech, and Language Process.*, vol. 23, no. 2, pp. 264–277, 2015 (DOI: 10.1109/TASLP.2014.2387382).

[5] J. P. Hosom, "Speaker-independent phoneme alignment using transition-dependent states", vol. 51, no. 4, pp. 352–368, 2008 (DOI: 10.1016/j.specom.2008.11.003).

[6] J. P. van Hemert, "Automatic segmentation of speech", *IEEE Transac. on Signal Process.*, vol. 39, no. 4, pp. 1008–1012, 1991 (DOI: 10.1109/78.80941).

[7] A. Ljolje, J. Hirschberg, and J. P. H. van Santen, "Automatic speech segmentation for concatenative inventory selection", in *Progress in Speech Synthesis*, J. P. H. van Santen, R. W. Sproat, J. P. Olive, and J. Hirschberg, Eds., New York: Springer, 1997, pp. 304–311 (DOI: 10.1007/978-1-4612-1894-4_24).

[8] B. L. Pellom and J. H. L. Hansen, "Automatic segmentation of speech recorded in unknown noisy channel characteristics", *Speech Commun.*, vol. 25, no. 1–3, pp. 97–116, 1998 (DOI: 10.1016/S0167-6393(98)00031-4).

[9] G. A. Esposito, "Text independent methods for speech segmentation", *Nonlinear Speech Model. and App., Lecture Notes in Computer Sci.*, G. Chollet, A. Esposito, M. Faundez-Zanuy, M. Marinaro, Eds., Berlin, Heidelberg: Springer, 2005, vol. 3445 (DOI: 10.1007/11520153_12).

[10] V. Khanagha, K. Daoudi, O. Pont, and H. Yahia, "Phonetic segmentation of speech signal using local singularity analysis", *Digital Signal Processing*, vol. 35, no. C, pp. 86–94, 2014 (DOI: 10.1016/j.dsp.2014.08.002).

[11] D. T. Toledano, L. A. H. Gomez, and L. V. Grande, "Automatic phonetic segmentation", *IEEE Transac. on Speech and Audio Process.*, vol. 11, no. 6, pp. 617–625, 2003 (DOI: 10.1109/TSA.2003.813579).

[12] O. Scharenborg, V. Wan, and M. Ernestus, "Unsupervised speech segmentation: An analysis of the hypothesized phone boundaries", *J. of Acoustical Society of America*, vol. 127, no. 2, pp. 1084–1095, 2010 (DOI: 10.1121/1.3277194).

[13] B. D. Sarma and S. R. Mahadeva Prasanna, "Acoustic-Phonetic Analysis for Speech Recognition: A Review", *IETE Technical Review*, vol. 35, no. 3, pp. 305–327, 2017 (DOI: 10.1080/02564602.2017.1293570).

[14] M. Ziółko, J. Gałka, B. Ziółko, T. Drwięga, "Perceptual wavelet decomposition for speech segmentation", in *Proc. of the Interspeech, 11th Annual Conf. of the Int. Speech Commun. Association*, Makuhari, Chiba, Japan, 2010, pp. 2234–2237 (DOI: 10.21437/Interspeech.2010-614).

[15] D.-T. Hoang and H.-C. Wang, "Blind phone segmentation based on spectral change detection using legendre polynomial approximation", *The J. of the Acoustical Society of America*, vol. 137, no. 2, pp. 797–805, 2015 (DOI: 10.1121/1.4906147).

[16] Ö. Batur Dinler and N. Aydin, "An optimal feature parameter set based on gated recurrent unit recurrent neural networks for speech segment detection", *Appl. Sci.*, vol. 10, pp. 1273, 2020 (DOI: 10.3390/app10041273).

[17] F. Kreuk, J. Keshet, and Y. Adi, "Self-supervised contrastive learning for unsupervised phoneme segmentation", *Proc. of the Interspeech*, pp. 3700–3704, 2020 (DOI: 10.21437/Interspeech.2020-2398).

[18] J. Wright, A. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation", *IEEE Transac. on Pattern Anal. and Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009 (DOI: 10.1109/TPAMI.2008.79).

[19] X. Gu, C. Zhang, and T. Ni, "A hierarchical discriminative sparse representation classifier for EEG signal detection", *IEEE/ACM Transac. on Comput. Biol. and Bioinform.*, vol. 18, no. 5, 2020 (DOI: 10.1109/TCBB.2020.3006699).

[20] Mh. Hajigholam, A. A. Raie, K. Faez, "Using sparse representation classifier (SRC) to calculate dynamic coefficients for multitask joint spatial pyramid matching", *Iranian J. of Sci. and Technol., Trans. Electr. Eng.*, vol. 45, pp. 295–307, 2021 (DOI: 10.1007/s40998-020-00351-3).

[21] T. N. Sainath, A. Carmi, D. Kanevsky, and D. Ramabhadran, "Bayesian compressive sensing for phonetic classification", *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Process. (ICASSP)*, Dallas, TX, USA, 2010, pp. 4370–4373 (DOI: 10.1109/ICASSP.2010.5495638).

[22] T. N. Sainath and D. Kanevsky, "Sparse representations for speech recognition", A. Carmi, L. Mihaylova, S. Godsill Eds., book section "Compressed Sensing & Sparse Filtering", Berlin, Heidelberg: Springer, 2014, pp. 455–502 (DOI: 10.1007/978-3-642-38398-4_15).

[23] G. S. V. S. Sivaram, S. K. Nemala, M. Elhilali, T. D. Tran, and H. Hermansky, "Sparse coding for speech recognition", *IEEE Int. Conf. on Acoustics Speech and Signal Process. (ICASSP)*, Dallas, TX, USA, 2010, pp. 4346–4349 (DOI: 10.1109/ICASSP.2010.5495649).

[24] A. Bhowmick, M. Chandra, and A. Biswas, "Speech enhancement using Teager energy operated ERB-like perceptual wavelet packet decomposition", *Int. J. Speech Technol.*, vol. 20, pp. 813–827, 2017 (DOI: 10.1007/s10772-017-9448-7).

[25] P. K. Sahu, A. Biswas, A. Bhowmick, and M. Chandra, "Auditory ERB like admissible wavelet packet features for TIMIT phoneme recognition", *Engineer. Sci. and Technol., an Int. J. (Elsevier)*, vol. 17, no. 3, pp. 145–151, 2014 (DOI: 10.1016/j.jestch.2014.04.004).

[26] H. Frihia and Ha. Bahi, "HMM/SVM segmentation and labelling of Arabic speech for speech recognition applications", *Int. J. of Speech Technol.*, vol. 20, no. 3, pp. 563–573, 2017 (DOI: 10.1007/s10772-017-9427-z).

[27] M. Javed, M. M. A. Baig, and S. A. Qazi, "Unsupervised phonetic segmentation of classical Arabic speech using forward and inverse characteristics of the vocal tract", *Arab. J. Sci. Eng.*, vol. 45, pp. 1581–1597, 2020 (DOI: 10.1007/s13369-019-04065-5).

[28] S. Dusan and L. Rabiner, "On the relation between maximum spectral transition positions and phone boundaries", *Proc. of INTERSPEECH/ICSLP*, Pittsburgh, PA, USA, 2006, pp. 645–648 [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.569.3209&rep=rep1&type=pdf

[29] P. B. Ramteke and S. G. Koolagudi, "Phoneme boundary detection from speech: a rule based approach", *Speech Commun.*, vol. 107, no. 3, pp. 1–17, 2019 (DOI: 10.1016/j.specom.2019.01.003).

[30] G. Almpanidis, "Phonemic segmentation using the generalised Gamma distribution and small sample Bayesian information criterion", *Speech Commun.*, vol. 50, no. 1, pp. 38–55, 2008 (DOI: 10.1016/j.specom.2007.06.005).

[31] P. Teng, X. Liu, and Y. Jia, "Text-independent phoneme segmentation via learning critical acoustic change points", *Intell. Sci. and Big Data Engineer., Lecture Notes in Computer Sci.*, Berlin, Heidelberg: Springer, 2013, vol. 8261 (DOI: 10.1007/978-3-642-42057-3_8).

[32] A. H. Abo Absa, M. Deriche, M. Elshafei-Ahmed, Y. M. Elhadj, and B. Juang, "A hybrid unsupervised segmentation algorithm for Arabic speech using feature fusion and a genetic algorithm", *IEEE Access*, vol. 6, pp. 43157–43169, 2018 (DOI: 10.1109/ACCESS.2018.2859631).

[33] Y.-H. Wang, Ch.-T. Chung, and H.-Y. Lee, "Gate activation signal analysis for gated recurrent neural networks and its correlation with phoneme boundaries", *Interspeech*, Stockholm, Sweden, 2017 (DOI: 10.21437/INTERSPEECH.2017-877).

[34] F. Kreuk, Y. Sheena, J. Keshet, and Y. Adi, "Phoneme boundary detection using learnable segmental features", *IEEE Int. Conf. on Acoustics, Speech and Signal Process. (ICASSP 2020)*, Barcelona, Spain, 2020, pp. 8089–8093 (DOI: 10.1109/ICASSP40776.2020.9053053).

[35] J. Franke, M. Mueller, F. Hamlaoui, S. Stueker, and A. Waibel, "Phoneme boundary detection using deep bidirectional LSTMs", *Proc. of the Speech Commun.: 12. ITG Symp.*, Paderborn, Germany, 2016, pp. 1–5 (ISBN: 9783800742752).

[36] L. Lu, L. Kong, Ch. Dyer, N. A Smith, and S. Renals, “Segmental recurrent neural networks for end-to-end speech recognition”, *Proc. of the Interspeech*, 2016, pp. 385–389 (DOI: 10.21437/Interspeech.2016-40).

[37] Y. H. Lee, J. Y. Yang, C. Cho, and H. Jung, “Phoneme segmentation using deep learning for speech synthesis”, *Proc. of the Conf. on Res. in Adaptive and Convergent Systems*, Honolulu, HI, USA, 2018, pp. 59–61 (DOI: 10.1145/3264746.3264801).

[38] J. F. Gemmeke, T. Virtanen, and A. Hurmalainen, “Exemplar-based sparse representations for noise robust automatic speech recognition”, *IEEE Transac. on Audio, Speech, and Language Process.*, vol. 19, no. 7, 2011, pp. 2067–2080 (DOI: 10.1109/TASL.2011.2112350).

[39] D. Baby, T. Virtanen, J. F. Gemmeke, and H. Van Hamme, “Coupled dictionaries for exemplar-based speech enhancement and automatic speech recognition”, *IEEE/ACM Transac, on Audio, Speech, and Language Process.*, vol. 23, no. 11, pp. 1788–1799, 2015 (DOI: 10.1109/TASLP.2015.2450491).

[40] V.-H. Duong, M.-Q. Bui, and J.-Ch. Wang, “Dictionary learning-based speech enhancement, active learning – beyond the future”, *IntechOpen*, 2019 (DOI: 10.5772/intechopen.85308).

[41] M. Hasheminejad and H. Farsi, “Frame level sparse representation classification for speaker verification”, *Multimedia Tools and App.*, vol. 76, pp. 21211–21224, 2017 (DOI: 10.1007/s11042-016-4071-1).

[42] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, “A Survey of sparse representation: algorithms and applications”, *IEEE Access*, vol. 3, pp. 490–530, 2015 (DOI: 10.1109/ACCESS.2015.2430359).

[43] B. K. Natarajan, “Sparse approximate solutions to linear systems”, *SIAM J. on Comput.*, vol. 24, no. 2, pp. 227–234, 1995 (DOI: 10.1137/S0097539792240406).

[44] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic decomposition by basis pursuit”, *SIAM J. on Scientific Comput.*, vol. 20, no. 1, pp. 33–61, 1999 (DOI: 10.1137/S1064827596304010).

[45] J. A. Tropp and S. J. Wright, “Computational methods for sparse solution of linear inverse problems”, *Proc. of the IEEE*, vol. 98, no. 6, 2010, pp. 948–958 (DOI: 10.1109/JPROC.2010.2044010).

[46] SPGL1: A solver for large-scale sparse reconstruction [Online]. Available: https://www.cs.ubc.ca/ mpf/spgl1/index.html

[47] Matlab Benchmark Scripts, L-1 Benchmark Package [Online]. Available: http://people.eecs.berkeley.edu/∼yang/software/l1benchmark/l1benchmark.zip

[48] O. J. Räsänen, U. K. Laine, and T. Altosaar, “An improved speech segmentation quality measure: the R-value”, *Interspeech, 10th Annual Conf. of the Int. Speech Commun. Association*, Brighton, United Kingdom, pp. 1851–1854, 2009.

[49] J. S. Garofolo *et al.*, “TIMIT acoustic-phonetic continuous speech corpus”, 1993 (DOI: 10.35111/17gk-bn40).

[50] Arabic Speech Corpus: a single-speaker, Modern Standard Arabic speech corpus made for high quality speech synthesis [Online]. Available: http://www.arabicspeechcorpus.com

[51] C. Lopes and F. Perdigao, “Phoneme recognition on the TIMIT database, speech technologies”, *IntechOpen*, 2011 (DOI: 10.5772/17600).

[52] N. Halabi, “Modern standard Arabic phonetics for speech synthesis”, *University of Southampton, Electronics & Computer Sci.*, Ph.D. Thesis, 2016 [Online]. Available: https://eprints.soton.ac.uk/409695/1/Nawar_Halabi_PhD_Thesis_Revised.pdf

[53] Praat: doing phonetics by computer [Online]. Available: https://www.fon.hum.uva.nl/praat/

[54] C. S. Burrus and J. E. Odegard, “Coiflet systems and zero moments”, *IEEE Transac. on Signal Process.*, vol. 46, no. 3, pp. 761, 1998 (DOI: 10.1109/78.661342).

Ihsan Al-Hassani is a Ph.D. student at Higher Institute for Applied Science and Technology HIAST, Damascus, Syria. She received her M.Sc. degree in Telecommunication Engineering in 2013 from HIAST. Her research interests are focused on digital signal processing based embedded systems specialized in speech processing and digital communication.

E-mail: ihsan.alhassani@hiast.edu.sy
Higher Institute for Applied Science and Technology
HIAST
Damascus
Syria

Oumayma Al-Dakkak received her Ph.D. in Electronics System from Grenoble Institute of Technology (Institut Polytechnique de Grenoble) in 1988. She is a Research Director and Head of Telecommunication Department, Higher Institute for Applied Science and Technology HIAST, Damascus, Syria. Her research interests include signal processing, speech processing, speech recognition, machine learning, cryptography and digital communication.

iD https://orcid.org/0000-0002-8842-0979
E-mail: oumayma.dakkak@hiast.edu.sy
Higher Institute for Applied Science and Technology
HIAST
Damascus
Syria

Abdlnaser Assami received the M.Sc. degree from the ENST, Paris, France in 1996 and the Ph.D. degree from the University of Cergy-Pontoise, France in 2009. He is currently an Assistant Professor and Head of Digital Electronics Lab at the Higher Institute for Applied Sciences and Technology, Damascus, Syria. His research interests include man-machine communication, mobile communication, and Turbo-processing and its application to communication systems.

iD https://orcid.org/0000-0002-2036-5264
E-mail: abdelnasser.assimi@hiast.edu.sy
Higher Institute for Applied Science and Technology
HIAST
Damascus
Syria