

Recognition of Human Speech using q -Bernstein Polynomials

Abdulkadir Karacı
Kastamonu University
Department of Computer and
Instructional Technology Education,
Education Faculty, Turkey

Ibrahim Büyükyazıcı
Gazi University
Department of Mathematics
Turkey

Muharrem Aktümen
Ahi Evran University, Department of
Mathematics Education, Education
Faculty, Turkey

ABSTRACT

Recognition of human speech has long been a hot topic among artificial intelligence and signal processing researchers. In this paper, Pattern recognition and q -Bernstein polynomials have been combined to create computer software designed for the recognition of human speech, and the main principles of speech recognition have been explained in a systematic content. q -Bernstein polynomials, which are mathematical operators, have been applied for pattern recognition, and a new method has thus been developed. Software has been prepared using the Delphi 7 programming language, and with this software, this method has been applied for the processing of verbal expression recognition. In the program as developed, 16 000 samples of 8-bit stereo images were computerized. Speech recognition tests were conducted for six words, and information related to the results of this test is provided in this paper.

Keywords

Speech recognition, pattern recognition, q -Bernstein polynomials

1. INTRODUCTION

Recognition of human speech is a problem with many solutions, but is still an open problem because none of the current methods are fast and precise enough to be comparable with the recognition capability of human beings. Several methods exist for recognition of human phonemes, including hidden Markov models (HMMs) [1], time-delay neural networks [2], support vector classifiers with HMM [3], independent component analysis [4], HMM and neural-network hybrids [5], and more.

Speech command recognition provides important advantages for communication between human and computer. Obtaining speech command data is very easy and does not require any special talent such as keyboard skills, in contrast to other data storage methods. Using speech commands, transferring texts to electronic form can be extremely fast. This method provides the users with freedom of movement and the possibility of using their hands for other purposes [6].

The first step in recognizing speech commands by computer is to transform voice signals into statistical form to take incidental characteristics of the sound out of the signal and to make a recognition procedure possible by providing it with the information obtained. In conveying the speech command to the recognition procedure, the transfer of the speech command to the computer by microphone is the first step. By transforming the

voice signal into statistical form, it is made ready for windowing, filtering, and other analyses. By these procedures, any noise existing in the voice signal is removed, and the signal is separated out from the elements which are dependent on the person or on the geographical area [7].

Two voice vectors are extracted in the pattern recognition method. The similarities and differences between the vectors are determined using various distance measurement methods. The first step in the pattern recognition procedure is to apply preparatory procedures to the raw voice data. In this study, in contrast to the usual pattern recognition procedure, after the voice data have been acquired and prepared, q -Bernstein polynomials are applied to the statistical voice data. The statistical voice data which have been processed through q -Bernstein polynomials come to resemble each other more closely, and this situation increases recognition performance.

In this paper, q -Bernstein polynomials, which are mathematical operators, have been applied to pattern recognition, and a new method has thus been developed. Software has been developed using the Delphi 7 programming language, and with the aid of this software, the method has been applied to process verbal expressions for recognition. Moreover, in the program as developed, 16 000 samples of 8-bit stereo images were computerized, speech recognition tests were conducted for six words, and results were obtained as reported later in this paper. We state that this study is an experimental work.

2. q -BERNSTEIN POLYNOMIALS

In recent years, q -Bernstein polynomials have attracted a great deal of interest because of their potential applications in approximation theory and numerical analysis, and many properties of these polynomials have been discovered [8, 9, 10, 11, 12].

Let $q > 0$. For each nonnegative integer n , the q -integer $[n]_q$ can be defined as

$$[n]_q = \begin{cases} (1-q^n)/(1-q) & \text{if } q \neq 1 \\ n & \text{if } q = 1 \end{cases}$$

and the q -factorial $[n]_q!$ as

$$[n]_q! = \begin{cases} [n]_q [n-1]_q \cdots [1]_q & \text{if } n \geq 1 \\ 1 & \text{if } n = 0 \end{cases}$$

For the integers n and k , with $0 \leq k \leq n$, the q -binomial coefficients are then defined as follows [9]:

$$\begin{bmatrix} n \\ k \end{bmatrix}_q = \frac{[n]_q!}{[k]_q! [n-k]_q!}$$

In 1997, Phillips [11] generalized Bernstein polynomials to q -Bernstein polynomials:

$$B_{n,q}(f;x) = \sum_{k=0}^n f\left(\frac{[k]_q}{[n]_q}\right) \begin{bmatrix} n \\ k \end{bmatrix}_q x^k \prod_{s=0}^{n-k-1} (1-q^s x), \quad (1)$$

$$0 \leq x \leq 1, n=1,2,3,\dots$$

where an empty product is taken to equal 1. Explicit expressions for $B_{n,q}(t^r;x)$ for $r=0,1,2$ can be obtained by direct calculation, giving

$$B_{n,q}(1;x) = 1$$

$$B_{n,q}(t;x) = x$$

$$B_{n,q}(t^2;x) = x^2 + \frac{x(1-x)}{[n]_q}$$

Note that if $f(x) \geq 0$ on $[0,1]$, this implies that $B_{n,q}(f;x) \geq 0$ on $[0,1]$ for $0 < q < 1$. These properties of q -generalized Bernstein operators enable the use of the Korovkin theorem [13] to obtain convergence results. Because for a fixed value of q with $0 < q < 1$, $\lim_{n \rightarrow \infty} [n]_q = \frac{1}{1-q}$, to ensure the convergence properties of (1), $q = q_n$ will be assumed to be a sequence such that $q_n \rightarrow 1$ as $n \rightarrow \infty$ for $0 < q_n < 1$.

Theorem 1. Let (q_n) be a sequence of real numbers such that $0 < q_n < 1$ and $\lim_{n \rightarrow \infty} q_n = 1$. Then, for $f \in C[0,1]$, $B_{n,q_n}(f;x)$ converges uniformly to $f(x)$ on $[0,1]$. The proof can be found in [11].

Theorem 2. Let (q_n) be a sequence of real numbers such that $0 < q_n < 1$ and $\lim_{n \rightarrow \infty} q_n = 1$. Then, for $f \in C[a,b]$,

$$B_{n,q_n}(f;x) = \sum_{k=0}^n f\left(a + \frac{[k]_{q_n}}{[n]_{q_n}}(b-a)\right) \begin{bmatrix} n \\ k \end{bmatrix}_{q_n} \left(\frac{x-a}{b-a}\right)^k \prod_{s=0}^{n-k-1} \left(1 - q_n^s \frac{x-a}{b-a}\right),$$

converges uniformly to $f(x)$ on $[a,b]$. Theorem 2 can be proved in the same way as Theorem 1.

Example 1. For $n=2,5,10,20,30$, the convergence of $B_{n,q_n}(f;x)$ to $f(x) = \frac{\sin x}{x}$ is illustrated in Fig. 1.

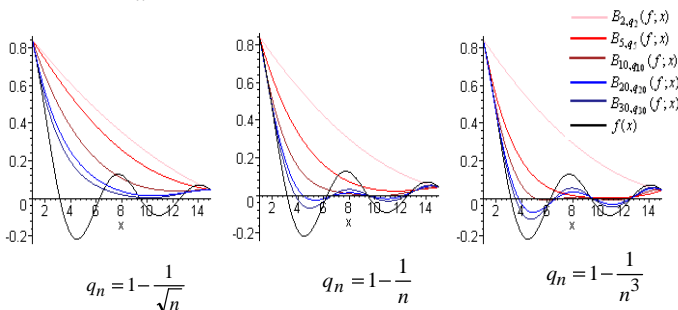


Fig. 1. Convergence of $B_{n,q_n}(f;x)$ to $f(x)$.

Example 2. For $n=2,5,10,20,30$, the convergence of $B_{n,q_n}(f;x)$ to $f(x) = x \cos x$ is illustrated in Fig. 2.

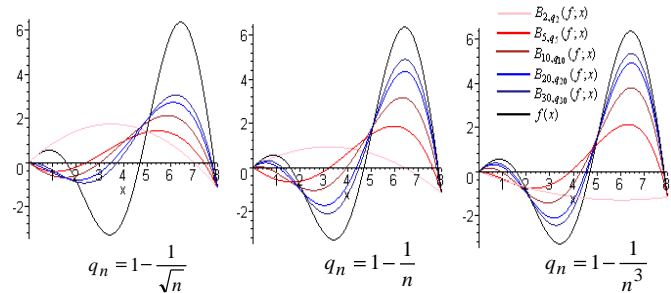


Fig. 2. Convergence of $B_{n,q_n}(f;x)$ to $f(x)$.

3. PREPARATIONS FOR SPEECH RECOGNITION

Much processing is needed between acquiring the voice signal and achieving speech recognition, which constitutes the speech recognition procedure. The recognition procedure can be approached by providing a basic overview of the preparations for speech recognition. The method obtained is illustrated in Fig. 3, step by step and generally [7].

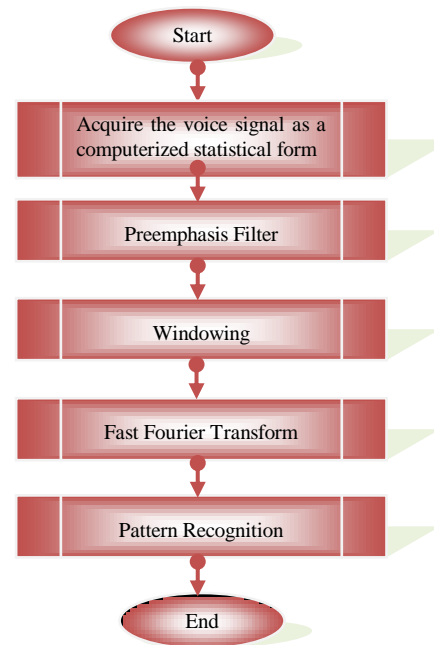


Fig. 3. A general view of speech recognition.

4. DESIGN OF THE SPEECH RECOGNITION SOFTWARE

The software developed in this study recognizes the limitation that spoken words vary from one person to another by incorporating q -Bernstein polynomials into the pattern recognition method. In the software, the user first creates a dictionary database with the words that he or she wants to be recognized. The software applies the Bernstein function to the words that the speaker said and all the words in the database

during word recognition. Then it performs a comparison procedure, calculating a recognition percentage for all the words in the database one by one and returning the word for which the measurement is most like the observed word. If further procedures need to be performed for the particular word found, the software executes them. The general sequence of the software as developed is shown in Fig. 4.

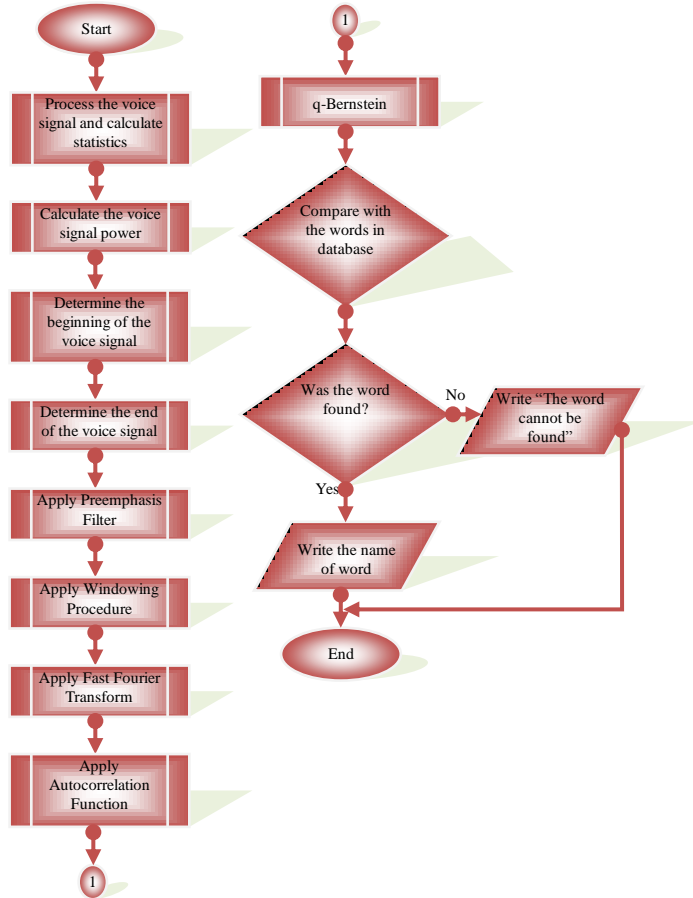


Fig. 4. General schematic of the software as developed.

4.1 Voice Acquisition

The voice signal is acquired using a Philips SHM3300 microphone. While the voice signal is being acquired, the codes described below are entered in the “audio 1” component of the data record. Moreover, in the software as developed, 16 000 samples are stored in the database as 8-bit stereo recordings.

```

procedure TForm1.Audio1Record(Sender: TObject; LP,
RP:pointer;BufferSize: Word);
begin
  defa:=defa+1;
  if defa=1 then liste1.items.add('Başla') else
liste1.items.add(inttostr(buffersize));
  gercekses:=lp; Veriyikutupla(lp,sonsize);
  if kutupkont.Checked=true then
Veriyigoruntule(sayisalsessinyali,image1,sonsize); Enerji;
sesyakala; end;

```

4.2 Data Polarization

Data polarization is used for processing the data samples from the voice card in a way which is independent of the 8-bit or 16-bit sampling procedure. The data as received are kept in 32-bit form for further processing. The data polarization subprocedure used here deducts 127 from the voice signal in the case of 8-bit data. This is necessary to identify the positive and negative oscillations of the voice. The results can be converted into 32-bit data and processed in a way which is independent of 8-bit or 16-bit processing. The subprocedure for data polarization is presented below.

```

procedure veriyikutupla(data:pointer;size:integer);
var
  p:array[0..50000] of integer; b:byte; i:integer; z:integer;
begin
  b:=data;
  for i:=0 to epenbuyukluk-1 do
    p[i]:=oncekiveri[i];
  for i:=epenbuyukluk to size-1 do
    p[i]:=b^[i-epenbuyukluk]-127;
  for i:=size-epenbuyukluk to (size-1) do oncekiveri[i-(size-epenbuyukluk)]:=p[i]; sayisalsessinyali:=@p; for z:=1 to 50000 do pheryerde[z]:=p[z];end;

```

4.3 Voice Signal Power Calculation Procedure

To detect the start and end of the voice signal, the intensity that represents the energy of the voice signal must be determined. Since 1970, energy parameters have been used to determine the end point of the voice signal. One of the basic characteristics of the voice signal is its power. A signal evaluation method frequently used in speech recognition systems determines the wave power as the sum of squared power values for every point of the wave or its density in an arbitrary time window [14]. This energy is calculated using a short-term energy formula. This energy value is used to calculate the energy in the voice signal over an arbitrary time period [15].

Sum of squares of energy:

$$E = \sum_{i=1}^N x(i)^2 \quad (2)$$

4.3.1. Short-term energy

Short-term energy is the energy that the voice signal has over a certain period of time. There are three different definitions used in short-term energy calculations [15]:

Logarithm of energy:

$$E = \sum_{i=1}^N \log x(i)^2 \quad (3)$$

Sum of squares of energy:

$$E = \sum_{i=1}^N x(i)^2 \quad (4)$$

Sum of absolute energy:

$$E = \sum_{i=1}^N |x(i)| \quad (5)$$

4.4 Procedure for Capturing the Voice Signal

It is very important to determine the voice signal accurately in noisy locations. A wrong determination of the voice ending point in speech recognition has two negative effects: first, a recognition failure occurs because of the false determination of the voice ending boundary; second, when the voice boundary is determined wrongly, the computation time increases because non-voice segments of the signal must be processed [16].

4.4.1. Procedure for establishing beginning and end of the voice signal

To establish the beginning and end of a voice signal, it is necessary to look at the energy values calculated. The value of the minimum energy threshold and the number of samples over which values greater than this threshold must be maintained must be determined. In the procedure used here, the number of samples is set to 50 and the threshold value to 40. The best threshold value can be found by experimentation. The program examines the voice energy value beginning at a predetermined point; if the energy reaches a value of 50 and then remains above that level for the next 40 samples, it means that the beginning of the voice signal has been found.

When the end of the voice signal is found, the energy values are looked at again. A reduction of energy to less than the maximum energy threshold value determined earlier indicates that the voice signal has finished and defines the number of samples to be provided to the program. These two parameters are optimized using a voice capture control window in the program. The default values of these parameters are 30 and 40, meaning that if over 30 samples, the energy remains less than 40, this means that the end of the voice signal has been established.

4.5 Preemphasis Filter

Spoken utterances have high energy values at low frequencies. The preemphasis filter is used to ensure that low-frequency components do not mask the high-frequency ones.

The preemphasis filter is usually described using the relation in equation 6:

$$P(z) = 1 - \mu z^{-1} \quad (6)$$

where μ squared varies between 0.9 and 0.95 [17], and z represents a standard delay [7].

The following program code implements a preemphasis filter:

```

Procedure      preemphasisfilter(rawdata:pointer;size:integer;
coefficient:real);
var
preemphasisdata:pointerintegerdizi; i:integer;
begin
preemphasisdata:=rawdata;
if preemphasisdata^0>10 then preemphasisdata^0:=0;
for i:=0 to size-1 do begin

```

```

preemphasisdata^[i+1]:=round(preemphasisdata^[i+1]-
coefficient*preemphasisdata^[i]);
end; end;

```

The pointer variable in the preemphasis filter procedure points to the voice data to which the preemphasis filter is being applied. When the preemphasis filter is applied, if the first element of the numeric voice data vector is greater than the cutoff value, it is zeroed by `preemphasis^0:=0;`. Then in a `For` loop, the coefficient is multiplied by the previous sample data (`preemphasisli^[i-1]`) and inferred from the sample data of that time period (`preemphasisli^[i]`), and the data vector with preemphasis filtration applied is thus obtained. This procedure continues until the sample number is -1. Figure 5 illustrates the application of a preemphasis filter to the voice signal corresponding to the word “Word.”

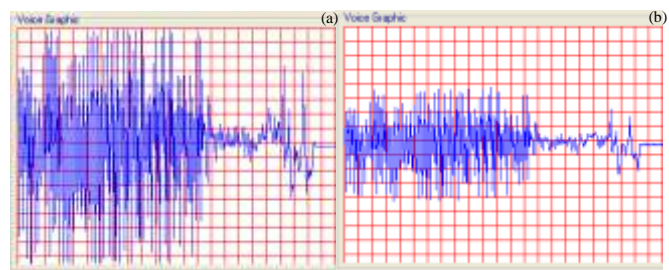


Fig.5. (a) Raw voice signal. (b) Voice signal after application of preemphasis filter.

4.6 Windowing

A captured voice signal must be windowed and a certain part selected for manipulation by Fourier analysis. The part selected must correspond to the smallest unit of meaning and enable Fourier analysis calculations to be performed in a short time (generally 30 ms [7]). Voice signal graphs, the applied raw voice signal, and Hamming windowing are shown in Fig. 6.

Windowing means the multiplication of the voice signal $y(n)$ by a time-varying function $W(n)$. Windowing can be implemented by starting a window at the point where the previous window finishes, so that the two adjacent windows do not overlap, or else the later window can be started a bit earlier, so that the two windows overlap. In this way, the hard limits in the procedure can be softened.

In the software developed here, different types of windowing such as Square, Blackman, Hanning, Hamming or Bartlett can be used according to the choice of the user. In this study, the Hamming windowing method has been used because it yields better results.

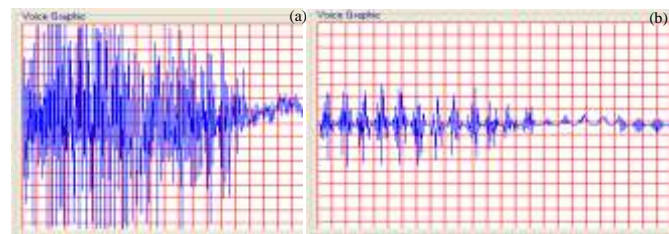


Fig.6. (a) Raw voice signal. (b) Voice signal with applied windowing.

4.7 Fast Fourier Transform (FFT)

The Fast Fourier Transform (FFT) divides a voice signal into its constituent frequencies. A normal voice signal is formed by combining voice components of various frequencies, like the colors in a rainbow. The voice signal in its raw form can create different impressions, although various speakers are saying the same word. It is difficult to characterize a voice signal using only its raw properties (time and amplitude). For this reason, the voice signal is manipulated in various ways to make recognition easier. FFT is one of these techniques [18].

The Fast Fourier Transform (FFT) has a long and interesting history. It was originally discovered by Gauss and then rediscovered by Cooley and Tukey and became widely known. The FFT is an algorithm for quick calculation of the Discrete Fourier Transform (DFT). The DFT is one of the main procedures used in digital signal processing for linear filtration, which is used in many applications. However, the calculation time required in DFT for a voice signal of finite size is greater than in other methods [19]. FFT is an efficient method for calculating the DFT. The reason that this model is considered efficient is that it can be used to solve a variety of problems, and therefore this technique has received great attention. Procedures can be implemented more quickly with FFT than with DFT. FFT has a computational complexity of order $N \log_2 N$ for sample size N , whereas DFT has a complexity of order N^2 . For instance, if $N=1024$, FFT will require time $N \log_2 N=10\ 240$, but DFT will need time $N^2=1\ 048\ 576$ [20].

The DFT function is defined in equation (7) [21]:

$$X(j) = \sum_{k=0}^{N-1} A(k)W^{jk} \quad , \quad j = 0,1,\dots, N-1 \quad , \quad (7)$$

where $W = e^{2\pi i / N}$.

The raw voice signal and the voice signal after application of FFT are shown in Fig. 7.

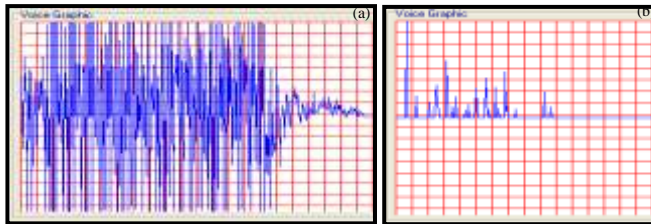


Fig.7. (a) Raw voice signal. (b) Voice signal after application of FFT.

4.8 Autocorrelation Function

The autocorrelation function identifies the main frequencies in a signal and makes recognition easier [7]. Discovering characteristic frequencies in noisy environments is a key technique for correct voice recognition. In many existing voice recognition systems, the recognition of characteristic frequencies depends on the quality of the voice data after the correction procedure. The autocorrelation function has been found to provide the best performance in noisy environments [22].

In the speech recognition program developed here, the Blackman-Tukey autocorrelation distance is used. The Blackman-Tukey autocorrelation distance is described by equation 8 [23]:

$$R(k) = (1/(N-k)) \sum_{i=0}^{N-1-k} x(i).x(i+k) \quad . \quad (8)$$

4.9 Recognition

This is the step where the manipulated voice signal from the speaker is compared with source signals saved in a database after previous manipulations. In this section, in contrast with the usual pattern recognition approaches, the q -Bernstein function is applied both to all the reference signals in the database and to the voice signal from the speaker. Then a comparison procedure is performed in an effort to find the closest reference signal. If the correspondence is lower than 80 percent, this means that the word has not been found. A subprogram which applies the q -Bernstein function to voice data is given below.

```

Procedure Tformyakalanan.qbernstein(size:integer);
var
pencere,pencsay,tuti,x,ne:integer; iyerine:integer; s,k,i,j:integer;
carp:double; toplamf,toplamg:real;
begin
penbuy:=1;
pencere:=size div penbuy;
for pencsay:=0 to pencere do begin
for x:=penbuy*pencsay to penbuy+penbuy*pencsay do begin
toplamg:=0;toplamf:=0;
for j:=0 to penbuy do begin
carp:=1;
for s:=0 to penbuy-j-1 do begin
carp:=carp*(1-(power(1-1/penbuy,s)*(x-
penbuy*pencsay)/(penbuy+penbuy*pencsay)));
end;
toplamg:=toplamg+qbinomial(penbuy,j)*power((x-
penbuy*pencsay)/(penbuy+penbuy*pencsay),j)*carp*ffttalanokun
an[j+x];
toplamf:=toplamf+qbinomial(penbuy,j)*power((x-
penbuy*pencsay)/(penbuy+penbuy*pencsay),j)*carp*ffttalan[j+x];
end;
topl[x]:=toplamf;
topg[x]:=toplamg;
end;end;end;

```

Various distance measurements can be used to calculate the degree of resemblance between voice signals. In the program developed here, the Oclid measurement is used. The Oclid measurement finds the geometric distance between two points in a vertical coordinate system and can be defined as:

$$d(X,Y)_{oclid} = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad . \quad (9)$$

The voice signal being recognized is compared with a reference signal in the database using the Oclid measurement. In this procedure, the Oclid measurement for every window is calculated by dividing the two voice signals into a window eight

times larger than the normal window size. If the calculated Oclid value is smaller than the acceptance threshold value determined by the program, the window size for the observed signal is increased by one. The procedure is repeated for every window. Finally, the recognition percentage is calculated as follows:

$$\text{Recognition Percentage} = \frac{\text{ObservedWindowSize} * 100}{\text{Total WindowSize}} \quad (10)$$

These procedures are repeated for all the words stored in the database.

After these procedures, the program returns whether or not the voice signal has been identified, the recognition percentage associated with the identification, and which word has been identified. If desired, further processing specific to the identified word can be performed. For example, for the word “word,” a routine specific to this word can be performed. Furthermore, the program lists the recognition percentage for every reference contained in the database. In Fig. 8, the recognition percentage of the word “word” as determined by the program is shown.

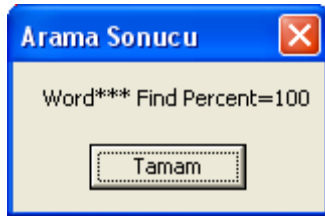


Fig. 8. Recognition percentage of the word “word” as determined by the program.

5. EXPERIMENTAL RESULTS

In this study, software has been developed for the recognition of spoken utterances using the q -Bernstein mathematical function as the pattern recognition method. At the end of the tests, it was seen that the voice recognition rate achieved by the software proved to be higher than that achieved by the pattern recognition method without q -Bernstein. Moreover, in many pattern recognition methods, words are sometimes confused with each other, but in the pattern recognition method with q -Bernstein, words are not confused.

If the voice signal as recognized by the voice recognition software is similar to the real voice signal at a level of more than 80%, it means that the word is recognized. The similarity ratio is obtained by comparing the window values to a threshold value. As a result of the tests, it was determined that the threshold value yielding the best result is 51. In this software, at the end of the tests held separately with single sampling, the minimum recognition rate was 100%. Moreover, it was observed that the greater the number of samples in the database, the higher the similarity ratio, which increases recognition performance.

Using the software as developed, the words *Word*, *Excel*, *Windows*, *Exit*, *Next*, and *Open* were recorded in a database, and ten tests were carried out for each word. It can be seen that the style of pronouncing the words and the tone of voice have an impact on recognition. The experimental results belonging to voice recognition program which has been developed by using the q -Bernstein function are shown in Fig.9 and the experimental results belonging to voice recognition program which has been

developed without using the q -Bernstein function are shown in Fig.10.

As it is shown in Fig.10, in accordance with the acquired test results from the voice recognition program which has been developed without the q -Bernstein function, the recognition rates of the words are low. Besides, some words are not recognized even false recognitions have been occurred. Due to the uniformly convergence, shape-preserving and q -generalization of the Bézier curves, giving them a practical application features of the q -Bernstein function, the problems like false recognition of words the or non-recognition of words are prevented.

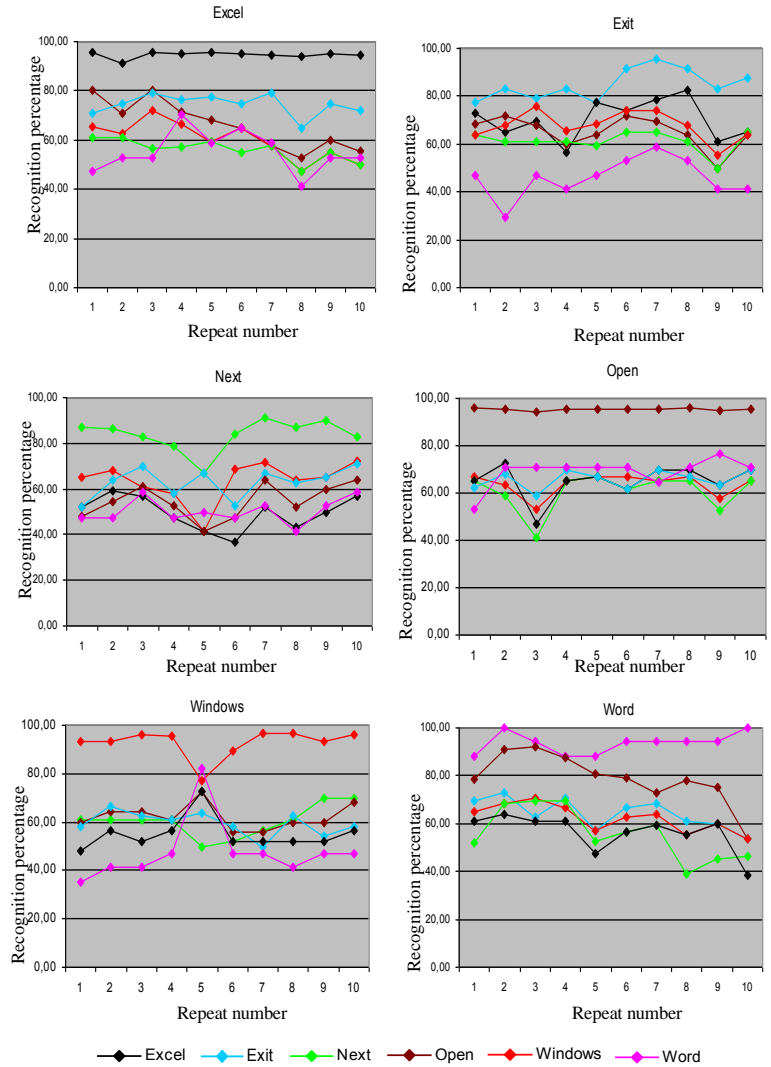
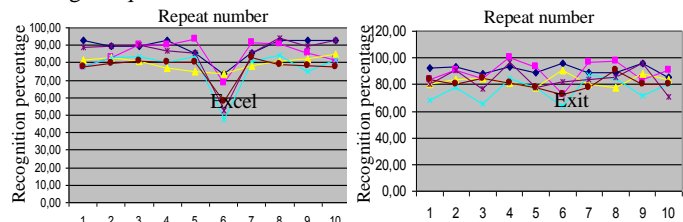


Fig. 9. The experimental results belonging to voice recognition program which has been developed by using the q -Bernstein function.



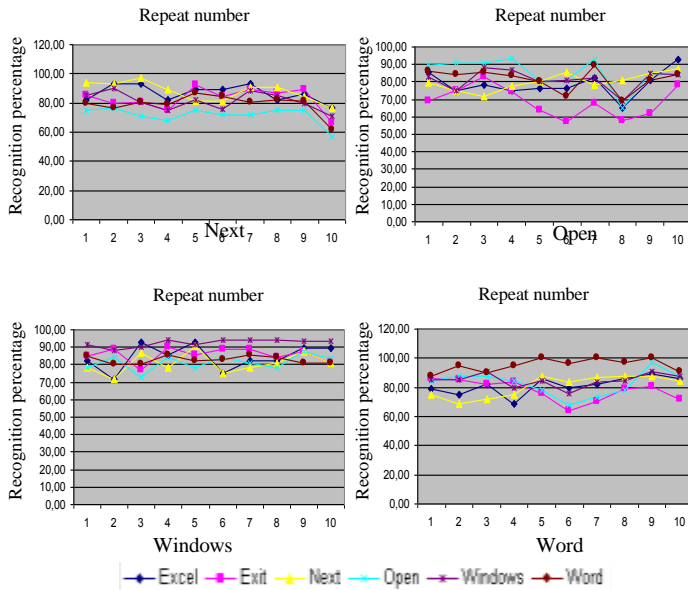


Fig. 10. The experimental results belonging to voice recognition program which has been developed by not using the q -Bernstein function.

6. CONCLUSIONS AND FUTURE WORK

In this study, q -Bernstein polynomials, which are mathematical operators, have been applied to pattern recognition, and thus a new method has been developed. This method was applied to six words using the software developed in this research. The data obtained have shown that the human speech recognition rate was higher using this method than with the pattern recognition method without q -Bernstein polynomials. Use of the same method for voice and fingerprint recognition technologies is under consideration.

7. REFERENCES

- [1] B. Babaali, H. Sameti, The Sharif speaker-independent large vocabulary speech recognition system. Proceedings, 2nd Workshop on Information Technology and its Disciplines, Kish Island, Iran, 2004.
- [2] M.R. Berthold, A time-delay radial basis function network for phoneme recognition. Proceedings of the IEEE International Conference on Neural Networks, 7, Orlando, 4470–4473, 1994.
- [3] S.E. Golowich, D.X. Sun, A support vector/hidden Markov model approach to phoneme recognition, ASA Proceedings of the Statistical Computing Section, 125–130, 1998.
- [4] O.W. Kwona, T.W. Lee., Phoneme recognition using ICA-based feature extraction and transformation. Signal Process. 84(6) (2004), 1005–1019.
- [5] P. Schwarz, P. Matejka, J. Cernocky, Phoneme recognition based on TRAPs, Proceedings of the Workshop on Multimodal Interaction and Related Machine Learning Algorithms, Switzerland, 2004.
- [6] E. Mengüoğlu, Rule-based design and realization of a Turkish verbal expression recognition system. Master's Thesis, Hacettepe University, Institute of Science, Ankara, 1999.

- [7] S. Doğan, Recognition of verbal instructions in PC medium. Master's Thesis, Marmara University, Institute of Science, Istanbul, 1999.
- [8] W. Heping, X.Z. Wu, Saturation of convergence for q -Bernstein polynomials in the case $q \geq 1$. J. Math. Anal. Appl. 337(1) (2008), 744–750.
- [9] V. Kac, P. Cheung, Quantum Calculus. Springer, New York, 2002.
- [10] S. Ostrovska, q -Bernstein polynomials and their iterates. J. Approx. Theory 123(2) (2003), 232–255.
- [11] G.M. Phillips, Bernstein polynomials based on the q -integers, Ann. Numer. Math. 4 (1997), 511–518.
- [12] V.S. Videnskii, On some classes of q -parametric positive operators, Operator Theory Adv.Appl. 158 (2005), 213–222.
- [13] F. Altomare, M. Campiti, Korovkin Type Approximation Theory and Its Applications, de Gruyter Series Studies in Mathematics, Vol.17, Walter de Gruyter, Berlin-New York, 1994.
- [14] B. Akçay, Recognition of Turkish speech through artificial neural networks. Master's Thesis, Hacettepe University, Institute of Science, Ankara, 1994.
- [15] H. Qiang, Z. Youwei, On prefiltering and endpoint detection of speech signals. Fourth International Conference on Signal Processing (ICSP 98), 749 – 752, 1998.
- [16] G.S. Ying, C.D. Mitchell, L.H. Jamieson, Endpoint detection of isolated utterances based on a modified Teager energy measurement, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 93) 2(1993), 732–735.
- [17] L.T. Burrows, Speech processing with linear and neural network models. Unpublished thesis, Cambridge University, United Kingdom, 1996.
- [18] Ö. Aydın, Development of a voice recognition system by using artificial neural networks. Master's Thesis, Trakya University, Institute of Science, Edirne, 2005.
- [19] S. Oraintara, Y.J. Chen, T.Q. Nguyen, Integer fast Fourier transform. IEEE Transactions on Signal Processing 50(3) (2002), 607-618.
- [20] W.T. Cochran, J.W. Cooley, What is the Fast Fourier transform? Proceedings of the IEEE, 55(10), 1664-1667, 1967.
- [21] J.W. Cooley, J.W. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comput. 19 (1965), 297-301.
- [22] H. Kobayashi, T. Shimamura, A weighted autocorrelation method for pitch extraction of noisy speech, Proceed. IEEE Inter. Conference on Acoustics, Speech and Signal Processing 3, 1307–1310, 2000.
- [23] M. Dendrinos, G. Carayannis, Spectrum analysis using a new autocorrelation measure. Acoustics, Speech and Signal Processing 4(1998), 2468–2471.