INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2011, VOL. 57, NO. 4, PP. 483–488
Manuscript received May 16, 2011; revised November 2011. DOI: 10.2478/v10177-011-0068-x

Estimation of Hardware Requirements for Isolated Speech Recognition on an Embedded Systems

Krzysztof Kłobucki and Tomasz Mąka

***Abstract*—In recent years, speech recognition functionality is increasingly being added in embedded devices. Because of limited resources in these devices, there is a need to assess whether the defined speech recognition system is feasible within given constraints, as well as estimating how many resources the system needs. In this paper, an attempt has been taken to define a technique for estimating hardware resources usage in the speech recognition task. To determine the parameters and their dependencies in this task, the two systems were tested. The first system utilized Dynamic Time Warping pattern matching technique, the second used Hidden Markov Models. For each case, the measurement of recognition rate and time, vocabulary database size and learning time has been performed. Obtained results have been exploited to define linear and polynomial regression models, and finally, an estimation algorithm has been developed using these models. After testing proposed approach, it was observed that even low-end mobile phones have sufficient hardware resources for realisation of isolated speech recognition system.**

***Keywords*—Isolated speech recognition, ASR, resources estimation.**

I. Introduction

NOWADAYS, computers in various forms can be found almost everywhere. They are present not only in mobile phones and media players, but also in cars, washing machines and many other appliances. One of the main differences is that many of these devices (e.g. mobile phones, sensors) are much smaller than well known desktop computers or laptops. Their size is limited by their application, e.g. mobile phones and handheld media players have to be small enough to fit into hand.

Number of different applications for isolated speech recognition in embedded system can be proposed [1], [2]. Currently, mobile phones give people vast number of different functionalities. It is difficult to use most of them using a small keyboard or small display. With voice interface there would be no need to go through all the menus to turn on or off a GPS or WiFi. It could be done with one simple voice command. Today, cars, similarly to mobile phones, are produced with increasing number of functionalities. In-Car Entertainment Systems, often called Infotainment Systems, are now becoming command centers not only for multimedia functions, but also for all other car functions like locking the door, opening the windows or controlling the air conditioning. They come with even greater need for a voice interface than mobile phones, as driver hands should be constantly occupied by steering wheel during driving. There is a need to do simple things in a car, like changing radio station or turning on air conditioning, safely – without taking hands off the steering wheel. This need is now more and more respected, especially in high-end cars. Speech recognition applications in digital homes are evident. With voice interface it would be possible to turn off the light in a room while lying in bed, turn on a television set without a remote controller or uncover blinds after waking up in the morning, to name just a few examples. Embedded devices have many limitations, among which processing power, memory size and battery life time are the most important [3], [4]. On the other side, complex applications like speech recognition require high computing power and big amount of memory. In most of embedded devices it is not possible to expand memory or increase the computing power – limitations of the device are constant [1], [5], [6]. Therefore, algorithms are needed to assess the possibility of being able to realise speech recognition task on a device with given constraints.

Resource estimation algorithm should analyse input data such as vocabulary size, number of speakers, CPU clock, memory size and requested word recognition rate. Based on these data, algorithm should be able to evaluate if speech recognition system, running in real time with requested recognition rate, is feasible on a device with given limitations.

II. Isolated Speech Recognition

Speech recognition can be a valuable addition to embedded systems, however embedded and mobile devices due to their size and battery power face many hardware constraints, of which the most important are limited computing power and memory space [5]. All these restrictions make implementation of speech recognition systems, on such devices, a difficult task. Algorithms, especially for continuous speech recognition, require both high computing power and lots of memory. Isolated speech recognition algorithms need less resources, but they are still demanding.

Three approaches of dealing with speech recognition in embedded system with limited resources were presented in [1]. Each of these methods differs in the amount of computation being done on the embedded system.

In the first of three approaches – Network Speech Recognition (NSR) – embedded system is only responsible for speech encoding (compression) and sending it to a remote server. Then on the server side, feature extraction and speech decoding takes place. This solution has many advantages, of which the most important are low usage of device resources and possibility of upgrading the speech recognition system,

K. Kłobucki and T. Mąka are with the Faculty of Computer Science and Information Technology, West Pomeranian University of Technology, Żołnierska 49, 71-210 Szczecin, Poland (e-mails: {kklobucki, tmaka}@wi.zut.edu.pl).

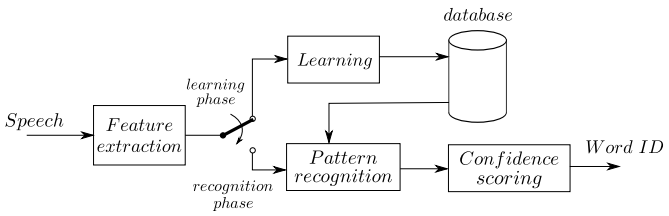

Fig. 1. Isolated speech recognition work flow.

being transparent for device users, as all important processing is done on a remote server. The biggest drawback of this method are data losses resulting from transmission and lossy compression [7]. Lossless compression can be utilized, however it has an impact on the size of data to transfer, and as there is no guaranteed transfer rate over a link between the device and a remote server recognition time can be affected. In Distributed Speech Recognition (DSR), first stage of general speech recognition process takes place on the device. Remote server is responsible for the second stage – speech decoding. The biggest advantage of this method over NSR is that there is no need for compression which can cause data losses. Moreover, recognition time is affected not much more than in NSR, because feature vectors are relatively small when compared to overall size of input data. Device resources are also not heavily occupied as most of feature extraction algorithms are mature enough to have fast implementations that do not consume a lot of processor cycles or memory space. Transmission data losses, recognition speed and need to upgrade the software, in case of change in speech recognition system, are the biggest disadvantages of this approach. In Embedded Speech Recognition (ESR), both feature extraction and speech decoding are done on a device. In this case remote server is not needed. As this method does not require any transmission, data loss problems are not relevant. Additional advantage is that speech recognition application does not depend on the network and is always ready to use. On the other side embedded devices have limited resources which strongly affect recognition time and dictionary size. Another drawback of this approach is that user has to take care of application updates.

The general work flow for isolated speech recognition system is depicted in Figure 1. In the learning phase, for each input word, a set of feature vectors is extracted and stored in the database with proper label. This phase is performed only once for defined vocabulary. The recognition stage utilize previously created database to compare with the feature vectors of input word and determine recognized word. There are mostly two approaches, exploited for isolated speech recognition, namely DTW (Dynamic Time Warping) [8] and HMMs (Hidden Markov Models) [2], [7]. The overall recognition accuracy is dependent on acquisition conditions, speakers' differentiation (age, gender, nationality) and the size of vocabulary.

The proposed estimation technique in this work is dedicated to systems working on devices with limited resources for speaker-dependent voice control tasks.

Therefore, the example list of 29 words, selected on the basis of three speech recognition applications in embedded systems has been prepared. All words are commands or item names in Polish language. The list was divided into three sets corresponding to these applications. This division is presented in Table I. For mobile phone interface application subset of 13 words was defined. Selected vocabulary is a set of commands for performing various operations on a mobile phones, such as viewing text messages or mails, controlling additional functionalities like WiFi, GPS or Radio, controlling phone display brightness and volume. Subset of 21 words was defined for infotainment system interface. This set includes previously described subset of 13 words for mobile phone application. Among 21 words voice commands were defined for controlling an audio system, air conditioning, electric windows and doors, light and highlighting inside a car. Whole set of 29 words was defined for digital home interface application. Among them there are commands for controlling blinds, air conditioning, light, audio-video devices, doors and windows.

TABLE I
EXAMPLE VOCABULARY DIVISION AMONG APPLICATIONS

No	Word (Polish)	Meaning	Applications
1	ciemniej	darker	Mobile Phone
2	jasniej	brighter	
3	ciszej	quieter	
4	glosniej	louder	
5	mniej	less	
6	wiecej	more	
7	nastepny	next	
8	poprzedni	previous	
9	otworz	open	
10	zamknij	close	
11	wlacz	turn on	
12	wylacz	turn off	
13	radio	radio	
14	cieplej	warmer	Infotainment
15	zimniej	colder	
16	drzwi	doors	
17	okno	window	
18	klimatyzacja	air conditioning	
19	swiatlo	light	
20	zapal	turn on (light)	
21	zgas	turn off (light)	
22	odslon	uncover	Digital Home
23	zaslon	cover	
24	ogrzewanie	heating	
25	telewizor	television set	
26	rolety	roller blinds	
27	zaluzje	venetian blinds	
28	zaslony	blinds	
29	temperatura	temperature	

In this paper two of the defined subsets, 13-word and 29-word, were used both in measurements of examined values and verification of models developed on basis of this measurements. A subset of the 21-word was used only for verification of these models.

III. RESOURCE USAGE ESTIMATION

In order to create technique estimating resources usage in embedded systems for speech recognition tasks, it was necessary to create models describing the variability of the basic parameters of such system. Therefore, two isolated speech recognition systems have been tested for the same vocabulary. The first system utilizes basic DTW technique described in [8].

TABLE II
IMMUTABLE SYSTEM PARAMETERS

Parameter Name	Symbol	Unit
Vocabulary Size	W	-
Number of Speakers	S	-
CPU Clock Speed	C	GHz

TABLE III
DEPENDENT PARAMETERS

Parameter Name	Symbol		Unit
	System 1	System 2	
Recognition Rate	R_{D_L}, R_{D_P}	R_{H_L}, R_{H_P}	-
Database Size	M_D	M_H	B (bytes)
Learning Time	T_{D_L}	T_{H_L}	s
Feature Extraction Time	T_{DE}	T_{HE}	s
Comparison Time	T_{DC}	T_{HC}	s

First Euclidean distances between test pattern and all templates from database were computed. Then based on distance values it was decided which word was recognized. The second system based on approach utilizing HMMs (implemented in HTK toolkit [9]). We used word based HMMs, not phoneme based. This means that each word was represented by one model. During recognition process, probability of generating given vector of features was computed for each model. Based on this results decision about recognized word was made. The number of states in model was constant and equal 5, from which 3 were emitting states. Both systems utilize the same features set – energy, static MFCCs, delta and acceleration coefficients were extracted from each frame [10]. Size of the frame was set to 32 ms with 50% overlap.

Estimation of resources for isolated speech recognition system requires to distinguish between immutable, dependent on the system, input characteristics and the dependent parameters. In the first group were vocabulary size, number of speakers and CPU clock speed. In the second were recognition rate, database size (memory consumption), learning time (time required to prepare acoustic models), feature extraction time and decision time (comparison time). Detailed information regarding this distribution is given respectively in Table II and III. Taking properties of speech recognition system parameters into consideration, we performed many tests for both systems (based on DTW and HMMs, hereafter named as system 1 and system 2 respectively) and different constraints. Based on obtained results, regression models [11] for variables listed in Table III were created.

Two regression models for recognition rate were calculated, namely linear and polynomial, as influence of vocabulary size and number of speakers on this variable is not linear. However, it turned out that in most of the cases linear models gave better results if evaluated to root mean square error (RMSE) as shown in Table IV. According to regression models for 13 and 29 words, two general regression models linear and polynomial, were created. Those models were not based on particular measurements, only on models presented earlier. In fact these two models are linear regression models of specific models coefficients. Verification of general models was done with recognition rate data collected for 21 words. Linear

TABLE IV
REGRESSION MODELS ERRORS FOR RECOGNITION RATE

Model	RMSE (DTW)	RMSE (HMM)
Linear (13 words)	2.4692	1.5937
Polynomial (13 words)	3.2727	3.777
General Linear (21 words)	3.3261	2.4505
General Polynomial (21 words)	3.4311	1.8017
Linear (29 words)	1.6666	2.0761
Polynomial (29 words)	2.065	2.2947

regression models for systems 1 and 2 are given by formulae 1 and 2 respectively. Similarly, polynomial regression models are defined by equations 3 and 4.

In database size models for both methods dependencies were linear, however they were not influenced by the same set of values. Before these dependencies will be presented, size of both method implementations will be discussed.

$$R_{D_L}(W,S) = (0.0170 \cdot W + 1.6234) \cdot S - (1.3064 \cdot W - 75.2924). \quad (1)$$

$$R_{H_L}(W,S) = (-0.025 \cdot W + 2.125) \cdot S + (0.0188 \cdot W + 78.1562). \quad (2)$$

$$\begin{aligned} R_{D_P}(W,S) = {} & (-0.0009 \cdot W + 0.0314) \cdot S^5 \\ & + (0.0236 \cdot W - 0.8148) \cdot S^4 \\ & - (0.2117 \cdot W - 7.4959) \cdot S^3 \\ & + (0.8094 \cdot W - 29.6357) \cdot S^2 \\ & - (1.2723 \cdot W - 50.6822) \cdot S \\ & - (0.5982 \cdot W - 47.4911). \end{aligned} \quad (3)$$

$$\begin{aligned} R_{H_P}(W,S) = {} & (-0.078 \cdot W + 0.1432) \cdot S^3 \\ & + (0.1953 \cdot W - 4.0391) \cdot S^2 \\ & - (1.5469 \cdot W - 36.4427) \cdot S \\ & + (3.625 \cdot W - 8.125). \end{aligned} \quad (4)$$

In addition to the data needed in the process of speech recognition, an important factor affecting the amount of required memory is the size of the application. To make this measurement, the two programs used were compiled under identical conditions. Next the size analysis of obtained code of both applications was taken. First approach, incorporating DTW algorithms, required 12696 bytes of memory. HMM-based method needed 592074 bytes, which is almost 50 times more than DTW-based. This values were added as constants to database size regression linear models.

DTW-based method was dependent on vocabulary size and number of speakers, which is reflected in methods' linear regression model defined by equation 5, where P_D is the application size.

$$M_D(W,S,P_D) = 6990 \cdot W \cdot S - 46913 + P_D \quad (5)$$

Database size in HMM-based method, on the other hand, depends on vocabulary size and number of state in hidden Markov models. Model is defined by equation 6, where Q is number of states in word-based hidden Markov model and P_H is size of application. Influence of number of states was not analysed, its impact was solely defined by HTK toolkit files review.

$$M_H(W, Q, P_H) = 1063 \cdot W \cdot Q - 18 + P_H \tag{6}$$

In case of learning time, both systems were dependent of the same set of values, namely vocabulary size, number of speakers and CPU clock speed. However, there were some differences in this dependencies. The system 1 had a linear dependency on number of samples (product of vocabulary size and number of speakers), while in system 2 such dependency was not observed. Linear regression model for learning time in system 1 is defined by equation 7, while the respective model for system 2 is given by equation 8.

$$\begin{aligned} T_{D_L}(W, S, C) &= (-0.9349 \cdot C + 2.1114) \cdot W \cdot S \\ &\quad -(7.8496 \cdot C - 17.2477). \end{aligned} \tag{7}$$

$$\begin{aligned} T_{H_L}(W, S, C) &= [(-0.0883 \cdot C + 0.3134) \cdot W \\ &\quad +(0.2052 \cdot C - 1.3897)] \cdot S \\ &\quad +[(-0.8088 \cdot C + 1.8369) \cdot W \\ &\quad -(0.8762 \cdot C - 2.5806)]. \end{aligned} \tag{8}$$

The feature extraction time in both systems was dependent solely on CPU clock speed. The difference between two systems was that for first approach three models were created, while for second approach only one. Models for system 1 are defined by equations 9–11, while model for system 2 is given by equation 12.

$$T_{DE_{min}}(C) = -0.472 \cdot C + 0.964, \tag{9}$$

$$T_{DE_{mean}}(C) = 0.6 \cdot C - 0.442, \tag{10}$$

$$T_{DE_{max}}(C) = -0.982 \cdot C + 2.225. \tag{11}$$

$$T_{HE}(C) = -0.008 \cdot C + 0.020 \tag{12}$$

Comparison time dependencies were different for each method. In system 1 comparison time was dependent on vocabulary size, number of speakers and CPU clock speed. Similarly to feature extraction time minimal, mean and maximal comparison time models were defined. These models are given by equations 13–15.

$$\begin{aligned} T_{DC_{min}}(W, S, C) &= (-0.0014 \cdot C + 0.0037) \cdot W \cdot S \\ &\quad +(0.0039 \cdot C - 0.0108), \end{aligned} \tag{13}$$

$$\begin{aligned} T_{DC_{mean}}(W, S, C) &= (-0.0017 \cdot C + 0.0045) \cdot W \cdot S \\ &\quad +(-0.0005 \cdot C - 0.0028), \end{aligned} \tag{14}$$

$$\begin{aligned} T_{DC_{max}}(W, S, C) &= (-0.0024 \cdot C + 0.0062) \cdot W \cdot S \\ &\quad +(0.0054 \cdot C - 0.0234). \end{aligned} \tag{15}$$

In case of system 2, comparison time was dependent on vocabulary size and CPU clock speed. Number of speakers had no impact on comparison time in this method. Similarly as in first approach, also for system 2, three models were defined. These models are given by equations 16–18.

$$\begin{aligned} T_{HC_{min}}(W, C) &= (-0.0011 \cdot C + 0.0026) \cdot W \\ &\quad -(0.0802 \cdot C - 0.1869), \end{aligned} \tag{16}$$

$$\begin{aligned} T_{HC_{mean}}(W, C) &= (-0.0010 \cdot C + 0.0026) \cdot W \\ &\quad -(0.0829 \cdot C - 0.1945), \end{aligned} \tag{17}$$

$$\begin{aligned} T_{HC_{max}}(W, C) &= (-0.0014 \cdot C + 0.0034) \cdot W \\ &\quad -(0.0933 \cdot C - 0.2221). \end{aligned} \tag{18}$$

Based on presented models, resource estimation can be performed providing wide range of valuable information. First of them is assessment of feasibility of given automatic speech recognition system within the given hardware constraints. Second is possibility to evaluate required resources, depending on input values, like size of dictionary or number of speakers. Finally, if hardware parameters are known, limits of given hardware platform can be evaluated.

In Figure 2, four-level decision tree for estimating the feasibility of ASR system is shown. The constraints are defined as R_{IN} (desired recognition rate), M_{IN} (memory consumption), $T_{R_{IN}}$ (recognition time) and $T_{L_{IN}}$ (learning time). The feasibility assessment process can use selected levels or whole tree. For each level, the parameters W (dictionary size) and S (number of speakers) have to be known. Additionally, for memory consumption, application size on target device (P_D, P_H) and in case of system 2, number of states (Q) should be known. To estimate learning and recognition times, the processor clock speed (C) has to be provided.

At the first level, the desired recognition rate is compared with linear models for both systems. To get more detailed evaluation of recognition rate, polynomial models could be used along with linear. Then, if desired level of recognition can be achieved memory consumption is verified. It is assumed that automatic speech recognition system should not consume more than 10% of available memory. The limit was set to 10%, but it should be even less, as considered devices like mobile phones or infotainment systems need memory for other applications, like navigation, which are also very demanding. For evaluation of memory consumption models given by equation 5 for DTW-based method and by equation 6 for HMM-based system are used. If user does not know the size of applications, then hard coded values have to be used. Finally recognition time, which is a sum of feature extraction time and comparison time, is taken into account. For feature extraction time models defined by equation 10 and by equation 12 are utilized. In case of system 1, maximum time model was selected in order to make the algorithm more strict. For HMM-based technique there was only one model for feature extraction time estimation. Maximum models were also exploited for estimation of comparison time. These models are defined by equation 14 in case of DTW-based technique and by equation 17 for HMM-based. The last level is connected with assessing of learning time. Models defined by equations 7 and 8 were used for both systems respectively.

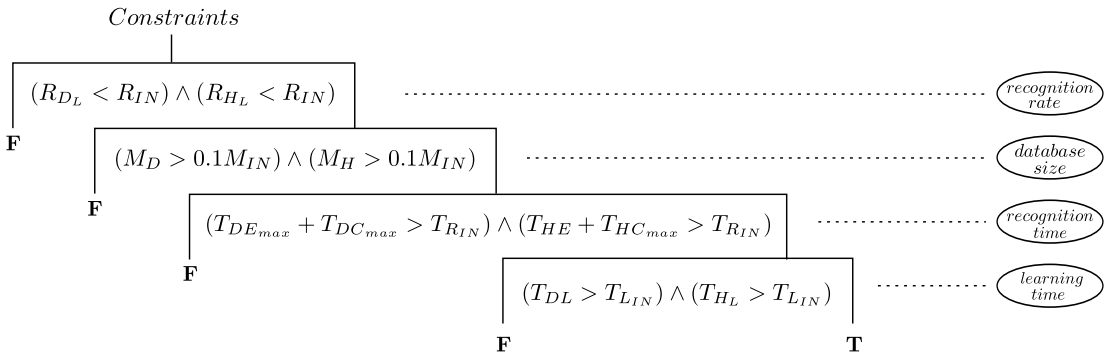

Fig. 2. Decision tree for isolated speech recognition system feasibility evaluation.

Presented estimation approach was tested for mobile phone application with two hardware configurations: device based on ARM 11 processor, 369MHz with 130Mb of memory („low-end" configuration) and device based on ARM Cortex-A8, 600MHz with 16Gb of memory („high-end" configuration). Estimation results for both configurations are given in Table V. It was assumed that there are 13 words and 10 speakers.

The first fact that could be observed are large differences between the results achieved by the DTW-based approach and those obtained with the approach incorporating HMMs. The reason is the general weakness of the DTW approach compared to methods of statistical modeling, which are represented in this case by the system 2, using HMMs. This weakness mainly affects the recognition rate. System 1 turned out to be also worse in the result related to the demand for memory, even with taking size of the application into account. In the DTW-based method the amount of memory needed is affected by the number of words and speakers, while in the HMM-based method, only by the number of words, as there is always only one model per word. Therefore, for a small number of speakers system 1 may be better in terms of memory consumption, but with an increasing number of speakers the memory needed for the DTW-based technique is growing faster than in the HMM-based.

Based on the results presented in Table V, it can be concluded that the proposed speech recognition system would be feasible using second approach. However, realisation with used implementation of DTW-based method probably would not meet most of the requirements. The main obstacle is the low recognition rate. In DTW approach, the results are obtained at level of 75%, while the expected minimum would probably at level of 90%. Another problem in the DTW-based method is the recognition time. In contrast to the learning time, which is not significant, since learning process is conducted only once, the recognition time is extremely important. It is the time in which the program responds to user actions. Recognition time of 2 seconds is noticeable to the user and thus too long. However, these results are caused by the lack of optimization in our implementation of the system 1 and there is room for further improvements. The memory consumption in both cases is on acceptable level, but it should be noted that it should not exceed 5% of available memory, because many other applications are also installed on mobile phones. Table VI presents the results of memory consumption for more demanding case, where the dictionary consists of 200 words (commands), but still there are 10 speakers. It can be observed that in DTW-based technique in low-end configuration memory consumption exceeds 10% of available memory, what probably would not be acceptable. From values presented in Table VI, recognition time for system 1 is definitely too long, which makes tested implementation unusable for such big dictionary. There is no recognition rate given in Table VI as this is the only variable with unclear dependencies. Therefore estimations for recognition rate are only true in analysed scope. Values from outside the scope cannot be estimated with reasonable accuracy. As it is clear from the above example, presented approach can be used both to assess the realisability of speech recognition system with given parameters and defined constraints, as well as to determine the limits of speech recognition systems, depending on the hardware platform.

IV. Conclusion

In this work, an approach for evaluation of the realisability of isolated speech recognition tasks, for embedded devices with limited resources, has been presented. Proposed estimation technique uses linear and polynomial regression models for estimation of recognition rate, database size (memory consumption), learning time, feature extraction time and comparison time. Based on this estimation and on input values, one can define if realisation of ASR system with given parameters is possible. The verification of regression models, created for the purpose of the technique, showed that in most cases they estimate the unknown values with only minor errors. After testing the estimation approach, it was observed that nowadays even the low-end mobile phones have sufficient resources to deal with isolated speech recognition. Undertaken research has proven that even for large set of words isolated speech recognition implementations can be realised on low-end mobile devices. All measurements undertaken for purpose of this work were done in similar noise conditions. In real world, noise conditions are variable, and therefore significant influence of environmental noise on recognition rate could be observed. Additional analysis of recognition rate with regards to impact of environmental noise could be performed.

TABLE V
RESOURCE ESTIMATION RESULTS FOR MOBILE PHONE APPLICATION

Configuration	Low-End		High-End	
System	1	2	1	2
Recognition Rate	76,75%	96,4%	76,75%	96,4%
Database Size [B]	874 787	669 077	874 787	669 077
Memory Consumption	0.67%	0.51%	0.005%	0.004%
Learning Time [s]	243.986	45.534	214.098	40.725
Feature Extraction Time [s]	1.863	0.017	1.636	0.018
Comparison Time [s]	0.670	0.225	0.599	0.199
Recognition Time [s]	2.532	0.242	2.234	0.2146

TABLE VI
MEMORY CONSUMPTION ESTIMATION FOR MOBILE PHONE FOR 200 WORDS

Configuration	Low-End		High-End	
System	1	2	1	2
Database Size [B]	13 946 087	1 662 982	13 946 087	1 662 982
Memory Consumption	10.73%	1.28%	0.09%	0.01%
Learning Time [s]	3 547	858	3 114	780
Feature Extraction Time [s]	1.863	0.017	1.636	0.015
Comparison Time [s]	10.607	0.764	9.500	0.678
Recognition Time [s]	12.470	0.781	11.136	0.693

REFERENCES

[1] Z.-H. Tan and B. Lindberg, *Automatic Speech Recognition on Mobile Devices and over Communication Networks*. Springer-Verlag, 2008, pp. 2–21.

[2] L. Rabiner and W. Schafer, *Theory and Applications of Digital Speech Processing*. Prentice-Hall, 2010, pp. 950–984.

[3] V. Amudha, B.Venkataramani, R. V. kumar, and S. Ravishankar, "Software/Hardware Co-Design of HMM Based Isolated Digit Recognition System," *Journal Of Computers*, vol. 4, no. 3, pp. 154–159, 2009.

[4] S. Grassi, M. Ansorge, F. Pellandini, and P.-A. Farine, "Implementation of Automatic Speech Recognition for Low-Power Miniaturized Devices," in *Proceedings of 5th COST 276 Workshop on Information and Knowledge Management for Integrated Media Communication*, Prague, Czech Republic, 2–3 October 2003, pp. 59–64.

[5] S. Jalali, *Trends and Implications in Embedded Systems Development*. TCS white paper, 2009.

[6] C. Levy, G. Linares, and J.-F. Bonastre, "GMM-Based Acoustic Modeling for Embedded Speech Recognition," in *INTERSPEECH 2006 – ICSLP, Ninth International Conference on Spoken Language Processing*, Pittsburgh, PA, USA, 17-21 September 2006.

[7] A. Peinado and J. Segura, *Speech Recognition Over Digital Channels: Robustness and Standards*. John Wiley & Sons, Inc., 2006, pp. 8–77.

[8] P. Senin, "Dynamic time warping algorithm review," University of Hawaii at Manoa, Tech. Rep., 2008.

[9] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book Version 3.4*. Cambridge University Press, 2006.

[10] L. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition*. Prentice Hall, 1993, pp. 219–226.

[11] D. Larose, *Data Mining Methods and Models*. Wiley-IEEE Press, 2006, pp. 36–98.