International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2015): 6.391

Implementation of CORDIC based SVM for Speaker Verification System

Pavithra R [1], Saritha N. R.[2]

[1]M.Tech Student, Department of ECE, Sree Buddha College of Engineering, Pathanathitta, Kerala, India

[2]Professor/ECE, Department of ECE, Sree Buddha College of Engineering, Pathanathitta, Kerala, India

Abstract: ***This brief presents the implementation of a support vector machine (SVM) for speaker verification system. The proposed system comprises of a Gaussian kernel unit and a scaling unit . The proposed system can be used inside a speaker verification system for the purpose of classification of a speaker to be true or an imposer. A Gaussian kernel processing elements (GK-PEs) and an exponential processing elements are included in the support vector machine. The Gaussian kernel processing element is responsible for evaluating the kernel value of the test vector and the supporting vector. The Gaussian kernel processing element is designed to process 4 supporting vectors simultaneously. The GK-PE is designed in a pipeline fashion so as to perform 2-norm and exponential operations .The SVM makes use of an enhanced CORDIC architecture in order to calculate the exponential value. . The scaling unit is designed to perform SVM decision value evaluation. The proposed system can be used inside a speaker verification system along with a speaker feature extraction (SFE) module and a decision module. The SFE module is responsible for performing autocorrelation analysis, linear predictive coefficient (LPC) extraction, and LPC-to-cepstrum conversion. The decision module then accumulates the frame scores that are generated by all of the test frames, and is then compared with a threshold to see if the test utterance is spoken by the claimed speaker***

Keywords: Support Vector Machine (SVM), Support Feature Extraction Module(SFE), Linear Predictive Cepstral Coefficient(LPCC).

1. Introduction

The term Biometrics derived from the Greek words "bio" meaning life and "metric" meaning to measure which is related to the use of unique human characteristics to identify an individual. Based on the biometric features extracted from the human beings, pattern recognition decision is made by the biometric system. These Biometric identifiers are the distinctive, measurable characteristics which are used to label and describe individuals. The categorization of Biometric identifiers can be done as physiological characteristics versus behavioral characteristics. Physiological characteristics are those which are the shape of the body for Eg: fingerprint, palm veins, face recognition, retina and odour/scent. Behavioral characteristics on the other hand are related to the pattern of behavior of a person, including typing rhythm , voice, gait.

A speaker recognition system can be divided in to two types: a speaker identification system or a speaker verification system. In a speaker identification system , an unknown speaker is being identified as one of the speakers in the database where as for a speaker verification system a person's identity is validated[1].

Various studies on speaker recognition has been extensively done in the last decades . The two essential issues in a speaker verification system are feature extraction and classifier design. Cepstral coefficients are the most frequently adopted cepstral coefficient. The extraction of cepstral coefficients can be performed by two dominant approaches [2]. One approach is based on linear predictive analysis which is a parametric approach, this is developed to match closely the resonant structure of the human vocal tract which produces the corresponding speech sound. The obtained linear predictive coefficients (LPCs) can be converted to LPC cepstral coefficients (LPCCs). The other one is based on modeling the human auditory perception system which is a nonparametric method. Mel frequency cepstral coefficients (MFCCs) are used for this purpose .

2. Support Vector Machine

A. SVM Theory

The support vector machine can be considered as a powerful tool for pattern classification. The inputs are mapped in to a high dimensional space and classes with largest margin are separated. SVMs perform a nonlinear mapping from an input space to an SVM feature space. The SVM have proven to be a new and effective method for speaker recognition because it discriminates between various classes and non linear decision boundaries can be effectively trained. Since the speaker verification is a 2 class problem, SVM can be effectively used in it. The design of the inner product, the kernel which is induced by high dimensional mapping is one of the critical aspect of using SVM successfully. The SVM finds application in speaker and speech recognition. In the case of speaker verification system, a decision in the hypothesis that whether the speech is produced by a speaker or whether it has been produced by someone else in the population has to be made. The kernel compares the various features of the test vectors and produces the measure of similarity.

In the case of speaker verification system, a decision in the hypothesis that whether the speech is produced by a speaker or whether it has been produced by someone else in the population has to be made. The kernel compares the various features of the test vectors and produces the measure of similarity. SVM works based on the idea of structural risk minimization induction principle which aims at minimizing a bound on the generalization error, rather than reducing the mean square error. For the optimal hyper plane $w \cdot x + b = 0$, $w \in RN$ and $b \in R$, for classifying an unknown point x , the SVM decision function is defined as

$$f(x) = wx + b = \sum_{i=1}^{Ns} \alpha i\ ti\ yi \cdot x + b \quad (1)$$

where NS is the support vector number; yi is the i th support vector; αi is the corresponding Lagrange multiplier; and $ti \in \{-1, +1\}$ describes, which class yi belongs to.

Almost in all cases , searching suitable hyper plane in input space is too restrictive to be of practical use. The solution to overcome this situation is by mapping the input space into a higher dimension feature space and then searching the optimal hyper plane in this higher dimensional feature space. Let $z = \phi(x)$ denote the corresponding feature space vector with a mapping ϕ from R^N to a feature space Z. It is not necessary to know about ϕ. a function $k(\cdot, \cdot)$ called kernel is provided, which uses the points in input space to compute the dot product in feature space Z, that is

$$\phi(xi) \cdot \phi(yj) = k(xi, yj). \quad (2)$$

Finally, the SVM decision function becomes

$$f(x) = \sum_{i=1}^{Ns} \alpha i\ ti\ yi \cdot x + b \quad (3)$$

Those functions which satisfies the Mercer's theorem can be used as kernels. Linear kernel, polynomial kernel, and Gaussian kernel are typical kernels.

B. SVM Module

The SVM module is used to evaluate the SVM decision function for a test point, which is a 10-D LPCC vector. The kernel evaluations with all of the support vectors that were obtained by SVM hyper plane during training must be carried out at an unknown test point **x**. To obtain satisfactory classification performance, a large number of support vectors must be used0. Thus completion of kernel evaluation is a computationally intensive task. In order to accomplish real time speaker verification and low power dissipation an efficient SVM architecture is essential . The proposed system consists of a Gaussian kernel which is based on

$$k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \quad (4)$$

where **x** and **y** are the two vectors whose kernel value is evaluated, and σ is the standard deviation.

In Support Vector Machine, each frame forms a test vector and these test vectors are evaluated with the supporting vectors using kernel evaluation. Fig. 1 shows the block diagram of the proposed SVM module. The SVM module mainly comprises of two computational units, a scaling unit and a Gaussian kernel unit.

c. Gaussian kernel unit.

To perform the Gaussian kernel evaluations of the test vectors and the supporting vectors a Gaussian kernel unit is used. Figure 2 Shows the block diagram of Gaussian kernel unit. The Gaussian kernel unit consists of four Gaussian kernel processing element(GK-PEs) and five serial to parallel unit(SPUs) [4]. The Gaussian kernel processing element performs the kernel evaluation of each test vectors with one of the supporting vectors. There are four Gaussian kernel units so as to perform the Gaussian kernel evaluation with 4 supporting vectors. SPU is used to transform serially input data to data that are input parallel for the GK-PE. Out of the 5 SPU , one SPU receives the test vector while the other four receives the corresponding supporting vectors

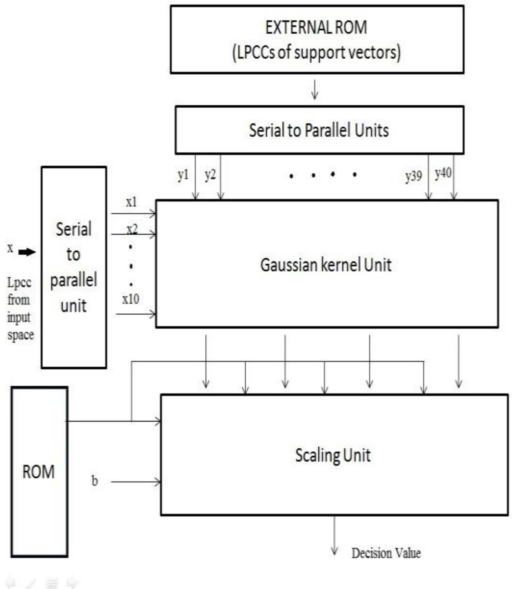

Figure 1: Block Diagram of SVM module

Each GK-PE incorporates a normal PE (Norm PE) and an exponential PE (Exp PE). The architecture of Gaussian kernel unit and exponential processing elements are described below

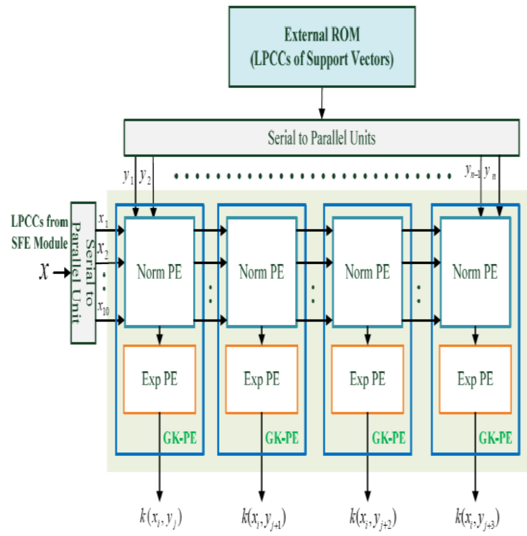

Figure 2: Architecture of Gaussian kernel unit

Normal processing element: The Norm PE is responsible for calculating $-\|x-y\|^2/(2\sigma)^2$, where $x = (x_1, x_2, \ldots, x_{10})$ denotes an LPCC test vector, and $y = (y_1, y_2, \ldots, y_{10})$ represents an LPCC support vector. The Norm PE first computes the norm square, $\|x - y\|^2 = (x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_{10} - y_{10})^2$, with an adder tree to sum the square values of difference in each dimension. The standard deviation is taken as one. Hence, only the compliment of a 2 and a right-shift operation are required to obtain $-\|x - y\|^2/2\sigma^2$ from $\|x - y\|^2$.

Exponential processing element: Out of the different hardware implementation options in exponential operation CORDIC method is used [6]. The CORDIC method based exponential processing element is used in the implementation of Exp PE . By using CORDIC based architecture the area of the hardware can be reduced as it uses only adders and shifters. By using the unfolding technique high speed can be achieved. The value of constant K is pre-calculated and is stored in the memory. Fig shows the detailed CORDIC circuit which is capable of angle updating process [7]. The value of $\tanh^{-1}(2^{-i})$ is also pre-calculated and stored in ROM.A counter is being used in order to generate the value of P . This value is then used as a selection line for the MUX.

Any number $z \in R+$ can be expressed as

$$z = z1 + p \ln 2 \qquad (5)$$

where $p \in Z+$ and $z1 \in [-1, 1]$. Where p is an integer , which equals fix (z/ ln2) Accordingly, performing the exponential operation on Z yields

$$e^{z} = e^{z1+p \ln 2} = e^{z1} \cdot e^{p \ln 2} = 2^{p} \cdot e^{z1} . \qquad (6)$$

Based on (6), e^{z} is replaced by e^{z1} with $z1 \in [-1, 1]$, and then a right shifting by P bits is conducted [5]. Accordingly, the architecture of Exp PE as shown in Fig. 4.

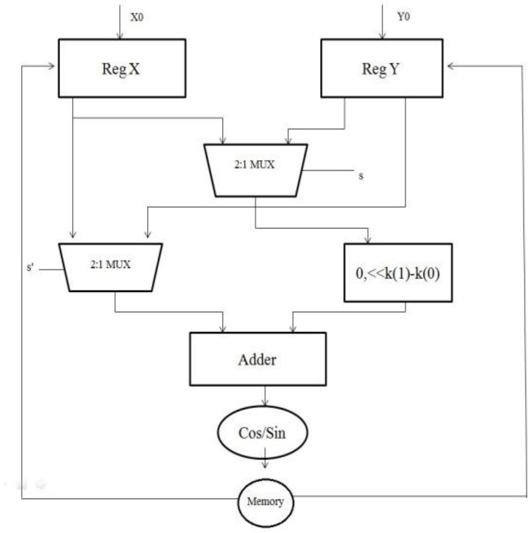

Figure 3: Architecture of CORDIC.

The scaling unit first multiply each k(xi, x) which are the resultant output of Gaussian kernel unit by scaling unit αi yi , which was obtained during training phase and is stored in ROM. Since the Gaussian kernel unit outputs 4 values in parallel four multipliers are required to perform the scaling multiplication on the 4 kernel values. The output of the multiplier is given to a two stage multiplier so as to produce the sum. Until the kernel values of the supporting vectors are processed an accumulator is used to accumulate the four scaling multiplication results. The stored bias constant is then added to the result of the scaling multiplication to generate SVM decision value.

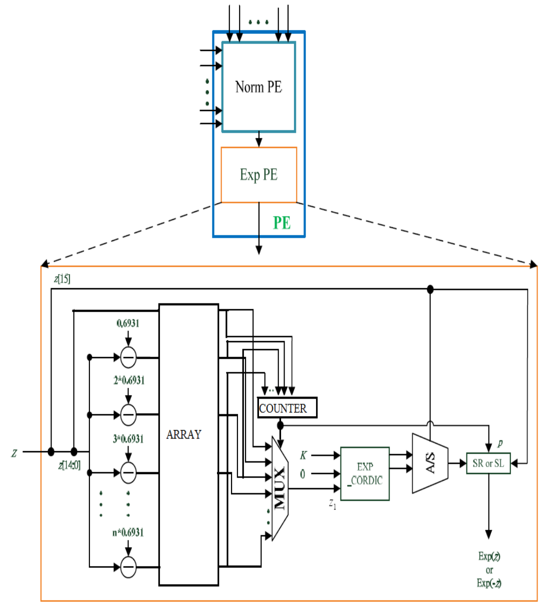

Figure 4: Architecture of Exp PE.

D. Scaling Unit

The scaling unit first multiply each k(xi, x) which are the resultant output of Gaussian kernel unit by scaling unit αi yi , which was obtained during training phase and is stored in ROM. Since the Gaussian kernel unit outputs 4 values in parallel four multipliers are required to perform the scaling multiplication on the 4 kernel values. The output of the multiplier is given to a two stage multiplier so as to produce the sum. Until the kernel values of the supporting vectors are processed an accumulator is used to accumulate the four scaling multiplication results. The stored bias constant is then added to the result of the scaling multiplication to generate SVM decision value.

3. Experimental Results

The model is simulated using Xilinx ISE Design Suite 13.2. Performance evaluation is done based on the throughput and area. Fig. 5 shows the simulation results Serial to Parallel Units , Fig. 6 shows the simulation results for Exponential PE and Fig. 7 shows the simulation result of Scaling unit.

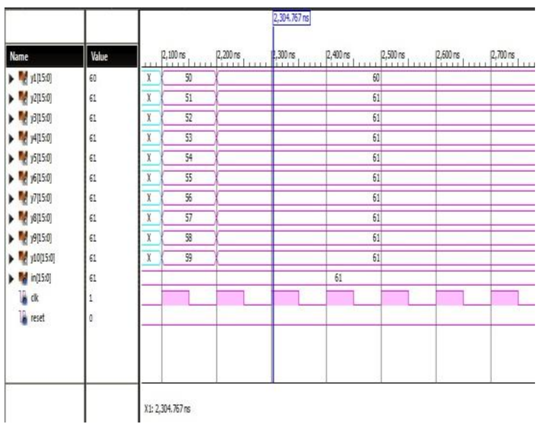

Figure 5: Simulation result of SPU's

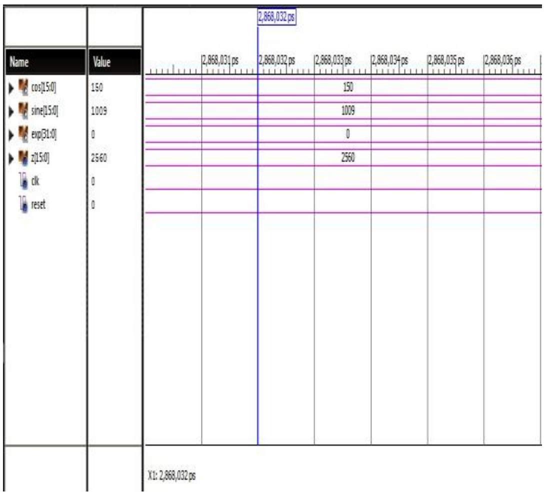

Figure 6: Simulation result of Exp PE

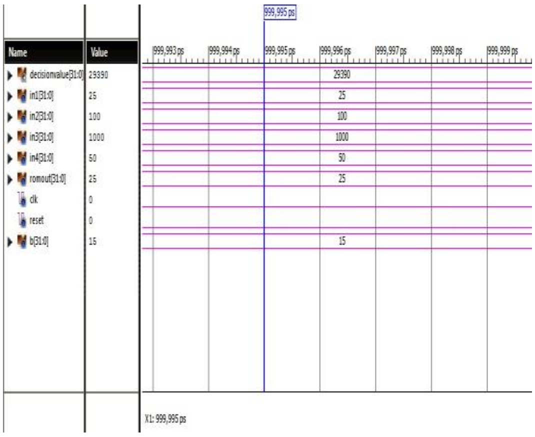

Figure 7: Simulation result of Scaling Unit

4. Conclusion

In this paper a Support vector Machine for supporting Speaker Verification System is presented. The architecture of SVM module consists of Serial to Parallel units , Gaussian kernel Units, Scaling units. The SVM module evaluates all the required kernels values, performs various scaling multiplications and completes the remaining operation of decision value evaluation. With efficient CORDIC architecture area of the hardware can be decreased.

5. Acknowledgment

This journey of self actualization would not have been fulfilled without the guidance and support of several individuals. My deep appreciation is extended to my Thesis Guide Prof. Saritha N.R., Department of Electronics and communication for his wisdom, guidance, encouraging, appreciation and design process. Authors wish to remark the great task carried out by the Xilinx user guide.

References

[1] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM supervectors for speaker verification, "*IEEE Signal Process. Lett.*, vol.13, no. 5, pp. 308–311, May 2006.

[2] B. H. Juang and T. H. Chen, "The past, present, and future of speech processing, " *IEEE Signal Process. Mag.*, vol. 15, no. 3, pp. 24–48, May 1998.

[3] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models, " *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 72–83, Jan. 1995.

[4] Jia-Ching Wang, Li-Xun Lian, Yan-Yu Lin, and Jia-Hao Zhao, "VLSI Design for SVM-Based Speaker Verification System" IEEE Trans. on very large scale integration (vlsi) systems, vol. 23, no. 7, July 2015.

[5] A. Bodabous, F. Ghozzi, M. Kharrat, and N. Masmoudi, "Implementation of hyperbolic functions using CORDIC algorithm, " in Proc. IEEE 16th Int. Conf. Microelectron.(ICM), Dec. 2004, pp. 738–741.

[6] V.Kantabutra, "On hardware for computing exponential and trigonometric functions, "IEEE Trans. Comput., vol. 45, no. 3, pp. 328–339, Mar. 1996.

[7] Pramod Kumar Meher and Sang Yoon Park, "cordic designs for fixed angle of rotation", IEEE Trans on very large scale integration (vlsi) systems, vol. 21, no. 2, February 2013.