QUICKPROP ALGORITHM ON VOICE RECOGNITION

Abdul Aziz[1], Ednawati Rainarli[2]

Informatics Engineering Study Program
[1,2] Faculty of Engineering and Computer Science, University Computer Indonesia
Jl. Dipati Ukur 114 Bandung
E-mail : aziz.rpl1@gmail.com[1], irene_edna@yahoo.com[2]

ABSTRACT

Speaker recognition is a speech recognition process based on a speaker. Angga setiawan and friends, have used backpropagation to do speaker recognition. The results of the research obtained an accuracy value of 83.99%. In addition Windra Swastika has also compared the use of quickprop with backpropagation for image recognition problems. The results showed that the quickprop method works better than the backpropagation method. Therefore, the classification algorithm that will be used in this study is the quickprop method. In this study, the voice data used came from 5 male speakers who said 5 words in Indonesian. The word used comes from research according to the Leipzig Corpora Collection which is stored in the form of .wav files. Before the recognition, voice data is first carried out by the feature extraction process with the MFCC method. The results of the feature extraction will be saved and then will be used as input for the Quickprop classification process. Based on K-Fold Cross Validation testing of the parameters used, the average accuracy is 92% and the selected word can be used in the recognition process.

Keywords : Speaker Recognition, Quickprop, Feature Extraction, MFCC.

1. INTRODUCTION

Speaker Recognition is a process of identification and verification to identify a speaker whose identity is identified based on the input data. To be able to recognize a person's voice data based on a spoken word, voice data will go through a feature extraction process, pattern matching, so that the information contained in the voice data can be used [1]. Artificial Neural Networks (ANN) is a method that can be used for the learning process as pattern matching in the case of voice recognition. There are many ANN methods, one of which is Learning Vector Quantization (LVQ), Backpropagation and Quickprop.

In previous research conducted by Angga Setiawan and friends, applying feature extraction with the Mel Frequency Cepstrum Coefficients (MFCC) method and pattern matching through ANN with LVQ learning methods the results obtained an average percentage of successful recognition of voice data by 83.99% [2]. In another study conducted by Yulia Nur Utami, et al. applying MFCC as feature extraction and using Backpropagation ANN as pattern matching, obtaining 80.23% accuracy results of success [3]. Besides Windra Swastika, in another case study comparing the Quickprop method with the Backpropagation method as a learning method in the case of image recognition, the Quickprop method can work better than the Backpropagation method [4]. From some of these studies Quickprop looks able to provide good results when used for pattern recognition, but the performance of Quickprop for the case of word speaker recognition in Indonesian is unknown.

Based on the description above, this research will apply the Quickprop learning method for voice recognition cases. In the feature extraction stage using MFCC as a voice feature extraction that is useful for the voice recognition process.

2. THEORETICAL BASIS

The speaker recognition system that will be built in this study consists of several processes, namely Feature Extraction, Normalization, then Quickprop (training and testing). In general, the description of the system to be built can be seen in Figure 1.

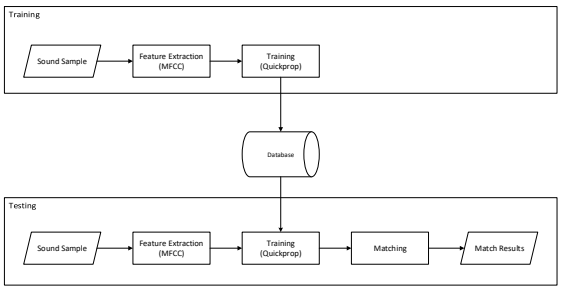

Figure 1. Overview of the system

The dataset (training data and test data) that is used as data enter the training process (training) and the testing process (testing) is the sound file from the recording of 5 speakers by mentioning the 5 most frequently used Indonesian words [5] namely (are,

can, belong, form, call) which will then be cut into one word that you want to use so that it lasts 1 second using the .wav format with a mono sound signal.

In the training process will be carried out the process of converting from analog signals to digital signals against selected sound signals which will then enter into feature extraction process. After obtaining a digital signal, the feature extraction method used is the Mel Frequency Cepstrum Coefficients (MFCC) method so that it can be used as training input materials using Quickprop.

In the testing process, the same process will be carried out as in the training process, but after the feature extraction process and normalization, an introduction process using Quickprop is carried out which has been trained in the training process. The results of this testing process will show the name of the speaker from the selected test data.

2.1 MFCC

Mel Frequency Cepstrum Coefficients (MFCC) is one method that is widely used in the field of speech technology, both speaker recognition and speech recognition [3]. The Mel Frequency Cepstrum Coefficients method consists of 8 process steps consisting of DC Removal, Pre-Emphasize Filtering, Frame Blocking, Windowing, Fast Fourier Transform (FFT), Mel-Frequency Wrapping, Discrete Cosine Transform (DCT) and Cepstral Liftering. Steps in the MFCC process can be seen in Figure 2.

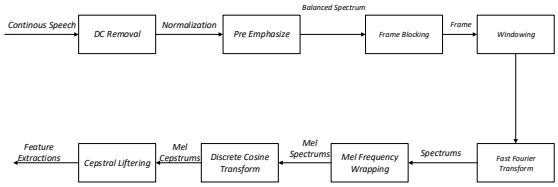

Figure 2. MFCC Diagram Block

2.1.1 DC-Removal

The DC removal process is performed to calculate the average of the sound sample data, and subtract the value of each sound sample by the mean value. The goal is to get normalization from input voice data [7].

$$dr_i = s_i - \bar{x}, 0 \leq i \leq N-1 \quad (1)$$

Where :

dr_i : a signal point resulting from the DC Removal process.

s_i : signal point-i.

$\bar{x}$: the average value of the signal point.

i : 0, 1, 2, …, N-1

N : signal length (many signal points).

2.1.2 Pre-Emphasize Filtering

After going through the DC-Removal process, the next step is Pre-Emphasize Filtering is one type of filter that is often used before a signal is processed further. This filter maintains high frequencies on a spectrum that are generally eliminated during the sound production process [3].

$$pf_i = dr_i - (dr_{i-1} \times \alpha) \quad (2)$$

Where :

pf_i : signal from the pre-emphasize filter to-i.

dr_i : signal before the pre-emphasis filter to-i.

α : *pre-emphasize filter*.

i : 0, 1, 2, …, N-1

N : signal length (number of signal points).

Then the equation to get the pre-emphasize filtering results can be represented in equation 3.

$$p_i = dr_i + pf_i \quad (3)$$

Where :

p_i : the new signal i-point.

dr_i : sample before pre-emphasize i (results of DC Removal).

pf_i : *pre-emphasize filter* in the i sample.

i : 0, 1, 2, …, N-1

2.1.3 Frame Blocking

In this step, the signal points are divided into frames. Each frame has the same size or length, and also each frame overlaps other frames. The overlap between frames that can be used is about ½ to 1/3 the length of the frame. In this study the overlap used is ½ of the frame length. The duration of the frame used in this study is 0.025 seconds, while the sample rate used is 16000 [3]. The process of calculating the number of frame blocking can be seen in Equation 4.

$$number\ of\ frame = ((NS - SP)/M) + 1 \quad (4)$$

Where :

NS : number of samples

SP : *Sample Points* in each frame (Sample Rate × frame length in seconds (s))

M : *SP*/2 (*Overlap*)

2.1.4 Windowing

This windowing process is carried out on each frame resulting from the frame blocking process. This is done to reduce spectral leakage or aliasing [3]. Aliasing is a new signal which has a different frequency than the original signal. This effect can occur because of the low number of sampling rates, or because of the frame blocking process which causes the signal to become discontinued. To reduce the possibility of spectral leakage, the results of the framing process must go through the window process [7]. The process for calculating can be seen in equation 5.

$$w_i = p_i \times fw_i \quad (5)$$

Where :

w_i : windowing result point signal value.

p_i : the results of the 1st pre-emphasize of the i frame.

fw_i : the 1st window function for the i frame.

NS : the number of sample points in each frame.

1 : *1st sample point in the frame (where values n = 0, 1, ..., NS-1).*

The most common window function used in speaker recognition applications is the Hamming Window. The following equation from the Hamming Window can be seen in equation 6.

$$fw_i = 0.54 - 0.46 \cos\frac{2\pi l}{SP-1} \quad (6)$$

Where :

fw_i : the hamming window function in the i frame.

SP : *Sample Point* (*frame length*).

l : 0, 1, …, SP-1

2.1.5 FFT

After going through the windowing stage, the next step is Fast Fourier Transform (FFT). Each frame is converted from the time domain to the frequency domain to get its frequency spectrum. This is done to facilitate computing and analysis [2]. The FFT equation can be seen as follows.

$$f_k = \sum_{i=0}^{SP-1} (w_i \cos\frac{2\pi ik}{SP}) - j\sum_{l=0}^{SP-1} \left(w_i \sin\frac{2\pi ik}{SP}\right), 0 \le k \le SP-1 \quad (7)$$

Where :

f_k : FFT spectrum to k.

N : the number of samples to be processed (N ∈ N).

l : 0, 1, 2, …, SP-1 (the number of samples in the frame).

k : 0, 1, 2, …, SP-1 (discrete frequency variable, is the result of FFT).

w_i : the value of the nth signal point (n windowing results on the frame).

j : imaginary number.

Where to get the results from FFT using equation 8.

$$\left|fft_k = [R^2 + I^2]^{\frac{1}{2}}\right| \quad (8)$$

Where :

R : real number (the calculation results $w_i \cos\frac{2\pi lk}{N}$).

I : imaginary number (the calculation results $j(w_i \sin\frac{2\pi lk}{N})$).

The results of the FFT will then be referred to as the FFT magnitude which will be mapped into the mel scale.

2.1.6 Mel-Frequency Wrapping

Magnitude that has been obtained from the FFT process will then enter the Mel-Frequency Wrapping stage. Mel Frequency Wrapping is generally done using a filter bank. Filterbank is one form of filter that is carried out with the aim to determine the energy size of a particular frequency band in the sound signal [5]. The equation used to get the results of the Mel-Frequency Wrapping can be seen as follows.

$$MF_m = \sum_{k=0}^{SP-1} fft_k \times H_{mk} \quad (9)$$

Where :

MF_m : the results of the Mel Frequency Wrapping to m.

fft_k : FFT results to k.

SP : the number of FFT results.

m : 0, 1, 2, …, *number of Mel Filterbank*

To find the mel filter bank coefficient can be made by making a triangular filter bank, using the following equation [2].

$$mel(f) = 2595 \times \log_{10}(1 + \frac{f}{700}) \quad (10)$$

Where :

mel(f) : *Mel Scale frequency*.

f : frequency used.

And the inverse mel scale equation can be seen in equation 11.

$$mel^{-1}(f) = 700 \times (10^{\frac{f}{2595}} - 1) \quad (11)$$

Where :

$mel^{-1}(f)$: inverse Mel Scale frequency.

f : frequency used.

Then for the filter bank boundary used can be seen in equation 12 below

$$fb_m = \left(\frac{SP}{FS}\right) \times mel^{-1}\left(mel(Nlow) + m\frac{mel(Nhigh) - mel(Nlow)}{Nfbank}\right) \quad (12)$$

Where :

fb_m : the result of the m-boundary filter boundary point.

SP : *Sample Point* (number of sample points).

FS : sampling frequency.

m : number of triangular filter banks.

Then after obtaining the value of the filter bank boundary, the following rules are carried out in equation 13 [8].

$$H_{mk} = \begin{cases} 0, for\ k < fb_m \\ \frac{k - fb_{m-1}}{fb_m - fb_{m-1}}, for\ fb_{m-1} \le k < fb_m \\ \frac{fb_{m+1} - k}{fb_{m+1} - fb_{m-1}}, for\ fb_m \le k < fb_{m+1} \\ 0, for\ k > fb_{m+1} \end{cases} \quad (13)$$

Where :

fb_m : the results of the boundary point filter.

H_{mk} : coefficient of mel filter bank.

2.1.7 DCT

Discrete Cosine Transform is the last step of the main process of MFCC feature extraction. The basic

concept of DCT is to declassify cepstrums to produce a good representation of local spectral properties. Basically the concept of DCT is the same as inverse fourier transform. But the results of DCT approach PCA (Principal Component Analysis). PCA is a classical statistical method that is widely used in data analysis and compression. This is why DCT often replaces inverse fourier transform in the MFCC feature extraction process [8]. Here is equation 14 to calculate DCT.

$$C_{koef} = \sqrt{\frac{2}{Nfbank}} \sum_{m=0}^{Nfbank-1} (\log MF_m) \cos\left[koef\left(\frac{2m-1}{2}\right)\frac{\pi}{Nkoef}\right] \quad (14)$$

Where :

MF_m : the results of the Mel Frequency Wrapping to m.

M : 0, 1, 2, …, SP-1 (the index results of the m filterbank).

Nfbank : number of Mel Filterbank (in this study 40).

Nkoef : expected number of coefficients (in this study 13).

The zero coefficient of the DCT will generally be eliminated, even though it actually indicates the energy of the signal frame. This is done because, based on studies that have been conducted, this zero coefficient is not reliable against speaker recognition [6].

2.1.8 Cepstral Liftering

The result of the DCT function is cepstrum which is actually the final result of the MFCC process. But to minimize the sensitivity of the MFCC coefficient that has been obtained, the cepstrum produced from DCT will be processed again in the cepstral liftering block. Cepstral liftering smoothes the resulting spectrum of the main processor so that it can be used better for pattern matching [7]. The equation 15 to calculate cepstral liftering is as follows.

$$Cepstral_n = \left\{C \times 1 + \frac{L}{2}\sin\left(\frac{n\pi}{L}\right)\right\} \quad (15)$$

Where :

L : number of cepstral coefficients.

n : index of cepstral coefficients.

C : dct results.

2.2 Nomalization

The results of the feature extraction process that has been carried out must go through the normalization process so that it can be processed by Quickprop because the training and testing process will be easier to do with discrete values compared to continuous values. Normalization is the scaling of values that enter a certain range. This is done so that the input values and output targets correspond to the range of activation functions used in the network. If the activation function used is a binary sigmoid, then the normalization equation 16 that can be used is [9].

$$normal_i = \frac{0.8(Cepstral_n - a)}{b-a} + 0.1 \quad (16)$$

Where :

$normal_i$: the results of the normalization of the i.

$Cepstral_n$: the nth Cepstral Liftering result.

a : minimum value on the frame.

b : maximum value on the frame.

2.3 Quickprop

Quickprop or Quickpropagation is one method of learning with ANN (Artificial Neural Networks). Quickprop comes from Backpropagation but uses different ways of updating synaptic weights. Quickprop is widely used in many cases, one of which is pattern recognition. Among the learning methods with ANN, QuickProp includes learning methods that are quite fast in learning and produce fairly high accuracy in recognition [4]. The following is the architecture of the quickprop network in Figure 3.

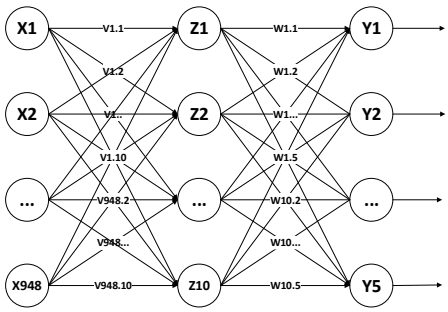

Figure 3. Quickprop architecture

2.3.1 Quickprop Training

Procedurally the steps in the Quickprop network training algorithm can be described as follows :

Step-1: Initialize all weights in the hidden and output layers, then set the activation function used for each layer.

Step-2: Calculate the output obtained from the neurons in the hidden layer.

$$v_j(p) = \sum_{i=1}^{r} x_i(p) \times w_{ij}(p) \quad (17)$$

Where :

v : output on the current layer.

j : neurons in the output layer.

x : input on the current layer.

r : the number of inputs to the neurons in the current layer.

w : weight

Step-3: Then activate the output in step 2 by using the sigmoid activation function

$$y_j(p) = \frac{1}{1+e^{-v_j(p)}} \quad (18)$$

Step-4: Perform the same calculation as in steps 2 and 3 to get the output value in the output layer.

Step-5: Calculate the difference between the output of step 3 and the desired target using the cost function as in equation 19.

$$E = \frac{1}{2}\sum_j^N (d_j - o_j)^2 \quad (19)$$

Where :

d_j : expected output

o_j : actual output.

Step-6: Change the weight or update the weights using equation 20 as follows.

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij} + \frac{\frac{\partial E}{\partial w_{ij}}(p)}{\frac{\partial E}{\partial w_{ij}}(p-1) - \frac{\partial E}{\partial w_{ij}}(p)} \Delta w_{ij}(p-1) \quad (20)$$

Where :

$w_{ij}(p+1)$: Changes in new weight.

$\Delta w_{ij}(p-1)$: delta weights on the previous epoch.

$\frac{\partial E}{\partial w_{ij}}(p)$: Derivative error.

$\frac{\partial E}{\partial w_{ij}}(p-1)$: Derivative error on the previous epoch.

Step-7: the training process will stop if the Error value is smaller with the specified value.

2.3.1 Quickprop Testing

Procedurally the steps in the Quickprop network testing algorithm are almost the same as the steps in training, but the initialization step is replaced by the results of the previous training process and the steps used are only steps 2 and 3.

3. RESEARCH METHOD

The research method used in this study is quantitative research. Called quantitative motede because the research data in the form of numbers and analysis using statistics. The sampling technique used is simple random sampling, said to be simple (simple) because the taking of sample members from the population is done randomly [10]. The flow of research that will be conducted in this study can be seen in Figure 4 below.

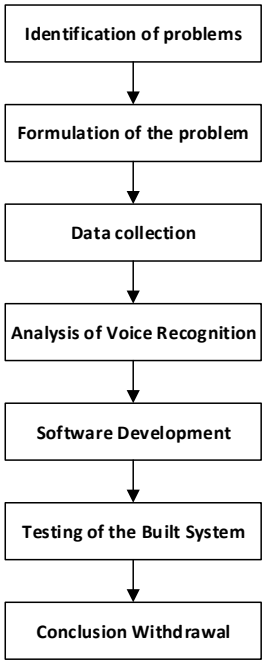

Figure 4. Research flow

4. RESULTS AND DISCUSSION

Testing is done using K-Folds Cross Validation, using the value K = 5 which shows the number of datasets, where the dataset is divided into 5 parts namely K1, K2, K3, K4, K5. The number of datasets used is 500 data. Each dataset (K) is 100 data.

In this test the optimal parameter values used are hidden layer = 10, learning rate = 0.3, error = 0.01 and epoch = 500. Here are some experiments from different parameters that are carried out in the training process and testing with the same dataset of 500 can be seen in table 1.

Table 1. Parameter combination

No	Learning Rate	Accuracy (%)
1	0.3	98.4
2	0.4	82.8
3	0.5	78.6

4.1 Test Result

After testing using 5-fold cross validation on the quickprop algorithm, the test is done 5 times, so we can recap the test results which can be seen in table 2.

Table 2. K-Fold Cross Validation Testing Results

Testing	True	False	Accuracy(%)
1	92	8	92
2	95	5	95
3	93	7	43

4	92	8	92
5	88	12	88
Average accuracy (%)			92

5. CONCLUSION AND SUGGESTION

5.1 Conclusion

Based on the results of the research, analysis and design, implementation and up to the testing stage, it can be concluded that the selected Indonesian word parameters have been able to be used for speaker recognition process on speaker recognition using the Quickprop method as a classification algorithm by utilizing feature extraction results MFCC. In the testing process with the optimal parameters used are hidden layer = 10, learning rate = 0.3, error = 0.01, maximum epoch = 500 has obtained the highest accuracy value of 95% while the lowest accuracy value is 88% and the average number of accuracy values is 92%. The accuracy value obtained will be different because it is influenced by training parameters, training data and test data used.

5.2 Suggestion

Based on the results of the research and testing that has been done, this research can still be developed in accordance with the needs of users. The suggestions for future research are :

1. Add filters that can reduce noise or interference and reduce sound data without reducing important information that will be carried out feature extraction process so as to improve accuracy and speed up the extraction process.
2. Add variations to the number of parameters that will be used such as the number of hidden layers, learning rate, errors that will affect the test results.
3. Another thing that needs to be added is that the system can recognize the name of the speaker and the word he said.

REFERENCES

[1] J. P. Campbell, "Speaker recognition: A tutorial," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1437-1462, 1997.

[2] A. Setiawan, A. Hidayatno, R. R. Isnanto, "Aplikasi Pengenalan Ucapan dengan Ekstraksi Mel-Frequency Cepstrum Coefficients (MFCC) Melalui Jaringan Syaraf Tiruan (JST) Learning Vector Quantization (LVQ) untuk Mengoperasikan Kursor Komputer," *TRANSMISI*, vol. 13, no. 3, pp. 82-86, 2012.

[3] Y. N. Utami, R. Rumani, N. Anbaranti, "Perancangan Speaker Recognition pada Sistem Kendali Lampu Berbasis Mikrokontroler," *e-Proceeding of Engineering*, vol. 2, no. 2, pp. 3332-3346, 2015.

[4] W. Swastika, "Quickprop Method to Speed up Learning Process of Artificial Neural Network in Money's Nominal Value Recognition Case," *AIP Publishing*, vol. 1825, no. 1, pp. 1-6, 2017.

[5] L. University, "Corpora and Language Statistics," Deutscher Wortschatz, 1998 - 2018. [Online]. Available: http://cls.corpora.uni-leipzig.de/en/ind_mixed_2013. [Accessed 3 Maret 2018].

[6] A. D. Andriana, "Perangkat Lunak Untuk Membuka Aplikasi Pada Komputer Dengan Perintah Suara Menggunakan Metode Mel Frequency Cepstrum Coefficients," *Jurnal Ilmiah Komputer dan Informatika (KOMPUTA)*, vol. 2 No. 1, 2013.

[7] D. Putra, A. Resmawan, "Verifikasi Biometrika Suara Menggunakan Metode MFCC dan DTW," *Lontar Komputer*, vol. 2 No. 1, pp. 8-21, 2011.

[8] F. Y. Leu, G. L. Lin, "An MFCC-based speaker identification system," *International Conference on Advanced Information AINA*, pp. 1055-1062, 2017.

[9] Z. Ramadhan, S. N. Endah, "Perintah Suara Berbahasa Indonesia untuk Membuka dan Menutup Aplikasi dalam Sistem Operasi Windows Menggunakan Metode Mel Frequency Cepstrum Coefficient dan Metode Backpropagation," *Seminar Nasional Ilmu Komputer*, pp. 33-41, 2016.

[10] P. D. Sugiyono, Metode Penelitian Pendidikan (Pendekatan Kuantitatif, Kualitatif, dan R&D), Bandung: Alfabeta, 2013.