

# SVM의 다중결정템플릿을 이용한 지문분류

## (Fingerprint Classification using Multiple Decision Templates with SVM)

민준기<sup>†</sup>      홍진혁<sup>†</sup>      조성배<sup>\*\*</sup>  
 (Jun-Ki Min)      (Jin-Hyuk Hong)      (Sung-Bae Cho)

**요약** 지문분류는 대규모 자동지문식별시스템에서 지문을 카테고리별로 나누어 매칭시간을 줄이는데 유용하다. 지문을 5가지 클래스로 분류하는 헨리시스템을 기반으로 신경망이나 SVM(Support Vector Machines) 등과 같은 다양한 패턴분류 기법들이 지문분류에 널리 사용되고 있다. 특히 최근에는 높은 분류 성능을 보이는 SVM 분류기를 이용한 연구가 활발하다. 이진분류기인 SVM을 지문분류문제에 적용하기 위해서 본 논문에서는 새로운 분류기 결합모델인 다중결정템플릿(Multiple Decision Templates, MuDTs)을 제안한다. 이 방법은 클래스 구분이 모호한 지문영상들의 분류에서 단일 결합모델들의 한계를 극복하기 위해, 하나의 지문클래스로부터 서로 다른 특성을 갖는 클러스터들을 추출하여 각 클러스터에 적합한 결합모델을 생성한다. NIST Database4 데이터로부터 추출한 핑거코드에 대해 실험한 결과, 5클래스와 4클래스 분류문제에 대하여 각각 90.4%와 94.9%의 분류성능(거부율 1.8%)을 획득하였다.

**키워드** : 지문분류, SVM, 클러스터링, 결정템플릿

**Abstract** Fingerprint classification is useful in an automated fingerprint identification system (AFIS) to reduce the matching time by categorizing fingerprints. Based on Henry system that classifies fingerprints into 5 classes, various techniques such as neural networks and support vector machines (SVMs) have been widely used to classify fingerprints. Especially, SVMs of high classification performance have been actively investigated. Since the SVM is binary classifier, we propose a novel classifier-combination model, multiple decision templates (MuDTs), to classify fingerprints. The method extracts several clusters of different characteristics from samples of a class and constructs a suitable combination model to overcome the restriction of the single model, which may be subject to the ambiguous images. With the experimental results of the proposed on the FingerCodes extracted from NIST Database4 for the five-class and four-class problems, we have achieved a classification accuracy of 90.4% and 94.9% with 1.8% rejection, respectively.

**Key words** : Fingerprint classification, Support vector machines, Clustering, Decision templates

### 1. 서론

지문분류는 전역적 특징을 이용하여 지문을 미리 정해진 카테고리로 분류하는 기술이다. 지문분류를 수행하는 다양한 기준들이 있지만 많은 연구들이 1900년에

Henry가 제안한 기준인 헨리시스템을 따른다[1]. 이것은 지문을 특이점(중심점과 삼각주)의 위치와 개수에 따라 그림 1과 같이 Whorl(W), Right loop(R), Left loop(L), Arch(A), Tented arch(T)의 다섯 가지 클래스로 분류하는데, 보통 대규모 자동지문식별시스템에서 식별시간 감소를 위해 지문매칭단계에 앞서 수행된다. 그림 2는 자동지문식별시스템의 처리과정을 보여준다.

지문분류는 사용되는 특징에 따라 크게 특이점기반 방법과 방향성기반 방법으로 나뉜다[2]. 특이점기반 지문분류는 Poincare와 같은 특이점추출 알고리즘을 이용하여 지문영상으로부터 추출한 중심점과 삼각주의 개수와 위상관계로 분류를 수행하는 방법으로서, 오래전부터 지문분류에 사용되어왔다[3,4]. 하지만 지문영상의 품질

· 이 연구는 한국과학재단(KOSEF)을 통해 생체인식연구센터(BERC)에서 지원받았음

· FingerCode데이터를 제공해준 A.K. Jain교수와 S. Prabhakar박사에게 감사를 포함

<sup>†</sup> 학생회원 : 연세대학교 컴퓨터과학과, 생체인식연구센터

loomlike@sclab.yonsei.ac.kr

hjinh@sclab.yonsei.ac.kr

<sup>\*\*</sup> 종신회원 : 연세대학교 컴퓨터과학과 교수

sbcho@cs.yonsei.ac.kr

논문접수 : 2005년 5월 18일

심사완료 : 2005년 8월 29일

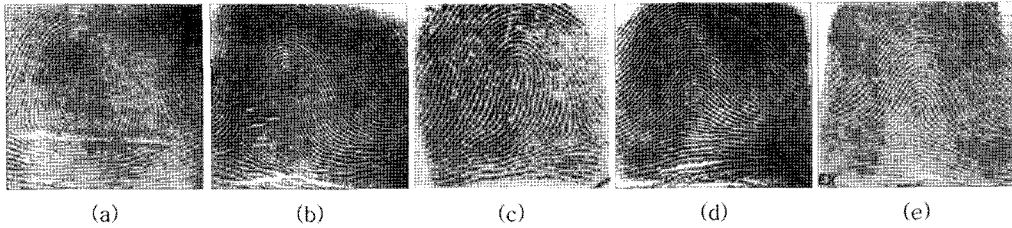


그림 1 헨리가 제안한 지문의 5가지 클래스. (a) Whorl, (b) Right loop, (c) Left loop, (d) Arch, (e) Tented arch

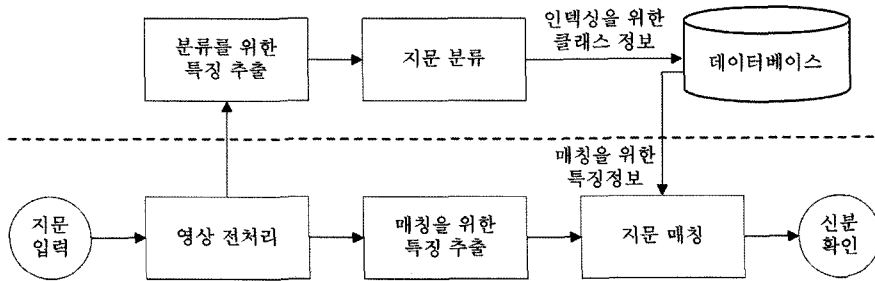


그림 2 자동지문식별시스템

이 안 좋은 경우 특이점을 정확히 추출하기 어려워 분류 성능이 저하된다[5].

방향성기반 지문분류는 지문의 용선 방향성으로부터 특징을 추출해 분류를 수행한다. 주로 사용되는 특징추출 방법으로 필터기반 방법, 구조기반 방법, 구문기반 방법 등이 있다. 필터기반 특징 추출은 지문영상에 가보필터와 같은 대역여과기를 이용하여 용선의 방향성정보를 추출하는 방법으로, 잡음에 둔감하여 영상품질이 안 좋은 데이터로부터 특징을 정확히 추출할 수 있지만 시간이 오래 걸린다. 이를 이용한 대표적인 데이터로는 Jain의 핑거코드(FingerCode)가 있다[6]. 구조기반 특징추출은 지문용선을 동일한 방향성을 갖는 영역들로 묶고, 이들 간의 관계를 구조적으로 나타내는 방법이다[7,8]. 각 클래스를 대표하는 구조를 학습 또는 생성한 뒤 입력된 지문 구조와 비교하여 분류를 수행한다. 구문기반 분류방법은 용선의 패턴을 단말기호로 표현하여 각 클래스별로 문법을 생성하고 패턴을 파싱하여 분류를 수행한다[9,10].

지문분류에는 신경망, KNN(K-Nearest Neighbor), SVM 등의 분류기들이 많이 사용되는데, 분류성능을 높이기 위하여 다수의 분류기들을 결합하는 방법들이 함께 연구되고 있다[2,6]. 특히 이진분류기인 SVM은 다른 분류기들에 비해 좋은 분류 성능을 보이는데, 지문분류와 같은 다중부류분류 문제를 풀기 위해서는 논리적 조합으로 개별 분류기를 생성한 뒤 이를 투표기반(Majority voting) 방법이나 ECC(Error Correcting Code) 등을 이용하여 결합하는 기술이 필요하다[7,11]. 다양한

결합방법들 중, 최근에는 분류기 출력 값들의 평균을 이용하여 각 클래스의 템플릿을 만들고 입력 값과의 유사도를 계산하여 가장 비슷한 클래스로 분류를 수행하는 단일결정템플릿(Decision Templates, DTs)이 좋은 성능을 보이고 있다[12]. 그러나 지문데이터는 그림 3과 같이 하나이상의 클래스 특징을 복합적으로 갖는 모호한 영상이 많기 때문에 클래스를 하나의 템플릿으로 축약시키는 단일모델분류방법은 한계가 있다[8,13]. 본 논문에서는 SVM결합을 위해 클래스 내의 다양한 특징들을 세분화하여 모델링하는 다중결정템플릿(Multiple Decision Templates)을 제안하고 이를 이용하여 지문분류를 수행한다.

제안하는 지문분류과정은 다음과 같다. 학습을 위한 지문데이터로부터 one-vs-all조합 SVM분류기들의 출력 값을 얻어 행렬형식의 결정프로파일을 생성한다. 각 클래스별 데이터의 결정프로파일을 SOM알고리즘으로

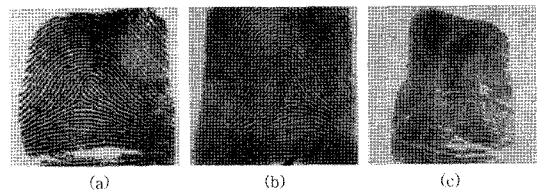


그림 3 분류가 모호한 지문영상. (a) Tented arch와 유사한 Right loop, (b) Tented arch와 유사한 Left loop, (c) Tented arch와 유사한 Arch

클러스터링하여 클래스 내에 다양한 특징들을 클러스터로 묶은 뒤, 각 클러스터의 결정템플릿을 생성한다. 분류단계에서는 입력샘플의 SVM출력 값인 결정프로파일을 클러스터들의 결정템플릿과 비교하고, 가장 유사한 클러스터의 클래스를 레이블로 결정한다. 본 논문에서는 제안하는 방법의 성능을 확인하기 위해서 NIST Database4로부터 추출된 핑거코드[6]데이터를 사용하여 실험하였다.

## 2. 배경연구

### 2.1 핑거코드

핑거코드는 Jain이 1999년에 제안한 필터기반 특징추출 방법으로, 이를 이용하여 NIST Database4 지문영상으로부터 특징벡터를 추출하여 핑거코드데이터를 생성한다[6]. 이 방법은 중심점을 기준으로 48개의 섹터로 지문영상을 나눈 뒤 0°, 45°, 90°, 135°의 방향성을 갖는 4개의 가보필터를 적용시킨다. 가보필터는 대역역과파이기 때문에 그림 4와 같이 동일한 방향성의 융선 영상은 선명해지고, 그렇지 않은 영상은 흐려진다. 변환된 영상은 각 섹터마다 픽셀 값의 표준편차가 계산되어 192차원의 특징벡터를 생성한다. Jain은 핑거코드를 K-NN과 신경망으로 2단계 분류를 수행하여 90%의 분류성능을 보였다[6]. Yao도 핑거코드데이터를 이용하여 지문분류를 수행하였는데, 지문의 구조적 특징을 추가로 이용하여 SVM의 ECC결합으로 90%의 분류성능을 보였다[7]. 핑거코드는 지문분류 외에도 지문매칭에 사용되기도 하였다[14].

### 2.2 Support Vector Machines

SVM은 패턴인식 분야에서 이진분류 문제를 해결하는 분류기술로서, 학습 데이터를 비선형 매핑함수  $\Phi(x)$ 를 이용하여 입력 샘플 공간  $X$ 로부터 선형 초평면이

만들어지는 고차원 특징 공간  $F$ 로 매핑하고 인식에러가 최소가 되게 하는 최적 초평면을 찾는다[11,15-17].

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

$n$ 개의 학습데이터 중에서  $i$ 번째 샘플( $i=1, \dots, n$ )인  $x_i$ 의 클래스 레이블을  $c_i \in \{1, -1\}$ 라고 할 때, 최적 초평면에 대한 결정 함수의 식은 다음과 같다.

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

식 (2)에서  $\alpha_i$ 는  $x_i$ 가 초평면을 구성하는 샘플인 경우 0이 아닌 값을 갖고, 그렇지 않을 경우에는 0의 값을 갖는 계수이다. 커널함수  $K(x, x_i)$ 는 비선형 매핑함수의 내적계산을 함수 형식으로 치환하여 계산을 간단하게 해주며, 표 1의 선형(Linear) 커널, 다항식(Polynomial) 커널, 가우시안(Gaussian) 커널, 시그모이드(Sigmoid) 커널이 주로 사용된다.

### 2.3 다중분류를 위한 SVM조합전략

SVM은 이진분류기이기 때문에 다중분류문제를 이진분류문제로 분해해서 풀어야한다. 이를 위한 대표적인 조합전략으로 one-vs-all, pairwise, complete-code가 있다[11,15].

1) **One-vs-all**: 가장 간단한 조합전략으로, 클래스의 수가  $M$ 개일 때 하나의 클래스와 나머지 클래스들을 분류하는  $M$ 개의 SVM을 이용한다.  $j$ 번째( $j=1, \dots, M$ ) SVM의 결정함수  $f_j(x_i)$ 는 식 (2)의 클래스  $c_i$ 를 다음 조건을 만족시키는 임시클래스  $t_i$ 로 치환한 식이 된다.

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

2) **Pairwise**: 다중분류문제의 클래스들을 두 개씩 짝지어 이진분류문제로 분해하는 전략이다.  $M$ 클래스 문

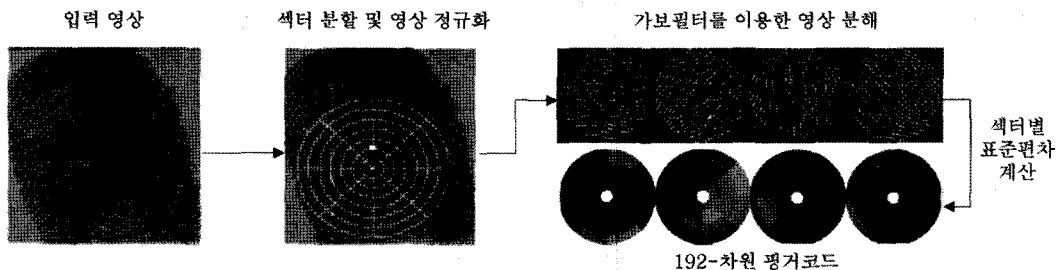


그림 4 핑거코드 특징벡터의 생성과정[6]

표 1 SVM의 커널함수

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

제의 경우  ${}_M C_2 = M(M-1)/2$  개의 분류기를 사용한다. 전체 학습 데이터 중에서 분류기를 구성하는 두 개의 클래스 레이블의 데이터만 학습에 사용하기 때문에 다른 조합전략들에 비해 학습속도가 빠르다.  $j, k \in \{1, \dots, M\}$ ,  $j \neq k$ 일 때, 결정함수  $f_{j,k}(x_i)$ 는 식 (2)의 클래스  $c_j$ 를 다음 조건에 따라 임시클래스  $t_i$ 로 치환한 식이 된다.

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

**3) Complete-code:** 전체 클래스를 두 그룹으로 나누는 모든 조합을 고려한 전략이다. 이는 one-vs-all 뿐만 아니라 two-vs-all 등 모든 이진조합을 포함한다. 전체 클래스를 두 그룹  $\{j\}, \{k\}$ 로 나뉘었을 때, 결정함수  $f_{j,k}(x_i)$ 는 식 (2)의 클래스  $c_j$ 를 식 (5)의 조건에 따라 임시클래스  $t_i$ 로 치환한 식이 된다.

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

## 2.4 분류기 결합방법

이진분류기를 이용하여 다중분류문제를 해결하거나 분류성능을 높이기 위해서 여러 개의 분류기를 결합하는 방법이 사용된다[18]. 이를 위한 대표적인 방법으로 Winner-takes-all, ECC, BKS(Behavior Knowledge Space), 단일결정템플릿 등이 있다.

**1) Winner-takes-all:** 투표기반 방법(Majority voting)은 전체 클래스 중 분류기들로부터 가장 많은 표를 얻은 클래스로 샘플의 레이블을 결정하는데, 여러 클래스가 동점이 되는 경우에는 정확히 분류할 수 없다. Winner-takes-all은 이를 보완한 방법으로, 분류기 출력 값의 합이 가장 큰 클래스를 샘플의 클래스 레이블로 결정한다.  $M$ 클래스 문제에 대해  $L$ 개의 분류기를 사용한다고 할 때, 샘플의 클래스 레이블  $c$ 는 다음 식과 같이 결정된다.

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

식 (6)에서  $\text{ind}_{i,j}(x)$ 는 클래스 레이블  $i$ 가  $j$ 번째 분류기에서 양의 값을 출력하는 클래스인 경우 1, 음의 값을 출력하는 클래스인 경우 -1이 되며, 그 외에는 0이 된다.

**2) ECC:** 이 방법은 클래스와 분류기의 수를 각각  $M$ 과  $L$ 이라 할 때, 클래스  $i$ 가  $j$ 번째 분류기에서 양의 값을 출력하는 클래스인 경우 1, 음의 값을 출력하는 클래스인 경우 -1, 학습에 사용되지 않은 경우 0으로 하여  $M \times L$ 의 coding matrix  $E \in \{-1, 0, 1\}^{M \times L}$ 을 만든다. 평가 시 입력샘플의 분류기출력 값들과  $E$ 의 각 행과의 유사도를 비교하여 가장 비슷한 행의 레이블을 클래스

로 결정한다. 식 (7)은 해밍거리(Hamming distance)를 사용했을 때의 ECC 클래스 결정 식을 나타낸다[15].

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

**3) BKS:** 단일 분류기가  $M$ 클래스를 분류한다고 하면,  $L$ 개의 분류기는  $K = M^L$ 개의 출력 패턴조합을 생성한다. BKS 방법은 이를 BKS-테이블  $T \in \{0, 1\}^{K \times L}$ 로 만들어 학습 데이터에 대한 모든 출력패턴을 클래스 레이블과 함께  $T$ 에 저장한다. 만약 서로 다른 클래스에 속하는 샘플들이 동일한 분류기출력패턴을 나타낼 경우, 그 패턴을 가장 많이 나타내는 클래스를 패턴의 레이블로 결정한다. 평가 시, 입력 샘플에 대한 분류기들의 출력패턴을  $T$ 에서 찾아 해당 레이블로 분류하는데, 패턴이  $T$ 에 없는 경우에 분류가 어렵다[19].

**4) 단일결정템플릿:** 이 방법은 학습데이터로부터 얻은 분류기 출력 값을 클래스별로 평균값을 계산하여 결정템플릿을 생성한다[12].  $M$ 클래스 문제에 대해  $L$ 개의 분류기를 사용한다고 하면,  $i$ 번째 학습데이터  $x_i$  ( $i = 1, \dots, n$ )의 분류기 출력 값들을 이용하여 식 (8)의 결정프로파일(decision profile)  $DP(x_i)$ 를 만든다. 식 (8)의  $y$ 행,  $z$ 열값인  $d_{y,z}(x_i)$ 는  $y$ 번째 분류기의 클래스  $z$ 에 대한 출력 값을 의미한다. 결정프로파일이 생성되면 이를 이용하여 식 (9)와 같이 클래스  $c$ 에 대한 결정템플릿  $DT_c$ 를 생성한다. 식 (9)의  $\text{ind}_c(x_i)$ 는  $x_i$ 의 클래스가  $c$ 인 경우 1이고 그 외에는 0의 값을 갖는다.

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

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

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

분류 시에는 입력 샘플의 결정프로파일과 각 클래스의 결정템플릿간의 유사도를 계산하여 가장 비슷한 클래스를 샘플의 클래스 레이블로 결정한다. Kuncheva는 [20]의 연구를 통해 단일결정템플릿 방법이 투표기반, BKS등의 기존 결합방법들에 비해 좋은 분류성능을 보임을 입증하였다.

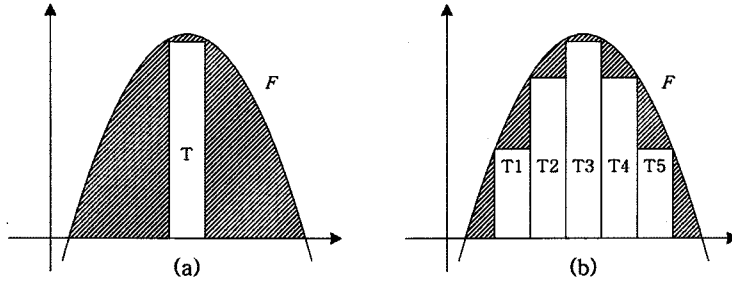


그림 5 한 클래스의 특징공간에서의 (a)단일결정템플릿과 (b)다중결정템플릿 오류영역(빚금)

### 3. 다중결정템플릿

#### 3.1 다중템플릿 모형

분류문제에서 한 클래스에 대한 특징공간을  $F$ 라고 하였을 때, 단일결정템플릿방법과 같이 클래스별로 하나의 템플릿을 갖는 경우 오류가 발생할 가능성이 있는 샘플 영역은 그림 5의 (a)와 같다. 클래스를 대표하는 템플릿을 특징공간에 적합하게 여러 개를 가질 경우 그림 5의 (b)와 같이 오류발생영역을 줄일 수 있다. 본 논문에서는 이를 위해 학습데이터의 분류기 출력값들을 클러스터링 하여 한 클래스를 여러 개의 템플릿으로 모델링하여 분류하는 다중결정템플릿 방법을 제안한다.

#### 3.2 다중결정템플릿 생성

다중결정템플릿을 생성하기 위해서 학습데이터를 이용하여 5개의 one-vs-all SVM (Whorl-vs-all, Right loop-vs-all, Left loop-vs-all, Arch-vs-all, Tented arch-vs-all)을 학습시키고, 이들로부터 출력 값을 얻어 결정프로파일을 구성한 뒤, 각 클래스별로 SOM알고리즘을 이용하여 클러스터링한다. SOM은 입력벡터들의 유사성에 따라 그룹을 나누는 대표적인 클러스터링 알고리즘으로,  $N \times N$ 의 클러스터 맵을 구성하면 입력벡터  $DP(x)$ 는 맵  $(i,j)$ 의 가중치가  $w_{i,j}$ 일 때 유클리드 거리식에 의해 식 (10)과 같이  $(k,l)$ 클러스터로 결정된다[21].

$$\|DP(x) - w_{k,l}\| = \min_{i,j=1,\dots,N} \{\|DP(x) - w_{i,j}\|\} \quad (10)$$

각 클러스터의 결정프로파일들로부터 식 (11)을 통해 클러스터별 결정템플릿을 생성한다.

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

$$dt_c^{k,l}(y,z) = \frac{\sum_{i=1}^n ind_c^{k,l}(x_i) d_{y,z}(x_i)}{\sum_{i=1}^n ind_c^{k,l}(x_i)} \quad (11)$$

식 (11)에서  $DT_c^{k,l}$ 은 클래스  $c$ 의  $(k,l)$ 클러스터의 결정템플릿을 의미한다. SVM은 이진분류기이기 때문에 식 (8)과 식 (11)의 결정프로파일/템플릿 행렬을 그림 6과 같이 양수와 음수를 이용하여 하나의 열로 구성할 수 있다. Whorl클래스 학습데이터의 결정프로파일들을  $4 \times 4$ 로 클러스터링 하였을 때 16개의 클러스터 결정템플릿들이 생성되고 이들은 Whorl클래스에 대한 다중결정템플릿을 구성한다. 나머지 클래스들에 대해서도 동일한 과정을 통해 다중결정템플릿을 생성한다. 알고리즘 1은 다중결정템플릿의 생성과정을 나타낸다.

알고리즘 1. decisionTemplate(data), 다중결정템플릿 생성

**Input:** 학습데이터(data[i].DP[])는  $i$  번째 학습데이터의 결정프로파일을 나타냄.  
**Output:** 결정템플릿(DT[m][k])는 클래스  $m$ 의 클러스터  $k$ 의 결정템플릿.  
**Notation:**  
 $M$  = 클래스 수  
 $K$  = 클러스터 수  
 $N$  = 학습데이터 수  
 $SOM(m, feature)$  = 클래스레이블  $m$ 과 샘플의 특징벡터  $feature$ 를 입력받아 SOM알고리즘으로 클러스터링하여 클러스터 레이블을 출력하는 함수

```

decisionTemplate (data) {
    DT[M][K];
    for (class=0; class<M; class++)
        for (i=0; i<N; i++)
            if (data[i].class==class)
                data[i].cluster = SOM(class, data[i].DP);
    for (class=0; class<M; class++)
        for (cluster=0; cluster<K; cluster++) {
            sum = 0;
            num = 0;
            for (i=0; i<N; i++)
                if ((data[i].class==class) and (data[i].cluster==cluster)) {
                    sum += data[i].DP;
                    num++;
                }
            DT[class][cluster] = sum/num;
        }
    return DT;
}
    
```

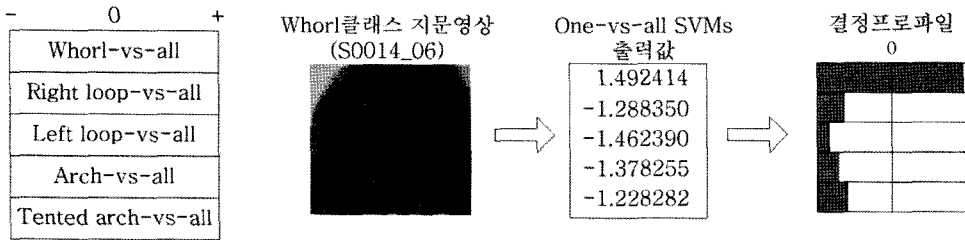


그림 6 One-vs-all조합 SVM의 결정프로파일과 지문영상에 대한 생성 예

3.3 다중결정템플릿을 이용한 분류

분류과정에서는 그림 7과 같이 입력샘플의 결정프로파일과 결정을템플릿과의 거리를 계산한다. 샘플과 템플릿의 거리가 가까운 것이 모델과 유사하다는 것을 나타내기 때문에 식 (12)와 같이 거리가 가장 가까운 클러스터의 결정템플릿을 찾고, 그 클러스터가 속해있는 클래스로 샘플을 분류한다. Kuncheva는 결정프로파일과 결정템플릿간의 유사도 계산을 위해 11가지 거리계산방법들을 사용하였다[20]. 본 논문에서는 이들 중 간단하면서도 좋은 성능을 보인 해밍거리(식 (13))와 유클리드거리(식 (14))방법을 사용한다. 알고리즘 2는 다중결정템플릿의 분류과정을 나타낸다.

$$dst_c^{i,j}(x) = \sum_{y=1}^L \sum_{z=1}^M |dt_c^{i,j}(y,z) - d_{y,z}(x)| \tag{13}$$

$$dst_c^{i,j}(x) = \sqrt{\sum_{y=1}^L \sum_{z=1}^M (dt_c^{i,j}(y,z) - d_{y,z}(x))^2} \tag{14}$$

4. 실험 및 결과

4.1 실험환경

본 논문에서는 제안하는 방법의 성능을 분석하기 위해 NIST Database4를 사용하였다. 이 데이터는 전문가에 의해 5가지 클래스(Whorl, Right loop, Left loop, Arch, Tented arch)로 분류된 지문이 클래스별로 400개씩 2회(F와 S) 등록되어 총 4000장의 지문영상으로

$$\min_{c=1,\dots,M} (\min_{i,j=1,\dots,n} dst_c^{i,j}(x)) \tag{12}$$

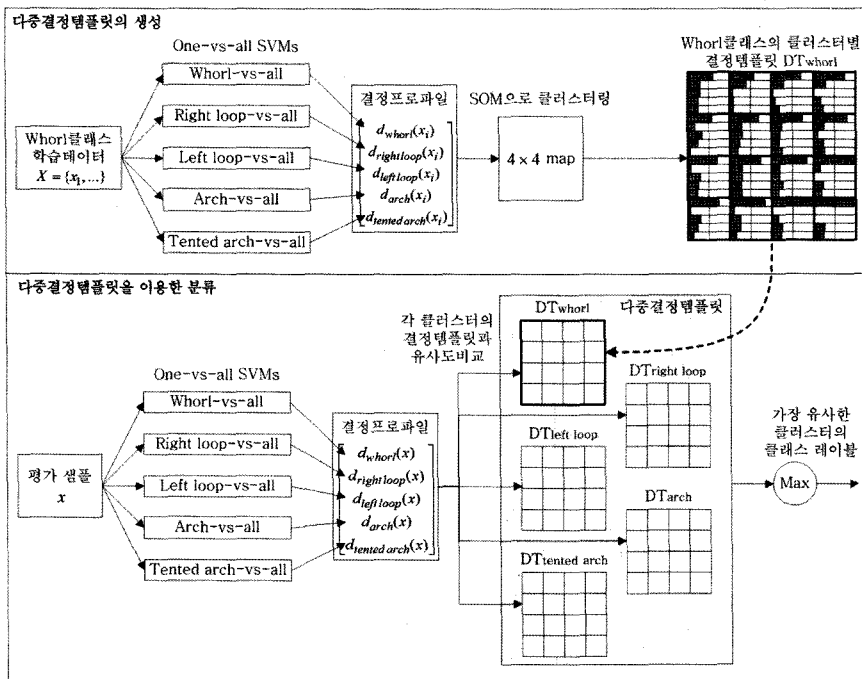


그림 7 4x4클러스터의 다중결정템플릿 생성과정(상)과 이를 이용한 분류과정(하)

**알고리즘 2.** classification(DT, DP), 다중결정템플릿을 이용한 분류

**Input:** 다중결정템플릿 DT와 평가샘플의 결정프로파일 DP  
**Output:** 평가샘플의 클래스 레이블 label(분류결과)  
**Notation:**  
 $M$  = 클래스 수  
 $K$  = 클러스터 수  
 similarity(DT[m][k], DP) = 클래스  $m$ 의 클러스터  $k$ 의 결정템플릿 DT[m][k]와 결정프로파일 DP간의 유사도를 계산하는 함수

```

classification(DT, DP){
    similar[M][K];
    for(class=0;class<M;class++)
        for(cluster=0;cluster<K;cluster++)
            similar[class][cluster] = similarity(DT[class][cluster], DP);
    for(class=0;class<M;class++)
        for(cluster=0;cluster<K;cluster++)
            if(similar[class][cluster] is max)
                label = class;
    return label;
}
    
```

구성되어 있으며, 각 지문영상은 잉크기반 지문데이터를 512×512 해상도로 스캔한 것이다. 본 논문에서는 실험을 위해 첫 번째 등록된 2000개의 지문(F0001~F2000)을 학습데이터로, 두 번째 등록된 2000개의 지문(S0001~S2000)을 평가 데이터로 사용하였다. 이들 중 350쌍(17.5%)의 지문은 하나 이상의 클래스 특성을 갖기 때문에 두 개의 클래스레이블을 갖도록 교차참조(cross reference)되어있으며(표 2), 실험에서는 이 데이터를 사용한 다른 연구들과 마찬가지로 학습 시에는 첫 번째 클래스 레이블을, 평가 시에는 두 가지 클래스를 모두 사용하였다[6,13].

표 2 NIST Database4의 교차참조에 따른 지문클래스 분포(행: 본 클래스, 열: 부 클래스)

	W	R	L	A	T
W	792	6	2	0	0
R	4	746	0	0	50
L	2	0	756	0	42
A	0	2	0	760	38
T	0	238	166	150	246

분류를 위한 특징은 Jain의 핑거코드 데이터를 +1에서 -1 사이의 실수 값으로 정규화하여 사용하였는데, 이 데이터는 추출과정에서 거부된 지문영상이 있기 때문에 학습데이터와 평가 데이터에 각각 1.4%와 1.8%의 거부

율이 적용되었다. SVM분류기는 LIBSVM<sup>1)</sup> 소스를 사용하였으며, 다양한 실험을 위해 one-vs-all(OVA), pairwise(PW), complete-code(COM)의 3가지 조합을 각각 다중결정템플릿으로 결합하였다. SVM의 커널함수는 많은 연구들과 마찬가지로 가우시안커널을 사용하였다[17]. 표 3은 SVM의 커널성능 확인을 위해 수행한 실험 결과로, one-vs-all조합을 winner-takes-all로 결합한 결과 가우시안커널이 다른 커널들보다 높은 성능을 보임을 확인하였다. 가우시안커널의 시그마값은 실험을 통해 0.0625를 선택하였다.

**4.2 클러스터 맵 크기 선택**

본 논문에서는 클러스터의 개수를 1×1부터 9×9까지 증가시키며 각각 10회씩 실험하여 적합한 맵을 선택하였다(1×1은 단일결정템플릿). 그림 8은 클러스터 개수별 오분류율을 나타내는 것으로, 모든 조합방법에서 단일결정템플릿보다 다중결정템플릿결합이 좋은 분류성능을 보였다. 그림에서 보여지는바와 같이 pairwise조합을 결합한 경우 성능향상이 가장 컸으며, one-vs-all조합을 4×4의 다중결정템플릿으로 결합하였을 때 가장 좋은 분류성능을 보였다. 결정프로파일을 클러스터링하여 결정템플릿을 만든 결과가 단일결정템플릿을 사용하였을 때보다 클래스의 내부 특징을 세부적으로 모델링하였는가를 분석하기 위해 클러스터별 CV(Coefficient of Variance)값을 측정하였다. CV값은 상호분산정도를 측정하기 위해 많이 사용되는 통계적 방법으로, 데이터

1) <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

표 3 one-vs-all SVM의 커널별 분류성능(winner-takes-all 방법사용)

커널 함수	선형 커널	다항식 커널( $d=3$ )	가우시안 커널( $\sigma^2=0.0625$ )	시그모이드 커널
분류율(%)	82.2	87.4	90.1	83.1

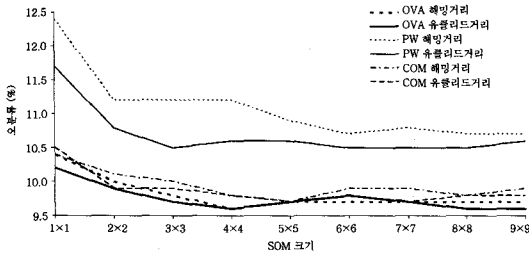


그림 8 SVM의 조합방법 별 클러스터 개수에 따른 다중결정템플릿 결합결과(분류율)

의 집중경향을 나타낸다.  $\sigma$ 와  $\mu$ 를 각각 데이터의 표준편차와 평균이라고 하면 CV값은 식 (15)와 같이 계산되며, 이 값이 작을수록 데이터가 집중되어 있는 것을 의미한다.

$$CV = \frac{\sigma}{\mu} \times 100\% \quad (15)$$

그림 9는 4×4로 클러스터링한 학습데이터에 대한 CV값을 나타내는데, 원본 학습데이터는 각 클래스별로 식 (15)를 통해 CV값을 계산하였고 클러스터링한 학습데이터는 클래스별 CV값을 계산하기 위해 클래스내의 각 클러스터별 CV값을 구한 뒤 이를 평균하였다. 클러스터링을 하지 않은 원본데이터의 경우 CV값이 높은 것을 통해 동일한 클래스 데이터내의 특징 편차가 큰 것을 확인할 수 있었고, 클러스터링기법을 이용하여 효과적으로 데이터들을 특징에 따라 세분화하여 CV값이 감소되는 것을 확인하였다.

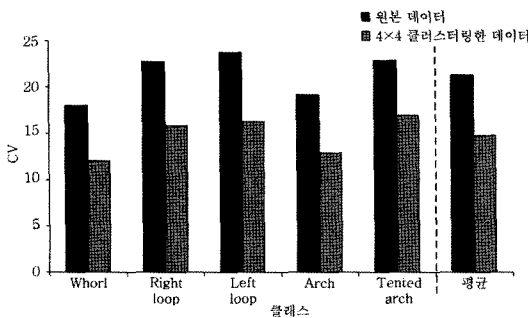


그림 9 원본 학습데이터와 이를 4×4로 클러스터링한 후의 각 클래스별 CV값

### 4.3 단일결정템플릿과 다중결정템플릿의 분류결과

앞의 실험을 통해 선택한 4×4의 다중결정템플릿을 이용하여 분류를 수행하였다. 1.8%의 거부율에서 5클래스 분류는 90.4%, 4클래스는 94.9%의 분류성능을 보였다. 표 4, 5의 혼동행렬은 one-vs-all SVM을 단일결정템플릿과 다중결정템플릿으로 결합한 분류결과를 나타

표 4 단일결정템플릿의 혼동행렬

	W	R	L	A	T
W	380	6	8	0	0
R	7	357	1	6	21
L	7	0	363	13	13
A	1	2	1	347	60
T	1	8	9	37	316

표 5 다중결정템플릿의 혼동행렬

	W	R	L	A	T
W	380	6	7	0	1
R	9	369	1	5	17
L	8	0	366	14	10
A	1	4	1	356	50
T	1	10	6	38	304

낸다. 혼동행렬의 각 행의 합은 4.1절에서 설명한 교차참조된 데이터 때문에 동일하지 않을 수 있다. 표 2에 나타난 것과 같이 교차참조가 많은 Tented arch(T)클래스의 경우 본 클래스가 아닌 부 클래스로 분류한 샘플 때문에 다중결정템플릿이 단일결정템플릿보다 적은 분류수를 나타내었으나, Tented arch 클래스의 오분류 샘플의 수는 동일하여 분류성능의 저하는 없음을 확인할 수 있다. 그림 10은 다중결정템플릿이 단일결정템플릿에 비해 클래스가 모호한 지문영상을 정확하게 모델링해주는 예를 보여준다.

### 4.4 여러 결합방법들과의 비교실험

다중결정템플릿과 기존의 결합방법들간의 성능 비교를 위해 one-vs-all, pairwise, complete-code SVM에 대해 winner-takes-all, ECC, BKS, 단일결정템플릿으로 결합실험을 하였다. 표 6은 각 결합방법에 대한 분류율을 나타낸다. 결정템플릿과 ECC는 해밍거리(HM)와 유클리드거리(EU)를 샘플의 유사도 계산에 사용하였다. BKS는 BKS테이블에 없는 분류기 출력패턴이 발생할 경우 winner-takes-all을 이용하여 분류하였다. 표 6과 같이 다중결정템플릿(유클리드거리)을 이용한 결합방법이 89.5~90.4%로 가장 좋은 분류결과를 보였다. 수행속도에 있어서는 다중결정템플릿의 학습단계에 클러스터링하는 과정이 추가되지만 SOM알고리즘 자체도 간단하고, one-vs-all의 경우 5차원 벡터를 클러스터링하기 때문에 전체 분류시간의 차이는 거의 없었다. 이를 실험으로 측정해본 결과 2000개의 학습데이터를 5개의 클래스별로 각각 4×4로 클러스터링 하는데 걸리는 총 시간이 약 60ms 정도로 분류기 학습시간에 비하면 무시할 수 있다. 표 7은 NIST Database4 데이터를 이용한 분류연구들로, 다중결정템플릿이 다른 분류기법들보다 높은 분류성능을 보임을 나타낸다.



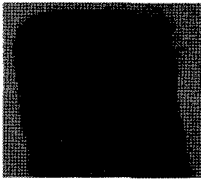
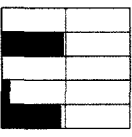
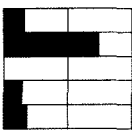
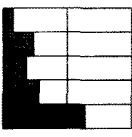
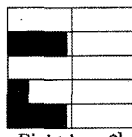
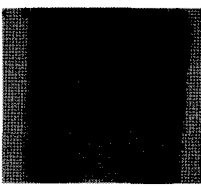
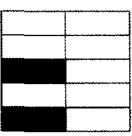
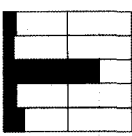
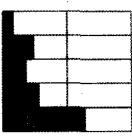
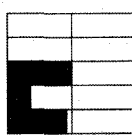

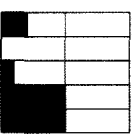
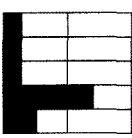
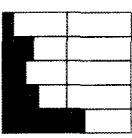
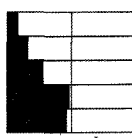
지문영상	결정프로파일	단일결정템플릿		다중결정템플릿
 Right loop (S1257_04)		 Right loop	 Tented Arch	 Right loop의 (4,1)클러스터
	유사도(거리)	1.91	1.41	0.94
 Left loop (S0244_09)		 Left loop	 Tented Arch	 Left loop의 (4,3)클러스터
	유사도(거리)	1.94	1.68	1.12
 Arch (S1998_07)		 Arch	 Tented Arch	 Arch의 (4,1)클러스터
	유사도(거리)	1.65	1.52	0.85

그림 10 모호한 클래스의 지문영상 분류(다중결정템플릿은 단일결정템플릿이 Tented arch등으로 오분류하는 지문을 정확히 분류한다)

표 6 SVM분류기에 대한 여러 결합방법의 분류율 비교(%)

	one-vs-all	pairwise	complete-code
Winner-takes-all	90.1	87.7	90.0
ECC:HM,EU	90.1	88.6	90.0
BKS	88.8	89.4	89.3
단일결정템플릿:HM	89.6	87.6	89.6
단일결정템플릿:EU	89.8	88.3	89.5
다중결정템플릿:HM	90.3	89.3	90.3
다중결정템플릿:EU	90.4	89.5	90.3

표 7 NIST Database4 데이터를 이용해 지문분류 실험을 한 기존 연구들과의 비교

저자 및 년도	클래스	오류율%	데이터 (거부율)	분류방법
Karu 1996 [5]	5	14.6	4000개 모두로 평가	특이점의 위상적 관계에 따른 규칙기반 분류
	4	8.9		
Jain 1999 [6]	5	10.0	F0001~F1000, S0001~S1000로 학습, F1001~F2000, S1001~S2000로 평가(1.8%)	필터기반 특징(평가코드)의 KNN+NN 2단계 분류
	4	5.2		
Senior 2001 [2]	4	11.4	Jain과 동일	음선의 패턴을 HMM과 DTree를 이용해 분류
Chang 2002 [10]	7	6.6	2000개(F0001~F2000)로 평가(5.1%)	음선의 패턴으로부터 추출한 문법구조(FSM)에 따라 분류
	5	5.2		
Yao 2002 [7]	5	10.0	Jain과 동일	구조적 특징과 필터기반 특징(평가코드)을 ECC SVM으로 분류
	4	5.3		
Q. Zhang 2004 [22]	5	15.7	4000개 모두로 평가(0.45%)	특이점과 의사음선을 분석하여 규칙기반으로 분류
	4	7.3		
제안하는 방법	5	9.6	앞의 2000개 학습, 나머지 2000개로 평가(1.8%)	SVM의 다중결정템플릿을 이용한 분류
	4	5.1		

## 5. 결론

본 논문에서는 여러 클래스의 특징을 동시에 가지고 있는 지문영상을 효과적으로 분류하기 위한 분류기 결합방법으로 다중결정템플릿을 제안하였다. 다중결정템플릿은 학습데이터의 SVM출력 값들을 SOM 알고리즘을 이용하여 클래스별로 클러스터링하여 각 클래스템플릿을 다양하게 생성한다. NIST Database4로 핑거코드 특징을 사용하여 실험한 결과, 5클래스 분류에 대하여 90.4%의 분류율(1.8% 거부)로, 투표방법이나 ECC, BKS, 단일결정템플릿 등의 기존 결합방법들보다 높은 분류성능을 보였다. 특히 클래스의 특징을 클러스터별로 세분화하여 결정템플릿을 생성함으로써 기존의 방법으로는 분류가 어려웠던 여러 클래스의 특징이 중복되어있는 모호한 지문영상들을 효과적으로 분류하였다. 추후에는 다중 템플릿의 이점을 극대화하기 위한 효과적인 분류기 조합방법과 데이터 개수에 따른 적절한 클러스터 맵 크기결정에 대한 연구를 수행할 계획이다.

## 참고 문헌

- [1] E. R. Henry, *Classification and Uses of Fingerprint Prints*, London: Routledge, 1900.
- [2] A. Senior, "A combination fingerprint classifier," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 10, pp. 1165-1174, 2001.
- [3] H. C. Lee and R. E. Gaensslen, *Advances in Fingerprint Technology*. Elsevier, 1991.
- [4] M. Kawagoe and A. Tojo, "Fingerprint pattern classification," *Pattern Recognition*, vol. 17, no. 3, pp. 295-303, 1984.
- [5] K. Karu and A. K. Jain, "Fingerprint classification," *Pattern Recognition*, vol. 29, no. 3, pp. 389-404, 1996.
- [6] A. K. Jain, et al., "A multichannel approach to fingerprint classification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348-359, 1999.
- [7] Y. Yao, et al., "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.
- [8] R. Cappelli, et al., "Fingerprint classification by directional image partitioning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 402-421, 1999.
- [9] K. Rao and K. Black, "Type classification of fingerprints: A syntactic approach," *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 2, no. 3, pp. 223-231, 1980.
- [10] J.-H. Chang and K.-C. Fan, "A new model for fingerprint classification by ridge distribution sequences," *Pattern Recognition*, vol. 35, no. 6, pp. 1209-1223, 2002.
- [11] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Trans. Neural Networks*, vol. 13, no. 2, pp. 415-425, 2002.
- [12] L. I. Kuncheva, et al., "Decision templates for multiple classifier fusion: An experimental comparison," *Pattern Recognition*, vol. 34, no. 2, pp. 299-314, 2001.
- [13] C. I. Watson and C. K. Wilson, *Fingerprint Database*. National Institute of Standards and Technology, Special Database 4, FPDB, 1992.
- [14] A. K. Jain, et al., "Filterbank-based fingerprint matching," *IEEE Trans. Image Processing*, vol. 9, no. 5, pp. 846-859, 2000.
- [15] R. M. Rifkin and A. Klautau, "In defence of one-vs-all classification," *Jnl. of Machine Learning Research*, vol. 5, pp. 101-141, 2004.
- [16] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.
- [17] S. 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.
- [18] L. I. Kuncheva, *Combining Pattern Classifiers*, Wiley-Interscience, 2004.
- [19] Y. S. Huang and C. Y. Suen, "A method of combining multiple experts for the recognition of unconstrained handwritten numerals," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90-94, 1995.
- [20] L. I. 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.
- [21] K. Obermayer and T. J. Sejnowski, *Self-Organizing Map Formation Foundations of Neural Computation*, The MIT Press, 2001.
- [22] 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.



민준기

2004년 연세대학교 기계전자공학부 정보산업전공 졸업. 2004년~현재 연세대학교 컴퓨터과학과 석사과정. 관심분야는 패턴인식, 인공지능, 진화계임



홍진혁

2002년 연세대학교 기계전자공학부 정보  
산업전공 졸업. 2002년~2004년 연세대  
학교 컴퓨터과학과 석사. 2004년~현재  
연세대학교 컴퓨터과학과 박사과정. 관심  
분야는 지능형 에이전트, 패턴인식, 바이  
오인포메틱스

조성배

정보과학회논문지 : 소프트웨어 및 응용  
제 32 권 제 1 호 참조