Journal of The Korea Society of Computer and Information
Vol. 21 No. 9, pp. 27-36, September 2016
www.ksci.re.kr
http://dx.doi.org/10.9708/jksci.2016.21.9.027

A Study on Grapheme and Grapheme Recognition Using Connected Components Grapheme for Machine-Printed Korean Character Recognition

Kyong-Ho Lee *

Abstract

Recognition of grapheme is a very important process in the recognition within 'Hangul(Korean written language)' letters using phoneme recognition. It is because the success or failure in the recognition of phoneme greatly affects the recognition of letters. For this reason, it is reported that separation of phonemes is the biggest difficulty in the phoneme recognition study.

The current study separates and suggests the new phonemes that used the connective elements that are helpful for dividing phonemes, recommends the features for recognition of such suggested phonemes, databases this, and carried out a set of experiments of recognizing phonemes using the suggested features.

The current study used 350 letters in the experiment of phoneme separation and recognition. In this particular kind of letters, there were 1,125 phonemes suggested. In the phoneme separation experiment, the phonemes were divided in the rate of 100%, and the phoneme recognition experiment showed the recognition rate of 98% in recognizing only 14 phonemes into different ones.

▸ Keyword : character segmentation, grapheme extraction, grapheme recognition

I. Introduction

문자 인식은 지난 수십 년 동안 많이 연구되어 온 분야이나, 인식률 향상을 위해 지금도 여전히 많은 연구들이 나오고 있다. 오현아 등은 문자 인식 향상을 위해 대상 문자열에 회전 알고리즘을 적용하여 인식률 향상을 꾀하고 있으며[1], 진문용 등은 인식률 향상을 위해 휴리스틱 분할 알고리즘을 적용하여 향상을 꾀하고 있다[2]. 김강산 등은 구글의 Open API를 이용하여 한글 문자 인식 향상을 위한 연구를 하였으며[3], 정규수는 한글 문자 주변에 심벌이 위치한 상황에서 인식률 향상을 위해 한글 문자 템플릿에 의존하지 않는 점진적 좌측 방향으로의 블럽 투사를 제시하였다[4]. 박경화 등은 한글 문자의 다양성 확대를 위해 다중 컬럼 딥 인공 신경망을 이용하여 한글 필기체 인식을 시도하는[5] 등 아직도 많은 연구들이 지속되고 있다. 이렇게 연구가 지속되는 이유는 한글 문자의 특수성 때문으로 한글은 영문자 알파벳과 달리 19개의 초성과 21개의 중성, 27개의 종성들이 2차원적 조합으로 되어 11,172개의 문자가 구성되기 때문에 인식해야할 대상이 많으며, 자소 간 구분이 작은 획 하나에 의해 구분될 정도로 유사성이 높아 인식에 대한 처리 과정이 알파벳에 비해 복잡하기 때문이다[6,7]. 또한 아직도 영문자 알파벳 등 보다 인식률이 낮기 때문에 많은 연구가 지속되고 있다.

그동안 연구된 한글 인식 방법들은 한글을 작성하는 방법에 따라 '인쇄체 인식'과 '필기체 인식'으로 분류할 수 있고 또 인식하는 단위에 따라 '문자 단위 인식'과 '자소 단위 인식'으로 분류할 수 있다. 문자 단위 인식은 문자 단위를 통으로 인식 대상으로 보는 것이며, 자소 단위 인식은 문자를 구성하고 있는 자소를 분리해 자소 단위로 인식을 하는 것으로 초성, 중성, 종성을 이루고 있는 소수의 자소 단위를 인식한 후 인식한 자소 정보와 문자를 구성하는 형식 정보를 이용하여 글자를 인식하는 방법이다. 두 가지 다 전처리 작업으로 입력된 영상에서 먼저 기울어짐 보정을 하고, 텍스트 영역 추출과 수평 분할, 수직

• First Author: Kyong-Ho Lee, Corresponding Author: Kyong-Ho Lee
*Kyong-Ho Lee(khlee@halla.ac.kr), School of Information & Communication, Broadcasting Engineering, Halla University
• Received: 2016. 08. 09, Revised: 2016. 08. 30, Accepted: 2016. 09. 20.

분할 등의 반복을 통해 문자 단위로 분할 한 후, 문자 단위에서는 각각의 방법으로 인식을 한다. 또한 대부분의 경우 추가되는 전 처리 작업으로 추출해 낸 문자의 크기를 자신들이 정한 규격으로 조절한 후 인식을 한다. 문자 단위 인식의 경우 정해진 규격 크기를 웨이블렛 변환을 하여 인식하거나, 신경망을 통한 인식에서는 문자를 구성하는 화소를 입력으로 하여 인식을 한다[8-11]. 그런데 현재 한글에서 구성할 수 있는 문자의 종류는 11,172자로 문자 단위 인식에서는 인식 대상 문자 수에 따라 인식률과 인식 속도, 메모리 요구량이 크게 달라지는 것으로 알려져 있다. 관련 논문들을 보면 실험에 수만 자를 인식한 경우라도 인식 대상에 중복되는 문자들 때문에 수만 자가 되는 것이고, 문자 별로 분류해 보면 그 가짓수는 조합형 11,172자에 터무니없이 적으며, 사용 빈도가 높은 완성형 2,350자에도 턱없이 부족한 수이다.

자소 단위 인식은 문자에서 자소를 분리하여 분리한 일정한 수의 자소만을 인식하므로 인식해야할 가지 수가 적어 성능의 변화가 상대적으로 적다고 알려져 있다. 물론 자소를 인식한 후에는 문자 구성 형식 정보, 오토마타, 신경망 등과 같은 적당한 방법으로 문자를 인식한다. 자소 단위 인식에서 자소의 분할 연산은 매우 중요한 과정이다. 그림 1은 현재 한글에서 사용하는 자소들을 사용 위치에 따라 분류한 것이다. 자소 단위 인식에서 자소 분리를 하다가 자소의 일부분이 소실될 수도 있고, 자소 구성 영상 이외의 잡영이 포함 될 수도 있다. 자소의 범위를 작게 잡으면 자소의 일부가 소실되고 소실을 막기 위하여 자소의 영역을 크게 잡으면 잡영이 들어오는데 소실과 잡영을 처리하는데 애를 먹기도 한다[7].

구분	내 용	갯수
초성	ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ ㄲ ㄸ ㅃ ㅆ ㅉ	19
중성	ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㅘ ㅝ ㅙ ㅞ ㅐ ㅒ ㅔ ㅖ ㅚ ㅟ ㅢ	21
종성	ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ ㄲ ㅆ ㄳ ㄵ ㄶ ㄺ ㄻ ㄼ ㄽ ㄾ ㄿ ㅀ	27

Fig. 1. Korean consonants and vowels according to the location

자소 분할에 애를 먹는 가장 큰 이유로는 한국어 문법에서 분류하고 있는 초성, 중성, 종성의 개념에서 벗어나지 못하고, 그 개념을 가지고 구성 형식 정보로 추정한 크기를 자소 추출 윈도우로 적용하여 인식하려는 방법이 자소 영역을 잘못 잡아 자소 정보 소실 또는 잡영 추가 문제를 일으킨다. 자소 분할 인식에서는 대부분의 경우 한글을 아래와 같이 6개의 패턴으로 이루어 졌다고 보고 분할을 시도하고 있다[7]. 그러나 수많은 글자들은 인쇄하였을 때 자소가 서로 붙어 인쇄되고 있어 위 패턴에 맞추어 기계적으로 분류하기가 매우 어렵다.

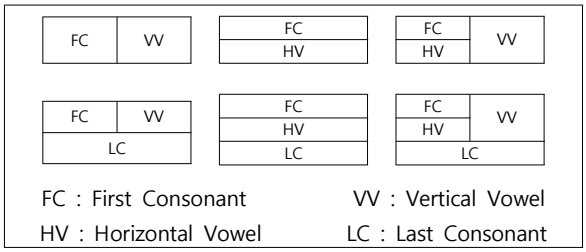


Fig. 2. Korean character syntactic form

맑은 고딕으로 많이 쓰이는 문자로 판명된 완성형 2350자를 인쇄하여 분석해 보면 649자(약 28%)의 문자가 자소가 서로 붙어 있어 자소별 기계적 분할이 어려울 것으로 사료되며, 앞에서 지적한 바와 같이 자소를 분류하기 위한 범위 설정 크기가 알맞지 않으면 인식해야할 자소 일부의 손실과 원치 않는 잡영의 유입 문제가 발생되어 자소 인식률에 많은 영향을 준다.

컴퓨터와 같은 기계를 이용한 문자 인식 자동화는 사람이 발화하는 음운을 기본으로 하는 초성, 중성, 종성 틀이 아닌 기계를 중심의 새로운 자소 인식 방법이 제공되어야 한다. 본 연구에서는 인쇄체 문자에서 자동 자소 단위 한글 인식에 쉬운 자소를 제안하고, 자동 자소 인식 방법을 제안하였다. 본 연구의 가치는 한국어 연구에서 음운론을 기본으로 하는 초성, 중성, 종성의 관점에서의 기존의 연구 방법과 다르게 자동 자소 인식을 유리하게 하기 위한 노력으로 한국어 음운학에서의 기본 자소 뿐 아니라 이들이 결합된 다양한 복합 자소도 새로운 자소로 제안하였고, 이런 자소들을 자동으로 인식하는 방법을 제안하였다. 본 연구에서 사용한 인쇄체는 맑은 고딕으로 하였으며, 본 연구에서 새로운 자소 추출과 새로운 자소에 인식을 위해 분석한 글자는 조합형 11,172자 모든 글자를 대상으로 하지 않고 많이 쓰이는 빈도로 구성된 완성형 2,350자를 인쇄 출력하고 이 인쇄물로 연구를 수행하였다.

II. 본 론

2.1 연구개요

본 연구는 차량 운행 시 안전과 원활한 소통을 위하여 차도 표면에 글자를 쓰는 작업인 도로 노면 문자 도색[13]을 위한 자동화 장비를 만드는 연구 과정에서 도로노면 문자도색을 위한 정자체의 인쇄 한글 인식의 필요로 본 연구를 추가 수행하였다. 정부 관련 기관의 관련 자료를 분석하여 규정상 문제없는 문자체로 인식 대상 문자체를 맑은고딕으로 선택하고 이로 연구를 수행하였다. 최종 목표는 자소 단위 인식 방법을 통한 문자 인식이다. 자소 인식 방법을 선택한 이유는 통 문자 인식 방법은 인식 대상 수가 많아지면 인식률은 떨어지고, 인식 속도는 느려지며, 메모리 요구량 매우 커지기 때문이다.

자소 단위 문자 인식에서 자소의 분할 연산은 매우 중요한 과정이다. 기존 연구에서는 사람이 발화하는 음운을 기본으로

하는 음운학을 바탕으로 초성, 중성, 종성의 틀로 6가지 틀을 만들어 이 틀에 의한 추정 자소 분리를 하다가 자소의 일부분을 소실하여 잘못 문자를 인식 하거나 자소 구성 영상 이외의 잡영을 포함시켜 문자를 잘 못 인식하기도 하였다. 즉 자소의 범위를 작게 잡으면 자소의 일부가 소실되고 자소의 소실을 막기 위하여 자소의 영역을 크게 잡으면 잡영이 들어와 자소 분할에 애를 먹는 것이다. 이런 잘못의 원인은 많은 연구가 자동화의 주역이며 일을 하는 기계 중심으로 한 전략이 아니라 결과물의 소비자인 인간의 행위를 기계에 적용한 전략 때문이다. 따라서 본 연구에서는 자동화 인식의 주체인 컴퓨터를 고려한 자소의 제안과 이 자소를 인식하는 방법을 제안하였다.

2.2 연결 성분 결합을 통해 추출된 자소

앞에서 밝힌 바와 같이 자소 단위 문자 인식에서 자소 분리 프로세스는 매우 중요한 작업이다. 본 연구에서 수행한 자소 분리 프로세스는 그림 3와 같다.

Fig. 3. Grapheme separation process

천연색 영상을 회색 영상으로 바꾸는 작업은 [13]을 참고하였고, 회색 영상을 2진화 영상으로 바꾸는 경계 값은 127로 하였다. 기울어짐 보정과 텍스트 영역 추출, 문자 분리는 [11]을 참고하였고, 자소 추출은 분리된 문자 내에서 연결 성분을 이용한 Two Pass Connected Components Labeling법을 이용하였다. 자소 분리를 위해 사용한 알고리즘은 그림 4와 같고, 동 알고리즘을 적용하여 인쇄체 문자를 분리한 예가 그림 5와 같으며, 추출된 자소들은 그림 6과 같다. 프로그램에서 자소들은 그림 5에서와 같이 먼저 선택되는 자소부터 준비된 색으로 분리되도록 하였으며, 본 연구에서는 그림 7과 같이 제안한 자소들을 분류하였다. 그림 6에서 '겨 in 계...'의 의미는 '계'와 같은 글자들 안에서 '겨'가 하나의 연결 성분으로 추출되어 '겨'를 하나의 자소로 판단하였음을 의미하고 'lower part of 포'는 '포'라는 글자가 'ㅍ'의 위(上) 가로 한 획이 나머지 부분과 분리되어 있어 이것도 하단 부분을 하나의 자소로 잡았음을 의미 한다. 본 연구에 의하여 제안된 자소는 95개 이다.

```
TPCCL( data[rows][cols] ) {
   linked[N] //연결 관계가 표현될 배열.
   num = 0 //새로운 요소에 붙일 라벨 번호 저장
   labels[rows][cols] //라벨링 결과가 들어갈 배열. 0으로 초기화
   < 패스 1 >
   행 우선으로 data 배열 처리 {
      if data[row][col]가 관심 요소라면 {
         //현 요소의 라벨링을 위해 이웃 요소를 검사.
         if 라벨된 이웃이 여럿이라면 {
            label[row][col]에 이웃 중 가장 작은 라벨 값을 저장
            관련된 이웃 라벨끼리 동등함을 linked에 저장
         }
         else //만일 이웃이 없다면
            현 요소에 유일한 라벨을 붙인다.
      }
   }
   < 패스 2 >
   행 우선으로 data 배열 처리 {
      if 요소가 백그라운드가 아니라면{
         labels[row][col]에 패스 1에서 동등함으로 저장된 값 중 가장
작은 라벨로 다시 라벨링 한다.
      }
   }
   return label[rows][cols]
}
```

Fig. 4. Grapheme extract algorithm

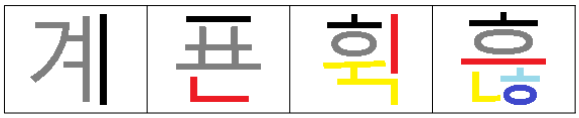


Fig. 5. Grapheme extract results

ㄱ/ㄴ/ㄷ/ㄹ/ㅁ/ㅂ/ㅅ/ㅇ/ㅈ/ㅊ/ㅋ/ㅌ/ㅍ/ㅏ/ㅑ/ㅓ/ㅕ/ㅗ/ㅛ/ㅜ/ㅠ/ㅡ/ㅣ/ㅐ/ㅒ/겨 in 계.../구/국/굳/굴/굵/굼/궁/규/귤/그/거 in 꺼.../꾸/꾹/꿀/꿈/꿍/뀨/끄/노/뇨/ᄀᆞᆨ in 눅.../ᄃᆞᆫ in 눋.../ᄃᆞᆯ in 눌.../ᄃᆞᆷ in 눔.../ᄃᆞᆼ in 눙.../ᄇᆞᆨ in 뉵.../ᄇᆞᆯ in 뉼.../ᄇᆞᆷ in 뉨.../ᄇᆞᆼ in 뉭.../도/됴/로/료/머/며/모/묘/ᄆᆞᆫ in 묻.../ᄆᆞᆯ in 묽.../ᄆᆞᆷ in 묾.../ᄆᆞᆺ in 뭇.../버/벼/보/뵤/ᄇᆞᆺ in 붓.../뽀 in 뽁.../뽀/ᄊᆞ in 싸.../어/여/오/요/ㅉ(고립점유) in 짜.../ㅉ(고립점무) in 쪼.../쿠/쿡/쿨/쿰/쿵/큐/큘/큼/크/토/툐/lower part of 포.../lower part of 표.../ᄑᆞᆺ in 훑...

Fig. 6. The 95 extracted graphemes

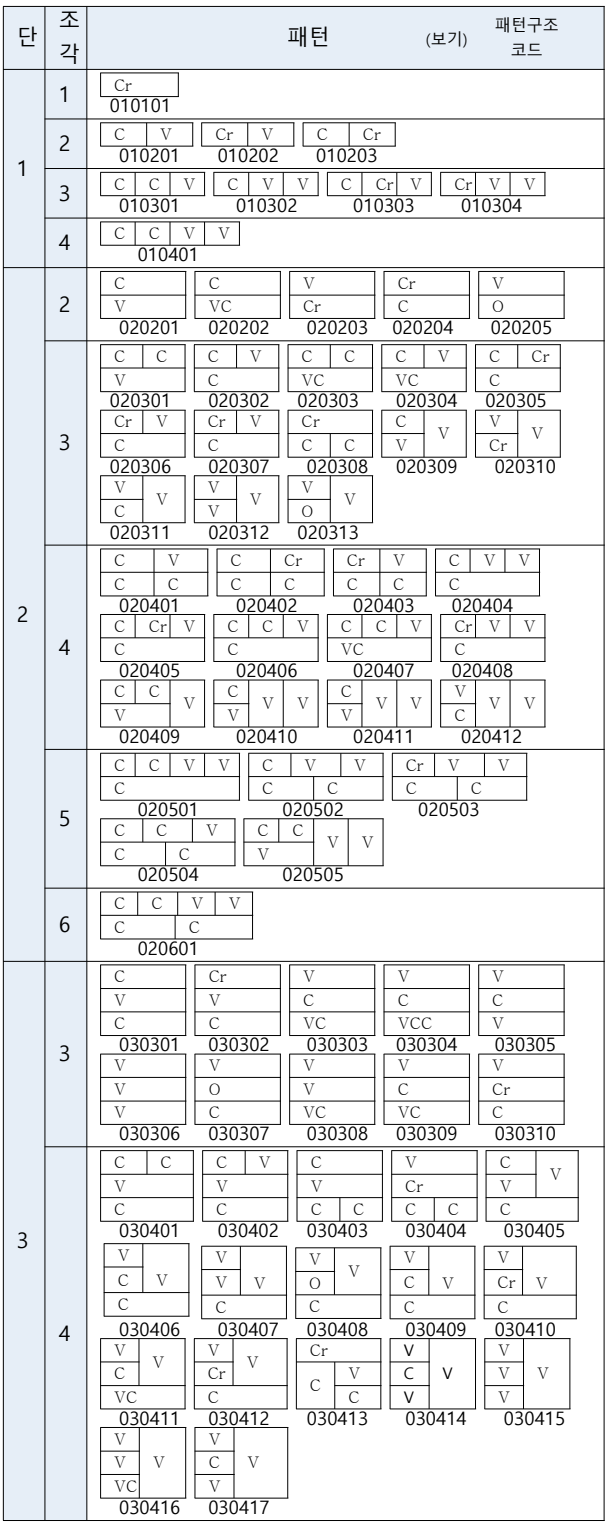
추출된 자소는 ㄱ/ㄴ/ㄷ/...과 같은 자음(Consonant)형 자소(C type으로 분류)와 ㅏ/ㅑ/ㅓ/... 같은 모음(Vowel)형 자소(V type으로 분류), 구/국/굳/...과 같은 문자(Character)형 자소(Cr type으로 분류), ᄀᆞᆨ/ᄃᆞᆫ/ᄃᆞᆯ/... 모음+자음 결합 형태 자소(VC type으로 분류), '포'나 '표'의 하단부와 같은 명칭이 어려운 기타(Other) 자소(O type으로 분류), ᄑᆞᆺ 모음+자음+자음 형태의 자소(VCC type으로 분류)으로 본 연구에서는 분류하였다.

분류	소속 자소
C	ㄱ/ㄴ/ㄷ/ㄹ/ㅁ/ㅂ/ㅅ/ㅇ/ㅈ/ㅊ/ㅋ/ㅌ/ㅍ/ㅆ in 싸/ㅉ(고립점유) in 짜.../ㅉ(고립점무) in 쪼...
V	ㅏ/ㅑ/ㅓ/ㅕ/ㅗ/ㅛ/ㅜ/ㅠ/ㅡ/ㅣ/ㅐ/ㅒ
Cr	겨 in 계.../구/국/굳/굴/굵/굼/궁/규/귤/그/거 in 꺼.../꾸/꾹/꿀/꿈/꿍/뀨/끄/노/뇨/도/됴/로/료/머/며/모/묘/버/벼/보/뵤/뽀 in 뽁.../뽀/어/여/오/요/쿠/쿡/쿨/쿰/쿵/큐/큘/큼/크/토/툐
VC	ᄀᆞᆨ in 눅.../ᄃᆞᆫ in 눋.../ᄃᆞᆯ in 눌.../ᄃᆞᆷ in 눔.../ᄃᆞᆼ in 눙.../ᄇᆞᆨ in 뉵.../ᄇᆞᆯ in 뉼.../ᄇᆞᆷ in 뉨.../ᄇᆞᆼ in 뉭.../ᄆᆞᆫ in 묻.../ᄆᆞᆺ in 붓...
O	lower part of 포.../lower part of 표...
VCC	ᄆᆞᆫ in 묻.../ᄆᆞᆯ in 묽.../ᄆᆞᆷ in 묾.../ᄑᆞᆺ in 훑...

Fig. 7. Grapheme classification

2.3 추출 자소를 이용한 문자 패턴

앞의 2.2에서 추출한 자소를 이용하여 문자 패턴을 분류하면 1단 1조각 문자 패턴부터 5단 6조각 패턴까지 다양한 패턴이 존재한다. 그림 8은 본 연구에서 분류한 자소에 의해 구성되는 문자 패턴을 나열한 것이다. 패턴의 하단에 숫자는 패턴에 부여한 코드이며, 패턴에 의한 코드별 소속문자는 그림 9와 같다.



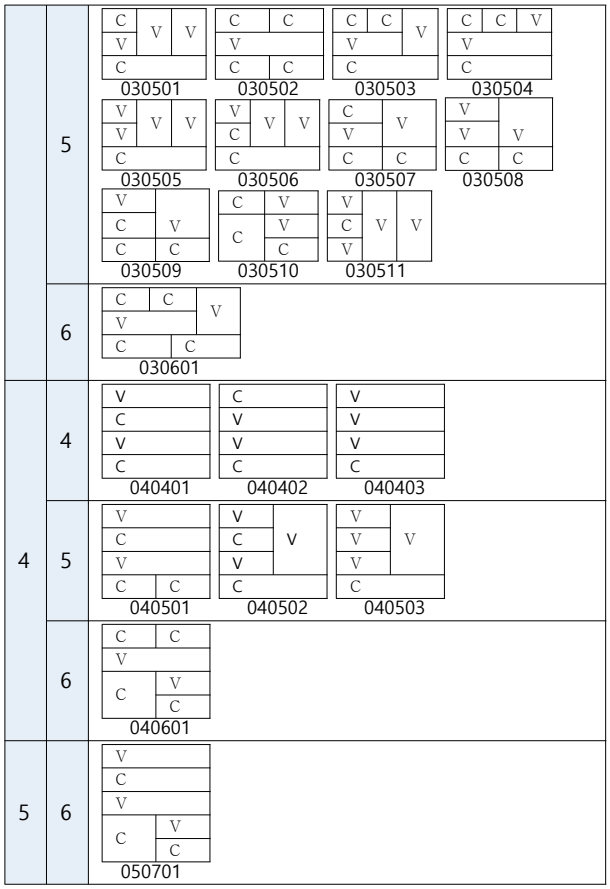


Fig. 8. Korean character syntactic form in new grapheme

코드	소속 문자
010101	구규그꾸뀨끄노뇨도됴로료머며모묘버벼보뵤뽀어여오요쿠큐크토툐국굳굴굵굼궁귤꾹꿀꿈꿍쿡쿨쿧쿵큘큠
010201	가개갸걔거겨기나내냐너녀니다대댜더뎌디라래랴러려리마매먀미바배뱌비사새샤서셔시싸쌔써씨아애야얘이자재쟈저져지짜째쨔쩌쪄찌차채챠처쳐치카캐캬커켜키타태탸터텨티
010202	계놔궈꿔뀌뇌돠돼되롸뢰메몌뫄뫼베볘봐봬뵈에예와왜외쿼퀴톼퇘퇴쾍괼귐뀔뀀퀵퀄큄
010203	꺼껴뻐뼈
010301	까깨꺄끼따때떠띠빠빼뺴삐
010302	네녜데뎨레례세셰쎄제졔쩨쳬케켸테톄
010303	께꼐뻬
010304	게궤귀꿰퀘
010401	떼뗘
020201	고고교누뉴느두듀드루류르무뮤므부뷰브소쇼수슈스우유으조죠주쥬즈쪼쭈쮸쯔초쵸추츄츠코쿄투튜트
020202	눅눋눌눙눔눞뉼뉨뉴뉳둑둘둠둥듈듐듀듋륙룰룸룽륙률륨륭묵묶문물묽묾몸뭉뭍뭇붙불붉붊붕붙붚붤뷤뷸뷺숙숟술숨숭숱슐슉슐슘슝욱울욹욺움웅육율윰융죽줄쥵줌중쥴쥼쭉쭌쭐쭘쭝축출춤충츌춈충툴톰통튤튬튱
020203	호효
020204	꾼꿉꿋끅끈끌끔끕끗끙끝녹논놀놈놉뇻농눔뇩뇬뇰뇹뇻뇽독돈돋돌돔돕돗동돛돝뚝론롤롬롭롯롱먹먼멀멈멉멋멍멋며면멸몃목몬몰몸몹못몽묜묠묩묫벅번벋벌범법벗벙벚벽변별볍볏병볕복본볼봄봅봇봉뵨뽁뽄뽈뽐뽑뽕뽱돌룐룔룩억언얻얼엄업엇엉엊엌엎역연열염엽엿영옅옆옥온올옴옵옷옹옻욕욘욜욤욥옷용쿤쿱쿳큭큰클큼큽킁톡톤톨톰톱톳통툔군굽굿궂균극근귿글금급긋긍꾼큔
020205	포표
020301	꾜꾜또뚜뜨뽀뿌쀼쁘쏘쑈쑤쓰
020302	각간갇갈감갑갓강갖갗같갚갞객갠갤갬갭갯갱갹간갈갓걍걔갠걱건걷걸검겁것겅겆겉겊격견겯결겸겹겻경곁긱긴긷길김깁깃깅깆깊낙난낟날남납낫낭낯낱낳낵낸낼냄냅냇냉냑냔냘냠냥넉넌널넘넵넛넝녁년녈념녑닉닌닐녕녁님닙닛닝닢닥단닫달담답닷당닺닻댁댄댈댐댑댓댕덕던덛덜덤덥덧덩덫덮뎐뎔뎡딕딘딛딜딤딥딧딩딪락란랄람랍랏랑랒랖랙랜랠램랩랫랭략랸럇량럭런럴럼럽럿렁력련렬렴렵렷령릭린릴림립릿링막만맏말맘맙맛망맞맡맥맨맬맴맵맷맹맺맻막말먕믹민믿밀밈밉밋밍및밑박반받발밤밥밧방밭백밴뱔뱀뱁뱃뱅뱋뱍뱐뱝빅빈빌빔빕빗빙빚빛삭산삳살삼삽삿상삳색샌샐샘샙샛생샥산샬샴샵샷샹샌샐샘생석선섣설섬섭섯성섶셕션셜셤셥셧셩식신싣실심십싯싱싶싹싼쌀쌈쌉쌍

	쌕쌘쌜쌤쌥쌩썅썩썬썰썸썹썽씩씬씰씸씹씻씽악안알암압앗앙앝앞액앤앨앰앱앳앵약얀얄얌얍얏양얕얜얠얩익인일임입잇잉잊잎작잔잗잘잠잡잣장잦잭잰잴잼잽잿쟁쟉쟌쟐쟘쟝쟨쟬적전절점접젓정젖젼졀졈졉직진짇질짐집짓징짖짙짚짝짠짤짬짭짯짱짹짼쨀쨈쨉쨋쨍쨘쨩쩍쩐쩔쩜쩝쩟쩡찍찐찔찜찝찡찢착찬찰참찹찻창찾책챈챌챔챕챗챙챤챨챰챵척천철첨첩첫청쳔칙친칟칠침칩칫칭칵칸칼캄캅캇캉캑캔캘캠캡캣캥캭캵컥컨컫컬컴컵컷컹켠켤켬켭켯켱킥킨킬킴킵킷킹탁탄탈탐탑탓탕택탠탤탬탭탯탱턍턱턴털텀텁텃텅텬틱틴틸팀팁팃팅
020303	뚝뚤뚬뚱뿍뿔뿜뿡쁑쑥쑬쑴쑹쓩툭
020304	뉠뉨뒬뒹뤽뤌륌륑뮐뷕뷜뷩쉭쉴쉼슁윅윌윔윙쥑쥘쥠췰췸췽튁튈튐튕
020305	꺽껀껄껌껍껏껑껸꼘꼇꼍뻑뻔뻗뻘뻠뻣뻥뼉뼘뼙뼛뼝
020306	꾑놘놜돤돨된될됨됩됫롼뢍뢴뢸룀뢥룃룅멕멘멜멤멥멧멩몡몣뫈뫙묀묄묍묏묑벡벤벧벨벰벱벳벵봔뵉뵌뵐뵘뵙엑엔엘엠엡엣엥옌옐옘옙옛왁완왈왐왑왓왕왝왠왬왯왱외왼욀욈욉욋욍퀀퀄퀑퀸퀍퀏퀭퇀퇸툇툉겐겔겝겟궈귓궉권궐궝궨궙궛꿜꿩뀐뉜뉠뉨뉩뉫벤
020307	끎놂돎롭롯룡멂뭇뭄벎볐볶욶읅읆읎었역엷엶엾였옭옮옰꿇긁몄
020308	과긔괘괴눠뉘늬둬뒤듸뤄뤼뭐뮈붜뷔솨쇄쇠숴쉬워위의좌좨죄줘쥐쫘쫴쬐쭤촤최춰취콰쾌쾨틔
020309	화헤홰회
020310	하허해햐허혀히
020311	파패퍄퍼펴피
020312	퐈푀
020401	갉갊읽값갔갰겂겄겪겼긺낚낡낢났냈넋넒넓넜녔닒닦닭닮닯닸댔덖덞덟덮딨랐랬렀렸맑맒맸밂밌밖밗밝밞밟뱄빎삯삵삶샀샜섞섟섦섧섰셨쌌쌋쌨쎎썼앉앍앎았앴얇읽읾있잖잚잤잿젊졌짊짢짧짰쨌쩠쪘찼챘첬쳤칡캤컸켰탉탔탰텳텄텼
020402	꺾꼈꼈뻤뼜
020403	놨됐뢨멨뫘벴봤뵀옜왔궜꿨
020404	섹센셀셈셉셋셍셴셸솅쎅쎌쎈젝젠젤젬젭젯젱쩽첵첸첼쳄쳅쳇쳉쳰쳴켁켄켈켐켑켓켕텍텐텔템텝텟텡톈겐겔겜겝겟겡넥넨넬넴넵넷넹녠덱덴델뎀뎁뎃뎅뎬렉렌렐렘렙렛렝롄롑롓
020405	뻥껙껜껨껫껭
020406	깍깐깔깜깝깟깡깥끽낀낄낌낍낏낑깩깬깰깸깹깻갱꺅꺌딱딴딸땀땁땃땅땃땍땐땔땜땝땟땡떡떤떨떰떱떳떵띤띨띰띱띳띵빡빤빨빰빱빳빵빽뺀뺄뺌뺍뺏뺑뺙뺨삑삔삘삠삡삣삥
020407	뙬뙴뙹
020408	궷꿱꿴꿸꿼꿽퀑
020409	꽈꽤꾀똬뙈뙤뛰띄뾔쏴쐐쐬쒀쒸씌쮜퉈튀
020410	눼뒈뭬붸쉐웨줴췌퉤
020411	페폐
020412	혜
020501	떽뗀뗄뗌뗍뗏뗑
020502	겠셌넸뎄뎠
020503	꿨
020504	꺆깖깠깼땄떪떫떴뗬빪빴뺐쐈쨌춌
020505	뛔뤠쒜
020601	뗐
030301	곡곤곧골곰곱곳공곶곤골곱곳눈눕눗눕늑는늘늠늡늣능늦늪둔둡둣둔득든듣들듬듭듯등룬룹룻륜륩륫륵른를름릅릇릉릊릍릎문뭅뭇뮨뮷므믈믐믓분붑붓뷴븃븍븐블븜븝븟속손솔솜솝솟송솥슉숀숄숌숍숏숑순숩숫숯숫슥슨슬슴습슷승운웁웃윤윱윳윷윽은을음읍읏응읒읓읔읕읖족존졸좀좁좃종좆좇죡죤죵준줍줏쥰즉즌즐즘즙즛증쪽쫀쫄쫌쫍쫏쫑쫓쫑쭌쭙쯤쯧쯩촉촌촐촘촙촛총춈춘춥춧츈측츤츨츰츱츳층콕콘콜콤콥콧콩툰툽툿튠특튼튿틀틈틉틋
030302	농멍영
030303	훅
030304	훝
030305	흐후휴흐
030306	푸퓨프
030307	폭폰폴폼폽폿퐁푠퓰퓹퓻
030308	푹푼풀풂품풍퓰픔픙
030309	훌훑훔훙훅휼흄흉
030310	혹혼홀홈홉홋홍홑횬횰횹횻
030401	꼭꼰꼴꼼꼽꼿꽁꽂꽃똑똔똘똥뚠뜩뜬뜯뜰뜸뜹뜻뿐뿟쁜쁠쁨쁩쏙쏜쏟쏠쏨쏩쏭쑨쑵쓱쓴쓸씀씁
030402	강겅넝당랑렁망쌍양쩡
030403	곪곬곯늙늚듦숚숢슭읊좖틂
030404	훑
030405	곽관괄괌괍괏광괜괠괩괭괵괸괼굄굅굇굉뉜뉩뉜뉠뒨뒵뒷륀륏뮌뮐뮙뮛뮌뷜뷘솩솬솰솽쇈쇌쇔쇗쉰쉴쉼쉽쉿쉰쉽쉿웍원월웜웝웡윈윕윗윈윌윔윗좍좔좝좟좡좽죈죌죔죕죗죙쥔쥡쥣쫙쫠쫠쬔쬘쬘쬠쬡쭹촨촬촹쵠쵤쵬쵭쵯쵱췬췹췻콱콴콸쾀쾅쾡쾰퇸퇼튄튈튐튑
030406	학힉힌힐힘힙힛힝
030407	팍판팔팜팝팟팡팥팩팬팰팸팹팻팽퍅퍽펀펄펌펍펏펑편펼폄폅평픽핀필핌핍핏핑
030408	퐝푄
030409	학한할함합핫항핵핸핼햄햅햇행향헉헌헐험헙헛헝혁현혈혐협혓형
030410	확환활홧황홱홴횃횅획횐횔횝횟횡
030411	휙휠휨휭
030412	헥헨헬헴헵헷헹
030413	굶끓욺
030414	희
030415	풔퓌
030416	퓔퓜
030417	훠휘희
030501	뒝쉑쉔쉘쉠쉥웩웬웰웸웹웽췐
030502	쑮씂
030503	꽉꽐꽝꽥꽹꾄꾈꾐꾑꾕똴뙨뛴뛸뛰띈띌띔띕쏵쏸쐰쐣쐴쐼쐽쒼씐쓸씜
030504	땋떻빻
030505	펙펜펠펨펩펫펭폘폡폣
030506	혤혭
030507	괆괬눴뒀뤘뷨쇘쉈웠좼줬쭸퉜
030508	퍆팖팠팼펐폈
030509	핥했헒혔
030510	많앓닳싫않잃쟎찮챦
030511	훼
030601	꽜
040401	훈훗휸휷흑흔흗흘흠흡흣흥흩
040402	뭉웅좋
040403	푼풉풋퓬퓻픈플픔픕픗
040501	흙흫흙
040502	훤훨훰훵훽휀휄휑휜휩휫흰흰힐힐힘힘힙힙힝힝
040503	퓡퓐퓟
040601	꿇뚫쓿
050601	흉

Fig. 9. Classified korean character

2.4 자소 자동 인식을 위한 자소 특징 점 추출

95개나 되는 자소를 자동 인식하기 위해 자소를 구분하기 위한 특징 점 추출이 필요하다. 자소 인식을 위한 특징 점은 한글의 구조를 고려하여 설정하였다. 한글은 선과 선의 결합, 원과 사각형의 구조를 가지고 있다. 따라서 본 연구에서는 자소 내에서 특징 점으로 끝 점, 2선 모임 점, 3선 이상 모임 점, 고립 점으로 구성하였다. 끝 점, 2선 모임 점, 3선 이상 모임 점의 특징은 자소를 세선화 한 후 추출해 내었고, 고립점은 역세선화 후 찾아내었다.

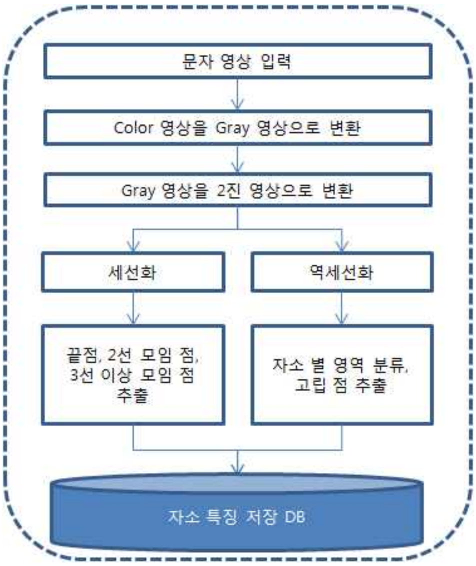


Fig. 10. Feature extraction process

끝점 추출은 아래 8패턴 중 하나와 일치점으로 하였으며, 끝점에 8방향성 정보를 추출할 수 있으니, 방향성 정보와 특징 점

의 종류, 글자 최인접 사각형 크기 정보와 이 안에서의 3x3 위치 정보가 저장되었다.

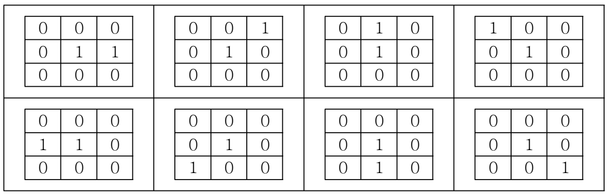


Fig. 11. Mask for end point

2선 모인 점은 위와 같은 패턴을 5x5 행렬로 구성하되 중심 점은 무관조건으로 하고, 2패턴의 or 결합 형태로 2선 모임점을 추출하되 패턴들을 동쪽 방향에서 부터 반시계 원형으로 나열하였을 경우 1 또는 2칸 거리의 패턴들의 결합만 허용하였다. 또한 끝점과 같은 형태의 정보들이 추출되어 저장되었다.

3선 이상 모임 점은 아래 패턴을 번호순으로 원형 나열하고, 16패턴 중 각각이 양 방향으로 거리가 3이상 차이 나는 3개 또는 4개의 패턴을 or 연산으로 결합하여 3선 모임 점과 4선 모임 점을 찾았고 앞에서와 같이 관련 정보를 저장하였다.

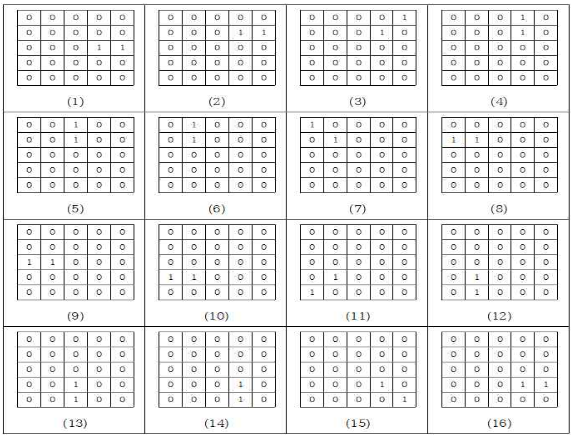


Fig. 12. Mask for three lines or more meeting point

요소	구역별 특징 점 분포표											
	끝점			2선 모임			3선 이상			고립점		
ㅇ												
											1	
ㅁ				1		1						
											1	
				1		1						
ㄱ	1					1						
			1									
ㄴ	1											
			1	1								
ㅂ	1		1									
							1		1			
				1		1					1	
ㄷ			1	1								
			1	1								
ㅂㅁ	1		1					1				
				1		1		1				
				1		1					1	
머			1	1	1							
								1	1	1		
			1	1	1							
모				1		1					1	
				1		1		1				
	1		1					1				
ㅍㅁ	1		1				1		1			
				1		1	1		1			
				1		1					1	
며			1	1	1			1	1			
								1	1	1		
			1	1	1							
묘				1		1					1	
				1		1	1	1				
	1		1				1	1				
ㄹ	1					1						
				1		1						
			1	1								
ㅎ	1		1					1				
								1				
											1	
어			1									
								1	1	1		
			1									
오											1	
								1				
	1		1					1				
여			1					1	1			
								1	1	1		
			1									
요											1	
							1	1				
	1		1				1	1			1	
ᄒ	1		1				1		1		1	
							1		1			
											1	
ㅡ												
	1		1									
ㅣ		1										
		1										
ㅋ	1					1						
	1								1			
			1									
그	1					1						
	1		1						1			
ㅌ			1	1								
			1				1					
			1	1								
ㄱㅁ	1					1						
	1		1	1		1		2	1			
				1		1					1	
ㅅ		1										
								1				
	1		1									

ㅏ	1											
			1				1					
	1											
ㅓ			1									
	1								1			
			1									
ㅗ		1										
	1		1					1				
ㅜ	1		1					1				
		1										
와	1											
	1		1					2	1			
											1	
구	1					1						
	1		1					1	1			
		1										
거	1		1		1							
								1	1			
	1		1									
노	1											
			1	1				1				
	1		1					1				
과	1		1					1				
	1					1		1				
			1									
크	1					1						
	1								1			
	1								1			
뉴	1											
			1	1			1	1				
	1		1				1	1			1	
국	1		1				1	1				
	1					1	1	1			1	
			1									
왜	2					1			1			
	1		1					2	1			
											1	
끄	1	1			1	1						
	1		1				1		1			
리	1		1					1				
			1	1				1				
			1	1								
너			1	1								
			1	1				1				
	1		1					1				
왕	1	1			1	1						
	1		1					2	1			
								1			1	
뇨			1	1								
			1	1			1	1				
	1		1				1	1			1	
또	1	1	1									
	1						1	2	1			
			1	1	1					1		
보	1		1				1		1			
				1		1					1	
	1		1									
뚀	1	1	1									
							1	3	2			
			1	1	1					1		
뵈	1		1				1		1			
				1		1	1		1		1	
	1		1				1		1		1	

뉘	1		1					1				
	1					1		1				
			1	2		1						
로	1			1		2						
				1				1				
	1		2					1				
료	1			1		2						
				1			1	1				
	1		2				1	1			1	
뢰	2		1					1	1		1	
						1		1	1			
			1	2		1						
쿰	2					1			1			
	1		1	1		1		2	1			
				1		1					1	
눼	1		1					1				
	1			1	1	1		3				
		1		1	1	1						1
뭄	1	1			1	1						
	1		1	1		1		2	1			
				1		1		1			1	
즈	1		1					1				
								1				
	1		1									
샤	1		1				1					
			1				1					
	1											
쉬	1		1						1			
	1								1			
			1									
쓰	1		1									
	1		1				1	1				
끄	1		1				1		1			
	1		1									
쁴	1		1									
							1		1			
	1		1									
프	1		1				1		1			
											1	
	1		1				1		1			
쁴	1		1				1		1			
							1		1		1	
	1		1									
뉴	1		1					1				
	1					1		2				
		1	1									
겨	1		1		1			1	1			
		1							1			
	1		1									
규	1					1						
	1		1				1		2			
		1	1									
큐	2					1			1			
	1		1					1	1			
		1										
국	1					1						
	2		1					2	1			
			1			1						
꾸	1	1			1	1						
	1		1					2	1			
		1										
뉠	1		1					1				
			2	1			1	1				
			1	1								

토			2			1	1						
			1			1		1					
	1		1					1					
툐			2	1									
			1	1									
	1		1									1	
굳	1					1							
	1		2	1				2	1				
			1	1									
뉘	1		1					1					
	1			1	1	1		2					
		1	1	1									
굴	1					1							
	2		1			1		2	1				
			1	2		1							
츠	1	1	1					1					
								1					
	1		1										
쓰	1		1										
							1		1				
	1	1	1					1					
쯔 (l)	1		1				1		1			1	
							1		1				
	1	1	1					1					
쿡	2					1			1				
	2		1			1		2	1				
			1										
큐	2					1			1				
	1		1				1		2				
		1	1										
뀨	1	1			1	1							
	1		1					2	2				
	1		1										
뽀	1	2	1				1	2	1				
				1		1		3		1		1	
	1		1					1					
꾹	1	1			1	1							
	1		1					2	1				
	1		1			1		1					
뾰	1	2	1				1	2	1				
				1	2	1		2		1		1	
	1		1					2					
뀱	1					1							
	2		1			1		3					
		1	1	2	1								
쿨	2					1			1				
	2		1			1		2	1				
			1	2		1							
꿀	1	1			1	1							
	2		1			1		3	1				
			1	2		1							
표 (L)	1		1										
	1		1				1		1				
	1		1				1		1			1	
쯔	1		1				1		1				
							1		1				
	1	2	1										
포 (L)	1		1										
	1		1				1	1	1				
	1		1					1					
푯	1		1					1					
	1		1					3					
	1		1					2				1	
퓟	1		1					4					
	1		2	1	1			1					
		1	1	1	1								

L:하단부분 l:고립점있음.

Fig. 13. Per grapheme features distribution

저장된 자소별 추출 특징 점은 앞의 설명과 같이 저장하되 끝 점, 2선 모임 점, 3선 이상 모임 점과 고립점은 자소의 가로와 세로를 3대 3으로 구역으로 나누었을 때 각 추출된 위치에 따라 그림 13과 같이 저장하여 인식에 이용하였다. 이렇게 저장하였을 때 저장 모습이 자소의 모습을 유추할 수 있었다.

2.5 자소 인식 실험

본 연구에서 천연색 영상을 회색 영상으로 바꾸는 작업은 [13]을 이용하였고, 회색 영상을 2진화 영상으로 바꿀 때 경계값은 127로 하였고, 기울어짐 보정과 텍스트 영역 추출, 문자 분리는 [11]을 참고하였다. 글자의 크기는 20포인트 글자 크기로 가정하여 글자의 높이가 100픽셀이 되도록 영상 크기를 조정하였다. 분리된 문자에서 자소 추출은 자소 범위 내에서 연결 성분을 이용한 추출로 수행하였고, 앞의 그림 6에서 보인 바와 같이 글자에서 자소는 완벽히 분리 추출됨을 확인할 수 있었다.

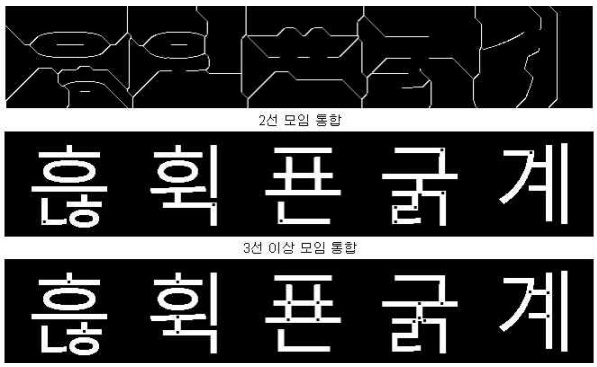


Fig. 14. Grapheme recognition test image

그림 14는 인식 실험을 위한 변환 과정과 특징점 추출과정이다. 자소 인식 실험에는 조합형 11,172개 중 난수 발생을 통해 무작위로 문자를 추출하도록 하여 350개 글자로 실험을 수행하였고 350개 글자에는 중 자소 1,125개가 있었으며, 실험에서는 자소 1125개 중 8개의 특징 점을 찾지 못하였고, 3개의 잘못된 특징 점을 추출하였다. 잘못된 특징 점을 찾는 경우는 오른쪽 삐침 부분이 직선으로 세선화 되지 못하고 한 번 절곡되어 세선화 될 경우에 나타났으며, 찾지 못한 특징 점은 2선 모임점이 부드럽게 세선화 될 경우 발생하였다.

자소 인식은 그림 13에서 보인 특징 점 정보를 데이터베이스화하여 끝 점의 수, 2선 모임 점의 수, 3선 이상 모임 점의 수, 고립점의 개 수 정보로 1차 분류한 후 각 특징 점의 분포를 이용하여 2차 분류하였다. 데이터베이스의 저장 값에 일치 하지 않을 경우 일치하지 않은 특징 점과 데이터베이스의 특징 점의 상대 거리가 작은 쪽의 자소로 판단되도록 하였고, 실험에서는 1,125개의 자소 중 14개를 다른 자소로 판단하였다.

앞의 상황을 참고하건데 세선화 과정 후 글자 구조를 감안한 적절한 세선화 글자의 스무딩 과정을 고안하거나 또 다른 연구에서와 같이 자소 분리를 한다면 세선 후 자소 분리 과정을 연구해 보면 어떤 방법이 도출 될 것으로 사료된다.

III. Conclusions

본 연구에서는 문자 인식 방법 중에서 자소 인식을 이용한 문자 인식 방법에서의 자소 인식의 어려움을 해소하기 위한 새로운 자소 인식에 관한 연구를 수행하였다.

기존 자소 인식을 이용한 문자 인식 연구에서는 자소 인식에 한국어 음운학에서 분리한 초성, 중성, 종성의 자소들을 인식 단위로 하여 인식을 하려고 시도한 것과 다르게 본 연구에서는 인쇄체 문자의 연결 성분을 그대로 인정한 새로운 자소를 구성하였고 제안하여 자소 분리에 매우 유리함을 보였다. 또한 이 자소들을 인식하기 위한 특징 점들을 한글의 구조를 감안하여 끝 점, 2선 모임 점, 3선 이상 모임 점, 고립점으로 제안하고 각 특징 점들을 추출하는 방법을 제시하였다. 또한 특징 점의 추출 위치도 함께 저장하고 본 연구에서 제안한 자소들을 인식하는 실험을 수행하였다. 본 연구에서는 비록 높은 인식률을 얻기는 하였으나 제안한 방법의 알고리즘 수행시간을 줄이려는 노력과 오인식과 미인식 특징 점들을 위한 개선 노력이 필요하다.

또한 본 연구의 후속 연구로는 본 연구에서의 인식한 자소들로 문자를 인식하기 위한 연구가 필요하다.

REFERENCES

[1] Hyuna Oh, EuGene Rhee, "Enhancement of Car Licence Plate and Security with Rotation Algorithm", Journal of Security Engineering, Vol. 13, No. 2, pp. 83~90, Apr. 2016

[2] Moon Yong Jin, Jong Bin Park, Dong Suk Lee, "Real-Time Vehicle Licence Plate Recognition System Using Adaptive Heuristic Segmentation Algorithm", KIPS Tr. Software and Data Eng., Vol. 3, No. 9, pp. 361-368, Mar. 2014

[3] Kang-San Kim, Seok-Cheon Park, Seok-Ho Oh, "Suggestion of Enhanced Korean Character Recognition Technique Using Google Tesseract Open API", Proceeding of Korean Society for Internet Information, Vol. 16, No. 1, Spring. 2015.

[4] Kyusoo Choung, "Text Area Detection of Road Sign Images based on IRBP Method", Journal of Intelligent Transportation System, Vol. 13, No. 6, Dec. 2014.

[5] Kyung-Wha Park, Byoung-Hee Kim, Dong-Sig Han, Seong-Ho Son, Woo-Yung Kang, Byoung-Tak Zhang, “Handwritten Hangul Recognition using Multi-column Deep Neural Networks", Proceeding of KIPS Spring Conference, Vol. 26, No. 1, 2016.

[6] Min-Soo Kim, Eun-Young Kang, Woo-Sung Kim, Sun-Hwa Han, Jin-Hyung Kim, "A Study on Implementation of Printed Character Recognition System And Performance Evaluation", Korea Information Processing Society, Vol. 7, No. 11, pp. 3584-3591, Nov. 2000.

[7] Kil Taek Lim, Ho Yon Kim, "A Study on Machine Printed Character Recognition Based on Character Type Classification", The Institute of Electronics and Information Engineers, Vol. 40, No. 5, pp. 26-39, Sep. 2003.

[8] Kil-Taek Lim, Gi-Seok Kim, "Reestimation of

Recognition Result of MLP Classifier for Machine Printed Hangul - Feasibility Study on Softmax Method", Journal of Information & Electronic Technology, Vol. 6, pp. 93-105, 2007.

[9] Kil Taek Lim, Ho Yon Kim, "A Study on Machine Printed Character Recognition Based onCharacter Type Classification", Journal of IEIE, Vol. 40, No. 5, pp. 266-279, Sep. 2003.

[10] Kil Taek Lim, Seon Hwa Jeong, Seung Ick Jang, Ho Yon Kim, "An Implementation Method of the Character Recognizer for th Sorting Rate Improvement of an Automatic Postal Envelope Sorting Machine", Journal of Korea Society of Industrial Information System, Vol. 12, No. 4, Dec. 2007.

[11] Duk-Ryong Lee, Woo-Youn Kim, Il-Seok Oh, "A Hangul Document Image Retrieval System Using Rank-based Recognition", Journal of The Korea Contents Association, Vol. 5, 2005.

[12] Kyong-Ho Lee,Jae-Joon Seong, “Study on Automation about Painting the Letters to Road Surface”, Proceeding of Korea Society of Computer Information(Summer), Vol. 24, No. 1, pp. 113-116, Summer 2016.

[13] A. A. Gooch, S. C. Olsen, J. Tumblin, and B. Gooch, "Color2gray: Salience-preserving color removal", ACM Transactions on Graphics, vol. 24, no. 3, pp. 634 - 639, 2005.

Authors

Kyong-Ho Lee received the B.S. degree in Computer Science from Korea National Open University and the M.S. degree in Information and Communication Engineering from Korea Advanced Institute of Science and Technology and the Ph.D. degrees in Electronic Engineering from Dankook University, Korea, in 1991, 1994 and 2008, respectively. Dr. Lee is currently a Professor in the School of Information & Communication, Broadcasting Engineering, Halla University. He is interested in HCI.