

# 규칙 정보를 이용한 은행 전표 상의 필기 한글 금액 인식

지 태 창<sup>†</sup> · 이 현 진<sup>††</sup> · 김 은 진<sup>†††</sup> · 이 일 병<sup>††††</sup>

## 요 약

한글 인식에 관한 기존의 연구는 한글 낱자 인식에 치우쳐 왔고, 실제 문서 인식 시스템 개발을 위한 연구는 거의 이루어지지 않았다. 그래서, 본 논문에서는 인식된 문자열의 오류 교정에 관한 연구로서 한글 금액열 인식기를 개발하였다. 한글 낱자 인식에서 문제가 되는 부분은 데이터의 방대함 때문에 발생한다. 컴퓨터상에서 표현될 수 있는 한글 낱자의 개수는 2000여 자 이상이다. 따라서, 기존의 연구들은 이러한 문제점을 해결하기 위해서 실생활에서 많이 쓰이는 낱자에 대해서만 실험을 했다. 하지만, 실험 대상 낱자의 개수를 1000여 자 정도로 줄였어도, 여전히 80%대 이하의 저조한 인식률을 보이고 있다. 이렇게 인식률이 저조한 범용 한글 낱자 인식기를 한글 금액 인식이라는 제한된 상황에서 사용하는 것은 적합하지 않다. 따라서, 본 연구에서는 한글 금액에 사용되는 16자의 한글 낱자만 인식할 수 있는 인식기를 제안하였다. 제안한 한글 낱자 인식기는 통계적 인식기를 사용한 다중 인식기 형태로 만들었고, 이를 통해 개별적인 특징으로 인한 인식률의 저하를 방지할 수 있다. 금액의 후처리하는 한글 금액열 내에 내재되어 있는 금액에 대한 구조적인 규칙 정보를 이용하였다. 이 규칙을 이용하여 한글 금액의 인식 단위에 대한 인식 결과의 오류 보정을 할 수 있다. 실험 결과 제안한 한글 낱자 인식기의 1후보까지 인식률은 95.49%였고, 4후보까지 인식률은 99.72%였다. 그리고, 후처리의 처리를 거친 금액열에 대해서는 신뢰도가 96.42%였다. 본 논문에서는 사용된 낱자의 개수가 적고, 구조안에 규칙 정보가 존재하는 한글 문자열의 경우에 제한된 글자를 인식하는 낱자 인식기와 오류를 교정할 수 있는 후처리로 문자열 인식의 신뢰도를 향상시킬 수 있는 방법을 제안하였다.

## Handwritten Korean Amounts Recognition in Bank Slips using Rule Information

Tae-Chang Jee<sup>†</sup> · Hyun-Jin Lee<sup>††</sup> · Eun-Jin Kim<sup>†††</sup> · Yillbyung Lee<sup>††††</sup>

### ABSTRACT

Many researches on recognition of Korean characters have been undertaken. But while the majority are done on Korean character recognition, tasks for developing document recognition system have seldom been challenged. In this paper, I designed a recognizer of Korean courtesy amounts to improve error correction in recognized character string. From the very first step of Korean character recognition, we face the enormous scale of data. We have 2350 characters in Korean. Almost the previous researches tried to recognize about 1000 frequently-used characters, but the recognition rates show under 80%. Therefore using these kinds of recognizers is not efficient, so we designed a statistical multiple recognizer which recognize 16 Korean characters used in courtesy amounts. By using multiple recognizer, we can prevent an increase of errors. For the Postprocessor of Korean courtesy amounts, we use the properties of Korean character strings. There are syntactic rules in character strings of Korean courtesy amounts. By using this property, we can correct errors in Korean courtesy amounts. This kind of error correction is restricted only to the Korean characters representing the unit of the amounts. The first candidate of Korean character recognizer show 95.49% of recognition rate and up to the fourth candidate show 99.72%. For Korean character string which is postprocessed, recognizer of Korean courtesy amounts show 96.42% of reliability. In this paper, we suggest a method to improve the reliability of Korean courtesy amounts recognition by using the Korean character recognizer which recognize limited numbers of characters and the postprocessor which correct the errors in Korean character strings.

\* 본 연구는 정보통신부 '산학연 공동 기술 개발 사업'의 지원으로 수행되었음.

† 정 회 원 : LG-EDS 시스템 기술연구부문 사원

†† 준 회 원 : 연세대학교 대학원 컴퓨터학과

††† 정 회 원 : 버추얼텍 품질관리팀 팀장

†††† 정 회 원 : 연세대학교 컴퓨터학과 교수

논문접수 : 2000년 1월 11일, 심사완료 : 2000년 7월 10일

## 1. 서 론

문서의 사용이 급속히 증가하고 있는 상황에서도, 종이에 작성된 대부분의 기존 문서를 적절히 처리하여 컴퓨터에 저장시키지 못하고 있다. 따라서, 대용량의 자료 입력이 필수적인 데이터베이스를 구축하는데 있어서도 대부분 수작업에 의한 입력이 사용되고 있는 실정이다. 따라서 텍스트, 그래픽, 영상 등 문서의 기본 구성 요소를 컴퓨터로 인식하고, 더 나아가 문서 자체의 구조를 분석하여 그 의미를 이해하고자 하는 연구를 수행할 필요가 있다.

문서의 기본 구성 요소를 컴퓨터로 인식하고자 하는 연구는 활발히 진행되고 있다. 인쇄체의 문자 단위 인식에 대한 연구는 상당한 수준에까지 진척되어 이미 실용화되고 있는 시스템이 소개된 바 있고[2], 요즘은 필기문자를 인식하고자 하는데 까지 그 영역을 넓히고 있다.

이에 반하여 실제 문서 인식 시스템 개발을 위해 필요한 문서의 구조를 분석하고 이해하는 연구, 인식을 효과적으로 수행하기 위해 영상의 질을 강화시키는 전처리에 대한 연구, 그리고 인식된 문자열의 오류 교정에 대한 연구 등은 상대적으로 미비한 형편이다. 하지만, 최근에는 그간의 기반 연구에 힘입어 그 관심이 크게 고조되고 있다. 1991년부터 문서분석 및 인식에 관한 국제 학술회의(ICDAR), 첨단 필기인식에 관한 국제 워크샵(IWFHR)이 격년으로 개최되고 있으며, 최근에는 우리 나라와 프랑스 간의 연구교류를 위한 한불 문자인식 워크샵이 개최되는 등 국내외적으로 활발한 연구개발이 진행되고 있다[2].

본 논문에서는 실제 문서 인식 시스템 개발을 위해 필요한 연구 중 하나인 한글 금액열 인식을 개발하고, 은행 전표 내의 한글 금액열을 대상으로 실험하였다.

금액 인식기를 만들기 위한 한글 금액열의 특징을 살펴보면 다음과 같다.

첫째, 한글 낱자의 개수가 제한적이다. 금액에 쓰이는 한글의 개수는 '일, 이, 삼, 사, 오, 육, 칠, 팔, 구, 십, 백, 천, 만, 억, 원, 정'의 16자로 상당히 적다. 그래서, 기존의 한글 낱자 인식 방법을 사용하여 쉽게 낱자 인식기의 인식률을 증가시킬 수 있다.

둘째, 금액열 자체에 금액에 대한 구조적인 규칙 정보가 내재되어 있다. 뒤에서 언급하겠지만, 금액이 이

루어지기 위해서는 일정한 규칙을 만족해야 한다. 이 규칙은 한글 금액열이 자체적으로 가지고 있는 성질이고, 이 성질을 이용하면 규칙 후처리가 가능하다.

위의 두 가지 특징을 통합해서 낱자 인식기, 규칙 후처리를 거치는 한글 금액열 인식기를 구성하였다.

본 논문의 2장에서는 한글의 낱자 인식과 한글의 후처리에 대한 연구동향에 대해서 설명한다. 3장에서는 한글 금액열 인식기의 구성과 한글 금액 낱자 인식기, 규칙 후처리에 대해서 설명한다. 그리고 4장에서는 제안하는 한글 금액열 인식기의 성능을 살펴본다. 마지막으로 5장에서 본 연구에 대한 결론을 내린다.

## 2. 관련 연구 동향

### 2.1 필기 한글 인식에 관련된 연구들

한글 낱자 인식을 위한 연구들은 글자의 분류 방법에 따라 구문 해석적 방법, 통계적 방법 및 신경 회로망을 이용한 방법으로 나누어진다[6].

구문 해석적 방법은 주어진 한글 글자에서 획을 추출한 후, 이들의 글자 내의 상대적 위치를 이용하여 한글 글자를 인식하는 방법이다. 구문 해석적 방법을 사용한 예로는 제약조건 만족 그래프를 사용하여 이미 분리된 자소들을 다시 결합하는 방법이 있고[11], 계층적인 무작위 그래프 모형을 이용한 방법이 있다[13].

통계적 방법은 특징 추출을 통해 주어진 글자의 특징 벡터를 구성한 후 이 벡터들을 통계적으로 분류하는 방법이다. 통계적 방법을 사용한 예로는 가로, 세로, 대각선의 네 방향 성분을 추출해서 MDC(Minimum Distance Classifier)를 이용해서 인식하는 방법이 있고[14], 다수의 HMM(Hidden Markov Model)을 이용한 분류 방법이 있다[18].

신경 회로망을 이용한 방법은 문자 영상을 그대로 이용하거나, 주어진 문자 영상으로부터 특징을 추출하여 특징 벡터를 구성해서 신경 회로망을 통해 인식하는 방법이다. 신경 회로망을 이용한 방법의 예로는 통계적 근집화를 이용한 대부분류와 신경 회로망을 이용한 상세 분류를 통한 인식 방법과[5], PCNN(Partially Connected Neural Network)을 변형한 MPCNN(Modular Partially Connected Neural Network)을 이용하여 인식하는 방법이 있다[12].

구문 해석적 방법은 인식률은 좋으나 인식 시간이

오래 걸리는 단점이 있다. 본 논문에서 다루는 시스템은 실시간 처리가 가능해야 하기 때문에 구문 해석적 방법으로 낱자 인식을 만드는 데는 어려움이 있다. 일반적으로 신경망을 이용한 방법은 인식률과 인식 시간면에서 좋은 결과를 보인다. 하지만, 본 논문에서 사용한 특징으로 실험하였을 때 제대로 수렴하지 못하였다. 따라서, 본 논문에서는 인식률과 인식 시간을 향상시키면서, 구현하기에도 쉬운 통계적인 방법으로 낱자 인식을 구현하였다.

### 2.2 후처리 시스템 관련 연구

개별적으로 인식된 문자들이 문자열을 형성할 때는 오인식된 문자를 교정하는 후처리를 적용해서 인식률을 높인다. 후처리 시스템은 크게 다양한 사전을 이용한 방법과 Binary N-gram과 형태소 분석을 이용한 방법 등으로 나눌 수 있다[5].

사전을 이용한 후처리는 입력 전표 상의 특정 필드에 나타날 수 있는 어휘가 한정된 경우 이를 사전 형태로 구성하여 입력 문자들에 대한 오인식을 교정한다. 예를 들면 은행 입출금 전표의 은행명과 지점명 필드 부분, 우편 봉투에서의 주소 부분 등이다. 일반적인 사전 후처리는 각 문자에 대한 인식 후보 문자열을 기반으로 이들을 조합하면서 조합된 단어가 사전 내에 존재하는지를 검증함으로써 가장 가능성이 높은 단어를 출력한다[3].

사전을 이용한 후처리의 예로는 HMM모델을 기반으로 해서 어휘정보 네트워크를 구성한 후 주소 DB와 비교해서 주소를 인식하는 방법[2]과, 각 낱자의 인식률을 이용해서 단어 후보를 생성하고 그 결과와 주소 사전을 비교해서 주소를 인식하는 방법[14]의 주소 인식을 위한 후처리와 명사 사전 내의 개별 단어에 대해 인식 후보 문자열을 참조하는 방법[5]의 소규모 명사를 위한 후처리 방법이 있다.

Binary N-gram과 형태소 분석을 이용한 후처리는 주어와 서술어를 제대로 갖춘 한국어 문장의 인식을 위한 후처리 방법으로 문맥적 지식을 사용하여 후처리를 한다. 이 방법의 예로는 한국어의 형태론적 품사 정보인 어절 정보와 표제어를 묘사하는 기능을 가진 N-gram 사전인 문자열 정보를 이용하여 후처리를 하는 방법이 있다[10].

다양한 사전을 이용한 방법과 Binary N-gram과 형태소 분석을 이용한 방법은 구성할 수 있는 단어의 종

류가 한정되거나 정확한 문장을 이룰 수 있는 경우에만 이용이 가능한데, 금액열은 종류가 한정되어 있지 않고 정확한 문장으로 만들 수도 없기 때문에 위의 두 가지 후처리 방법은 사용이 불가능하다.

그래서, 본 논문에서는 앞 절에서 언급한 한글 금액열의 고유한 규칙 정보를 사용한 새로운 후처리 방법을 개발하여 사용하였다.

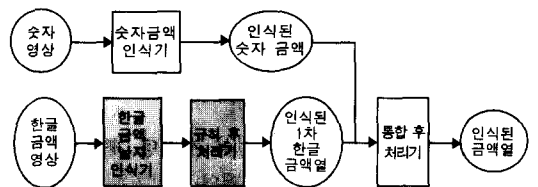
## 3. 한글 금액 인식기

### 3.1 한글 금액 인식기의 전체 구조

한글 금액 인식기의 전체 구조는 (그림 1)과 같다. 우선, 은행 전표에서 한글 금액 영상에 해당되는 부분만이 추출되어 낱자 단위로 분리된 후, 한글 금액 인식기에 들어와서 한글 낱자 인식의 입력으로 사용된다.

낱자 인식기의 인식 결과는 규칙 후처리기의 입력으로 사용되어 1차 한글 금액열의 출력을 보인다. 은행 전표에 한글 금액열만 있는 경우에는 이 결과가 금액으로 결정된다. 하지만, 숫자 금액이 같이 있는 경우에는 숫자 금액 인식 결과와 1차 한글 금액의 인식 결과를 통합 후처리기의 입력으로 사용해서 최종 인식 결과를 생성한다.

본 논문에서는 한글 금액 인식기의 여러 모듈 중 처음 부분에 위치한 한글 금액 낱자 인식기와 규칙 후처리기에 대하여 연구하였다.



(그림 1) 한글 금액 인식기의 전체 구조

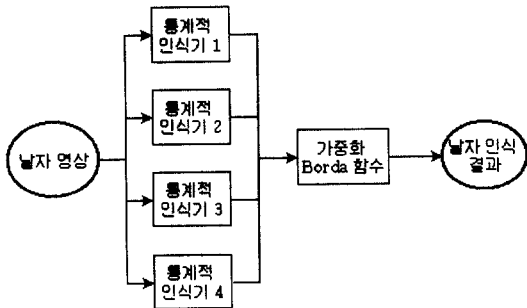
### 3.2 한글 금액 낱자 인식기

#### 3.2.1 한글 금액 낱자 인식기의 구성

한글 금액 낱자 인식기의 구성은 (그림 2)와 같다. 통계적 인식기와 가중화 Borda 함수는 기존의 알고리즘을 그대로 사용하였다.

(그림 2)에서 보는 바와 같이 낱자 인식기는 낱자 영상에서 4가지 특징을 추출한 뒤 각각 통계적 인식기에서 인식을 수행한다. 그 후 그 값을 가중화 Borda

함수의 입력으로 하여 출력을 내는 다중 인식기의 구조이다. 각각의 통계적 인식기는 모두 MDC(Mean Distance Classification) 알고리즘을 사용하였고, 각 인식기에서 인식의 기준이 되는 표준 벡터는 LVQ 1 알고리즘으로 보정한 후 사용하였다.



(그림 2) 한글 금액 남자 인식기의 구성도

### 3.2.2 통계적 인식기에서 사용되는 특징들

고태석은 특징 추출 방법을 통계적 특징, 영상 변환에 의한 특징, 지역/위상적 특징의 3가지로 중 분류하고, 각 중분류별로 대표적인 4~5가지 특징 추출 방법을 선정해서 한글의 초성자음, 횡모음, 종모음별로 인식 실험을 수행하였다[1]. 그 결과 지역/위상적 특징 분류가 다른 특징 분류들보다 인식 성능이 우수하였고, 지역/위상적 특징 분류 중 특히 윤곽선 위상 변화 특징 추출 방법과 모퉁성 특징 추출 방법이 각각 자음과 모음에서 성능이 우수하였다.

따라서, 지역/위상적 특징 분류 중 세 가지 특징 추출 방법을 본 논문의 특징 추출 방법으로 선택하였다. 그 중에서 인식률이 가장 좋은 윤곽선 위상 변화 특징 추출 방법은 방법을 달리하여 두 가지 방법으로 특징을 추출해서 모두 네 가지 특징 추출방법을 사용하였다.

특징은 기본적으로 동적 그물망을 사용하여 추출하였다[9]. 그 이유는 필기습관에 따른 변형으로 영상 안의 유효한 정보가 한 쪽에 편중된 경우가 있기 때문이다. 따라서, 동적 그물망을 사용해서 정보의 치우침을 어느 정도 보정하였다. 논문에서 사용한 네 가지 특징 추출방법을 자세히 살펴보면 다음과 같다.

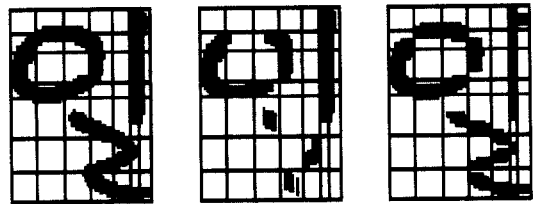
첫 번째 특징 추출 방법은 가로-세로 방향 성분의 특징 추출 방법이다[9].

입력 영상을 7×7의 동적 그물망으로 나누고, 가로획과 세로획의 영상 지도를 구한다. 각 영역에 대하여

영역 내에 분포하는 가로획이나 세로획의 런길이(run-length)가 각 그물망의 길이의 1/2 이상이면 특징이 존재하는 것이다. 각 그물망에서 특징으로 인식된 흑화소의 개수를 그물망의 크기로 나누는 정규화 과정을 거쳐서 특징 벡터를 구한다.

한글 글자체의 형태를 보면, 가로, 세로, 대각선(45°, 315°), 원의 5가지 기하학적인 형태를 가지고 있다. 이 중에서 특히 많은 형태를 차지하는 것이 가로획과 세로획의 성분이다. 따라서, 이런 가로획과 세로획의 성분을 추출하여 특징을 추출하는 것이 한글 인식에 있어서 중요하다.

특징 벡터의 크기는  $7 \times 7 \times 2 = 98$ 이다. (그림 3)은 '일'에 대한 가로-세로 방향 성분 특징 추출의 예를 보여주고 있다.



(원 영상) (세로 방향 성분) (가로 방향 성분)

(그림 3) '일'에 대한 가로-세로 방향 성분 특징 추출

두 번째 특징 추출 방법은 윤곽선의 위상 변화(Gradient(1))를 이용한 특징 추출 방법이다[1].

문자 영상의 윤곽선에 소벨(Sobel) 연산자를 적용하여 윤곽선의 방향성분을 추출한다. 방향성분은 360도를 60도씩 나누어 총 6개의 값으로 나타낸다. 방향성분을 구한 뒤 문자 영역을 44의 동적 그물망으로 나누고, 각 영역 내에 들어 있는 방향성분의 분포를 0~1 사이의 값으로 정규화 하여 특징벡터로 사용한다.

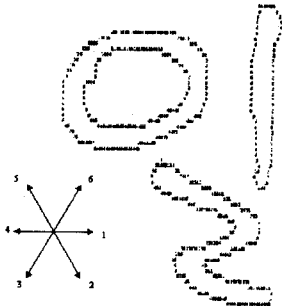
이 특징은 패턴의 에지에 많은 정보가 존재한다는 연구 결과에 근거한다. 그리고, 첫번째 특징이 직선 성분에만 중점을 두는 단점을 피하고, 'ㅇ', 'ㅎ'을 잘 구별할 수 있게 된다. 고태석의 연구를 보면 이 특징이 자음에 대해서 가장 좋은 인식률을 보이고 있다[1].

일반적인 영상처리 분야에서는 4-, 8- 방향 성분을 특징 추출할 때 사용하는 것이 일반적이다. 하지만, 본 논문에서는 'ㅇ'이나 'ㅎ'과 같이 직선 성분만을 이용해서는 찾아낼 수 없는 특징을 찾기 위해서 윤곽선의 위상 변화를 이용한 특징 추출 방법을 사용하였기 때문

에 직선 만을 찾는 4방향 성분은 이용하지 않았다. 그리고, 8방향 성분은 네번째 특징 추출 방법으로 사용하였다.

윤곽선의 위상 변화를 이용한 특징 추출을 두 가지 사용한 이유는 원에 대한 특징을 조금 더 잘 찾아보고자 하기 위함이다. 45도나 60도의 기울기 하나로 원을 완벽하게 찾아낼 수 있다라고 말하기는 힘들다. 반면에, 정확도를 높이기 위해 30도의 기울기를 사용하면, 특징 벡터의 수가 너무 늘어나게 되어, 다른 특징 추출 방법과 형평성의 문제가 생기게 된다. 따라서, 특징 벡터의 수를 정당히 하면서 원 성분을 조금 더 잘 찾아내기 위하여 8방향 성분과 6방향 성분의 두 가지 특징 추출 방법을 사용하였다.

특징 벡터의 크기는  $4 \times 4 \times 6 = 96$ 이다. (그림 4)는 '일'에 대한 6방향 코드 윤곽선의 위상 변화를 이용한 특징 추출의 예를 보여 주고 있다.



(그림 4) '일'에 대한 6방향 코드 윤곽선의 위상 변화를 이용한 특징 추출

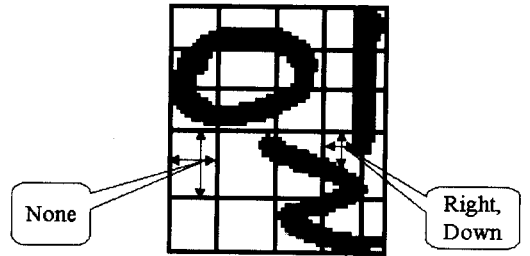
세 번째 특징 추출 방법은 오목성 특징(Up Down Left Right Hole : UDLRH) 추출방법이다[1].

입력 영상을  $5 \times 5$ 의 동적 그물망으로 나눈 후, 각 영역별로 모든 백화소를 기준점으로 하여 상하좌우로 영상의 끝까지 주사한다. 주사선이 흑화소와 만나는 경우, 주사선의 기준점이 주사 방향으로 특징점을 갖는 것으로 한다. 상하좌우 모든 방향에서 흑화소와 만나는 경우는 원형 특징을 갖는 것으로 나타낸다. 그래서, 한 그물망 내에 상, 하, 좌, 우, none의 다섯 가지 특징 벡터를 가지게 된다.

고태석의 연구를 보면 이 특징이 종모음과 횡모음에 대해서 다른 특징들 보다 월등한 성능을 보이고 있다 [1]. 그래디언트 특징은 자음에 대해서 좋은 인식률을 보이지만, 모음에 대해서는 오목성 특징보다는 떨어진다. 가로, 세로 방향 성분은 기울어 쓰게 되는 경우 인

식물이 떨어질 수 있지만, 기본적으로 오목성 특징은 직선 성분에 대해 좋은 특징 벡터를 산출하고, 한글을 기울어 쓴다고 해도 직선 성분이라는 사실에는 변함이 없기 때문에 기울어짐에 강한 면을 보인다.

특징 벡터의 크기는  $5 \times 5 \times 5 = 125$ 이다. (그림 5)는 '일'에 대한 오목성 특징 추출의 예를 보여주고 있다.

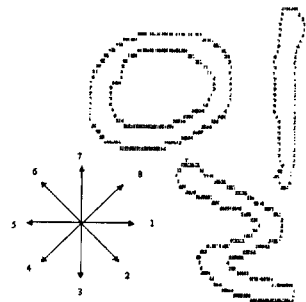


(그림 5) '일'에 대한 오목성 특징 추출

네 번째 특징 추출 방법은 윤곽선의 위상 변화(Gradient(2))를 이용한 특징 추출방법이다[1].

문자 영상의 윤곽선에 소벨(Sobel) 연산자를 적용하여 윤곽선의 각 점에서의 방향성분을 추출한다. 방향 성분은 360도를 45도씩 나누어 총 8개의 값으로 나타낸다. 방향성분을 구한 후 문자 영역을  $5 \times 5$ 의 동적 그물망으로 나누고, 각 영역 내에 들어 있는 방향성분의 분포를 0~1 사이의 값으로 정규화 하여 특징벡터로 사용한다.

특징 벡터의 크기는  $5 \times 5 \times 8 = 200$ 이다. (그림 6)은 '일'에 대한 8방향 코드 윤곽선의 위상 변화에 의한 특징 추출의 예를 보여 주고 있다.



(그림 6) '일'에 대한 8방향 코드 윤곽선의 위상 변화에 의한 특징 추출

### 3.2.3 통계적 인식기

통계적 인식기의 인식 방법은 MDC 알고리즘을 사용하였다[2].

전체 패턴이  $C$ 개의 클래스( $A = \{1, \dots, C\}$ )로 이루어져 있고, 각 클래스에  $k$ 개의 입력 집합이 있을 때, MDC 알고리즘은 클래스  $c$ 에 대해 특징 벡터  $x_c$ 의  $i$  번째 원소의 평균  $m_{c(i)} = \sum_{j=1}^k x_{c(i)j} \div k$ 을 구하고, 이 평균과 입력 영상의 특징 벡터와의 거리의 차가 최소가 되는 것을 인식 결과로 결정하는 방법이다[16]. MDC 알고리즘으로 최소값  $F$ 를 정하는 것은 식 (1)과 같다.

$$F = \min_{c \in A} \|x - m_c\|$$

단,  $\|x - m_c\| = \left[ \sum_{i=1}^n (x_{(i)} - m_{c(i)})^2 \right]^{1/2}$  (1)

$n$ 은 벡터의 원소의 수

### 3.2.4 LVQ 알고리즘

보통 평균을 집단을 대표하는 값으로 사용하고 있지만, 통계학에서는 정확도를 높이기 위해서 평균 대신에 Mean, Median, Trimmed Mean 등의 다른 대표값을 사용한다. 그래서, 본 논문에서는 통계적 인식기로 구한 평균 벡터만으로 인식을 수행하지 않고 Learning Vector Quantization(LVQ) 알고리즘[15]을 사용해서 평균 벡터를 정규화한 표준벡터를 가지고 인식을 수행하였다.

LVQ 알고리즘은 클래스의 정보를 이용해서 평균 벡터를 조금씩 이동시켜서 인식기의 성능을 증가시키는 교사학습 방법이다. 학습이 이루어지는 동안에 입력 벡터  $x$ 는 입력 공간에서 무작위로 선택되어진다. 입력 벡터  $x$ 의 클래스 레이블과 가장 가까운 평균 벡터  $m_c$ 가 같은 클래스에 속하면, 평균 벡터  $m_c$ 가 입력 벡터  $x$ 의 방향으로 미리 정의된 양만큼 이동을 한다. 반대의 경우에는 평균 벡터  $m_c$ 가 입력 벡터  $x$ 로부터 미리 정의된 양만큼 멀어지게 된다. LVQ 알고리즘은 Kohonen이 1986년에 처음으로 제안했고, [15]에는 세 가지 종류의 LVQ 알고리즘이 소개되고 있다. 본 논문에서 사용한 것은 LVQ 1 알고리즘이다.

LVQ 1 알고리즘은 위에서 언급한대로 가장 가까운 평균 벡터와 입력 벡터가 같은 클래스에 속하면 입력 벡터에 다가가고, 다른 클래스에 속하면 입력 벡터에서 멀어지는 것으로 수식으로 나타내면 다음과 같다.

$$\begin{cases} m_c(t+1) = m_c(t) + \alpha(t) [x(t) - m_c(t)] \\ \quad \text{ } x \text{와 } m_c \text{가 같은 클래스에 속해 있으면,} \\ m_c(t+1) = m_c(t) - \alpha(t) [x(t) - m_c(t)] \\ \quad \text{ } x \text{와 } m_c \text{가 다른 클래스에 속해 있으면,} \end{cases} \quad (2)$$

단,  $0 < \alpha(t) < 1$

본 논문에서는  $\alpha(t)$ 를 0.1 이하의 값으로 하였으며 시간에 불변하게 고정시켰다.

### 3.2.5 가중화 Borda 함수

앞의 낱자 인식기에서 사용된 특징에서 언급했듯이 각각의 특징은 자음이나 모음 등 한글을 이루는 어느 한 자소에 대해서만 좋은 인식률을 보이고 있다. 아직까지 자음과 모음을 모두 잘 표현할 수 있는 특징이 개발되지 못 하였다. 이런 상황에서 자소 단위의 인식이 아닌 낱자에 대한 인식을 수행할 때 기존의 특징 중 어느 한가지 특징만 사용해서 정확한 인식 결과를 내기는 어렵다.

따라서 자음과 모음의 인식률이 모두 좋은 완벽한 특징을 찾아내거나, 개개의 특징을 특징 단계에서 결합하거나 인식기 단계에서 결합하여 인식 성능을 향상시키는 것이다. 필기 한글 글자에 대한 최적의 특징을 찾아내는 것에 대해 여러 곳에서 연구가 이루어지고 있지만 아직 최적의 특징을 찾아내지는 못하고 있다[1, 8]. 개개의 특징을 특징 단계나 인식기 단계에서 결합하는 방법도 여러 곳에서 연구되고 있고, 하나의 특징을 사용하는 것보다는 좋은 결과를 보이고 있다[2, 7].

본 논문에서는 개개의 특징을 통합하는 방법을 사용하였는데 특징 단계에서 결합하는 방법보다는 인식기 단계에서 결합하는 방법의 실험 결과가 더 좋아서, 다중 인식기의 형태를 이용하여 한글 낱자 인식기를 완성하였다.

다중 인식의 결합기로 사용되는 방법은 하나의 인식 결과만 출력하는 추상 레벨과 순위의 형태로 출력하는 순위 레벨, 그리고 순위와 더불어 신뢰값까지 함께 출력하는 측정치 레벨 등으로 분류된다. 이 중에서 순위 레벨에 의한 결합이 다른 방법보다 더 좋은 결과를 보이고 있다[16]. 순위 레벨에 의한 결합은 결합을 위해 제공되는 정보가 많으며, 인식기의 과거 성향에 대한 정보는 거의 사용하지 않는다. 순위 레벨의 결합 방법으로는 최고 순위 방법, Borda 함수, 가중화 Borda 함수, Condorect 함수, Nanson 함수 등이 있고, 이 중에서는 가중화 Borda 함수가 가장 좋은 인식 결과를 보인다[7]. 따라서, 본 논문에서는 가중화 Borda 함수를 사용하여 한글 낱자 인식기를 완성하였다.

전체 패턴이  $M$ 개의 클래스( $A = \{1, \dots, M\}$ )로 이루어져 있을 때, Borda 함수는 클래스  $i$ 에 대하여 낱자 인식기  $e_k$ 가 출력한 순위  $r_{ik}$ 에 따라 역순위  $M - r_{ik}$

합당하고, 모든 인식기의 역순위 값을 합산한 것을 그 클래스의 Borda 점수로 정하고, 이 점수가 큰 순서대로 순위를 결정하는 방식이다. 가중화 Borda 함수는 Borda 함수에 낱자 인식기의 중요도  $W_k$ 에 따라서 별도의 가중치를 부여하는 방식이다. 본 논문에서는 가중치  $W_k$ 를 훈련 데이터에 대한 각 개별 인식기의 인식률을 참조하여 (1, 10) 사이의 값으로 변환하여 사용하였다. 가중화 Borda 함수의 결과  $F(x)$ 는 다음과 같이 정의할 수 있다.

$$F(x) = \max_{i \in A} (B_i(x))$$

$$\text{단, } B_i(x) = \sum_{k=1}^K W_k \times (M - r_k^i(x)) \quad (3)$$

3.3 한글 금액 규칙 후처리기

규칙 후처리기는 한글 금액에 내재되어 있는 구조적인 규칙 정보를 이용한다. 1장에서 언급했듯이 금액을 잘 보면 금액이 이루어질 때 어떤 규칙이 존재한다는 것을 알 수 있다. 이 규칙을 이용해서 규칙 정보를 만들고, 규칙 정보에 의해서 후처리를 하는 것이 규칙 후처리기이다.

한글 금액의 구조적인 규칙 정보를 살펴보면 다음의 <표 1>과 같다.

<표 1> 한글 금액의 구조적인 규칙 정보

구조적인 규칙 정보	
한글	한글 숫자가 연속해서는 나올 수 없고, 각 단계에서 세 숫자 번까지 나올 수 있다.
십	각 단계에서 한 번만 나올 수 있다.
백	각 단계에서 한 번만 나올 수 있고, 십의 앞에 나온다.
천	각 단계에서 한 번만 나올 수 있고, 십, 백의 앞에 나온다.
만	한 금액열에 한 번만 나올 수 있다.
억	한 금액열에 한 번만 나올 수 있고, 만의 앞에 나온다.
원	한 금액열에 한 번만 나올 수 있고, 금액열의 마지막이 나 두 번째에만 나타날 수 있다.
정	한 금액열에 한 번만 나올 수 있고, 금액열의 마지막에 만 나타날 수 있다.

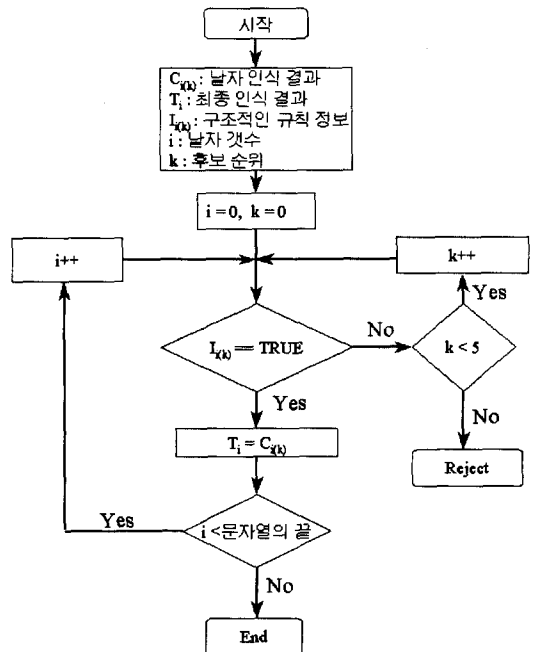
위 <표 1>을 보면 단계라는 말이 나온다. 한글 금액의 구조를 보면 금액의 특성상 한 금액열이 여러 단계로 나누어진다. 한국의 금액 뿐만 아니라 모든 나라의 금액은 약간의 차이가 있지만, 모두 단계가 존재한다. 한국의 금액은 일, 십, 백, 천의 네 단위가 하나의 단계를 이루는 요소고, 각 단계를 나누는 기준은 만,

억 단위이다. 영어에서는 ‘,’로 구분되는 부분 즉, one, ten, hundred가 단계를 이루는 요소이고, thousand, million, billion이 각 단계를 나누는 기준이 된다. 이를 그림으로 보면 다음 (그림 7)과 같다. (그림 7)에서 색이 다른 부분이 단계로 나누어지는 부분이다.

구백팔십이만팔천원정

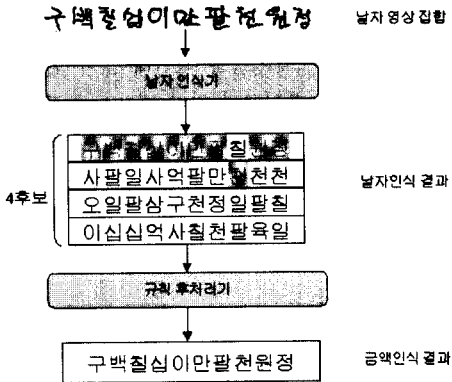
(그림 7) 금액에서 단계를 구분하는 예

위 <표 1>의 구조적인 규칙 정보를 이용한 규칙 후처리 알고리즘에 의해서 처리되는 과정을 도식적으로 살펴 보면 다음 (그림 8)과 같다. 낱자의 인식 결과는 상위 4개의 후보를 가져가게 된다. 처음에 1후보에 대해 규칙을 제대로 따르고 있는지 검사한다. 규칙이 정당하면 다음 자리로 넘어가고, 규칙에 어긋나면 그 자리의 2후보에 대해 규칙을 검사한다. 2후보에 대한 규칙이 정당하면, 인식 결과를 1후보가 아닌 2후보로 바꾸고, 다음 자리에 대해 검사를 시작한다. 2후보도 규칙에 어긋나면, 차례대로 3, 4후보에 대해 검사한다. 마지막, 4후보에 대한 검사도 실패하면, 금액열에 대한 판단을 사람에게 넘기도록 reject시킨다.



(그림 8) 규칙 후처리기의 처리 과정

이상과 같이 한글 낱자 인식기, 규칙 후처리기의 두 단계를 거치면서 한글 금액열이 인식되게 된다. 다음 (그림 9)는 실제 한글 금액 인식 과정의 예를 보여주고 있다.



(그림 9) 한글 금액 인식 과정의 예

#### 4. 실험 결과 및 분석

##### 4.1 실험 환경

실험에 사용한 시스템은 중앙처리장치는 Pentium III 500 MHz, 주 메모리는 128MB였다. Windows98 환경에서 C++ 언어를 사용하여 구현하였다.

본 논문에서는 고려대학교에서 구축된 오프라인 한글 글씨 데이터베이스 KU-1을 사용하였다[4]. KU-1 데이터베이스는 KS C 완성형 한글 사용 빈도순 상위 1,500자에 대하여 다양한 계층, 직업, 지역 분포를 고려한 1,000명 이상의 필기자가 정서체와 본인의 평소 자유 필체로 필기한 1,000벌의 명도 한글 글씨 영상으로 구성되어 있다.

낱자 인식 실험을 위해 KU-1 데이터베이스 1000set에서 각각 16자씩 추출하고, 500set은 학습 집합으로 500set은 실험 집합으로 사용하였다. 대표적인 입력 영상은 다음 (그림 10)과 같다.

일이삼사오육칠팔구십백천만억원정  
 일이삼사오육칠팔구십백천만억원정  
 일이삼사오육칠팔구십백천만억원정  
 일이삼사오육칠팔구십백천만억원정  
 일이삼사오육칠팔구십백천만억원정

(그림 10) 대표적인 입력 영상

한글 금액열은 아직 표준 데이터가 없기 때문에, 수집한 은행 전표 50장에 적혀 있는 금액과 일치하도록, KU-1 데이터베이스의 낱자를 조합하여 금액열을 만드는 방법을 취하였다. 사용한 금액열은 1set 당 50개이고, 50개 금액열에 사용된 낱자의 개수는 320자이며, 한 금액열당 평균 6.4자가 사용되었다. KU-1 데이터베이스를 이용하여, 한글 낱자 학습 집합 500 set 중 100 set을 금액열 학습 집합으로 정하고, 마찬가지로 한글 낱자 실험 집합 500 set 중 100 set을 금액열 실험 집합으로 정하여 금액열을 추출하였다.

#### 4.2 실험 결과

##### 4.2.1 한글 금액 낱자의 인식 결과

한글 금액 낱자 16자(일, 이, 삼, 사, 오, 육, 칠, 팔, 구, 십, 백, 천, 만, 억, 원, 정)에 대해 실험에 사용된 네 개의 통계적 인식기의 성능을 비교해 보면 <표 2>와 같다.

<표 2> 4가지 통계적 인식기의 성능 비교

통계적 인식기	인식률			
	1후보	2후보	3후보	4후보
가로-세로 방향 성분 특징	82.44%	92.64%	95.87%	97.40%
Gradient(1) 특징	82.73%	92.31%	95.66%	97.45%
UDLRH 특징	83.50%	93.46%	96.81%	97.98%
Gradient(2) 특징	91.09%	96.85%	98.45%	99.04%

1, 2, 3, 4후보는 통계적 인식기의 인식 값을 순서대로 나열했을 때의 순위를 가리킨다. 즉, 1후보의 인식률은 인식 값이 가장 큰 경우의 결과에 대한 정 인식률을 나타낸다. 2후보의 인식률은 인식 값이 가장 큰 상위 두 개의 결과에 대한 인식률이고, 3, 4후보에 대한 인식률은 비교하는 대상이 차례대로 누적되게 된다. <표 2>의 인식 결과를 보면 Gradient(2)의 인식률이 가장 좋았다. 그 이유는 특징 벡터의 크기가 다른 세 개의 특징은 100개 정도인데 반하여 Gradient(2) 특징은 200개로 벡터 공간이 커지기 때문이다.

위의 통계적 인식기를 가중화 Borda 함수로 통합한 한글 낱자 인식기의 인식 결과는 <표 3>과 같다. 그리고, MDC를 사용한 방법과 MPCNN(Modular Partially Connected Neural Network)을 사용한 방법과 성능을 비교하였다.

MDC를 사용한 방법은 개개의 특징을 통합하는 방



법 중 특징 단계에서 결합하여 인식하는 방법으로[2], 본 논문에서 사용한 네 가지 특징을 하나의 특징 벡터로 만들고 MDC 알고리즘을 사용해서 인식하였다. MPCNN 알고리즘은 입력층을 10개의 수용 영역으로 분할하고, 입력층과 은닉층 사이를 부분 연결한 인공 신경망의 한 종류이다. MPCNN 알고리즘을 사용한 방법은 한글 금액열에 쓰이는 18자(본 논문에서 사용한 16자에 '조', '금' 추가)를 인식하였다[12].

<표 3> 한글 낱자 인식기의 성능 비교

인식기		인식률			
인식 방법	실험 데이터	1후보	2후보	3후보	4후보
가중화 Borda	KU-1	95.49%	99.12%	99.48%	99.72%
MDC	KU-1	92.27%	97.58%	98.75%	99.11%
MPCNN	360명	95.30%	—	—	—

<표 3>에서 보는 바와 같이 4가지 특징 벡터를 특징 단계에서 결합하여 MDC 알고리즘으로 인식하는 것 보다 본 논문에서 제안한 특징 벡터를 인식기 단계에서 가중화 Borda 함수로 결합하여 인식한 인식 결과가 1후보에 대해 인식률이 3% 정도 향상된 것을 알 수 있다. MPCNN 방법은 첫 번째 후보에 대해서만 인식 결과를 보이고, 실험 데이터가 임의의 360명이 실험에 사용될 18자를 6번 적은 38880개의 낱자를 수집해서 사용하였다. 따라서, 실험 데이터가 KU-1 데이터베이스를 사용한 본 논문과 다르고, 인식할 대상인 한글 낱자의 개수가 본 논문보다 2자가 많기 때문에 본 논문에서 제안한 방법과 정확한 비교를 할 수는 없지만, 그래도 MPCNN 방법보다는 본 논문에서 제안한 방법이 향상된 인식 결과를 보였다.

낱자 인식기의 후보로 4개를 선택했다. 그 이유는 4번째 후보의 99.72%라는 인식률이 충분히 좋은 인식률이고, 5개 이상으로 후보를 더 늘려도 인식률이 눈에 띄게 증가하지 않고, 그에 따른 인식 시간만 늘어나서 좋은 효과를 볼 수 없었기 때문이다. 이 4후보가 금액 인식기의 입력으로 들어간다. 한글 낱자 인식기에서 낱자 한 자를 인식하는데 걸린 시간은 약 0.035초였다.

4.2.2 한글 금액열의 인식 결과

위에서 실험한 금액 낱자 인식기의 1후보의 인식률은 95.49%이다. 실험 환경에서 언급했듯이 실험에 사용된 금액열의 평균 길이는 6.4자이다. 산술적으로 한

낱자당 95.49%의 인식률을 가지는 6.4개의 낱자로 이루어진 문자열의 인식률을 보면  $95.49\%^{6.4} = 74.43\%$ 의 상당히 작은 인식률이 나타난다. 그래서 이 인식률을 높이기 위해서 규칙 후처리기의 후처리 과정을 거치게 된다.

금액의 구조적인 규칙을 이용한 규칙 후처리기의 실험 결과를 후처리기의 성능을 확인해 보기 위해서 가중화 Borda 함수로 통합하기 전의 네가지 특징에 의한 인식 결과와 함께 아래 <표 4>에 금액열에 대한 인식률로 보였다.

<표 4> 금액열 인식기의 규칙 후처리 후 금액열에 대한 인식 결과

인식 방법	후처리 전	규칙 후처리 후			
		정 인식률	오 인식률	기각률	신뢰도
가로-세로 방향 성분 특징	49.04%	79.82%	16.76%	3.42%	83.24%
Gradient(1) 특징	47.72%	76.82%	19.56%	3.62%	80.44%
UDLRH 특징	48.36%	79.58%	17.26%	3.16%	82.74%
Gradient(2) 특징	69.42%	93.54%	5.46%	1.00%	94.54%
가중화 Borda 함수	83.18%	95.76%	3.58%	0.66%	95.42%

후처리 전과 비교해 보면 금액열에 대해서 신뢰도가 많게는 38% 정도에서 적게는 13% 정도까지 눈에 띄게 상승했다는 것을 알 수 있다.

앞에서 보였듯이 낱자가 문자열로 확장되면 인식률이 급속히 감소하게 된다. <표 4>를 보면 알 수 있듯이 낱자 인식률이 85.94%였던 Gradient(1) 특징이 금액열로 가서는 42.72%로 떨어졌다. 하지만, 규칙 후처리를 거치면서 80.44%로 높은 신뢰도의 향상을 보였다.

그러나, 가중화 Borda 함수나 Gradient(2) 특징을 사용한 인식 방법의 신뢰도가 95%에 이르는 것과 비교하면 많이 떨어진다는 것을 알 수 있다. 그 이유는 금액의 숫자에 대한 후처리가 이루어질 수 없기 때문이다. 규칙 후처리기는 금액의 단위에 대한 후처리만 가능하다. 하지만, 금액에는 단위뿐만 아니라 숫자도 존재하고 있다. 그래서, 후처리 전의 인식률이 나쁠 경우 금액의 숫자에 대한 인식률도 떨어져서 규칙 후처리를 거친 후에도 신뢰도가 약간은 나쁘게 되는 것이다.

규칙 후처리기에서 처리되는 시간은 한 금액열 당 약 0.23초 정도였고, 한 금액열당 평균 6.4개의 낱자가 존재하므로 낱자 인식기의 인식 시간을 제외한 규칙 후처리기에 의해서만 걸리는 시간은 약 0.076초 정도

였다.

### 5. 결 론

본 논문에서는 한글 금액열을 인식하기 위한 시스템을 개발하였다. 우선, 통계적 인식기와 가중화 Borda 함수를 사용하여 제한된 수의 한글을 인식할 수 있는 한글 낱자 인식기를 만들고, 금액의 구조적인 규칙 정보를 이용하여 규칙 후처리를 수행하여 최종 인식 결과를 생성해 내도록 하였다.

실제 시스템을 개발하여 은행 전표상에서 실험한 결과 낱자 인식율은 95.49%가 나왔고, 금액열은 96.42%의 비교적 높은 신뢰도를 보였다

본 시스템의 의의는 낱자 인식기의 인식 결과가 95% 내외로 비교적 높게 나오긴 하지만, 금액열 단위로 보면 신뢰도가 80%대로 극도로 저하되는 상황에서 규칙 후처리를 사용하여 오인식률을 최소화 했고, 본 시스템의 낱자 인식기의 인식 결과가 80%대로 낮아도 꽤 믿음직한 신뢰도를 보인다는데 있다.

### 참 고 문 헌

[1] 고태석, 김종렬, 정규식, “오프라인 필기체 한글 자소 인식에 있어서 특징성능의 비교”, 인지과학, 제7권 제1호, pp.57-74, 1996.

[2] 권진옥, 이관용, 변혜란, 이일병, “한글 주소 인식 시스템”, 한국정보과학회 '97 봄 학술발표논문집, 제24권 제1호, pp.529-532, 1997.

[3] 권혁철, 이영식, 황호정, “한글 철자 검사기/교정기 이식 및 글자 인식을 위한 후처리 기술에 관한 최종 연구 보고서”, 삼성전자, 1993.

[4] 김대인, 이성환, “오프라인 한글 글씨 데이터베이스 KU-1의 글씨 품질 자동 평가”, 한국정보과학회 '98 봄 학술발표논문집, 제25권 제1호, pp.707-709, 1998.

[5] 김수형, 도정인, 김준호, 이상규, “통계적 군집화 및 신경망 기법을 이용한 필기체 문자의 오프라인 인식”, 제2회 문자 인식 워크샵 논문집, pp.133-142, 1994.

[6] 김은진, “인간의 한글 지각 처리에 근거한 한글 인식 방법에 관한 연구”, 박사학위 논문, 연세대학교, 1997.

[7] 백종현, 조성배, 이관용, 이일병, “이중 결합 구조를

갖는 다중 인식기 시스템”, 한국정보과학회 '96 봄 학술발표논문지, 제23권 제1호, pp.281-284, 1996.

[8] 윤성수, 변혜란, 이일병, “인간의 정보처리 방법에 기반한 특징추출 및 필기체 문자인식에의 응용”, 인지과학, 제9권 제1호, pp.1-11, 1998.

[9] 중앙대학교, 기아정보시스템, “필기체 문자인식 기술개발”, 정보통신부 제조업 경쟁력 강화사업 연구 보고서, Sep., 1995.

[10] 홍남희, 이원일, 이종혁, 이근배, “어절 정보와 문자열 정보를 이용한 문자 인식에서의 오인식 수정 기법에 관한 연구”, 제1회 문자 인식 워크샵 발표 논문집, pp.109-113, 1993.

[11] 황영섭, 방승양, “계약 조건 만족 그래프를 이용한 필기 한글 인식”, 한국정보과학회 1997년 봄 학술 발표 논문집, Vol.24(1), pp.521-524, 1997.

[12] Y.-M. Baek, K.-T. Lim, S.-I. Chien and J.-S. Park, “Off-line Handwritten Hangul Recognition Based on Multiple Features and Modular Partially Connected Multi-layer Perceptron,” Proceedings of 6<sup>th</sup> IWFHR, pp.269-278, 1998.

[13] H. Y. Kim and J. H. Kim “Handwritten Korean Character Recognition Based on Hierarchical Random Graph Modeling,” Proceedings of 6<sup>th</sup> IWFHR, pp.577-586, 1998.

[14] S. H. Kim “Handwritten Korean Address Recognition by Combining Minimum Distance Classification and Dictionary Look-up,” Proceedings of 6<sup>th</sup> IWFHR, pp.439-448, 1998.

[15] T. Kohonen, et al., “LVQ\_PAK - the learning vector quantization program package Version 3.1,” April, 1995.

[16] D-S. Lee., S. N. Srihari, “Handprinted Digit Recognition : A Comparison of Algorithms,” Proceedings of 3<sup>rd</sup> IWFHR, pp.153-162, 1993.

[17] A. S. Pandya, R. B. Macy, “Pattern Recognition with Neural Networks in C++,” CRC Press, pp. 214-230, 1996.

[18] H. S. Park, S. W. Lee, “Off-line Recognition of Large-set Handwritten Characters with Multiple Hidden Markov Models,” Pattern Recognition, 29-2, pp.231-244, 1996.



### 지태창

e-mail : garura@csai.yonsei.ac.kr  
1997년 연세대학교 컴퓨터과학과  
졸업(학사)  
1999년 연세대학교 대학원 컴퓨터  
과학과 졸업(석사)  
1999년~현재 LG-EDS 시스템  
기술연구부문 사원

관심분야 : 신경회로망, 패턴인식, 문자인식, Data Mining



### 김은진

e-mail : ejkim@virtualek.co.kr  
1987년 연세대학교 전산학과  
졸업(학사)  
1990년 연세대학교 대학원 전산  
학과 졸업(석사)  
1997년 연세대학교 대학원 컴퓨터  
과학과 졸업(박사)

2000년~현재 버추얼텍 품질관리팀 팀장  
관심분야 : 신경회로망, 패턴인식, 문자인식



### 이현진

e-mail : dryad@csai.yonsei.ac.kr  
1996년 순천향대학교 전산학과  
졸업(학사)  
1998년 연세대학교 대학원 컴퓨터  
과학과 졸업(석사)  
1998년~현재 연세대학교 본  
대학원 컴퓨터과학과  
박사과정

관심분야 : 신경회로망, 패턴인식, 규칙 추출



### 이일병

e-mail : yblee@csai.yonsei.ac.kr  
1976년 연세대학교 전자공학과  
졸업(학사)  
1980년 University of Illinois  
전산과학과 졸업(석사)  
1985년 University of Massachu-  
setts 전산정보과학과  
졸업(박사)

1985년~현재 연세대학교 컴퓨터과학과 교수  
1995년~1999년 연세대학교 소프트웨어 응용 연구소 소장  
관심분야 : 신경회로망, 문서인식, Computer Vision, Data Mining