Paper

Recognition of the Numbers in the Polish Language

Anna Plichta[a], Tomasz Gąciarz[a], and Tomasz Krzywdziński[b]

[a] *Institute of Computer Science, Tadeusz Kościuszko Cracow University of Technology, Cracow, Poland*
[b] *Comarch SA, Cracow, Poland*

Abstract—Automatic Speech Recognition is one of the hottest research and application problems in today's ICT technologies. Huge progress in the development of the intelligent mobile systems needs an implementation of the new services, where users can communicate with devices by sending audio commands. Those systems must be additionally integrated with the highly distributed infrastructures such as computational and mobile clouds, Wireless Sensor Networks (WSNs), and many others. This paper presents the recent research results for the recognition of the separate words and words in short contexts (limited to the numbers) articulated in the Polish language. Compressed Sensing Theory (CST) is applied for the first time as a methodology of speech recognition. The effectiveness of the proposed methodology is justified in numerical tests for both separate words and short sentences.

Keywords—Automatic Speech Recognition, Compressed Sensing, Sparse Classification.

1. Introduction

Automatic Speech Recognition (ASR) can be defined as a real time computer-driven transcription process of a spoken language into a readable text. The main idea of ASR research is to define efficient speech recognition methodologies, implemented and automatically driven by computers, which work in a real time and allow to transform spoken language pronounced by native speakers. The accuracy of the speech recognition technology depends usually on the size of the implemented vocabulary, noise or speaker accent and individual pronunciation features. If the vocabulary size is sufficiently large and the system is properly trained to learn an individual speaker's voice, The recognition accuracy can exceed 90%. There are three most important features of the ASR systems: large vocabularies, continuous speech capability and speaker independence. Most of the speech recognition technologies has been initially designed for the visually disabled people as voice recognition support, which may be helpful for those patients, who suffer from eye diseases or some cerebral impairment.

In the recent years, thanks to intelligent and mobile technology, advanced speech recognition methods have been used by the users of mobile devices. Automatically created messages or documents from dictation – ASR technology makes their life easier. ASR has been also used for practical purposes and over the last several years it has been an area of great interest and activity to the signal processing. It seems that the perspective of development of ASR systems is optimistic – future applications of automatic speech recognition will contribute substantially to the quality of life of the society [1]–[3].

There are two main categories of speech recognition technologies: acoustic-phonetic approach, and pattern recognition-based approach. In the former, a continuous speech spectrum can be divided into several segments, which are defined as phonetic units (with unique labels), based on particular speech features. Speech stream is therefore defined as the continuous stream of such phonetic units. Then, the sequences of phonemic units are mapped into the sequences of words through a lexical decoding.

In the pattern recognition-based approach, the basic speech units are modeled by using acoustic lexical description of all words in the vocabulary. The acoustic-phonetic mapping is generated as a result of a finite discrete training process of a set of utterances. This mapping generates speech units which are defined based on the acoustic descriptions of linguistic units represented in the words occurring in the training set. This methodology seems to be the most effective speech recognition technique so far [1], [3]–[5].

Speech recognition problem is very complex for solving by using classical learning or pattern recognition methodologies. First, signal segmentation process generates a huge number of phonetic units, that must be processed. There are several methods of digital parametric representation of the speech signals, e.g., short-time spectral envelope method, Linear Predictive Coefficients (LPC) method, or Mel-Frequency Cepstral Coefficients (MFCCs) methodology, which can support the complex signal processing and classification. However, this support seems to be insufficient in the light of many types of the noise factors and individual speaker's articulation abilities, and differences between the spoken and written language or difficulties in the recognition of the words in the speech context [6].

Although speech recognition is a difficult classification task, Polish language morphology makes this problem even more challenging in the case of the Polish language, also in the simple cases of limitation of the analysis to the specific types of words such as numerals. Speech recognition applications used to recognize Polish languages usually include voice user interfaces such as voice dialing, appliance remote controlling, searching specific information, preparation of structured documents or speech-to-text processing. Algorithms which are used in those systems are based on several formal and heuristic learning-based or optimization methods, such as neural networks (usually as hybrid method with

Markov models), statistics (N-grams) and rule-based (Finite State Traducers) language models or dynamic Bayesian networks [4], [7]. Developing an effective model for the Polish language recognition can be useful for other recognition systems in which sounds are not yet classified [6].

The main objective of this work is to develop a speech recognition model based on the *Compressed Sensing* and *Sparse Classification* methods for the analysis and pattern recognition of Polish numerals, both as separated words and as words in a context. In this case Sliding Window technique [8] is used as support mechanism. All above mentioned methods have been previously known as effective tools for face recognition [9]. The developed technology allows to recognize speech of virtually any speaker. Its efficiency has been justified in series of numeral tests presented in the experimental work section.

This paper is structured as follows. In Section 2 the recent solutions for ASR with a special focus on the Polish languages approaches are surveyed. Details of the Automatic Speech Recognition general problem with a short explanation of the main steps in speech recognition process are defined. In Section 3 the authors defined the proposed methodology and implementation details. Section 4 presents the results of simple empirical analysis. The paper ends with conclusions and general future research remarks.

2. Automatic Speech Recognition

Speech recognition is a type of pattern recognition problem. The design of a pattern recognition system essentially can be realized in the following four steps [2]:

- data acquisition and preprocessing,
- data representation, necessary for pattern recognition,
- training, for instance imparting pattern class definition into the system, usually by showing a few typical examples of the pattern,
- decision-making process, generation of a pattern class by means of a training set of examples.

2.1. Problem Statement

The ASR process can be realized in the following three main steps [2]–[3], [10]:

- preliminary processing (pre-processing),
- parametric encoding,
- classification.

2.1.1. Pre-processing

The signal pre-processing stage is usually recorded and subjected to pre-treatment, such as the use of pre-emphasis filter and removal of the silence at the beginning and end of the recorded words. Then, the signal is divided into short time periods, called frames, by using of windowing method (such as Sliding Window technique). The preliminary processing stage precedes the process in which these features are isolated from the signal, and they are later used for classification. Pre-processing stage is crucial, because it usually has a great impact on the efficiency of the whole speech recognition system. The pre-processing phase flow is presented in Fig. 1.

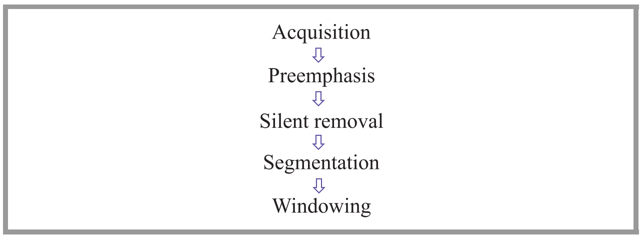

Fig. 1. Preliminary processing stage.

The first step in the pre-processing is an acquisition (recording the signal). The analog speech signal is usually recorded by a microphone and stored at the computer hard disc. However, before digital conversion, the strength of the signal may be improved by filtering through anti-aliasing filters. The analog-to-digital conversion processes is therefore defined as a sequence of sampling, quantization and signal encoding procedures.

After the acquisition, the signal strength is modulated by using a pre-emphasis filter. In the signal spectrum, the high frequency units are reinforced. Pre-emphasis is applied in order to reduce the phenomenon that the higher frequency has lower amplitude than the lower frequency (spectral tilt).

In the process of segmentation the speech signal is divided into short segments called frames. Each frame contains specified number of samples. The frames can overlap. The typical length of the frame is 20–40 ms, which gives 320–640 samples at sampling 16,000 Hz [2].

In the segmentation process some gaps may occur in the processed signal. Those gaps may be the main reason of disruptions in Fourier transform (the additional harmonic components appear in the temporary spectrum). In order to cope with that, the value of each frame should be multiplied by a *window function*. It makes the time-frame at the end of frames more smooth. The most commonly used window function is *Hamming window mapping*, which can be defined as follows [3]:

$$w(n) = 0.54 - -0.46 \cdot cos\left(2 \cdot \pi \cdot \frac{n}{N-1}\right), \qquad (1)$$

where

$$0 \leq n \leq N-1\,. \qquad (2)$$

2.1.2. Parametric Encoding

In the parametrization stage the characteristic features are isolated from the signal. A *feature vector f* is defined for each frame created during the preliminary processing stage f. The frame containing time course of the amplitude undergoes spectral analysis. This is usually difficult to obtain proper features of the frames which would not be dependent on the particular speaker. Indeed, the energy flow for the same word can be very different for various speakers.

There are two main methods of such frequency analysis: linear prediction model, and analysis based on Mel-frequency Cepstral Coefficients (MFCC) [11]. The authors used the second method in this work.

Mel-frequency Cepstral Coefficients (MFCC)

The purpose of the mel-cepstral analysis is to represent how the speech signal is considered by the human auditory system. First, the signal is transformed into the field of frequencies by using of the Fast Fourier Transformation (FFT). Then the signal is filtrated through the regular triangular filters in mel-scale, which reflects the nonlinearly perception of the frequencies (typical of human ear). Then, the discrete cosine transform is used to step into the cepstral field.

Mel-scale reflects, how human hear perceives the speech signal (how it reacts nonlinearly to the frequencies of the signal). At low frequencies (below 1 kHz) the changes are easier to detect than at higher frequency ranges. The higher the frequency, the lower the accuracy. The gaps between the series of ranges should be expanded in order to compensate for the nonlinearly. This may be achieved by using a proper set of filters for the successive frequency ranges. The filters are nonlinear according to the mel-scale [6].

2.1.3. Classification

The classification stage is defined as a comparison process of the signals' characteristics with respect to the recognition of characteristics stored in a classification local database, where a signal sample is assigned to a specific class. One of the simplest methods of classification algorithm is "k-nearest neighbors" methodology. Some other example can be sparse classification.

2.2. Efficiency Measures

There are various methods of the measurement of the efficiency of speech recognition techniques. The most popular metrics can be defined as follows [1].

Word Error Rate (WER) defined as

$$WER = 100\frac{S+D+I}{N}[\%]. \tag{3}$$

Word Accuracy (WAcc) defined as

$$WAcc = 100 - WER = 100\frac{N-(S+D+I)}{N}[\%], \tag{4}$$

where: S – denotes a number of substitutions, D – stands for the number of deletions, I – is the number of insertions, N – stands for the number of words to recognize.

WER defines how much the recognized series is different to the original one. It is based on the Levensthein distance (the degree of similarity of two sign chains) [3]. However, instead of letters words are used as the units to measure. What is interesting, is the fact WER can exceed 100%, so the value of WAcc can be less than zero.

The metrics defined above are based on the following recognition errors indicated for series of words in the context:

- *substitutions* – in the series of words the word is recognized as another word,
- *deletions* – the word appearing in the original series does not appear in the recognized series,
- *insertion* – the recognized series contains a new word between two words of the original series.

3. Implementation Details

In this section we define methods and algorithm we have developed for the implementation of our Polish numerals recognition system.

3.1. Database of Samples

We created a database of the 825 signal samples, which consists of wave type files containing numerals spoken by 80–83 speakers (depending on number there were 80 or more speakers). Experiments have been provided for each sample of from this database. To provide the independence of the speaker recognition, the samples were disposed which have already been uttered by the currently tested speaker. All tests except for comparing the effectiveness of various mitigation methods were carried out using the Least Angle Regression (LARS) Least Absolute Shrinkage and Selection Operator (LASSO) method.

The validation process is performed before classification. The results of the validation is usually a simple classification of the audio samples, which can be defined in the following way:

- properly validated samples – samples in which SCI rate exceeded the assumed threshold,
- correctly classified samples – samples that have been assigned to the correct class, excluding the validation,
- samples properly validated and classified,
- incorrectly validated samples – samples in which SCI did not exceed the assumed threshold, although it belongs to one of the classes from the database,
- samples that were incorrectly classified with the correct validation – samples that were incorrectly classified even though the validation proved positive,
- samples that were incorrectly classified and rejected with validation – samples, which in the absence of validation would be misclassified.

Table 1
Results of the recognition of sequences of the numerals

Number of iterations	30			50			70		
SCI	0.15	0.35	0.50	0.15	0.35	0.50	0.15	0.35	0.50
All samples	825	825	825	825	825	825	825	825	825
Properly validated samples	813	684	516	793	570	327	765	438	140
Correctly classified samples	767	767	767	777	777	777	777	777	777
Samples properly validated and classified	761	672	512	757	567	326	739	437	140
Incorrectly validated samples	12	141	309	32	255	498	59	387	140
Samples that were incorrectly classified with the correct validation	52	12	4	36	3	1	27	1	0
Incorrectly classified samples	58	58	58	48	48	48	48	48	48
Samples that were incorrectly classified and rejected with validation	6	46	54	12	45	47	21	47	48

Table 1 presents results of the recognition of sequences of the numerals – the results taking account of the SCI and the number of the iterations in which the LARS LASSO method was used.

3.2. Detection of the Initial and Terminal Elements in the Word

The isolated words are recognized and classified in the speech detection process. If the speech sequence is not divided into units it is hard to determine the boundary between the words. In both cases one should prepare the model samples to reflect the pure signal. The initial and final silence should be removed from the recording. The background noise makes it very difficult to determine the initial and final part of the word, because it disrupts the proper speech signal [7].

There are many algorithms for Voice Activity Detection. In the respective research the silence removing algorithm was used which makes use of signal energy, and the algorithm making use of zero crossing density. They can be used together. In both methods the first step is to divide the signal into equally long frames. Then for each frame the decision rule is made up in order to classify the segment as the speech signal or silence (background noise).

Signal energy algorithm. That particular algorithm is based on assumption that the speech signal energy in much higher than the background noise energy. For each frame the energy is calculated according to the formula [7]:

$$E_n = \sum_{n=1}^{N} [s(n)]^2 \,, \tag{5}$$

where N is the number of samples is the timeframe.

Then the result is compared to the threshold – if it is higher than the assumed value, the frame is classified as the part containing speech signal.

Zero crossing density algorithm. In that algorithm, the number of so called zero crossings of the signal is calculated for each frame, according to the [2]:

$$ZCR_n = \sum_{n=1}^{N} \Big| sgn\big[x(n)\big] - sgn\big[x(n-1)\big] \Big| \,, \tag{6}$$

where

$$sgm[x(n)] = \begin{cases} 1 & x(n) \geq 0 \\ -1 & x(n) < 0 \end{cases} \,. \tag{7}$$

One can assume, that if the zero crossing number is high the signal is background noise. Otherwise, it is speech.

3.3. Mel-filters Pool Used in the Implementation

For the purposes of the tests one use the pool of 40 FB-40 triangular filters which covers the frequency range [133, 6854] Hz at the assumed sampling frequency 16 kHz. Middle frequencies of the first 13 filters are distributed according to the linear relation (linear scale) in the range between [200, 1000] Hz in the interval of 66,67 Hz. Other 27 filters are distributed according to the logarithmic relation (logarithmic scale) in the range between [1071, 6400] with the step logStep = 1.0711703 [12]:

$$\text{logStep} = \exp\left(\frac{\ln\left(\frac{f_{c40}}{1000}\right)}{\text{numLogFilt}}\right) \tag{8}$$

where f_{c40} denotes the middle frequency of the last of logarithmic filters, numLogFilt is the numbers of logarithmic filters.

3.4. Compress Sensing Methodology – Main Concept

Compressed Sensing (or Compressive Sampling) is an emerging field that has attracted considerable research interest over the past few years and then has already become a key concept in various areas of applied mathematics, computer science, and electrical engineering.

Compressed Sensing has recently emerged as an efficient technique for sampling a signal with fewer coefficients than the number dictated by classical Shannon/Nyquist theory. The assumption underlying this approach is that the signal to be sampled is sparse or at least compressible, i.e., it must have a concise representation in a convenient basis. In Compressed Sensing, sampling is performed by taking a number of linear projections of the signal onto pseudorandom sequences. Therefore, the acquisition presents appealing properties such as low encoding complexity, since the basis in which the signal is sparse does not need to be computed, and universality, since the sensing is blind to the source distribution. Reconstruction of a signal from its projections can be done e.g. using linear programming, with a complexity that is $O(N^3)$, with N the number of samples to be recovered [13]–[15].

3.5. Implementation-Compressed Sensing Used in the Process of Classification

The classification method by means of compressed sensing is called sparse classification. The idea is to represent the feature vector of the examined sample as the linear combination of the feature vectors of relatively few samples taken from the teaching set. Then, the quality of that representation for each class is examined.

Because the samples can have various length but they are divided into frames of equal length, feature vectors can be given various length (and they are usually are given). In sparse classification we need feature vectors of equal length. In the implementation we used a method of changing the vector length (normalization) based on the linear interpolation.

3.5.1. Sparse Representation

Sparse representations are representations that account for most or all information of a signal with a linear combination of a small number of elementary signals.

The examined sample y can be represented as the linear combination of the samples taken from the teaching set $d_{i,n}$ where the first index $(1 \leq i \leq l)$ stands for one of the I classes, whereas the second $((1 \leq n \leq N_i)$ refers to the particular sample taken from and i class which consists of N_i samples [16]–[18]:

$$y = \sum_{i=1}^{1} \sum_{n=1}^{N} \alpha_{i,n} d_{i,n} \,. \tag{9}$$

As each sample is represented by k-elements feature vector, the base can be represented as the matrix.

3.5.2. Il-minimization

Usually, the number of teaching samples is much more numerous than the length of the feature vector for the single sample. In other words, it has more unknowns than equations [13]. Such a system of equations, called undetermined, has infinite number of solutions. We assumed, however, that x is a sparse vector (it contains mainly zero elements). Therefore we choose only the most “sparse” results from the all available, according to the compressed sensing theory. Because minimalization of such a solution in NP-complete problem we stick to the minimalization in the norm l^1. In other words, we strive for making the sum of absolute values of the x vector as small as possible [19]. In the perfect case, we achieve a vector which have nonzero components at positions pertaining to only one class. There are many methods how to find the sparse result of the set of equations. In our experiment, the so called LASSO method was used (Least Absolute Shrinkage and Selection Operator).

3.5.3. LASSO Solutions

LASSO solutions can be calculated by standard numerical algorithms. In the tests the method Least Angle Regression (LARS) was used with the modification that allows for solving the problem of the recognition of Polish numerals.

LARS algorithm is a modification of Forward Stepwise Regression [9], [17]. One single modification of the enables solving the LASSO problem. If the non-zero coefficient exceeds zero, the predictor is removed from the set of predictors and the direction is calculated once again. LARS method seek for the solution iteratively. Each step it adds one value to the vector of solutions. Therefore, after k-steps only k-elements of the resulting vector x are non-zero.

Before applying the LASSO, the y vector and every matrix A column are normalized. Each element of the y vector is divided by the vector length:

$$y_{norm} = \frac{y}{||y||} = \frac{y}{\sqrt{y_1^2 + \ldots + y_N^2}} \,, \tag{10}$$

where N stands for the number of elements of the vector.

LASSO is useful for us to solve the linear regression problem, yet it restricts the solution – the sum of absolute values of the resulting vector must be less than the given value. The model of the linear regression is determined as [8]:

$$y_i = a_i x_{i1} + \ldots + a_n x_{in} + \varepsilon_i = A x_i + \varepsilon_i, \quad i = 1, \ldots, k. \tag{11}$$

Vector of the prediction is defined as

$$\hat{\mu} = \sum_{j=1}^{n} a_j \hat{x}_j = A\hat{x}, \tag{12}$$

and the square error of the prediction is defined as

$$S(\hat{x}) = \left\| y - \hat{\mu} \right\| = \sum_{i=1}^{k} (y_i - \hat{\mu}_i)^2 \,. \tag{13}$$

May $T(\hat{x})$ be the L_1 of the vector $(\hat{x})$

$$T(\hat{x}) = \sum_{j=1}^{n} \left| \hat{x}_j \right| \tag{14}$$

The result of the LASSO method is the vector $(\hat{x})$, which minimizes the square error $S(x)$ with respect to the condition on $T(\hat{x})$:

$$minimizeS(\hat{x})forT(\hat{x}) \leq t\,. \tag{15}$$

If the value of the parameter t is high, it does not affect the above-mentioned restriction. But for the smaller values of t most of x_i coefficients have mostly zero value (their number is affected by t parameter). Therefore, sparse vector x being the solution of the set of equations is obtained. We will make use of that vector during the subsequent stages (validation and classification).

3.5.4. Classification

Having already sparse solution of equations, can be properly classified. For each class, a check is made as to its exact samples can play test sample. This error is called the residue. To retrieve to the sample with copies of only one class is defined function $\delta_i(x)$, which leaves unchanged the elements vector x associated with the class, and resets the remaining. The reconstituted vector can be defined as follows:

$$y_{i^R} = A\delta_i(x)\,, \tag{16}$$

where reproduction error is the difference between the test vector y and the vector reconstituted y_{i^R} in standard L^2.

3.5.5. Classification of Continuous Speech

Classification of continuous speech is a task much harder than the classification of the isolated words. In conducted tests, this would require taking a break while making each part number, which would be an unnatural phenomenon for the speaker. In order to identify all the spoken numbers without having to take breaks, we apply a sliding window method, called sliding window [8], while the same classification we use sparse classification method, which was presented in the previous parts of this work.

The sliding window technique consists in the fact that the speech feature vector of the test select fragment, called a window, comprising the features specified amount of time frames and the this fragment is subject to classification. At the beginning of the first T is selected frames of speech, then the window is moved to the end of the vector, taking into account the specific step Δ. It is important that the coefficients T and have a significant impact on the effectiveness of recognition. Longer step reduces the required amount of computation, while reducing effectiveness. The classification for each window, based on the values of SCI and residuum. It is easy to calculate that the number of windows, and thus the maximum number of classification is W = I – T + 1. In practice, the number of classifications is much smaller, because not all windows are validated.

For samples of each word conduct identical process as in the case of the test sample. Window sizes in both cases must be the same, while a step shift can vary, e.g., to reduce the number of copies of the training we can choose to move more than one frame.

To sum up: each part of the number is treated as a single word. A word can be spoken by many speakers. Each word consists of a number of copies produced in the process of moving the window on the vector characteristics of speech.

3.6. Validation and Sparsity Concentration Index

Validation is the process which allows for checking if the given voice sample is the proper sample from the base. Validation precedes the classification process, as only validated samples undergo classification. The SCI coefficient is applied to validate samples. Sparsity Concentration Index stands for the level of concentration of the non-zero coefficients of the x vector pertaining to one class. SCI = 1 means, that the tested sample is represented only by items from one class. SCI = 0 means, that the non-zero coefficients are set evenly for all classes.

In the experiments the threshold was adopted $\tau \in (0,1)$, which determines whether a sample is classified as unrecognized. If SCI$(x) < \tau$, the sample must be rejected, otherwise it should be classified [8].

4. Empirical Analysis

Testing methodology – isolated numbers. All test programs were implemented in the C++ language. The base of samples consisted of wave files containing the numbers spoken by speakers, which gives the total number of 825 files. The tests were conducted for each sample from the test base by means of cross-validation. During the recognition of the given sample from the test base, all the samples formerly recorded by the tested speaker were removed in order to ensure, that the recognition level is not dependent on the particular speaker. All the test, except the comparison of the efficiency of various minimalization methods, were conducted by virtue of LARS-LASSO method.

4.1. Testing Methodology – Series of Numbers

We tested series of numbers (0-9) spoken by 16 speakers. The test base was the same as for the isolated numbers, but with some additional samples reflecting background noise (silence), cut from the series of numbers. It turned out that the following parameters affected the recognition efficiency:

- window length,
- shifting the window for the samples for recognition,
- shifting the window for the model samples,
- number of iterations for the LARS LASSO method,
- SCI threshold enabling for determining if the given window should be ascribed to a class or if it should be removed as a gap between words (classes).

4.2. Experimental Results

The results according to the SCI threshold and the number of LARS Lasso method iterations. All the samples used in that test were proper samples taken from the base. The feature vector length was 35 frames. According to the conducted tests, SCI threshold = 0.35 at 70 iterations is enough for the validated sample to be correctly recognized (at least, it is highly probable). However, the higher is the threshold, the more severe are the validation criteria. As a result, the number of correct samples which are removed increases (Fig. 2).

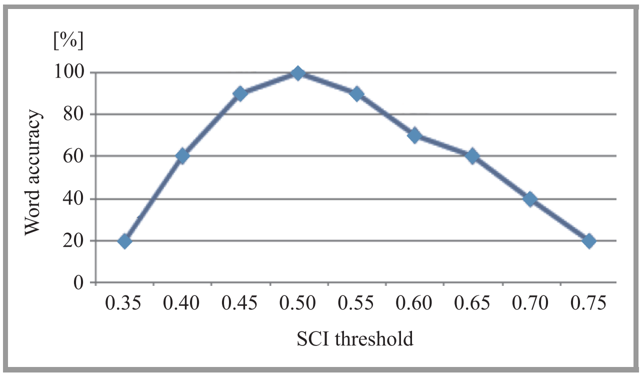

Fig. 2. The effect of SCI threshold on the efficiency of recognition.

4.2.1. Comparison of Sparse Classification to the *k*-nearest Neighbors

Figure 3 presents the percentage of the correctly and incorrectly recognized samples for two classification methods, namely Sparse Classification – minimization by virtue of the LASSO method at max. iteration number = 50, and *k*-nearest neighbors – for $k = 1$ or seeking only one nearest neighbor [20]. In both methods the feature vectors had the same length (35 frames). The classification was preceded by validation.

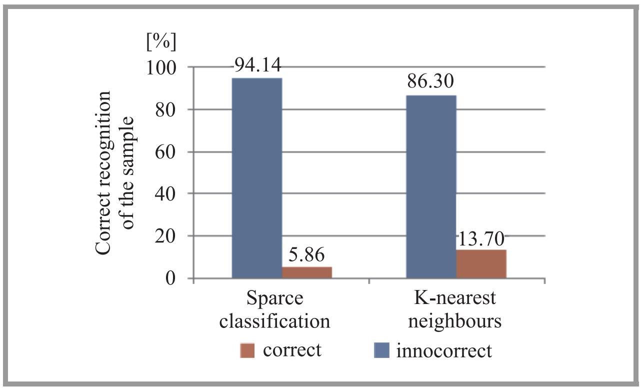

Fig. 3. The comparison of the various classification methods.

Sparse Classification method turned out to be more efficient than the standard *k*-nearest neighbors algorithm. It is because in the SC the model samples are considered more globally, whereas in the *k*-nearest neighbor method single incorrect sample from the base can lead to recognition error. The LARS LASSO method provides us with much better results, then. But there is also additional advantage of that algorithm. It is the time factor, which is so crucial in speech recognition.

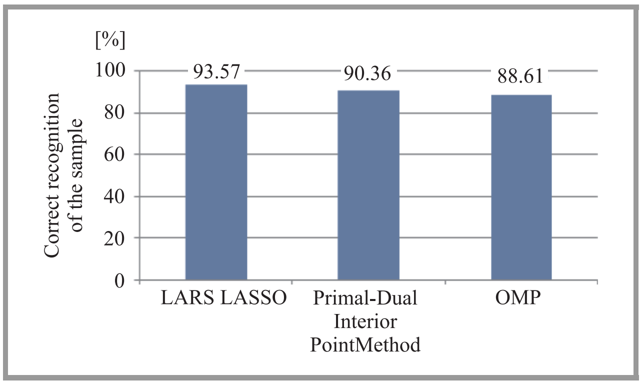

Fig. 4. The comparison of the efficiency of the various minimization.

Computation by means of LARS LASSO method, at the given parameters, took 0.14 s. OMP algorithm is faster (0.11 s) but its efficiency is much worse (Fig. 4). For the available Primal-Dual Interior Point Method it took terrible 11.3 s, despite the fact, we significantly shortened the feature vector.

4.2.2. Results of the Recognition of Sequences of the Numerals

The recognition of sequences of the numerals experiments have been conducted under the set of the key parameters presented in Table 2.

Table 2
Key parameters for the recognition of sequences of the numerals

Parameter	Value
The window size	45
Move the main window	2
Move the window of the samples	5
The maximum number of iterations of LARS LASSO	50
Threshold of SCI	0.5

Detailed results of the recognition os sequences of the numerals are presented in Table 3.

The technique of “sliding window” in conjunction with the method of classification “Sparse Classification” and validation based on the ratio of SCI, gives good results for the recognition of strings of digits.

Table 3
Results of the recognition of sequences of the numerals

Filename	WAcc [%]	Recognized string
Adrian_G_80514_22_10.wav	80	0 1 2 3 4 9 6 7 4 8 9
Adrian_S_80577_21_0_10.wav	90	0 1 2 3 4 5 6 7 8 4 9
Krzysztof_U_80582_22_0_10.wav	80	0 1 2 3 4 5 6 7 8 5 9 5
Lukasz_K_80534_21_0_10.wav	80	0 1 2 3 4 6 9 5 6 7 8 9
Maciek_M_80541_21_0_10.wav	90	0 1 2 3 4 5 6 7 8 9 5
Marcin_K_80523_21_0_10.wav	80	0 2 3 4 9 6 7 8 9
Marek_W_80589_21_0_10.wav	70	0 2 4 5 6 7 8 9 5
Michal_J_80519_21_0_10.wav	100	0 1 2 3 4 5 6 7 8 9
Michal_M_82432_23_0_10.wav	100	0 1 2 3 4 5 6 7 8 9
Michal_P_80558_22_0_10.wav	80	0 1 2 3 4 5 6 7 1 8 9 5
Pawel_M_80545_22_0_10.wav	80	0 1 3 4 5 6 5 8 9
Piotr_B_80497_22_0_10.wav	80	0 1 2 3 4 9 5 6 7 7 8 9
Radoslaw_S_80567_22_0_10.wav	70	0 4 2 4 4 9 6 7 8 9
Witold_P_80556_22_0_10.wav	100	0 1 2 3 4 5 6 7 8 9
Witold_U_80583_22_0_10.wav	70	0 2 3 4 6 7 8 5
Wojciech_Z_80602_21_0_10.wav	70	0 1 2 4 6 6 7 8
Average:	83	

5. Conclusions

According to the experiment results, the Compressed Sensing and Sparse Classification methods are very efficient in recognition of the isolated words and can find multiple applications f.ex. in recognizing voice commands in remotely controlled appliances.

Moreover, the SCI coefficient-based validation is quite effective. The application of the above-described validation method and Sparse Classification in the voice identification systems is also worth the separate research. One can also examine so called Sparse Imputation technique enabling for the recognition of words in a very noisy background. It is crucial, how to prepare the pool of model samples. The number of good quality speech samples should be vast. According to the experiment, the more model samples in the base, the more efficient is the recognition system.

References

[1] A. M. Peinado and J. C. Segura, *Speech Recognition over Digital Channels. Robustness and Standards*. Chichester, England: Wiley, 2006.

[2] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. New Jersey: Pearson Education, 2009.

[3] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. New Jersey: AT&T, 1993.

[4] P. Walendowski, *An Application of SVM Artificial Neural Networks in Speech Recognition*. Wrocław: Politechnika Wrocławska, 2008 (in Polish).

[5] B. Plannerer, "An introduction to speech recognition", tutorial, University of Munich, Germany [Online]. Available: http://www.speech-recognition.de/ 2008.

[6] W. Kasprzak, "Image and speech recognition", E-lecture notes, Warsaw University of Technology, 2011, updated version 2012 [Online]. Available: www.ia.pw.edu.pl/∼wkasprza/PAP/EIASR_2012.pdf

[7] R. G. Bachu, S. Kopparthi, B. Adapa and B. D. Barkana, "Separation of voiced and unvoiced using zero crossing rate and energy of the speech signal", American Society for Engineering Education (ASEE), Zone Conference Proceedings, 2008, pp. 1–7.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[9] J. F. Gemmeke and B. Cranen,"Using sparse representations for missing data imputation in noise robust speech recognition", in *Proc. 17th Eur. Sig. Proces. Conf. EUSIPCO 2009*, Glasgow, Scotland, 2009, pp. 1755–1759.

[10] J. Szabatin, *Theory of the sygnals*, Warszawa: Wydawnictwa Komunikacji i Łączności, 1982 (in Polish).

[11] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various MFCC implementations", in *Proc. 10th Int. Conf. Speech & Comp. SPECOM 2005*, Patras, Greece, 2005, pp. 191–194.

[12] J.-L. Starck, F. Murtagh, and J. M. Fadili, *Sparse Image and Signal Processing: Wavelets, Curvelets, Morphological Diversity*. Cambridge University Press, 2010.

[13] D. Needeel, "Topics in compressed sensing", Ph.D. thesis, University of California, Davis, 2009.

[14] N. Vaswani and W. Lu , "Modifed-CS: Modifying compressive sensing for problems with partially known support", *IEEE Trans. Sig. Proces.*, vol. 58, no. 9, pp. 4595–4607, 2010.

[15] E. Candes,"The restricted isometry property and its implications for compressed sensing", *Compte Rendus de l'Academie des Sciences*, vol. 346, no. 9–10, pp. 589–592, 2008.

[16] J. F. Gemmeke and B. Cranen, "Noise robust digit recognition using sparse representations", in *Proc. ISCA Tutor. Res. Worksh. Speech Anal. Proces. Knowl. Discov.*, Aalborg, Denmark, 2008, pp. 1–4.

[17] J. F. Gemmeke, "Classification on incomplete data using sparse representations: Imputation is optional", in *Proc. Benelearn 2008*, Spa, Belgium, 2008, pp. 71–72.

[18] J. Wright *et al.*, "Robust face recognition via sparse representation", *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, 2009.

[19] D. Salomon, *Data Compression: The Complete Reference*. Springer, 2007.

[20] N. Bhatia and Vandana,"Survey of nearest neighbor techniques", *Int. J. Comp. Sci. Inform. Secur.*, vol. 8, no. 2, pp. 302–305, 2010.

Anna Plichta graduated comparative literature at the Jagiellonian University in 2007. She also graduated computer science at Cracow University of Technology in 2010. Currently, she is an Assistant Professor at Tadeusz Kościuszko Cracow University of Technology. The main topics of her research are pattern recognition, artificial intelligent systems and e-learning technologies.

E-mail: aplichta@pk.edu.pl
Institute of Computer Science
Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Tomasz Gąciarz was studied computer science at the AGH University of Science and Technology and obtained his degree in 1994. Currently, he works as a teaching fellow at the Tadeusz Kościuszko Cracow University of Technology. The main topics of his research are artificial intelligent systems.

E-mail: tga@pk.edu.pl
Institute of Computer Science
Cracow University of Technology
Warszawska st 24
31-155 Cracow, Poland

Tomasz Krzywdziński was studied computer science at the Tadeusz Kościuszko Cracow University of Technology and obtained his degree in 2010. Currently, he works at the Comarch company.

E-mail: tkrzywdzinski@gmail.com
Comarch SA
Jana Pawła II av. 39a
31-864 Cracow, Poland