Speech Recognition Native Module Environment Inherent in Mobiles Devices

Blanca E. Carvajal-Gámez[1](✉), Erika Hernández Rubio[2],
Amilcar Meneses Viveros[3], and Francisco J. Hernández-Castañeda[2]

[1] Instituto Politécnico Nacional, Unidad Profesional Interdisciplinaria y Tecnología Avanzada, México D.F., Mexico
becarvajal@ipn.mx

[2] Instituto Politécnico Nacional, SEPI-ESCOM, México D.F., Mexico
ehernandezru@ipn.mx

[3] Departamento de Computación, CINVESTAV-IPN, México D.F., Mexico
ameneses@cs.cinvestav.mx

Abstract. Applications on mobile devices have been characterized for their usability. The voice is a natural means of interaction between users and mobile devices. Traditional speech recognition algorithms work in controlled media are targeted to specific population groups (e.g. age, gender or language to name of few), and also require a lot of computational resources so that the algorithms are effective. Therefore, pattern recognition is performed in mobile applications as web services. However, this type of solution generates high dependence on Internet connectivity, so it is desirable to have an embedded module for this task that does not consume many computational resources and have a good level of effectiveness. This paper presents an embedded mobile systems for voice recognition module is presented. This module works in noisy environments, it works for any age of users and has proved that it can work for several languages.

1 Introduction

Applications on mobile devices have been characterized for their usability [1]. Developers try that interaction means for mobile devices are natural to the user [2]. Actually, the most common means of interaction based on gestures on touch screens [3]. It has explored the interaction through voice applications such as search and on tasks requiring multimodal interactions [4,5].

In hardware complies with those demands of speech recognition applications, because we can utilize parallel and pipelined architectures [6], which either reduce power with low operation frequency or speed up there cognition. With such knowledge, comprehensible human-like speech recognition can be obtained [6].

In automatic speech recognition systems (ASR) can be sorted into the following three categories [7]:

M. Antona and C. Stephanidis (Eds.): UAHCI 2015, Part I, LNCS 9175, pp. 267–278, 2015.
DOI: 10.1007/978-3-319-20678-3_26

1. First the ASR model converts the speech signal into a sequence of phonemes o words, and after wards the natural language processing (NLP) attempts to understand the given words.
2. The ASR model outputs more than one possible representation of the speech signal. These are then analysed with an NLP and the best one is chosen.
3. The ASR model and the NLP are combined, such that the ASR model can make use of the information and constraints provided by the NLP.

A popular application, which consists of a list of all possible words that one might encounter in a particular application. Current spoken language systems have limited vocabularies, since this is dependent on the available power and memory space of the central processing unit being used. As a result, one might encounter out-of-vocabulary (OOV) words, which the ASR system will either rejector consider it as an error [8].

Mobile devices have restrictions Battery, Memory and CPU [9,10]. Although a part of CPU has been thought that the increase in cores and incorporating parallel programs can help reduce energy consumption [11], although it is not clear that the incorporation of parallel programs on mobile devices really save energy [12]. Various techniques have been developed for offloading as a form of energy savings, improve response time, and avoid problems of storage and memory on the mobile device [9].

It has sought strategies that allow offloading the task of speech recognition. This has supported infrastructure for distributed speech recognition [13]. There are also a lot of support from below cloud-like structures [14,15]. However this creates a high dependency on the internet [10].

Additionally, there are restrictions on the systems automatic speech recognition (ASR) and performance problems with human-machine iteration are requiring extra learning to use. However, the biggest challenge for voice-based interfaces require: a more natural communication possible [8].

The main weakness of ASR systems is the use of the statistical principle: which looks for the best scenario of all possible candidates given a pre-defined dictionary [reference]. This gives rise to a problem called a grammarian OOV (out-of-vocabulary, OOV) [8]. This system takes words not recorded and added to the system. This system, however; is not always the best way in systems where the vocabulary is very large [8]. During the design phase of the dialogue should be clearly identified in the following conditions: Outreach, Level of naturalness, Strength and Length dialog [8].

In this paper the development of a native voice recognition module for mobile devices is presented. This type of solution avoids reliance on internet connectivity. The algorithm used has low complexity. It is inherent in the medium, recalling that the environment in which mobile devices are used are noisy. Not supervised so requires no training. Recognizes word of at least two languages: English and Spanish. It does not depend on the user type (adult, young, boy, man or woman).

2 Proposed Solution

The most significant problems in speech recognition systems are related to the individuality of the human voice (such as age or gender to name a few), dialect, speaking rate, the context of phonics, noise background, the characteristics of voice acquisition device (microphone), and the directional characteristics of the source speech signal, among others [16–18]. All these problems are considered in the proposed method, because the objective of this research is that the speech recognition is inherent to the environment and the people who use the application. The speech recognition module, based on the DWT-Haar, comprises three main blocks: encoding and compression, feature extraction and recognition, as shown in Fig. 1.

2.1 Pre-processing and Reduction Modules

Pre-processing module and reduction with a DWT. This reduces the complexity of calculation and does inherent to the medium. DWTs take into consideration the temporal information that is inherent in speech signals, apart from the frequency information. Since speech signals are non-stationary in nature, the temporal information is also important for speech recognition applications [7].

In this block, two vectors are obtained: The approximation vector (AV) and the fluctuations vector (FV). These vectors are obtained when DWT-Haar is applied in the original speech signal. The size of the vectors AV and FV is half the size of the original speech vector. The AV vector contains the low frequency components of the voice signal. The vector FV contains high frequency components of the speech original signal. For this work the AV was chosen, because this vector has the largest amount of information about the speech original signal. To encode speech signal through native methods of recording, signal compression is obtained through the AV with WAV format. Compression has a rate of 22050 samples per second with 16 bits per sample without encoding pulse code modulation (PCM).

2.2 Estimation Module

To increase the robustness of the designed system under noisy conditions [19]. We propose the CLCES analysis, when there cognition system is corrupted by noisy speech signals. This statement is confirmed through an evaluation made on four different noisy environments with different measure: standard deviation, variance, energy and mean value.

This block obtains the corresponding features of each input voice signal. This extraction is performed in the AV obtained in the previous block. Acquired characteristics are energy, Eq. 1, the standard deviation Eq. 2, variance Eq. 3 and the center frequency of the speech signal in the VA.

$$E[y_{Lo\_D}[n]] = \sum_{m=1}^{n} |y_{Lo\_D}[m]|^2 \tag{1}$$

$$\sigma_{y_{Lo\_D}} = \sqrt{\sum_{m=1}^{n} \frac{\left(y_{(Lo\_D)_m}\bar{y}_{Lo\_D}\right)^2}{n}} \tag{2}$$

$$\sigma^2_{y_{Lo\_D}} = \sum_{m=1}^{n} \frac{\left(y_{(Lo\_D)_m}\bar{y}_{Lo\_D}\right)^2}{n} \tag{3}$$

where $\bar{y}_{Lo\_D} = \sum_{m=1}^{n}(y_{Lo\_D})/n$ represent de mean value of AV, and $n = w/2$.

2.3 Classification Module

Three renowned methods that were used at the classification stage of ASR systems are the HMM (Hide Markov Models), the ANN (Artificial neural network) and the SVMs (Suport Vector Machine) [7]. These three classifiers usually have prior training which is sometimes tedious and energy-intensive. In this paper, simple classification techniques were incorporated, such as fuzzy logic classifier and K-neighbors distance variations and techniques to obtain additional distance. These techniques are supported although simple largely this stage prior to preprocessing. Thus ensure satisfactory accuracy with low power consumption.

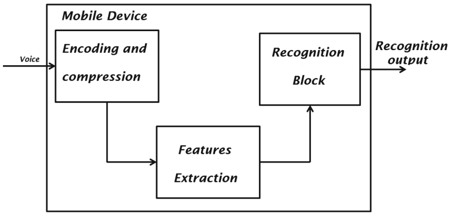
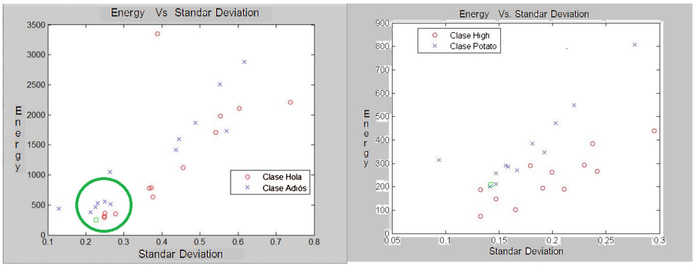


Fig. 1. Block diagram of the proposed method

This block is determined by the spoken word. The words used are suggested: "*Hola*" and "*Adios*", and also "*High*" and "*Potato*". The words "*Hola*" and "*Adios*" in particular were chosen in order to show that the system works with high statistical dependence as shown in Fig. 3(a), contained in the green circle overlapping of these two words is observed.

Unlike the words "*High*" and "*Potato*", which in Fig. 3(b), the statistical independence is observed. This task is accomplished by using speech recognition. The entries in this block are the characteristics of the speech signal.

Fuzzy Logic Method. A fuzzy logic system (FLS) is unique in that it is able to simultaneously handle numerical data and linguistic knowledge [20]. It is a nonlinear mapping of an input data (feature) vector into a scalar output. Fuzzy set theory and fuzzy logic establish the specifics of the nonlinear mapping. For many problems two distinct forms of problem knowledge exist: (1) objective knowledge, which is used all the time in engineering problem formulations (e.g., mathematical models), and (2) subjective knowledge, which represents linguistic



a)

b)

Fig. 2. Graphic of energy vs. standard deviation, (a) words "*Hola*" and "*Adios*", (b) "*High*" and "*Potato*".

information that is usually impossible to quantify using traditional mathematics (e.g., rules, expert information, design requirements) [20].

Fuzzy logic was used for word recognition in speech signal. This technique allows determining whether a spoken word in the set of speech signal features vector.

Fuzzy logic system (FLS), Fig. 2, maps crisp inputs into crisp outputs. It contains four components: rules, fuzzifier, inference engine, and defuzzifier. Once the rules have been established, a FLS can be viewed as a mapping from inputs to outputs (the solid path in Fig. 2, from "Crisp Inputs" to "Crisp Outputs"), and this mapping can be expressed quantitatively as $y = f(z)$.

Rules may be provided by experts or can be extracted from numerical data. In either case, engineering rules are expressed as a collection of `IF THEN` statements e.g.

`IF` "*Hola*" is very near "*Adios*" is very far, `THEN` turn somewhat to the right.

This one rule reveals that we will need an understanding of: (1) linguistic variables versus numerical values of a variable; (2) quantifying linguistic variables, which is done using fuzzy membership functions; (3) logical connections for linguistic variables (e.g., "and", "or", etc.); and (4) implications, i.e., "`IF` A `THEN` B". Additionally, we will need to understand how to combine more than one rule.

The fuzzifier maps crisp numbers into fuzzy sets. It is needed in order to activate rules which are in terms of linguistic variables, which have fuzzy sets associated with them.

The inference engine of the FLS maps fuzzy sets into fuzzy sets. It handles the way in which rules are combined. Just as we humans use many different types of inferential procedures to help us understand things or to make decisions, there are many different fuzzy logic inferential procedures. Only a very small number of them are actually being used in engineering applications of FL.

In many applications, crisp numbers must be obtained at the output of a FLS. The defuzzifier maps output sets into crisp numbers. In a controls application, for example, such a number corresponds to a control action. In a signal processing

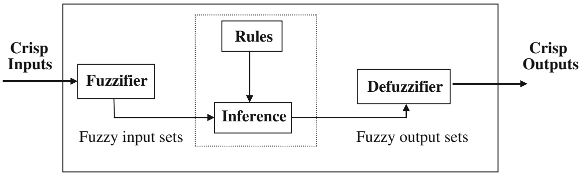


Fig. 3. Fuzzy logic system

application, such a number could correspond to the prediction of next year?s sunspot activity, a financial forecast, or the location of a target.

Crisp Sets. Recall that a crisp set A in a universe of discourse U (which provides the set of allowable values for a variable) can be defined by listing all of its members or by identifying the elements $x \in A$; thus A can be defined as $A = \{x|x \text{ meets some condition}\}$. Alternatively, we can introduce a zero-one membership function (also called a characteristic function, discrimination function, or indicator function) for A, denoted $\mu_A(x)$ such that $A \Rightarrow \mu(x) = 1$ if $x \in A$, and $\mu_A(x) = 0$ if $x \notin A$. In this paper the universe is the user's speech block, i.e. the input signal. In this universe must define members that belong to the class.

Fuzzy Sets. A fuzzy set F defined on a universe of discourse U is characterized by a membership function $\mu_F(x)$ which takes on values in the interval $[0, 1]$. A fuzzy set is a generalization of an ordinary subset (i.e., a crisp subset) whose membership function only takes on two values, zero or unity. A membership function provides a measure of the degree of similarity of an element in U to the fuzzy subset. Fuzzy logic is performed using the membership functions [21] for each of the features extracted from the Eqs. (1), (2) and (3). The membership function indicates the degree to which each element of a given universe belongs to a set. If the set is crisp, the membership function (characteristic function) take the values $\{0, 1\}$, while if the set is blurred, it will take in the interval $[0, 1]$. If the result of the membership function is equal to 0, then the element is not in the set. In contrast, if the result of the membership function is 1, then the element belongs to the set completely [21]. Gaussian membership, Eq. (4), is used for the purpose of this word recognizer in the speech signal by using fuzzy logic. The Gaussian membership is specified by two parameters c, σ to determine the boundaries of the speech signal, and also to determine where the greatest amount of information is presented in the spectral content of the word to be identified [22].

$$Gaussian(x; c, \sigma) = e^{-\frac{1}{2}(\frac{x-c}{\sigma})^2} \tag{4}$$

The Gaussian function is determined by the values they take σ and c. Where c represents the center of the function and σ is the standard deviation (2). For this case, c is the mean value, and each value is the mean average of each standard sample, and σ is the standard deviation (2) of each test pattern [22].

From define boundaries of membership function can determine whether or not the word issued by the user is the search word in the inference system.

K-Nearest Neighbors Method. In many pattern recognition problems, the classification of an input pattern is based on data where the respective sample sizes of each class are small and possibly not representative of the actual probability distributions, even if they are known. In these cases, many techniques rely on some notion of similarity or distance in feature space, for instance, clustering and discriminant analysis [3,23]. This decision rule provides a simple nonparametric procedure for the assignment of a class label to the input pattern based on the class labels represented by the K-closest (say, for example, in the Euclidean sense) neighbors of the vector.

The nearest neighbor classifiers require no preprocessing of the labeled sample set prior to their use. The crisp nearest-neighbor classification rule assigns an input sample vector y, which is of unknown classification, to the class of its nearest neighbor [24].

This idea can be extended to the K-nearest neighbors with the vector y being assigned to the class that is represented by a majority amongst the K-nearest neighbors. When more than one neighbor is considered, the possibility that there will be a tie among classes with a maximum number of neighbors in the group of K-nearest neighbor exists. One simple way of handling this problem is to restrict the possible values of K [24]. A means of handling the occurrence of a tie is as follows. The sample vector is assigned to the class, of those classes that tied, for which the sum of distances from the sample to each neighbor in the class is a minimum. This could still lead to a tie, in which case the assignment is to the last class encountered amongst those which tied, an arbitrary assignment. Clearly, there will be cases where a vector's classification becomes an arbitrary assignment, no matter what additional procedures are included in the algorithm [24].

3 Test and Results

The test module for voice recognition were performed on a smartphone android, words that have used statistical dependence and a population was considered with varied age. Also sought experiments were made in noisy environments.

The module for speech recognition embedded was implemented on a mobile device. This module identifies words, two in Spanish Languagem "*Hola*" and "*Adiós*", and two in English language, "*High*" and "`Potato`", applying Fuzzy Logic and KNN methods. The words "*Hola*" and "*Adiós*" have high statistical dependence. It seeks to do the tests with this type of words to prove that this does not affect the method proposed in this paper. The English words were chosen for their statistical independence.

A population with diverse gender and age were chosen for testing. Samples are shaped with a total population of 12 people, 6 men and 6 women. The ages are classified as follows. People over 50 years old: 3. People between 30 and 50 years old: 3. And people between 20 and 30 years old: 6.

The speech recognition system was tested in a smartphone with Android 4.3, 1 GB of RAM, Quad-core processor Qualcomm Snapdragon 400 MSM8226 @1200 Mhz. To execute the embedded module, the audio is acquired through the microphone of the mobile device, with active noise cancellation. It should consider the disadvantages of this type of device on a mobile device, where its main objective is the correct management of energy consumption. Since microphones with active noise cancellation is that they require external power to cancel outside noise with continuous presence as people laughing noise, sounds of cars, music, etc. So you have as the detector active noise within the mobile device applies more energy for canceling outside noise, reduces efficiency calculations in the mobile processor. So this is one of the main challenges to overcome in identifying words without dependence on internet for any gender and age of the user.

The speech files that were used have different environmental conditions, because they want to test the robustness and accuracy of the identification of the word within the audio file. The module was also tested in different languages. Tables 1 and 2 show the results obtained from the two algorithms applying in the embedded module. In these tables you can see the quantitative results of this algorithm. Also a comparison between the signal acquired and processed audio signal is shown.

Table 1. Results obtained by the embedded module to identify "`Hola`" and "`Adiós`"

	Embedded module							
	Fuzzy logic method				Knn method (0.15)			
Age range	Processing time (seg)	Sp (%)	Se (%)	ACC (%)	Processing time (seg)	Sp (%)	Se (%)	ACC (%)
Age > 50	0.109	69.35	93.33	71.08	0.128	60.12	63.33	56.66
$30 \leq$ Age ≤ 50	0.116	71.87	96.66	66.66	0.114	66.66	51.66	57.76
$20 \leq$ Age < 30	0.106	69.49	100	70.55	0.122	67.29	56.30	58.33
Avegrage	0.110	70.23	96.66	69.43	0.121	64.49	57.09	57.58

Table 2. Results obtained by the embedded module to identify "`High`" and "`Potato`"

	Embedded module							
	Fuzzy logic method				Knn method (0.15)			
Age range	Processing time (seg)	Sp (%)	Se (%)	ACC (%)	Processing time (seg)	Sp (%)	Se (%)	ACC (%)
Age > 50	0.107	68.25	98.90	73.34	0.150	46.66	51.33	53.33
$30 \leq$ Age ≤ 50	0.120	64.82	97.56	72.22	0.130	81.33	73.09	80.00
$20 \leq$ Age < 30	0.120	63.22	100	73.33	0.119	54.33	54.33	65.00
Avegrage	0.115	65.43	98.82	72.96	0.133	60.77	59.58	66.11

Performance results are calculated from the speech signal processing through the mobile device. To test the performance of the proposed method, we consider four cases: two for correct classifications and two for misclassification. The classifications are: true positive (TP), false positive (FP), false negative (FN) and true negative (TN). By using these different measures of performance metrics as the following relation is obtained [19]:

$$Specificity = \frac{TN}{TN + FP} \tag{5}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{6}$$

$$Precision = \frac{TP + TN}{samplesofspeechsignal} \tag{7}$$

Specificity (Sp) is the ability to detect samples that do not correspond to the audio signal. The sensitivity (Se) reflects the ability of an algorithm to detect the sample audio signal. Accuracy (ACC) measures the ratio between the total number of correctly classified samples (sum of true positives and true negatives) by the number of samples of audio signal [19]. The positive predictive value, and accuracy rate, gives the proportion of samples of the audio signal identified which are true. That is, the probability that a sample of the audio signal is identified as true positive. From the results of tests concentrated in Tables 1 and 2, we note the following. The module embedded on the mobile device detects up to 70 % of search word (Sp) and up to 96.66 % for detecting those who have the search word in the audio file (Se) for the Spanish language. For English language, the embedded module detects up to 65.43 % the search word (Sp) and up to 98.82 % for detecting those who have the search word in the audio file (Se). In the case of the Spanish language, an accuracy greater than 69.43 % is obtained at a time of 0.110 s. And for English, an accuracy of 72.96 % is obtained in a time of 0.115 s, with applying the Fuzzy logic method.

The module embedded on the mobile device detects up to 64 % of search word (Sp) and up to 57.09 % for detecting those who have the search word in the audio file (Se) for the Spanish language. For English language, the embedded module detects up to 60.77 % the search word (Sp) and up to 59 % for detecting those who have the search word in the audio file (Se). In the case of the Spanish language, an accuracy greater than 57 % is obtained at a time of 0.133 s. And for English, an accuracy of 66.11 % is obtained in a time of 0.133 s, with applying the Knn method.

4 Discussion

From the results, it can be seen that although there are variations in gender and age, the precision of the embedded system remains constant. This causes that the system will not need a re-training every time to capture the variations of voice that can occur with age. This job requires increasing values of Sp and Se, this is in progress with the investigation.

The range of ages, regardless of gender, who presented the best accuracy for word recognition within the mobile device is above 50 years old, for the Spanish language, Table 1. In the case of English language, the same behavior was presented. The greatest accuracy was obtained with the elderly 50 years old, Table 2. In the case of the Spanish words that begin or end with the same phoneme, such as cases that were used in this research as "*Hola*" and "*Adiós*" as well as the noisy environment of the cell itself and the age range of the people, causing the rate of Sp, and Acc is, decrease its performance.

In [22], a module for recognizing isolated words, in real time, on a mobile device is presented. In this study conducted with a sample population of 10 people in total. This population consists of 5 men and 5 women, and the age range of the members of the population is not mentioned. Replays of the audio signals on the mobile performed 3 times to create an average error. The tests in this work were attacked with white noise with SNR, Eq. 8, between 15 and 30 db [22].

$$SNR = \frac{\bar{y}_{Lo\_D}}{\sigma_{y_{Lo\_D}}} \tag{8}$$

The SNR has a uniform distribution, so that altering the signal audio not affected and a substantial modification of this signal is taken. This is different to present audio signal to ambient noise, because in this case the signal is affected by impulsive noise. And this kind of noise no predictable distribution, generating in certain sections of audio are altered significantly. The results presented in [22], are returned in a time of 11.61 ms and with an average of 61.4 % in the worst case, when the signal is changed to white noise. And an average of 90.2 % is obtained when the audio file is in a controlled environment.

The average processing time for the words used is in a range between 0.11 and 0.133 s. This is relevant because the voice processing does not exceed 0.2 s required for the user to have an answer during interaction with the mobile device.

5 Conclusion

The proposal presented in this research for isolated word recognition in a mobile device in uncontrolled environments gives yields higher than 70 % in less than the time 0.120 s, in the Spanish language. In the case of American English language, results over 66.11 % are obtained in an average time of 0.133 s. The voice recognition system is implemented on a mobile device with a microphone type active voice cancellation. This causes, as the tests are performed in uncontrolled ways, some mobile resources are limited by the energy due to design own mobile forces you to pay attention to the noise cancellation. In some works present results but not so bi-lingual, unlike the proposal presented here. The processing time is good for the tasks of interaction between the mobile device and the user.

This embedded word recognizer module requires no prior training or generation of a dictionary as those currently commercially. Also the so-embedded module works offline, i.e.; not require a connection to the network in order to perform their job recognition. This streamlines its use and management, adding portability and the generation of an App to be used as a tool in voice commands or support systems in any treatment such as Luria tests. We note also that being a working embedded system offline use so the battery is not as affected in the performance of this. The word recognition system achieves work for any genre and any age group not presenting any difficulty, to be altered or amended by voice acuity or severity of the tone of speech signal for the gender of the user, as well as the possible echo the voice generated by age. Concluding finally that although in some cases the performance is not expected, than the results shown in [22], embedded system mounted on an FPGA, where the processing is done faster and transparent manner.

Finally, for the voice recognition module is not necessary voice retraining determined by a period of time. This behavior is due to natural variations in the voice over time.

References

1. Love, S.: Understanding Mobile Human-Computer Interaction. Elsevier, Amsterdam (2005)
2. Jacko, J.A.: Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, 3rd edn. CRC Press Inc., Boca Raton (2012)
3. Bragdon, A., Nelson, E., Li, Y., Hinckley, K.: Experimental analysis of touch-screen gesture designs in mobile environments. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 403–412. ACM (2011)
4. Turk, M.: Multimodal interaction: a review. Pattern Recogn. Lett. **36**, 189–195 (2014)
5. Tzovaras, D.: Multimodal User Interfaces: From Signals to Interaction. Signals and Communication Technology. Springer, Heidelberg (2008)
6. Choi, J., You, K., Sung, W.: An fpga implementation of speech recognition with weighted finite state transducers. In: 2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1602–1605. IEEE (2010)
7. Cutajar, M., Gatt, E., Grech, I., Casha, O., Micallef, J.: Comparative study of automatic speech recognition techniques. IET Sig. Process. **7**(1), 25–46 (2013)
8. Fábián, T.: Confidence Measurement Techniques in Automatic Speech Recognition and Dialog Management. Der Andere Verlag, Tönning (2008)
9. Kumar, K., Liu, J., Lu, Y.H., Bhargava, B.: A survey of computation offloading for mobile systems. Mob. Netw. Appl. **18**(1), 129–140 (2013)
10. Kumar, K., Lu, Y.H.: Cloud computing for mobile users: can offloading computation save energy? Computer **43**(4), 51–56 (2010)
11. Hill, M.D., Marty, M.R.: Amdahl's law in the multicore era. IEEE Comput. **41**(7), 33–38 (2008)
12. Isidro Ramírez, R., Meneses Viveros, A., Hernándes Rubio, E., Torres Hernández, I.M.: Differences of energetic consumption between java and jni android apps. In: International Symposium on Integrated Circuits (ISIC 2014). IEEE (2014)
13. Pearce, D.: Enabling new speech driven services for mobile devices: an overview of the etsi standards activities for distributed speech recognition front-ends. In: AVIOS 2000: The Speech Applications Conference, pp. 261–264 (2000)

14. Bahl, P., Han, R.Y., Li, L.E., Satyanarayanan, M.: Advancing the state of mobile cloud computing. In: Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services, pp. 21–28. ACM (2012)
15. Di Fabbrizio, G., Okken, T., Wilpon, J.G.: A speech mashup framework for multimodal mobile services. In: Proceedings of the 2009 International Conference on Multimodal Interfaces, pp. 71–78. ACM (2009)
16. Husnjak, S., Perakovic, D., Jovovic, I.: Possibilities of using speech recognition systems of smart terminal devices in traffic environment. Procedia Eng. **69**, 778–787 (2014)
17. Oviatt, S.: Multimodal interfaces. In: The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications, pp. 286–304 (2003)
18. Ons, B., Gemmeke, J.F., et al.: Fast vocabulary acquisition in an nmf-based self-learning vocal user interface. Comput. Speech Lang. **28**(4), 997–1017 (2014)
19. Carvajal-Gamez, B.E., Gallegos-Funes, F.J., Rosales-Silva, A.J.: Color local complexity estimation based steganographic (clces) method. Expert Syst. Appl. **40**(4), 1132–1142 (2013)
20. Mendel, J.M.: Fuzzy logic systems for engineering: a tutorial. Proc. IEEE **83**(3), 345–377 (1995)
21. GEORGE, J.K., Bo, Y.: Fuzzy sets and fuzzy logic, theory and applications (2008)
22. Carvajal-Gamez, B.E., Hernándes Rubio, E., Meneses Viveros, A., Hernandez-Castaneda, F.J.: Feature extraction for word recognition on a mobile device based on discrete wavelet transform. In: Advances in Computing Science, vol. 83. Instituo Politénico Nacional (2014)
23. Sears, A., Jacko, J.A.: The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications. CRC Press, USA (2007)
24. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nearest neighbor algorithm. IEEE Trans. Syst. Man Cybern. **4**, 580–585 (1985)