

포섭구조 일대다 지지벡터기계와 Naive Bayes 분류기를 이용한 효과적인 지문분류

(Effective Fingerprint Classification using Subsumed One-Vs-All Support Vector Machines and Naive Bayes Classifiers)

홍진혁[†] 민준기[†] 조웅근[†] 조성배^{††}
 (Jin-Hyuk Hong) (Jun-Ki Min) (Ung-Keun Cho) (Sung-Bae Cho)

요약 지문분류는 사전에 정의된 클래스로 입력된 지문을 분류하여 자동지문인식 시스템에서 비교해야 할 지문의 수를 줄여준다. 지지벡터기계(support vector machine; SVM)는 패턴인식 분야에서 널리 사용되고 있을 뿐만 아니라 지문분류에서도 높은 성능을 보이고 있다. SVM은 이진클래스 분류기이기 때문에 다중클래스 문제를 지문분류를 위해서 적절한 분류기 생성과 결합 기법이 필요하며, 본 논문에서는 일대다(one-vs-all; OVA) 방식으로 구성된 SVM을 naive Bayes(NB) 분류기를 이용하여 동적으로 구성하는 분류방법을 제안한다. 지문분류에서 대표적으로 사용되는 특징인 FingerCode와 지문의 구조적 특징인 특이점과 의사용선을 사용하여 OVA SVM과 NB 분류기를 학습하고, 포섭구조의 분류기를 구성하여 효과적인 지문분류를 수행한다. NIST-4 데이터베이스에 제안하는 방법을 적용하여 5클래스 분류에 대해서 90.8%의 높은 분류율을 획득하였으며, OVA 전략의 SVM을 다중클래스 분류문제에 적용할 때 발생하는 동점문제를 효과적으로 처리하였다.

키워드 : 지문분류, SVM, FingerCode, naive Bayes 분류기, 특이점, 의사용선, 동적 분류

Abstract Fingerprint classification reduces the number of matches required in automated fingerprint identification systems by categorizing fingerprints into a predefined class. Support vector machines (SVMs), widely used in pattern classification, have produced a high accuracy rate when performing fingerprint classification. In order to effectively apply SVMs to multi-class fingerprint classification systems, we propose a novel method in which SVMs are generated with the one-vs-all (OVA) scheme and dynamically ordered with naïve Bayes classifiers. More specifically, it uses representative fingerprint features such as the FingerCode, singularities and pseudo ridges to train the OVA SVMs and naïve Bayes classifiers. The proposed method has been validated on the NIST-4 database and produced a classification accuracy of 90.8% for 5-class classification. Especially, it has effectively managed tie problems usually occurred in applying OVA SVMs to multi-class classification.

Key words : fingerprint classification, support vector machine, FingerCode, naïve Bayes classifier, singularity, pseudo ridges, dynamic classification

1. 서론

본 연구는 생체인식연구센터(BERC)를 통해 한국과학재단(KOSEF)에서 지원받았음.

[†] 학생회원 : 연세대학교 컴퓨터과학과
 hjinh@sclab.yonsei.ac.kr
 loomlike@sclab.yonsei.ac.kr
 bearoot@sclab.yonsei.ac.kr

^{††} 종신회원 : 연세대학교 컴퓨터과학과 교수
 sbcho@cs.yonsei.ac.kr

논문접수 : 2005년 11월 15일

심사완료 : 2006년 9월 4일

지문분류는 자동지문인식시스템에서 매우 중요한 부분으로, 많은 사람이 시스템에 등록된 경우 일대일 매칭의 횟수를 줄여 사용자 인식을 빠르게 수행하도록 돕는다[1,2]. 많은 연구에서 지문을 whorl(W), left loop(L), right loop(R), arch(A)와 tented arch(T)의 5클래스로 분류한 Henry 시스템을 채택한다[3]. Henry 시스템은 지문을 중심점과 삼각주의 상대적 위치와 수에 따라 분류하기 때문에, 많은 연구진들이 용선의 흐름으로부터 특이점을 정확히 추출하고자 한다[4]. Karu와 Jain은 특

이점을 바탕으로 하는 휴리스틱 분류 알고리즘을 제안하였고[5], Nyongesa 등은 중심점과 삼각주의 상대적 위치를 사용하였다[2]. Zhang과 Yan은 지문을 분류하기 위해 특이점과 함께 의사용선을 정의하여 사용하기도 하였다[6]. 비록 특이점은 지문분류에 명확한 기준이 되지만, 많은 경우 지문영상의 품질이 떨어지기 때문에 특이점을 정확히 추출하는 것은 매우 어렵다[6,7].

지문분류에서 높은 분류율을 얻기 위해서 FingerCode, 용선분포, 방향성 영상 등의 다양한 특징이 제안되었다. Jain 등은 Gabor 필터를 이용하여 용선의 방향성 흐름을 추출하여 정의한 FingerCode를 제안하였고 [8], Park 등은 고속 Fourier 변환으로 여과된 방향성 정보를 사용하였다[9]. Chong 등은 지문의 용선으로부터 추출한 기하학적 정보를 사용하였고[10], Cappelli 등은 지문을 그래프 모델로 정의한 방향성 영상을 제안하였다[3]. Nagaty는 지문으로부터 블록화된 방향성 영상을 이용한 일련의 기호를 추출하였으며[7], Chang과 Fan은 10개의 기본적인 용선의 패턴을 정의하고 이들의 분포를 지문분류에 사용하였다[11]. 이러한 지문특징들은 특이점 보다는 많은 정보를 포함하고 있기 때문에 지문분류의 성능을 높이는 역할을 한다.

최근에는 보다 강건한 지문분류기를 생성하기 위해 다양한 특징과 분류 기술을 결합하는 연구가 시도되고 있다[4,12]. Senior는 지문의 용선구조를 인식하기 위해 은닉 Markov 모델과 결정트리를 사용하였고[4], Yao 등은 회귀적 신경망과 SVM을 이용하여 다양한 특징을 결합하였다[12]. 다양한 종류의 특징과 방법을 효과적으로 결합한다면 지문분류는 보다 정확하고 안정적인 뿐만 아니라 영상에 포함된 다양한 잡음에 대해서도 높은 성능을 얻을 수 있다.

본 논문에서는 서로 다른 지문특징으로 학습된 NB 분류기와 SVM을 효과적으로 결합하는 지문분류 방법을 제안한다. 높은 분류성능을 위해서 FingerCode를 이용하여 SVM을, 특이점을 사용하여 NB 분류기를 OVA 전략으로 학습한다. 입력된 지문영상에 대해 NB 분류기

로부터 계산된 각 클래스의 확률에 따라 SVM을 순차적으로 구성하여 분류를 수행하기 때문에, 다수의 SVM을 사용할 때 발생하는 동점문제를 해결한다.

NIST 데이터베이스 4를 대상으로 제안하는 방법의 유용성을 검증하였다. 특이점을 이용한 NB 분류기의 경우 85.4%의 분류정확률을, FingerCode를 이용한 SVM의 경우 1.8%의 거부율에 대해서 90.4%의 분류정확률을 획득하였다. 제안하는 방법은 1.8%의 거부율에 대해서 90.8%의 분류정확률을 획득하였으며, 입력된 지문에 대해 다수의 지지벡터기계가 만족하는 동점의 경우를 효과적으로 처리하는 것을 확인하였다.

2. FingerCode기반 지지벡터기계를 이용한 지문분류

2.1 FingerCode

Jain이 1999년에 제안한 FingerCode는 NIST 데이터베이스 4에서 추출된 대표적인 지문 특징으로 지문분류와 인식에서 많이 사용되고 있다[8]. 추출 알고리즘은 먼저 지문영상을 중심점을 기준으로 부채꼴 모양의 48개 영역으로 분할하고, 분할된 지문영상에 4방향(0°, 45°, 90°, 135°)의 Gabor 필터를 적용하여 각 방향에 평행한 용선을 강화한다. 필터 방향과 평행하지 않은 용선은 그림 1과 같이 흐려진다. 필터 처리 후, Gabor 필터의 각 방향에 대해 48개 구역의 표준편차를 계산하여 총 192차원의 특징벡터인 FingerCode를 생성한다. FingerCode를 기반으로 Jain 등은 k 최근점 이웃과 신경망을 이용하여 1.8%의 거부율에서 90%의 분류정확률을 획득하였으며[8], Yao 등은 오류정정부호(error-correcting code; ECC) 전략을 이용한 SVM을 사용하여 90%의 분류정확률을 획득하였다[12].

2.2 SVM을 이용한 지문분류

통계적 학습이론을 기반으로 하는 SVM은 패턴인식이나 회귀모델 등에서 활발히 연구되고 있다[14,15]. SVM은 수식 (1)과 같이 비선형적 변환함수를 이용하여 입력된 샘플을 고차원의 특징공간으로 투사하고 학습

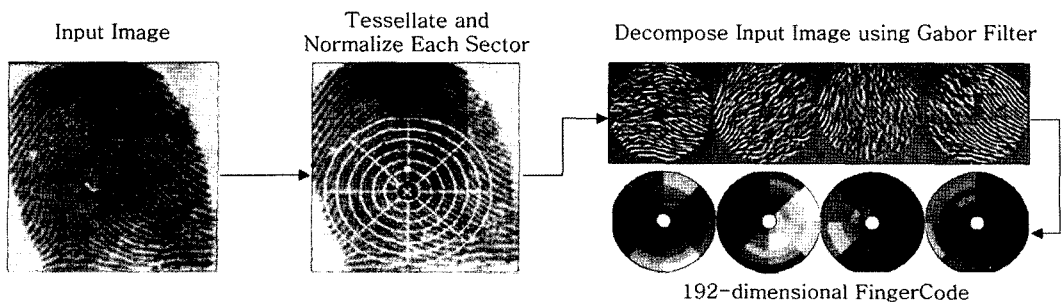


그림 1 FingerCode 생성과정[8]

데이터에 대한 인식오류를 최소화하는 최적 초평면을 찾는다[15].

$$X : x = (x_1, \dots, x_n) \rightarrow F : \Phi(x) = (\Phi_1(x), \dots, \Phi_n(x)) \quad (1)$$

학습 데이터의 수가 n 일 때, 클래스 레이블 $c_i \in \{1, -1\}$ 를 가지는 샘플 x_i 에 대해 SVM은 수식 (2)와 같이 샘플과 초평면 사이의 거리를 계산한다.

$$f(x) = \sum_{i=1}^n \alpha_i c_i K(x, x_i) + b, \quad K(x, x_i) = \Phi(x) \cdot \Phi(x_i) \quad (2)$$

수식 (2)의 α_i 는 x_i 가 초평면을 구성하는 지지벡터일 때에는 0이 아니며, 커널 함수 $K(x, x_i)$ 는 비선형 매핑 함수의 내적으로 쉽게 계산된다. 표 1은 SVM에서 많이 사용되는 커널함수를 보여준다.

표 1 SVM에서 사용되는 대표적인 커널함수

선형	다항식	가우시안	시그모이드
$(x \cdot x_i)$	$(x \cdot x_i + \gamma)^d$	$\exp\left(-\frac{\ x - x_i\ ^2}{2\sigma^2}\right)$	$\tanh(x \cdot x_i + \gamma)$

SVM은 기본적으로 이진분류기이기 때문에, 지문분류와 같이 다중클래스문제에서는 OVA(one-vs-all), 쌍별(pairwise)이나 완전코드(complete code)와 같은 다수의 분류기를 생성하는 전략이 필요하다[16,17].

1) OVA 전략: M (클래스 수)개의 SVM을 학습하며, 각 SVM은 샘플을 대응하는 클래스와 나머지 클래스로 분류한다. j 번째 SVM의 결정함수 $\hat{a}_j(x)$ 는 수식 (2)의 c_i 를 수식 (3)의 t_i 와 교환하여 얻어진다.

$$t_i = \begin{cases} +1 & \text{if } c_i = j \\ -1 & \text{if } c_i \neq j \end{cases} \quad (3)$$

2) 쌍별 전략: 하나의 SVM이 한 쌍의 클래스를 분류하며, 클래스의 모든 쌍의 수인 $M_C2 = M(M-1)/2$ 만큼의 SVM을 필요로 한다. 각 SVM의 학습에 사용되는 학습 데이터의 수가 적기 때문에 학습이 빠르다. 분류하고자 하는 클래스 $j, k = \{1, \dots, M\}$ (단, $j \neq k$)에 대해서 결정함수 $\hat{a}_j(x)$ 는 수식 (2)의 c_i 를 수식 (4)의 t_i 와 교환한 것과 같다.

$$t_i = \begin{cases} +1 & \text{if } c_i = j \\ -1 & \text{if } c_i = k \end{cases} \quad (4)$$

3) 완전코드 전략: 분류하고자 하는 클래스의 모든 이진조합에 대해 SVM을 구성한다. 특히 5클래스 분류 문제에서는 OVA와 이대다(two-vs-all) 전략을 모두 고려한 것이 된다. 모든 클래스가 두 개의 집합 $\{j\}$ 와 $\{k\}$ 으로 나뉘어질 때, 결정함수 $\hat{a}_j(x)$ 는 수식 (2)의 c_i 에 수식 (5)의 t_i 를 대입하여 얻어진다.

$$t_i = \begin{cases} +1 & \text{if } c_i \in \{j\} \\ -1 & \text{if } c_i \in \{k\} \end{cases} \quad (5)$$

분류에 필요한 SVM의 구성이 완료되면, 복수의 분류기 출력을 결합하는 방법이 필요하며, 대표적으로는 다수결투표(majority voting), 승자독식(winner-takes-all), ECC, 행동지식공간(behavior knowledge space; BKS)과 결정 템플릿(decision template; DT) 등이 있다.

1) 다수결투표: 개별 분류기가 가장 많이 선택한 클래스로 입력된 샘플을 분류한다[18]. Condorect의 이론을 통해 검증되었으며[19], 다양한 이론적 분석이 [20,21]에 있다. 단순하지만 좋은 성능을 보여주는데, 분류기의 결과가 동점을 내는 경우를 처리하지 못하는 한계가 있다.

2) 승자독식: 다수결투표의 한계점을 해결하기 위해서 제안되었으며, 수식 (6)에서와 같이 분류기 중에서 가장 높은 값을 가진 분류기의 결과를 입력된 샘플의 클래스로 선택한다. '최대치'로도 알려져 있으며, $ind_{i,j}(x)$ 는 지시자로 레이블 i 가 j 번째 SVM의 양의 클래스이면 1, 음의 클래스이면 -1이고, 그렇지 않으면 0을 값으로 갖는다.

$$c = \arg \max_{i=1, \dots, M} \sum_{j=1}^L ind_{i,j}(x) d_j(x) \quad (6)$$

3) ECC: 이 방법은 코드행렬 $E \in \{-1, 0, 1\}^{M \times L}$ (M : 클래스 수, L : 분류기 수)를 기반으로 동작한다[22,23]. E_{ij} 는 행렬 E 의 i 번째 열, j 번째 행의 요소를 가리키는데, ($E_{ij} = -1$ 또는 1)는 j 번째 분류기를 학습할 때 클래스 i 의 샘플이 양이나 음의 레이블을 가지는지는 의미한다. 만약 E_{ij} 가 0이면 i 번째 클래스를 가지는 샘플은 j 번째 분류기 학습에 사용되지 않는다. 테스트 샘플은 수식 (7)에서와 같이 코드행렬과의 거리를 비교하여 가장 작은 값을 가지는 레이블로 분류된다.

$$c = \arg \min_{i=1, \dots, M} \sum_{j=1}^L \frac{1 - \text{sign}(E_{i,j} d_j(x))}{2} \quad (7)$$

4) BKS: 이 방법은 분류기 출력의 모든 가능한 조합을 테이블 $T \in \{-1, 1\}^{M \times L}$ (M : 클래스 수, L : 분류기 수)에 저장하는데, 각 항별로 학습 데이터에서 가장 빈도가 높은 하나의 클래스 레이블을 가진다. 만약 학습 데이터가 없는 경우에는 레이블을 가지지 않는다. 테스트 샘플은 테이블 T 에서 분류기들의 출력과 똑같은 패턴을 가지는 항의 클래스로 분류된다[24]. 테이블 T 에서 해당하는 패턴이 학습되어 있지 않은 경우에는 분류에 실패한다.

5) DT: 학습 데이터로부터 결정 프로파일을 계산한 다음 이 값을 평균화하여 각 클래스에 결정 템플릿을 생성한다. L 개의 분류기를 가지고 M 클래스 분류를 수행하려 할 때, i 번째 샘플의 결정 프로파일은 수식 (8)과

같이 계산된다.

$$DP(x_i) = \begin{bmatrix} d_{1,1}(x_i) & \cdots & d_{1,M}(x_i) \\ \vdots & d_{y,z}(x_i) & \vdots \\ d_{L,1}(x_i) & \cdots & d_{L,M}(x_i) \end{bmatrix} \quad (8)$$

이때, $d_{y,z}(x_i)$ 는 z 클래스에 속한 샘플 x_i 에 대해 y 번째 분류기가 가지는 분류값을 나타낸다. 학습 데이터로부터 결정 프로파일이 생성되면, 수식 (9)에서와 같이 각 클래스에 대해서 이들 프로파일을 평균화하여 결정 템플릿을 계산한다. $ind_c(x_i)$ 는 샘플 x_i 의 클래스가 c 이면 1의 값을 가지고 그렇지 않으면 0을 값으로 갖는다 [25,26].

$$DT_c = \begin{bmatrix} dt_c(1,1) & \cdots & dt_c(1,M) \\ \vdots & dt_c(y,z) & \vdots \\ dt_c(L,1) & \cdots & dt_c(L,M) \end{bmatrix},$$

$$dt_c(y,z) = \frac{\sum_{i=1}^n ind_c(x_i) d_{y,z}(x_i)}{\sum_{i=1}^n ind_c(x_i)} \quad (9)$$

테스트 단계에서는 입력된 샘플의 결정 프로파일과 각 클래스의 결정 템플릿과의 유사도를 계산하며, 가장 유사한 결정 템플릿의 클래스로 분류한다. Kuncheva는 해밍 거리와 유클리드 거리를 포함한 11가지 유사도 측정 기준을 DT에 적용하였고, 기존의 대표적인 결합 방법인 다수결 투표나 BKS 등보다 높은 성능을 획득하였다[25].

3. 동적 지문분류

정적인 분류모델에 기반한 기존의 방법과는 달리, 본 논문에서는 SVM을 입력 샘플의 확률적 분류결과에 따라 적절히 구성하는 동적 지문분류 방법을 제안하여 다중클래스 분류에 대해 기존의 OVA SVM이 가지는 불명확함인 동점문제를 효과적으로 해결한다. 특이점, 의사용선과 FingerCode 등의 다양한 지문 특징을 이용할 뿐만 아니라 다수의 SVM을 포섭구조로 구성하여 분류 의사 결정을 명확히 한다. 포섭구조는 다중의 출력이 활성화될 경우 하나의 적절한 출력을 선택하기 위해 로봇이나 에이전트 분야에서 많이 사용되는 방법으로[27], 본 논문에서는 다수의 분류기로부터 획득된 분류 결과를 결합하는데 사용한다. 각 분류기로부터 얻어진 분류 결과의 평가 순서는 NB 분류기로부터 얻어진 각 클래스에 대한 확률값에 따라 결정된다.

제안하는 방법은 그림 2와 같이 NB 분류기와 OVA SVM으로 구성된다. NB 분류기는 입력된 지문 영상의 각 클래스에 대한 사후 확률 $prob = \{p_w, p_L, p_R, p_A, p_T\}$ 을 특이점과 의사용선을 이용하여 계산하고, OVA SVM은 2장에서 설명한 것과 같이 FingerCode를 이용하여 샘플의 각 클래스에 대한 소속도 $o\text{-svm} = \{m_w, m_R, m_L, m_A, m_T\}$ 을 측정한다. 동점과 거부의 경우를 처리하기 위해서 본 논문에서는 포섭구조의 형태로 OVA SVM을 순차적으로 선택하여 평가하는데, 이 순서는 NB 분류기에서 측정된 각 클래스에 대한 사후 확

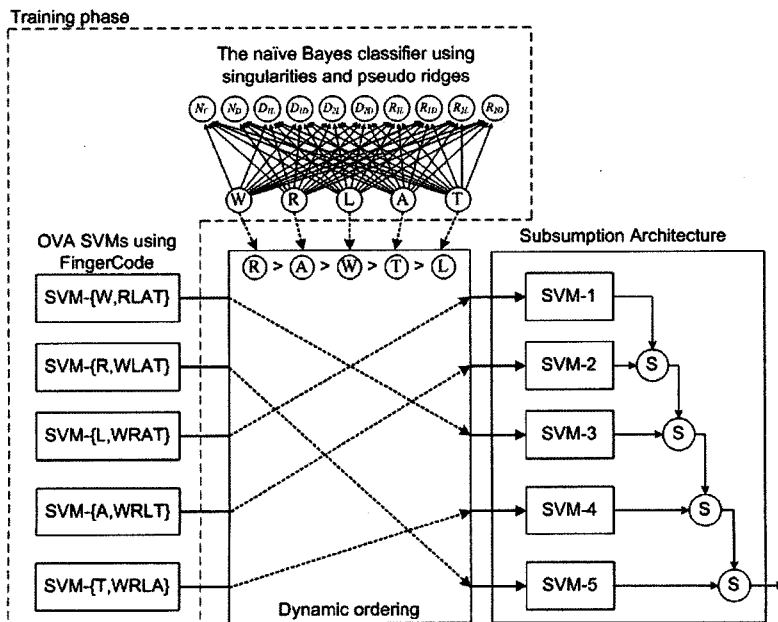


그림 2 제안하는 동적 분류방법

률에 의해 결정된다. 따라서 NB 분류기에 의해 보다 높은 확률을 가진 클래스에 대한 OVA SVM이 그렇지 않은 것보다 먼저 평가된다.

새로운 샘플이 입력되면, 먼저 각 클래스에 대한 소속 확률을 NB 분류기를 이용하여 측정한다. 측정된 확률에 따라 OVA SVM을 포섭구조로 구성하고, 만족하는 OVA SVM이 있을 때까지 그 순서에 따라 평가를 진행한다. 하나의 OVA SVM이 입력된 샘플을 대응하는 클래스에 속한다고 결정하면 그 샘플은 해당 클래스로 분류된다. 만약 어떤 OVA SVM도 만족하지 않을 경우에는 NB 분류기에서 계산된 확률 중 가장 높은 값을 가진 클래스로 분류된다. 그림 3은 제안하는 방법이 동작하는 의사코드를 보여준다. 입력된 샘플에 대해 OVA SVM이 동적으로 적절히 구성되어 동점이나 거부 의 경우를 효과적으로 해결한다.

NB 분류기는 지문의 대표적인 구조적 특징인 특이점과 의사용선을 이용하여 학습한다. 특이점은 중심점과 삼각주로 지문의 기본적인 특징인데, 중심점은 용선의 곡률이 최대인 점으로 보통 지문 중심에 위치하고 삼각주는 세 개의 다른 방향의 용선이 만나는 지점으로 삼각형의 모양을 보인다[5,6]. 전통적으로 중심점과 삼각주의 수와 위치는 지문 분류의 기준으로 사용되어 왔다. 이들 특이점은 보통 지문영상의 방향성 지도를 바탕으로 중심점과 삼각주의 위치를 계산하는 Poincare 색인을 이용하여 추출한다[1,6]. 보통 특이점은 지문영상의 중심부에 위치하기 때문에, 본 논문에서는 NIST 데이

타베이스 4에 대해 다음과 같은 제약조건을 고려한다.

- 512×480 영상의 가장자리는 보통 배경영역을 포함하기 때문에 가장자리로부터 40 픽셀까지는 특이점을 뽑지 않는다.
- 영상의 가장자리로부터 80 픽셀까지는 중심점을 뽑지 않는다.
- 삼각주는 보통 영상의 아래 부분에 위치하기 때문에 위에서 160 픽셀까지의 영상은 고려하지 않는다.
- 만약 중심점이 가장 가까운 삼각주로부터 8 픽셀 이내에 위치하면 이들 중심점과 삼각주는 제거된다.
- 중심점과 삼각주는 최대 2개씩만 뽑으며, 영상의 중심에서부터 가까운 순서대로 선택한다.

중심점과 삼각주의 수는 각각 NC 와 ND 로 표시하고, 영상 중간에 가장 가까운 중심점을 C 로 명한다. 중심점이 없는 경우에는 C 는 영상의 중간을 의미한다. 그림 4(a)는 두 개의 중심점을 가진 지문영상을 보여주며, 아래의 중심점이 C 로 사용된다. 삼각주는 왼쪽에서부터 차례로 $D1$ 과 $D2$ 로 정의된다.

중중 지문영상의 품질이 떨어져 특이점이 정확히 뽑히지 않을 경우가 있는데, 많은 연구들에서 특이점 이외에 다른 특징도 함께 사용한다. Zhang 등은 특이점기반 지문분류의 한계를 극복하기 위해 의사용선을 제안하였다[6]. 의사용선은 C 로부터 양방향으로 용선의 방향에 따라 연결된 점들로 구성되며, 본 논문에서는 100개의 점으로 의사용선을 정의한다. 의사용선을 추출하기 위해서 먼저 그림 5와 같이 16방향으로 지문영상의 방향성

```

prob[5] = {pw, pr, pL, pA, pT} // prob[]: NB 분류기가 계산한 사후 확률
order[5] = {0, 1, 2, 3, 4}
o-svm[5] = {mw, mr, mL, mA, mT} // o-svm[]: OVA SVM이 계산한 소속 정도

// OVA SVM의 평가 순서 결정
for(i=0; i<5; i++)
for(j=i+1; j<5; j++)
    if(prob[i] < prob[j])
    {
        int iTemp = prob[i];    prob[i] = prob[j];    prob[j] = iTemp;
        iTemp = order[i];    order[i] = order[j];    order[j] = iTemp;
    }

// 포섭구조의 OVA SVM을 이용한 지문분류
if(prob[order[0]] < r1) // r1: 거부 임계치
return reject;
for(i=0; i<5; i++)
{
if(o-svm[order[i]] >= a) // a: OVA SVM 만족 임계치
{
if(o-svm[order[i]] < r2) // r2: 거부 임계치
return reject;
return order[i];
}
}
return order[0];

```

그림 3 확률적으로 구성되는 OVA SVM을 이용한 동적 지문분류의 의사코드

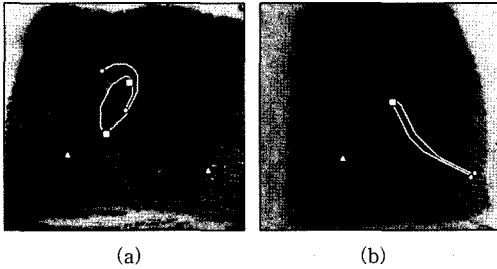


그림 4 특이점과 의사용선(□=중심점, △=삼각주, ○=의사용선 끝점)

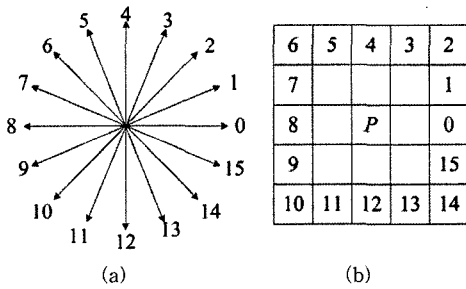


그림 5 의사용선에 사용된 16방향 지표

을 측정한다.

dir 이 현재 위치 p 에서의 용선방향을, dir_b 는 이전의 방향을 의미할 때, 다음 용선방향 dir_n 은 수식 (10)과 같이 계산된다.

$$dir_n = \begin{cases} dir, & \cos^{-1}(\vec{V}_{dir_b} \cdot \vec{V}_{dir}) > \cos^{-1}(\vec{V}_{dir_b} \cdot \vec{V}_{(dir+8)\%16}) \\ (dir+8)\%16, & otherwise \end{cases} \quad (10)$$

용선의 진행방향이 계산되면 p 는 그림 5(b)와 같이 그 방향에 따라 다음 점으로 이동하여 용선을 추적한다. 만약 d_p 가 수식 (12)를 만족하면 의사용선은 회전(turn)으로 간주한다.

$$d_p = \begin{cases} dir - dir_b + 16, & dir - dir_b < -8 \\ -dir + dir_b + 16, & dir - dir_b > 8 \\ dir - dir_b, & otherwise \end{cases} \quad (11)$$

$$\left| \sum_{p=0}^n d_p \right| > 15 \quad (12)$$

의사용선의 추적은 100번 반복하거나 의사용선이 '회전'으로 결정되면 완료된다. 그림 4(b)와 같이 의사용선이 오른쪽으로 흐르면 'left loop'로 분류되고, 그림 4(a)와 같이 의사용선이 '회전'으로 간주되면 'whorl'로 분류된다.

특이점과 의사용선을 지문분류에 사용하기 위해서, 점 C 와 다른 점들 사이의 상대적 위치와 거리를 사용하며, 점 C 와 특정 점 P 의 상대적 위치 L 는 다음과 같이 정의한다.

1) $C_y > P_y$:

$$L = \begin{cases} 0, & C_x > P_x \\ 4, & C_x \leq P_x \end{cases} \quad (13)$$

2) $C_y \leq P_y$:

$$L = \begin{cases} 1, & C_x > P_x \ \& \ |C_y - P_y| < 4.0|C_x - P_x| \\ 2, & |C_y - P_y| \geq 4.0|C_x - P_x| \\ 3, & C_x \leq P_x \end{cases} \quad (14)$$

두 점 사이의 거리 D 는 다음과 같이 정의한다.

$$dis = |(C_x - P_x)^2 + (C_y - P_y)^2| \quad (15)$$

$$D = \begin{cases} 1, & dis \leq 10 \\ 2, & 10 < dis \leq 20 \\ 3, & otherwise \end{cases} \quad (16)$$

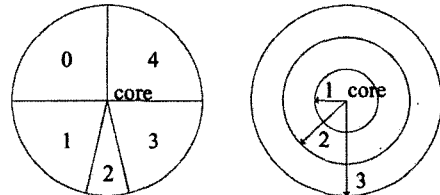


그림 6 점 C 와 P 의 상대적 위치와 거리

본 논문에서는 중심점과 삼각주의 수(N_C, N_D), 이들의 상대적 위치와 거리($D_{1L}, D_{1D}, D_{2L}, D_{2D}$)와 중심점과 의사용선 끝점과의 상대적 위치와 거리($R_{1L}, R_{1D}, R_{2L}, R_{2D}$)를 NB 분류기를 이용한 지문분류에 사용한다. NB 분류기는 5개의 상호독립적인 클래스(W,R,L,A,T)와 앞서 정의한 10개의 지문특징($N_C, N_D, D_{1L}, D_{1D}, D_{2L}, D_{2D}, R_{1L}, R_{1D}, R_{2L}, R_{2D}$)으로 구성되며, 각 클래스 노드는 그림 2에서와 같이 모든 특징들과 연결되어 있다. 표 2는 NB 분류기 구축에 사용된 지문특징에 대해서 보여준다.

NB 분류기는 샘플로부터 관측된 값을 바탕으로 각 클래스의 사후 확률을 계산하는데[28], 이를 위해 변수들의 사전 확률분포와 특징과 클래스 사이의 조건부 확

표 2 NB 분류기에 사용된 특징

특징	정의	상태
N_C, N_D	중심점과 삼각주의 수	0,1,2
D_{1L}, D_{2L}	삼각주의 위치	0,1,2,3,4,무
R_{1L}, R_{2L}	의사용선 끝점의 위치	0,1,2,3,4,회전
D_{1D}, D_{2D}	점 C 와 삼각주 사이의 거리	1,2,3,무
R_{1D}, R_{2D}	점 C 와 의사용선 끝점과의 거리	1,2,3,회전

를분포를 먼저 설계해야 한다[29]. 확률분포는 n_T 개의 영상으로 구성된 학습 데이터로부터 계산되는데, 변수 A 의 i 번째 상태를 A_i 이고 $\text{count}(A_i)$ 는 변수 A 가 i 번째 상태를 가지는 경우의 빈도를 나타낼 때, 사전 확률 $P(A_i)$ 는 수식 (17)과 같이 계산된다.

$$P(A_i) = \frac{\text{count}(A_i)}{n_T} \quad (17)$$

만약 변수 A 가 B 를 부모 노드로 가지면, 조건부 확률 $P(A_i|B_j)$ 는 수식 (18)과 같이 계산된다.

$$P(A_i|B_j) = \frac{\text{count}(A_i, B_j)}{\text{count}(B_j)} \quad (18)$$

Bayes 이론에 따라 n 개의 특징값이 증거로 주어진 때 각 클래스의 사후확률은 수식 (19)와 같이 측정된다.

$$P(C|F_1, \dots, F_n) = \frac{P(C)P(F_1, \dots, F_n|C)}{P(F_1, \dots, F_n)} \quad (19)$$

수식 (19)의 분모는 클래스의 사후확률 계산에서 항상 동일하기 때문에 분자만을 고려한다면 클래스의 사후확률은 특징들 사이의 독립성 가정에 따라 다음과 같이 표현된다.

$$\begin{aligned} P(C)P(F_1, \dots, F_n|C) \\ = P(C)P(F_1|C)P(F_2|C)...P(F_n|C) \\ = P(C)\prod_{i=1}^n P(F_i|C) \end{aligned} \quad (20)$$

4. 실험 및 결과

4.1 실험 환경

본 논문에서는 제안하는 방법을 평가하기 위해 지문 분류의 대표적인 평가 데이터인 NIST 데이터베이스 4를 사용하였다[13]. 이 데이터는 4,000장의 스캔된 영상(해상도: 512x512)으로 구성된다. 2,000개의 손가락에서 각각 두 장의 영상을 취득하였고, whorl (W), right loop (R), left loop (L), arch (A)와 tented arch (T)의 5개의 클래스가 균등하게 분포한다. 지문영상의 모호성 때문에 350개의 지문(17.5%)은 두 개의 클래스 레이블을 가지기 때문에, SVM과 NB 분류기의 학습에는 첫 번째 레이블만을 사용하였고, 테스트에서는 두 개의 레이블을 함께 고려하였다. 실험에서는 각 지문의 첫 번째 영상인 (F0001~F2000)를 학습 데이터로, 나머지 영상을 테스트 데이터로 사용하였다. Jain이 제안한 FingerCode는 -1에서 1로 정규화하여 사용하였으며, FingerCode 생성에서 발생하는 거부로 인해 학습 데이터에는 1.4%의 거부영상, 테스트 데이터에는 1.85%의 거부영상이 포함되었다. LIBSVM 패키지(<http://www.csie.ntu.edu.tw/~cjlin/libsvm>)로 SVM 모듈을 구성하였고 $\sigma^2(=0.0625)$ 값의 가우시안 커널을 사용하였다[30].

4.2 FingerCode를 이용한 SVM기반 지문분류

다중의 SVM을 생성하고 결합하는 기존 방법의 성능을 먼저 분석하였다. 승자독식, 해밍 거리와 유클리드 거리를 사용한 ECC와 DT, BKS를 사용하였고 표 3은 그 분류결과를 보여준다. 승자독식과 ECC가 90.1%의 분류정확률로 다른 방법에 높은 결과를 얻었으며, OVA 전략으로 SVM을 구성하였을 때 가장 높은 성능을 획득하였다. 표 4는 OVA SVM을 이용한 승자독식 전략의 혼동행렬을 보여준다.

표 3 다양한 SVM기반 분류방법 비교

결합전략	OVA 전략	쌍별 전략	완전코드 전략
승자독식	90.1	87.7	90.0
ECC (HM)	90.1	88.6	90.0
ECC (EU)	90.1	88.6	90.0
BKS	88.8	89.4	89.3
DT (HM)	89.6	87.6	89.6
DT (EU)	89.8	88.3	89.5

표 4 승자독식기반 OVA SVM의 혼동행렬

실제 클래스	분류 결과				
	W	R	L	A	T
W	382	6	6	0	0
R	7	365	2	5	17
L	10	0	365	13	10
A	3	4	2	356	47
T	2	8	12	40	302

4.3 제안하는 방법

NB 분류기와 SVM을 이용한 제안하는 방법은 지문 사이의 미묘한 차이를 다루어 지문분류 성능을 높이고자 하였다. 주된 분류는 성능이 뛰어난 SVM을 이용하고 NB 분류기는 SVM을 동적으로 구성하는 보조자의 역할을 한다. 그림 7에서와 같이 NB 분류기만을 이용하였을 때에는 성능이 높지 않지만, 동적으로 분류모델 구성에 이용될 때 90.8%의 분류정확률을 획득하여 제안하는 방법이 다른 방법보다 높은 성능을 획득하였다. NB 분류기와 SVM의 분류결과를 내적으로 결합한 경우 90.2%의 분류정확률을 얻었다. 이는 제안하는 방법이 SVM과 NB 분류기를 효과적으로 결합하였다는 것을 보여주며, 다양한 지문특징을 함께 고려할 때 보다 높은 분류 결과를 얻을 수 있었다. 표 5는 제안하는 방법의 혼동행렬을 보여준다.

그림 8은 정확히 분류하기 어려운 애매모호한 영상을 보여준다. 기존의 승자독식을 이용하는 OVA SVM은 'tented arch'로 잘못 분류한 반면에, 제안하는 방법은 NB 분류기가 SVM을 효과적으로 보완하여 이 영상을 'arch'로 정확히 분류하였다.

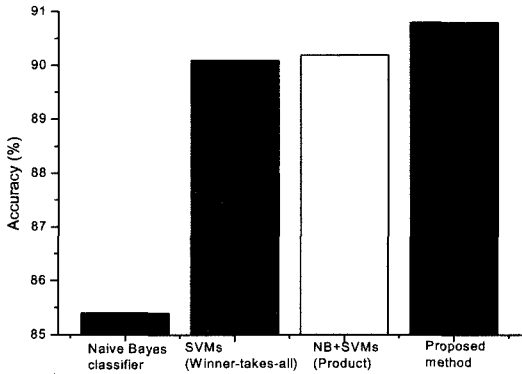


그림 7 분류 성능 비교

표 5 제안하는 방법의 혼동행렬

실제 클래스	분류 결과				
	W	R	L	A	T
W	373	10	10	0	0
R	4	374	0	6	15
L	5	0	377	8	9
A	0	6	4	365	40
T	1	8	15	39	295

4.4 관련연구와의 성능비교

본 논문에서는 Jain 등[8], Yao 등[12], Park[9], Zhang

과 Yan[6]이 제안한 기존의 대표적인 지문분류 방법과의 성능비교를 수행하였다. 그림 9는 NIST 데이터베이스 4에 대한 기존 방법의 분류정확률과 제안하는 방법의 분류정확률을 보여주며, 결과는 각 방법의 거부율에 대응하는 분류정확률을 표시하였다. 제안하는 방법은 대부분의 거부율에서 다른 방법에 비해 높은 분류 성능을 획득하였다. 특이점과 의사용선을 사용하는 Zhang과 Yan의 방법은 동작이 매우 빠르지만 지문영상에 포함된 잡음에 약한 성능을 보였지만[6], 제안하는 방법은 지문의 구조적 특징과 함께 FingerCode를 사용하여 영상에 포함된 잡음에 대한 영향을 최소화하였다. 제안하는 방법은 Jain의 연구에 대해서는 특이점을 추가로 고려하면서 OVA SVM의 성능을 극대화하였으며, 상대적 관계 그래프 모델을 사용한 Yao의 방법에 비해서는 특징을 쉽게 추출하였다.

5. 결론

본 논문에서는 NB 분류기와 OVA SVM을 효과적으로 결합한 지문분류 방법을 제안하였으며, NIST 데이터베이스 4에 대해 기존의 방법에 비해 높은 분류 성능을 획득하였다. 특이점, 의사용선, FingerCode 등의 다양한 지문특징을 이용하여 5클래스 분류 문제에 대하여 90.8%의 분류 정확률을, 4클래스 분류에 대해서 94.9%

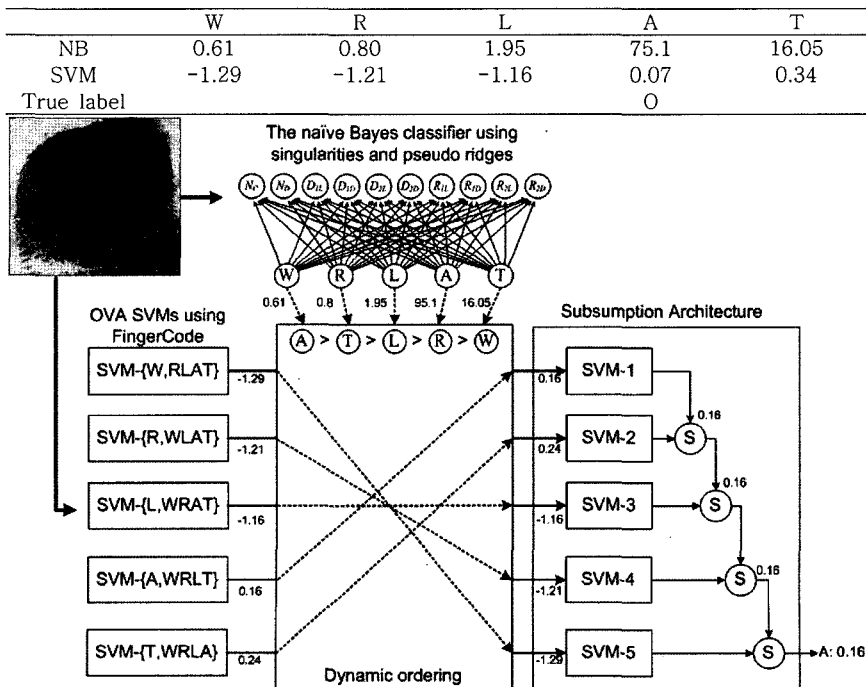


그림 8 제안하는 방법의 애매모호한 지문영상 분류과정

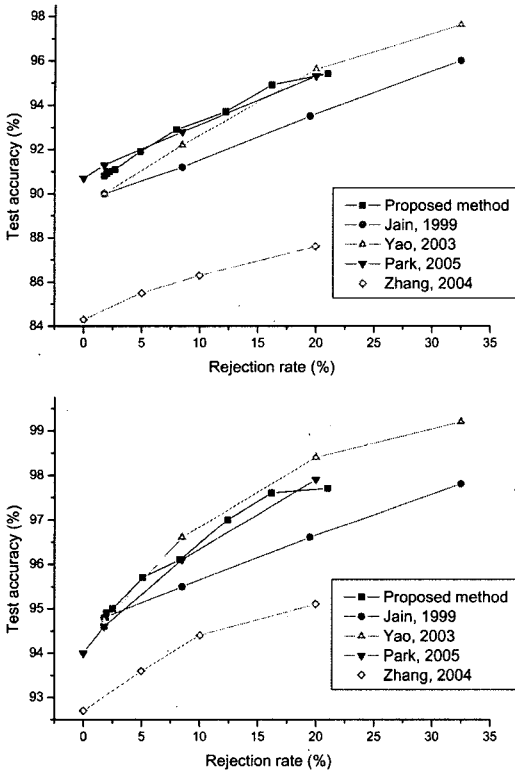


그림 9 NIST 데이터베이스 4에 대한 성능비교

의 분류 정확도를 획득하였다. 이는 단일 분류모델을 통해 얻은 분류 성능에 비해 향상된 것이며, OVA SVM이 가지는 동점의 문제도 효과적으로 해결하였다. 향후에는 다양한 다중부류 문제에 제한하는 방법을 적용하여 보다 범용적인 성능을 보이고자 한다.

참고문헌

- [1] N. Yager and A. Amin, "Fingerprint classification: A review," *Pattern Analysis and Applications*, vol. 7, no. 1, pp. 77-93, 2004.
- [2] H. Nyongesa, S. Al-khayatt, S. Mohamed and M. Mahmoud, "Fast robust fingerprint feature extraction and classification," *J. Intelligent and Robotic Systems*, vol. 40, no. 1, pp. 103-112, 2004.
- [3] R. Cappelli, A. Lumini, D. Maio and D. Maltoni, "Fingerprint classification by directional image partitioning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 402-421, 1999.
- [4] A. Senior, "A combination fingerprint classifier," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1165-1174, 2001.
- [5] K. Karu and A. Jain, "Fingerprint classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389-404, 1996.
- [6] Q. Zhang and H. Yan, "Fingerprint classification based on extraction and analysis of singularities and pseudo ridges," *Pattern Recognition*, vol. 37, no. 11, pp. 2233-2243, 2004.
- [7] K. Nagaty, "Fingerprints classification using artificial neural networks: A combined structural and statistical approach," *Neural Networks*, vol. 14, no. 9, pp. 1293-1305, 2001.
- [8] A. Jain, S Prabhakar and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348-359, 1999.
- [9] C. Park and H. Park, "Fingerprint classification using fast Fourier transform and nonlinear discriminant analysis," *Pattern Recognition*, vol. 38, no. 4, pp. 495-503, 2005.
- [10] M. Chong, T. Ngee, L. Jun and R. Gay, "Geometric framework for fingerprint image classification," *Pattern Recognition*, vol. 30, no. 9, pp. 1475-1488, 1997.
- [11] J. Chang and K. Fan, "A new model for fingerprint classification by ridge distribution sequences," *Pattern Recognition*, vol. 35, no. 6, pp. 1209-1223, 2002.
- [12] Y. Yao, G. Marcialis, M. Pontil, P. Frasconi and F. Roli, "Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines," *Pattern Recognition*, vol. 36, no. 2, pp. 397-406, 2003.
- [13] C. Watson and C. Wilson, *Fingerprint Database*, National Institute of Standard and Technology, Special Database 4, *FPDB*, April 1992.
- [14] E. Bredensteiner and K. Bennett, "Multicategory classification by support vector machines," *Computational Optimization and Applications*, vol. 12, no. 1-3, pp. 53-79, 1999.
- [15] D. Sebald and J. Bucklew, "Support vector machines and the multiple hypothesis test problem," *IEEE Trans. Signal Processing*, vol. 49, no. 11, pp. 2865-2872, 2001.
- [16] C. Hsu and C. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415-425, 2002.
- [17] L. Xu, A. Krzyzak and C. Suen, "Methods of combining multiple classifiers and their applications to handwriting recognition," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418-435, 1992.
- [18] D. Ruta and B. Gabrys, "Classifier selection for majority voting," *Information Fusion*, vol. 6, no. 1, pp. 63-81, 2005.
- [19] P. Boland, "Majority systems and the condorcet jury theorem," *Statistician*, vol. 38, pp. 181-189, 1996.

1989.

- [20] J. Kittler, M. Hatef, R. Duin and J. Matas, "On combining classifiers," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, 1998.
- [21] L. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 281-286, 2002.
- [22] K. Crammer and Y. Singer, "On the learnability and design of output codes for multiclass problems," *Machine Learning*, vol. 47, no. 2-3, pp. 201-233, 2002.
- [23] T. Gestel, J. Suykens, G. Lanckriet, A. Lambrechts, B. Moor and J. Vandewalle, "Multiclass LS-SVMs: Moderated outputs and coding-decoding schemes," *Neural Processing Letters*, vol. 15, no. 1, pp. 45-58, 2002.
- [24] S. Raudys and F. Roli, "The behavior knowledge space fusion method: Analysis of generalization error and strategies for performance improvement," *Lecture Notes in Computer Science*, vol. 2709, pp. 55-64, 2003.
- [25] L. Kuncheva, et al., "Decision templates for multiple classifier fusion: An experimental comparison," *Pattern Recognition*, vol. 34, no. 2, pp. 299-314, 2001.
- [26] L. Kuncheva, "Using measures of similarity and inclusion for multiple classifier fusion by decision templates," *Fuzzy Sets and Systems*, vol. 122, no. 3, pp. 401-407, 2001.
- [27] R. Brooks, "A robust layered control system for a mobile robot," *IEEE J. of Robotics and Automation*, vol. 2, no. 1, pp. 14-23, 1986.
- [28] J. Liu, B. Li and T. Dillon, "An improved naïve Bayesian classifier technique coupled with a novel input solution method," *IEEE Trans. Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 31, no. 2, pp. 249-256, 2001.
- [29] M. Ramoni and P. Sebastiani, "Robust Bayes classifiers," *Artificial Intelligence*, vol. 125, no. 1-2, pp. 209-226, 2001.
- [30] S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Computation*, vol. 15, no. 7, pp. 1667-1689, 2003.



민 준 기

2004년 연세대학교 기계전자공학부 정보산업전공 졸업. 2004년~2006년 연세대학교 컴퓨터과학과 석사. 2006년~현재 연세대학교 컴퓨터과학과 박사과정. 관심분야는 패턴인식, 진화게임, 영상처리, 생체인식



조 응 근

2005년 2월 연세대학교 컴퓨터과학과(학사). 2005년 3월~현재 연세대학교 컴퓨터과학과 석사과정. 관심분야는 생체인식, 패턴인식.

조 성 배

정보과학회논문지 : 소프트웨어 및 응용
제 33 권 제 3 호 참조

홍 진 혁

정보과학회논문지 : 소프트웨어 및 응용
제 33 권 제 3 호 참조