

논문 2004-41SP-3-19

에스 브이 엠을 이용한 화자인증 알고리즘의 하드웨어 구현 연구

(A Hardware Implementation of Support Vector Machines for Speaker Verification System)

최 우 용*, 황 병 희**, 이 경 희*, 반 성 범*, 정 용 화***, 정 상 화**

(W.-Y. Choi, B.-H. Hwang, K. Lee, S. B. Pan, Y. Chung, and S.-H. Chung)

요 약

화자인증이 생체인식 방법 중의 하나로, 사람의 목소리를 이용하여 사용자를 인증하는 방법이다. 현재까지 가장 많이 사용되는 화자인증 알고리즘으로는 HMM(Hidden Markov Model)과 DTW(Dynamic Time Warping)를 들 수 있는데, 이들 알고리즘은 사용자의 등록 및 인증을 위해 많은 수의 특징벡터를 필요로 하므로 스마트카드와 같은 메모리가 제한된 시스템에는 적용하기 어려운 단점이 있다. 본 논문에서는 SVM(Support Vector Machine)을 이용함으로써 적은 양의 메모리와 적은 계산량으로 화자인증을 수행할 수 있는 방법을 제안하였으며, 이의 실시간 처리를 위해 하드웨어 구조를 제시하였다. 한국어 4연숫자 데이터베이스를 이용하여 제안한 알고리즘의 성능을 평가한 결과, 기존 알고리즘에 비해 약간의 에러율 증가가 있었으나 수행시간 및 모델크기에서는 상당한 감소를 나타내었다. SVM을 이용한 화자인증 알고리즘을 하드웨어로 구현한 결과, 소프트웨어로 구현한 경우에 비해서 훈련시간은 175분의 1, 인증시간에서는 6분의 1의 감소를 나타내었다.

Abstract

There is a growing interest in speaker verification, which verifies someone by his/her voices. There are many speaker verification algorithms such as HMM and DTW. However, it is impossible to apply these algorithms to memory limited applications because of large number of feature vectors to register or verify users. In this paper we introduces a speaker verification system using SVM, which needs a little memory usage and computation time. Also we proposed hardware architecture for SVM. Experiments were conducted with Korean database which consists of four-digit strings. Although the error rate of SVM is slightly higher than that of HMM, SVM required much less computation time and small model size.

Keywords : 화자인증, SVM, 하드웨어

I. 서 론

정보통신 기술이 급속도로 발전하고 인터넷의 이용이 확산됨에 따라 사용자 인증에 대한 관심이 높아지고 있다. 최근까지 사용자 인증 수단으로 사용되던 패스워

* 정희원, 한국전자통신연구원 정보보호연구단
(ETRI Information Security Research Division)

** 정희원, 부산대학교 컴퓨터공학과
(Dept. of Computer Engineering, Pusan National Univ.)

*** 정희원, 고려대학교 컴퓨터정보학과
(Dept. of Computer Information, Korea Univ.)

접수일자: 2003년7월21일, 수정완료일: 2004년3월8일

드나 PIN(Personal Identification Number) 등은 타인에게 노출되거나 잊어버리는 등의 문제점을 가지고 있어 이를 대체하거나 보완하기 위한 방법으로 개인의 고유한 생체정보를 이용한 사용자 인증 방법에 관한 연구가 진행되고 있다. 생체인식에 사용되는 생체정보로는 지문, 얼굴, 음성, 홍채, DNA 등이 있는데, 이러한 생체정보 중에서 사람의 음성을 이용하는 화자인식은 비교적 값이 싸고 손쉽게 구할 수 있는 마이크를 입력 장치로 사용하며, 다른 생체인증 방법에 비해서 거부감이 적다는 장점이 있다.

화자인증 시스템은 인증문장의 고정 여부에 따라서 문맥종속 시스템과 문맥독립 시스템으로 나눌 수 있다. 전통적으로 많이 사용되어온 화자인증 방법으로는 DTW(Dynamic Time Warping), HMM(Hidden Markov Model), VQ(Vector Quantization) 및 GMM(Gaussian Mixture Model) 등이 있는데, 이 중 DTW와 HMM은 문맥종속 시스템에, VQ와 GMM은 문맥독립 시스템에 주로 사용되는 알고리즘이다.^{[1]-[5]}

DTW는 두 음성패턴을 비교하는데 있어서, 음성구간별 발생속도에 따라 정렬함으로써 동일한 음성구간끼리의 비교가 가능한 방법이다. 따라서 발생속도의 차이에서 오는 패턴의 불일치를 효과적으로 보상해 줄 수 있다. HMM은 음성패턴을 분석하는데 있어서 가장 널리 알려진 통계적 방법으로, 관측치의 사후확률을 최대로 하는 파라미터를 찾는 방법이다. 그러나 이러한 파라미터를 직접 구할 수 있는 계산식은 알려져 있지 않고, EM(Expectation Maximization) 알고리즘과 같은 반복계산을 통하여 구할 수 있다.

본 논문에서는 최근 패턴 인식 분야에서 활발하게 연구가 진행되고 있는 SVM(Support Vector Machine)을 이용한 화자인증 알고리즘을 제안하였다. 제안한 알고리즘의 성능분석을 위하여 한국어 데이터베이스를 이용하였으며, 인식 성능과 모델크기, 수행시간 등을 DTW 및 HMM과 비교하였다. 또한 화자인증용 SVM 알고리즘의 하드웨어 구조를 설계하였고, 이를 VHDL(Very high speed integrated circuit Hardware Description Language)로 모델링하여 FPGA(Field Programmable Gate Array)에 탑재하여 개발한 시스템이 실시간으로 화자인증을 수행함을 확인하였다. 제안한 시스템의 인식 성능은 DTW나 HMM에 비해 약간의 인식을 감소를 나타냈지만, 모델의 크기 및 인증 시간에서는 매우 우수한 성능을 나타내었다. 현재 SVM을 이용한 다양

한 생체인식 알고리즘 연구가 활발히 진행되고 있으므로 앞으로 본 논문에서 개발한 SVM 인식 엔진으로 다중 생체인식 알고리즘 처리도 가능할 것이다.

본 논문의 구성은 II장에서 SVM을 이용한 화자인증 알고리즘을 설명하고, III장에서 한국어 데이터베이스를 이용하여 DTW, HMM 및 SVM의 성능을 비교하였다. IV장에서는 SVM을 이용한 화자인증 시스템의 하드웨어 구현에 대하여 설명하였으며, V장에서는 화자인증 하드웨어 시스템의 실험 결과에 대해 분석하고, 마지막으로 VI장에서 결론 및 향후 과제를 제시하였다.

II. SVM을 이용한 화자인증 알고리즘

화자인증 알고리즘은 인증하고자 하는 문자열의 고정여부에 따라 문맥종속 시스템과 문맥독립 시스템으로 나눌 수 있다. 일반적으로 DTW와 HMM은 문맥종속 시스템에, VQ와 GMM은 문맥독립 시스템에 많이 사용되는 알고리즘이다. 본 장에서는 SVM을 문맥종속 화자인증 시스템에 적용하는 방법에 대해서 설명한다.

SVM의 목적은 서로 다른 두개의 클래스를 분류하는 초평면(hyperplane)을 설계하는 것으로, structural risk minimization 기법에 그 기초를 두고 있다.^{[6][7]} 최근에 패턴인식 분야에서 활발하게 연구되고 있는 SVM 알고리즘을 간단히 설명하면 아래와 같다.

훈련데이터가 다음과 같이 주어졌다고 가정하자.

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N) \in \mathbb{R}^d \times \{\pm 1\} \quad (1)$$

여기서 \mathbf{x}_i 는 입력패턴이고, y_i 는 그 결과값이다. 만약 두 클래스가 선형적으로 분리가능하다면 식 (2)와 같은 초평면에 의해서 두 클래스를 구분할 수 있다.

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (2)$$

SVM은 이러한 초평면 중에서 초평면과 가장 가까운 데이터와의 거리를 최대로 하는 초평면을 찾는 것으로, 식 (4)와 같은 제약조건 하에서 식 (3)을 최소로 하는 가중치 벡터 \mathbf{w} 를 구하는 것과 같다.

$$\min \frac{1}{2} \mathbf{w}^T \mathbf{w} \quad (3)$$

$$y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1, i = 1, \dots, N \quad (4)$$

위 식을 Lagrange 승수를 이용하여 다시 쓰면 다음

과 같이 식 (6)의 제약조건 하에서 식 (5)를 최대로 하는 최적화 문제를 푸는 것과 같게 된다.

$$\max Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (5)$$

$$\sum_{i=1}^N \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0 \text{ for } i = 1, \dots, N \quad (6)$$

$Q(\alpha)$ 를 최대로 하는 α 를 α_o 라고 하면 최적 가중치 벡터 \mathbf{w}_o 는

$$\mathbf{w}_o = \sum_{i=1}^N \alpha_{o,i} y_i \mathbf{x}_i \quad (7)$$

가 되고, 최적 바이어스 b_o 는

$$b_o = 1 - \mathbf{w}_o^T \mathbf{x}^{(s)} \text{ for } y^{(s)} = 1 \quad (8)$$

를 이용하여 구할 수 있다. 여기서 $\mathbf{x}^{(s)}$ 는 Lagrange 승수가 0이 아닌 입력 패턴이다.

지금까지는 입력공간에서의 데이터들이 선형적으로 분리 가능한 경우에 대해서 살펴보았는데, 일반적으로는 데이터들이 선형적으로 분리 가능하지 않은 경우가 대부분이다. 이러한 경우에는 입력공간에서의 데이터를 보다 높은 차원의 공간(특징공간)으로의 비선형변환을 통하여 데이터를 선형적으로 분리가능하게 만들 수 있다(Cover's theorem on the separability of patterns). 입력공간에서 특징공간으로의 비선형 변환을 $\varphi(\mathbf{x})$ 라 두면 특징공간에서의 최적의 초평면은

$$\sum_{i=1}^N \alpha_i y_i \varphi^T(\mathbf{x}) \varphi(\mathbf{x}_i) = 0 \quad (9)$$

와 같이 구할 수 있다. 그러나 특징공간은 매우 높은 차원의 공간이므로 특징공간에서 데이터를 처리한다는 것은 현실적으로 불가능하다. 이러한 경우에는 내적(inner-product) 커널을 이용함으로써 특징공간에서 직접 연산을 수행하지 않고 입력공간에서 데이터를 처리할 수 있다. 커널 함수를

$$k(\mathbf{x}, \mathbf{x}_i) = \varphi^T(\mathbf{x}) \varphi(\mathbf{x}_i) \quad (10)$$

라 두면 식 (5)와 (6)은 다음과 같이 나타낼 수 있다.

$$\max Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \quad (11)$$

$$\sum_{i=1}^N \alpha_i y_i = 0 \text{ and } \alpha_i \geq 0 \text{ for } i = 1, \dots, N \quad (12)$$

위 식을 풀어서 정리하면

$$\mathbf{K}' \alpha' = \mathbf{y}' \quad (13)$$

가 되고, Gauss-Jordan 소거법을 사용하여 α' 를 구할 수 있다. 여기서,

$$\mathbf{K}' = \begin{bmatrix} 0 & y_1 & y_2 & \dots & y_N \\ 1 & k_{11} & k_{12} & \dots & k_{1N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & k_{M1} & k_{M2} & \dots & k_{MN} \end{bmatrix}, \alpha' = \begin{bmatrix} \lambda \\ \alpha_1 \\ \vdots \\ \alpha_N \end{bmatrix}, \mathbf{y}' = \begin{bmatrix} 0 \\ y_1 \\ \vdots \\ y_N \end{bmatrix} \quad (14)$$

이며, $k_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ 이고, 식 (14)의 λ 는 식 (12)의 제약조건에 대한 Lagrange 승수이다. 이렇게 구한 α_i 중에서 0이 아닌 α_i 에 해당하는 \mathbf{x}_i 를 SV(Support Vector)라고 부르고, 최적의 초평면은 SV와 그에 해당하는 Lagrange 승수로 표현되는데, 그 식은 다음과 같다.

$$\sum_{i=1}^N \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) = 0 \quad (15)$$

이후의 인증과정은 식 (15)의 좌변의 \mathbf{x} 에 테스트 벡터를 대입하여 그 값이 양이면 수락하고, 음이면 거절

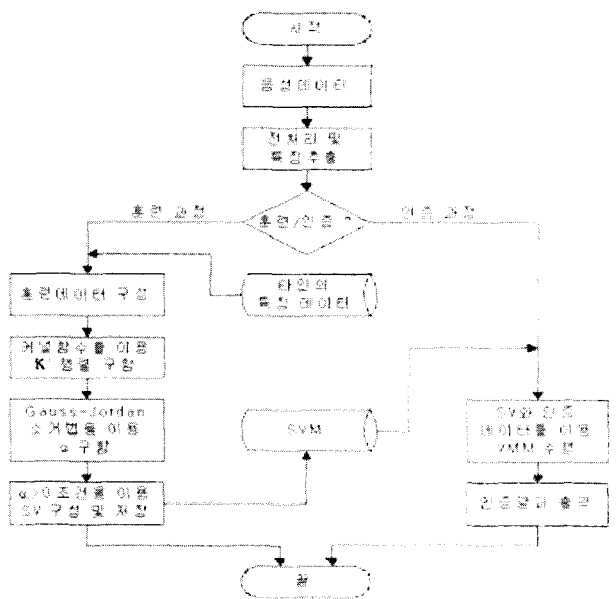


그림 1. SVM을 이용한 화자인증 알고리즘의 순서도
Fig. 1. Flowchart of speaker verification algorithm using SVM

하게 된다.

본 논문에서는 SVM에 사용되는 커널 함수로 다음과 같은 RBF(Radial-Basis Function) 커널을 사용하였다.

$$k(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|^2\right) \quad (16)$$

여기서 σ 는 커널들의 폭으로 선형적으로 결정된다. 그림 1에 훈련 과정과 테스트 과정으로 구성되는 SVM을 이용한 화자 인증 시스템의 순서도를 나타내었다.

훈련과정에서는 먼저 본인의 음성을 입력받아 전처리 과정을 거쳐 특징벡터를 추출한 후, 미리 저장된 타인의 화자 특징 데이터와 함께 훈련 데이터를 구성한다. 다음으로 커널 함수를 이용하여 식 (14)의 \mathbf{K}' 행렬을 구한 후, 생성된 \mathbf{K}' 행렬에 Gauss-Jordan 소거법을 적용하여 Lagrange 승수 α 를 구한다. 구해진 α_i 중에서 $\alpha_i > 0$ 조건을 만족하는 \mathbf{x}_i 를 이용하여 SV 테이블을 구성하는데, SV 테이블에는 SV와 목표값, 그리고 Lagrange 승수를 저장해 둔다. 그리고 인증 과정에서는 음성을 입력받아서 훈련과정에서와 같이 전처리 과정을 거쳐 특징벡터를 추출하고 추출된 데이터와 훈련 과정에서 저장해 둔 SV와의 VMM(Vector Matrix Multiplication) 연산을 통하여 사용자를 인증하고 그 결과를 출력한다.^[8]

III. 화자인증 알고리즘 인식 성능 분석

본 장에서는 DTW, HMM 및 SVM을 이용한 화자인증 시스템의 성능을 비교하였다. 실험에 사용된 데이터 베이스는 사무실 환경에서 녹음된 음성으로 4연숫자, 고립단어 및 단문으로 구성되어 있다. 훈련용 데이터는 각 문장을 27명(남성 17명, 여성 10명)이 각각 6회씩 발성한 음성데이터를 사용하였고, 인증용 데이터는 동일인이 각 문장을 각각 6회씩 발성한 음성데이터를 사용하였다. 음성특징으로는 12차 MFCC(Mel-Frequency Cepstral Coefficient)와 delta MFCC로 구성된 24차원 벡터를 사용하였으며, 20msec 윈도우를 10msec씩 이동하면서 추출하였다. 모든 음성데이터는 16kHz로 샘플링 되었으며 16bit로 양자화 하였다. SVM 실험을 위한 특징벡터로는 모든 프레임의 시간평균을 사용하였다. 따라서 하나의 음성을 표현하는데 있어서 DTW나 HMM에 비해서 매우 적은 수의 특징벡터로 표현이 가능하

다. 예를 들어, 1.5초 길이의 음성을 표현하기 위해서 DTW나 HMM의 경우 3,600개의 소수 값이 필요하지만 SVM은 단지 24개의 값만으로 표현할 수 있다. 이는 DTW나 HMM에 비해 150분의 1로 줄어든 값이며 음성의 길이가 길어지면 그 차이는 더 커지게 된다. 따라서 SVM은 기존의 알고리즘에 비해 모델의 크기나 수행시간 면에서 탁월한 이점을 가질 수 있다.

표 1에서 RBF 커널의 파라메타(σ)에 따른 SVM의 인식성능을 FAR(False Acceptance Rate)와 FRR(False Rejection Rate)로 나타내었다. 파라메타 값이 커질수록 FAR은 증가하고 FRR은 감소하며, 파라메타 값이 2.0 일 때 FAR과 FRR이 비슷한 값을 나타내었다.

표 2는 DTW와 HMM의 에러율을 나타낸 표이다. HMM이 DTW보다 조금 낮은 에러율을 나타내었고,

표 1. RBF 커널의 파라메타에 따른 SVM의 성능 비교

Table 1. Performances of SVM with various parameters

파라메타 값	FAR (%)	FRR (%)
1.0	2.8	13.0
1.5	3.8	10.2
2.0	6.3	6.4
2.5	9.7	4.3
3.0	13.9	2.9

표 2. DTW 및 HMM의 인식 성능

Table 2. Equal error rates of DTW and HMM

알고리즘	EER (%)
DTW	5.1
HMM	4.7

표 3. SVM, DTW 및 HMM의 수행시간

Table 3. Execution times of SVM, DTW and HMM

알고리즘	수행시간 (msec)	
	훈련시간	인증시간
SVM	2,455	2.0
DTW	278	28.3
HMM	12,515	17.8

표 4. SVM, DTW 및 HMM의 모델크기

Table 4. Model sizes of SVM, DTW and HMM

알고리즘	모델크기 (KBytes)
SVM	0.8
DTW	8.6
HMM	2.3

DTW나 HMM 모두 SVM보다 조금 더 높은 성능을 나타내었다. 그러나 본 논문에서 사용한 음성의 평균길이는 1.75초이므로 SVM이 처리해야할 데이터는 DTW와 HMM의 175분의 1에 불과하다.

SVM, DTW 및 HMM의 수행시간과 모델크기를 표 3과 표 4에 각각 나타내었는데 표 3을 보면 SVM의 훈련시간은 DTW의 약 9배, HMM의 5분의 1 수준이고, 인증시간은 DTW의 14분의 1, HMM의 9분의 1로 나타났다. 화자인증 시스템이 일반적으로 훈련 과정은 오프라인에서 이루어지고 인증과정은 실시간으로 온라인상에서 처리되므로 SVM을 이용함으로써 실생활의 적용에 가장 적합한 화자인증 시스템을 구현할 수 있다. SVM의 모델크기는 0.8KByte로 DTW의 11분의 1, HMM의 3분의 1 정도의 크기를 가진다. 따라서 SVM을 이용한 화자인증 시스템은 메모리가 제한된 시스템에서 화자인증을 구현함에 있어서 다른 알고리즘에 비해 탁월한 이점을 가질 수 있다.

IV. 화자인증 시스템의 하드웨어 구성

SVM을 이용한 화자인증 알고리즘의 하드웨어 구조는 그림 2에 나타낸 것과 같이 크게 훈련 VMM 제어 모듈, 인증 VMM 제어 모듈, 커널함수 계산 모듈, Gauss Jordan 제어 모듈, SV 테이블 구성 모듈의 다섯 가지 논리적 단위로 구성되어 있다. 훈련 VMM 제어 모듈은 본인 음성 데이터와 미리 등록된 타인 음성 데이터를 포함하는 훈련데이터를 커널함수 계산 모듈에 입력으로 준다. 커널함수 계산 모듈은 훈련 VMM 제어 모듈로부터 입력받은 훈련 데이터를 이용하여 소수

(float) 계산기를 통해 식 (16)의 커널 값들을 계산한다. Gauss Jordan 제어 모듈은 Gauss Jordan 알고리즘을 하드웨어로 구현한 모듈로서, 최대값 찾기 모듈, 열 교환 모듈, 최대값 열 계산 모듈, 행렬 계산 모듈로 구성되어 각각의 모듈을 순서대로 반복하여 수행함으로써 식 (13)의 Lagrange 승수(α_i)를 계산하여 그 결과를 훈련 VMM 제어 모듈로 반환한다. SV 테이블 구성 모듈은 구해진 Lagrange 승수가 양인 값을 가지는 벡터를 이용하여 SV 테이블을 구성하는 역할을 담당한다. 인증 VMM 제어 모듈에서는 구해진 테이블과 테스트 벡터를 사용하여 인증을 위한 계산이 수행되도록 관리하는 역할을 담당한다.

1. 훈련과정

훈련 과정은 사용자로부터 입력받은 등록 음성 데이터와 미리 저장된 타인의 음성 데이터를 이용하여 식 (14)의 Lagrange 승수(α')를 구하는 과정으로, 훈련 VMM 제어 모듈, 커널함수 계산 모듈, Gauss Jordan 제어 모듈 및 SV 테이블 구성 모듈에 의해서 실행된다.

가. 커널함수 계산 모듈

본 논문에서는 여러 가지 커널 함수 중에서 RBF 커널을 사용하였는데, RBF 커널은 식 (16)과 같이 지수함수 계산이 수행된다. 그러나 하드웨어로 실수의 지수함수 계산을 하기에는 다소 어려운 점이 있으므로 지수함수 계산값을 테이블 화하여 사용하였다. 커널 함수는 훈련 VMM 제어 모듈에서의 행렬 K' 의 구성과 인증 VMM 제어 모듈에서 테스트 벡터의 인증을 위한 계산시에 사용된다. 커널 함수의 주 계산은 벡터의 내적이

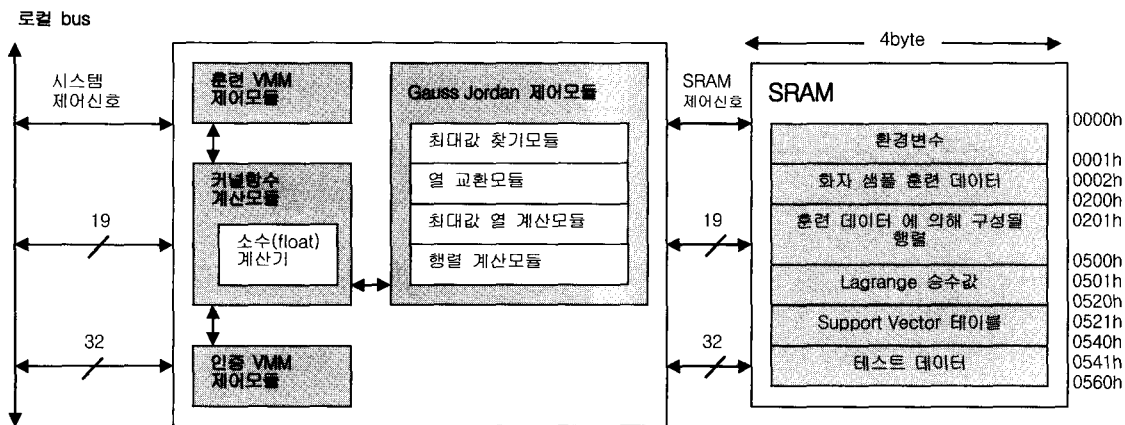


그림 2. 전체 블록 다이어그램
Fig. 2. Block diagram of hardware system

므로 그 계산은 소수 계산기를 통해 이루어진다.

나. Lagrange 승수

2장에서 설명한 것처럼 SV를 구하기 위해서는 식 (11)~(15)에서 나타내는 α_i 인 Lagrange 승수를 계산해야 한다. Gauss Jordan 제어 모듈이 이 역할을 담당한다. 훈련 데이터 설정 및 커널 값들의 계산을 마치게 되면 Gauss Jordan 제어 모듈이 동작하게 되는데, 그 동작 방법은 다음과 같다. SRAM에 저장된 행렬 K' 로부터 최대값 찾기 모듈을 통해 최대값을 가지는 데이터를 찾은 후 열 교환 모듈에서 최대값을 포함하는 열과 그 행에 대응하는 열의 벡터를 서로 바꾸는 역할을 수행한다. 최대값 열 계산 모듈에서는 최대값을 가지는 열에 대해 계산을 수행하고 행렬 계산 모듈에서는 나머지 열에 대해서 계산을 수행한다. 이 과정을 행렬 K' 의 행의 수만큼 반복하면 Gauss Jordan 제어 모듈에 대한 동작은 완료되어 Lagrange 승수를 구할 수 있다.

다. SV 테이블의 구성

Gauss Jordan 제어모듈이 동작을 마치면 SV 테이블 구성 모듈에서는 SRAM에 저장되어 있던 Lagrange 승수의 값을 읽어와 32번째 비트가 0인 데이터에 대응하는 훈련 벡터를 SV로 지정하고, SV로 지정된 벡터와 그 Lagrange 승수로 SV 테이블을 구성한다. SV 테이블 구성 모듈은 SV 테이블의 시작주소 및 크기를 SRAM에 저장한다.

2. 인증 과정

인증 과정의 동작은 인증 VMM 제어 모듈에 의해 수행되는데, 인증 VMM 제어 모듈은 SRAM에 저장된

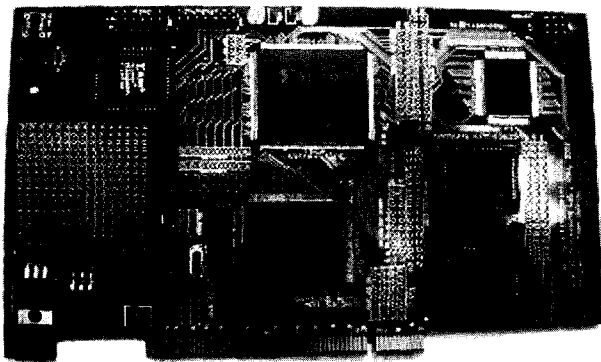


그림 3. SVM을 이용한 화자 인증 시스템
Fig. 3. Speaker verification system using SVM

사용자의 ID에 대응하는 SV의 시작주소와 크기를 읽어와 SV 테이블에 접근하여 해당 사용자의 SV와 Lagrange 승수를 읽어온다. 이렇게 읽어 들인 SV 및 Lagrange 승수와 테스트 벡터를 식 (15)의 좌변에 대입하여 그 값이 양이면 사용자를 수락하고, 음이면 거절한다.

V. 화자인증 하드웨어 시스템 성능 분석

SVM을 이용한 화자 인증 시스템 구현을 위하여 PCI I/F를 이용하여 호스트와 통신하며 ARM9, Xilinx FPGA와 SRAM 모듈로 구성된 시스템을 그림 3과 같이 구현하였다. 실험은 훈련 데이터를 이용하여 SV를 아내고 SV 테이블을 제대로 구성하는지를 검증하였으며, 테스트 벡터를 입력시켜 인증결과가 올바르게 출력되는지를 확인하였다.

실험에 사용된 훈련용 음성 특징벡터는 24차원 벡터 31개를 사용하였는데, 이 중 6개는 본인의 음성 특징벡터로, 나머지 25개의 벡터는 타인의 음성 특징벡터로 구

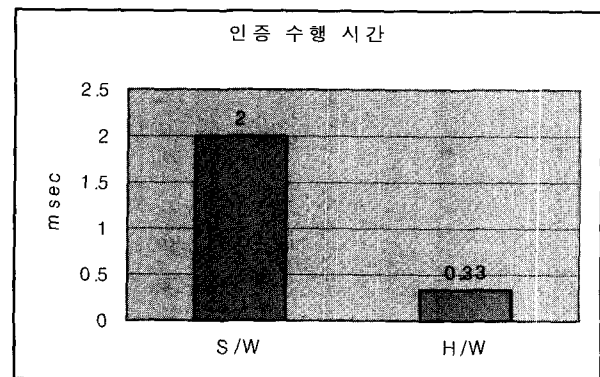
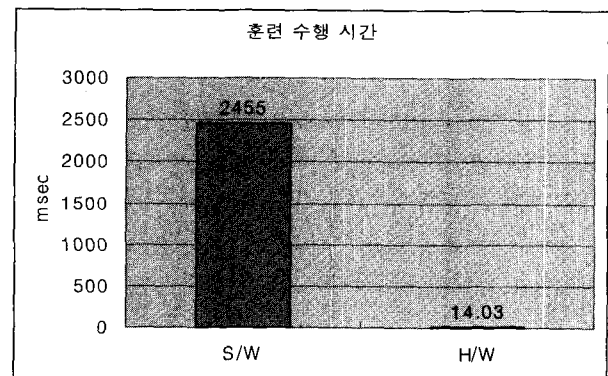


그림 4. 소프트웨어와 하드웨어의 훈련 및 인증 수행 시간 비교

Fig. 4. Training and verification times of software and hardware

성하였다. 이상의 데이터를 이용하여 하드웨어로 SVM을 구축하여 소프트웨어로 구현한 SVM과 비교한 결과, SV 테이블이 정상적으로 구성되었음을 확인할 수 있었다. 또한 인증과정의 확인을 위해서 본인 및 타인의 테스트 벡터를 입력했을 경우 인증 결과값이 소프트웨어로 구현한 결과와 동일함을 확인할 수 있었다.

FPGA는 Xilinx Virtex 600e를 이용하여 90%의 슬라이스 사용률, 508,845 게이트를 이용하였다. 그림 4는 그림 3의 시스템에서 50MHz로 동작할 때, 훈련 과정과 테스트 과정의 계산 시간을 측정하여 3장에서 측정된 소프트웨어의 수행 시간과 비교한 그래프이다.

그래프에서 나타내는 바와 같이 하드웨어로 구현했을 때가 소프트웨어로 구현했을 때보다 훈련 시간은 175분의 1로, 인증 시간은 6분의 1로 줄어들음을 확인할 수 있었다. 따라서 하드웨어로 구현한 시스템이 소프트웨어로 구현한 시스템보다 뛰어난 성능을 나타냄을 알 수 있으며, 훈련 시간에 비해서 인증 시간의 감소가 적은 것은 SVM 알고리즘 자체가 인증 수행 시의 계산량이 매우 적기 때문이라고 할 수 있다. 또한 소프트웨어와 하드웨어의 인증 시간이 각각 2msec와 0.33msec로 나타나므로 그림 5에서와 같이 초당 처리 가능한 인원수는 S/W에서는 약 500명, H/W에서는 약 3030명으로 인증시에 하드웨어가 소프트웨어보다 약 6배 이상의 성능향상을 보임을 알 수 있다.

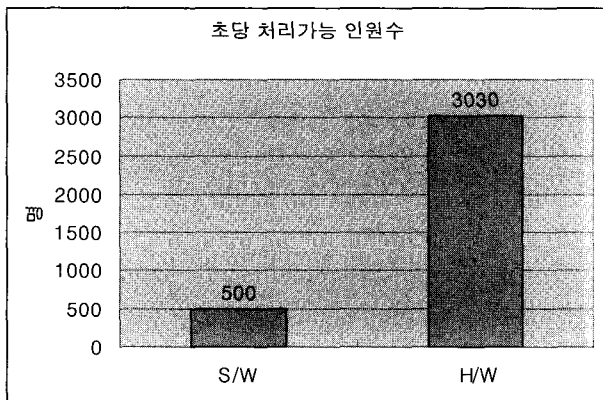


그림 5. 소프트웨어와 하드웨어의 초당 처리가능 인원수 비교

Fig. 5. Processing power of software and hardware

VI. 결 론

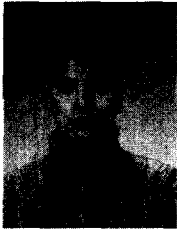
본 논문에서는 SVM을 이용한 화자인증 알고리즘을 제안하였고, 이의 실시간 처리를 위해 하드웨어 구현에 대하여 설명하였다. 제안한 화자인증 알고리즘은 특징벡터의 크기가 175분의 1로 감소했음에도 불구하고 약 1.5% 정도의 인식률 저하를 가져왔을 뿐이며, 모델의 크기나 수행시간에서는 탁월한 강점을 나타내었다. 그리고 제안한 하드웨어 구조를 PCI I/F를 이용하여 구현한 FPGA 보드에서 실험한 결과 초당 약 3,030명의 인증 과정을 처리함을 확인하였다.

앞으로의 연구는 화자인증뿐만 아니라 인증이 필요한 다른 신체적 특징, 즉 얼굴이나 지문, 홍채 인증 등으로 확장하는 연구가 진행되어야 할 것이며, 제안한 하드웨어의 속도 향상을 위해 VMM의 디자인에 있어 병렬적 처리와 FPGA의 게이트 수를 줄여 집적도 면에서 향상을 보일 수 있도록 하는 연구가 진행될 것이다.

참 고 문 헌

- [1] J. P. Campbell, "Speaker recognition: a tutorial," Proc. of the IEEE, Vol. 85, No. 9, pp. 1437-1462, Sep. 1997.
- [2] Q. Li, B.-H. Juang, C.-H. Lee, Q. Zhou and F. K. Soong, "Recent advancements in automatic speaker authentication," IEEE Robotics and Automation Magazine, pp. 24-34, Mar. 1999.
- [3] J. He, L. Liu and G. Palm, "A new codebook training algorithm for VQ-based speaker recognition," Proc. ICASSP, Vol. 2, pp. 1091-1094, 1997.
- [4] C. M. delAlamo, F. J. C. Gil, C. dela Torre Munilla and L. H. Gomez, "Discriminative training of GMM for speaker identification," Proc. ICASSP, Vol. 1, pp. 89-92, 1996.
- [5] S. Haykin, Neural networks, Prentice Hall, 1999.
- [6] W.-Y. Choi, K. Lee and Y. Chung, "Support vector machines for robust speaker verification," Proc. of the AICSS, pp. 262-267, 2002.
- [7] B. Scholkopf, C. J. C. Burges and A. J. Smola, Advances in kernel methods, The MIT Press, 1999.
- [8] R. Genov and G. Cauwenberghs, "Charge-mode parallel architecture for vector-matrix multiplication," Proc. of the IEEE, Vol. 48, No. 10, Oct. 2001.

저 자 소 개



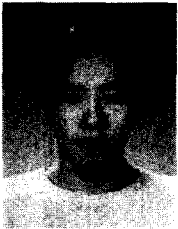
최 우 용(정회원)
 1998년 부산대학교 통계학과
 학사 졸업
 2000년 부산대학교 전자공학과
 석사 졸업
 2000년 2월~2001년 1월
 L&H Korea 연구원

2001년 2월~현재 한국전자통신연구원 연구원
 <주관심분야: 생체인식, 신호처리, 음성인식>



반 성 범(정회원)
 1991년 서강대학교 전자공학과
 학사 졸업
 1995년 서강대학교 전자공학과
 석사 졸업
 1999년 서강대학교 전자공학과
 박사 졸업

1999년 2월~현재 한국전자통신연구원 선임연구원
 <주관심분야: 생체인식, 신호처리, 정보보호>



황 병 희(정회원)
 2004년 부산대학교 컴퓨터공학과
 석사 졸업,
 2004년 2월~현재 LG CNS 연구원
 <주관심분야: 생체인식, 패턴인식, SVM>



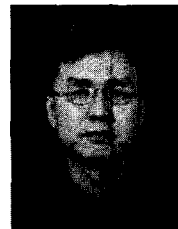
정 용 화(정회원)
 1984년 한양대학교 전자통신
 공학과 학사 졸업
 1986년 한양대학교 전자공학과
 석사 졸업
 1997년 미국 남가주대학교 컴퓨터
 공학과 박사 졸업

1986년 2월~2003년 8월: 한국전자통신연구원
 책임연구원
 2003년 8월~현재 고려대학교 컴퓨터정보학과
 부교수
 <주관심분야: 정보보호, 생체인식>



이 경 희(정회원)
 1993년 연세대학교 컴퓨터과학과
 학사 졸업
 1998년 연세대학교 컴퓨터과학과
 석사 졸업
 2004년 연세대학교 컴퓨터과학과
 박사 졸업

1993년~1996년 LG소프트(주) 연구원
 2000년~현재 한국전자통신연구원 선임연구원,
 <주관심분야: 생체인식, 얼굴인식, 패턴인식, 영상
 처리>



정 상 화(정회원)
 1985년 서울대학교 전기공학과
 학사 졸업
 1988년 미국 아이오와 주립대학교
 컴퓨터공학과 석사 졸업
 1993년 미국 남가주대학교 컴퓨터
 공학과 박사 졸업

1993년~1994년 미국 센트럴플로리다대학교 교수
 1994년~현재 부산대학교 컴퓨터공학과 부교수
 <주관심분야: 컴퓨터구조, 병렬처리>