

한글 음식 메뉴 인식을 위한 OCR 기반 어플리케이션 개발

이규철¹ · 유지상^{2*}

Development an Android based OCR Application for Hangul Food Menu

Gyu-Cheol Lee¹ · Jisang Yoo^{2*}

^{1,2*}Department of Electronic Engineering, Kwangwoon University, Seoul 01897, Korea

요 약

본 논문에서는 스마트폰으로 음식 메뉴를 촬영한 영상으로부터 글자를 인식하는 안드로이드 기반의 한글 음식 메뉴 인식 어플리케이션을 설계하고 구현한다. Optical Character Recognition (OCR) 기술은 크게 전처리, 인식 그리고 후처리 과정으로 구분된다. 전처리 과정에서는 Maximally Stable Extremal Regions (MSER) 기법을 이용하여 글자를 추출한다. 인식 과정에서는 무료 OCR 엔진인 Tesseract-OCR을 이용하여 글자를 인식한다. 후처리 과정에서는 음식 메뉴에 대한 사전 DB를 이용하여 잘못된 결과를 수정한다. 제안하는 기법의 성능을 평가하기 위해 실제 메뉴판을 DB로 이용하여 인식 성능을 비교 측정하는 실험을 진행하였다. 구글 플레이스토어에 있는 글자 인식 어플리케이션인 OCR Instantly Free, Text Scanner 그리고 Text Fairy와 인식을 측정 실험을 진행하였으며 실험 결과 제안하는 기법이 다른 기법보다 평균적으로 14.1% 높은 인식률을 보여주는 것을 확인하였다.

ABSTRACT

In this paper, we design and implement an Android-based Hangul food menu recognition application that recognizes characters from images captured by a smart phone. Optical Character Recognition (OCR) technology is divided into preprocessing, recognition and post-processing. In the preprocessing process, the characters are extracted using Maximally Stable Extremal Regions (MSER). In recognition process, Tesseract-OCR, a free OCR engine, is used to recognize characters. In the post-processing process, the wrong result is corrected by using the dictionary DB for the food menu. In order to evaluate the performance of the proposed method, experiments were conducted to compare the recognition performance using the actual menu plate as the DB. The recognition rate measurement experiment with OCR Instantly Free, Text Scanner and Text Fairy, which is a character recognizing application in Google Play Store, was conducted. The experimental results show that the proposed method shows an average recognition rate of 14.1% higher than other techniques.

키워드 : OCR, 문자 인식, 한글 인식, 영상 분할, MSER

Key word : OCR, Text recognition, Hangul recognition, Image segmentation, MSER

Received 27 December 2016, Revised 30 December 2016, Accepted 14 February 2017

* Corresponding Author Jisang Yoo(E-mail:jsyoo@kw.ac.kr, Tel:+82-2-940-5112)

Department of Electronic Engineering, KwangWoon University, Seoul 01897, Korea

Open Access <https://doi.org/10.6109/jkiice.2017.21.5.951>

print ISSN: 2234-4772 online ISSN: 2288-4165

©This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright © The Korea Institute of Information and Communication Engineering.

I. 서 론

현재 방한 외래객은 약 1,400만 명으로 매년 평균 약 9%의 성장률을 보이고 있다[1]. 문화체육관광부 자료에 따르면 중국인 관광객의 비중이 2010년도부터 꾸준히 증가하여 전체 방한 외래객 중 약 40%를 차지할 뿐만 아니라 국적 별 평균 지출 금액도 2,271달러로 1위를 차지하고 있다[2]. 관광객들이 겪는 주요 불편사항으로는 언어 소통이 가장 크게 대두되고 있고, 특히 음식점에서 음식 메뉴판에 그림 설명 없이 한국어로 되어있는 음식 메뉴를 접했을 때의 어려움을 토로하고 있다[3]. 이와 같은 문제를 해결하기 위해 여러 가지 방법들이 연구되고 있으며, 그중에서도 스마트폰 환경에서 Optical Character Recognition (OCR) 기술을 활용하여 텍스트를 인식하는 분야가 활발히 연구되고 있다[4].

OCR 기술은 패턴 인식의 한 분야로, 사람이 쓰거나 기계로 인쇄한 문자의 영상을 인식하여 기계가 읽을 수 있는 문자로 변환하는 것을 의미한다[5]. OCR 기술은 크게 필기체로 작성된 글자를 인식하는 필기체 인식 방법과 폰트로 작성된 글자를 인식하는 인쇄체 인식 방법으로 나눌 수 있다[6]. 필기체 인식 방법은 사람마다 다양한 필체가 존재하기 때문에 문자의 형태를 인식하는 것이 쉽지 않다. 반면에 인쇄체 인식 방법은 폰트가 정형화되어 있기 때문에 필기체 인식 방법보다 높은 인식률을 보이고 있고 많은 기업에서 상용화 제품을 출시하고 있다.

한글 음식 메뉴를 인식하는 과정은 크게 전처리, 인식 그리고 후처리 과정으로 나눌 수 있다[7]. 전처리 과정은 글자 인식 성능을 향상시키기 위하여 입력된 영상을 보정하는 단계이다. 전처리 기술로는 배경 제거, 이진화, 블러 제거 그리고 기울임 보정 기술 등 다양한 기법들이 존재한다.

배경 제거 기술은 인식하고자 하는 글자에 배경이 겹쳐서 존재하는 경우 인식을 저하의 원인이 되기 때문에 blob 검출을 통해 배경을 제거함으로써 인식 성능을 향상시키는 전처리 기술이다[8]. 이진화 기술은 임계값을 이용하여 입력 영상을 0과 1로 이루어진 이진화 영상으로 만드는 기술이다. 이 기술은 크게 전역적 이진화와 적응적 이진화로 구분할 수 있다[9]. 전역적 이진화 기법은 영상 전체에 하나의 임계값을 사용하기 때문에 밝기 값이 일정하지 않은 영상의 경우 정보의 손실을 가

져올 수 있다. 적응적 이진화 기법은 관심 화소의 주변 화소들의 정보를 이용해 화소 단위의 임계값을 결정하여 영역마다 다른 임계값으로 이진화를 하기 때문에 입력 영상의 명도가 불균형할지라도 비교적 정확한 이진화가 가능하다. 블러 복원 기술은 사용자의 손 떨림으로 발생한 블러를 보정하는 기술이다[10]. 기울임 보정 기술은 기울어진 상태에서 촬영한 영상의 인식률을 향상시키기 위해 호모그래피를 이용하여 영상을 복원하는 기술이다[11].

OCR 기술을 이용한 제품은 전통이 오래 된 만큼 현재 많은 제품이 출시된 상태이다. 디오텍은 카메라 스캔 방식의 실시간 번역 앱인 'DioLense'를 통해 텍스트를 인식한 후 내장된 전자사전과 온라인 번역 엔진을 통해 바로 검색하는 기술을 보유하고 있다[12]. ABBYY는 OCR 기술을 기반으로 6개의 모바일 앱 및 문서인식 PDF 변환, 명함 인식 등 다양한 제품을 출시하였다[13]. Google은 2014년 'Word Lens Translator'이라는 가상현실 번역 앱을 개발한 퀘스트 비주얼 인수하였으며 2015년에는 구글 번역 앱에 사진이나 이미지 속의 문자를 번역해주는 기능을 추가하였다[14]. 하지만 현재의 문자인식 알고리즘은 대부분 인식하려는 대상이 문서의 형태이기 때문에 스마트폰에 적용하는 경우 카메라의 잡음, 블러, 기울어짐 문제 등으로 인하여 인식률이 저하되는 문제가 발생한다. 또한 스마트폰 환경에서 실시간으로 동작하기 위해서는 알고리즘의 복잡도 또한 고려되어야 한다.

본 논문에서는 스마트폰 카메라를 이용하여 음식 메뉴를 촬영한 영상에서 한글 텍스트를 인식하는 OCR 기법을 제안한다. 먼저, 스마트폰 영상에서 발생하는 노이즈를 다운샘플링을 통해 제거한다. 그리고 한글을 추출하기 위해 blob 검출 기법인 Maximally Stable Extremal Regions (MSER)[15] 기법을 이용한다. 추출한 각 글자에 대하여 침식 및 팽창 연산을 이용하여 글자의 에지 영역을 획득하고 이진화와 수평방향의 모폴로지 연산을 통해 글자 영역을 추출한다. 추출한 글자 영역은 OCR 오픈소스인 Tesseract-OCR[16]을 통해 인식한다. 인식된 결과의 오류를 수정하기 위해 사전 DB 기반의 후처리 기법을 이용하여 인식률을 향상시킨다.

본 논문은 다음과 같이 구성된다. 2장에서는 제안하는 기법에 대하여 설명한다. 3장에서는 제안한 방법을 구현한 메뉴 인식 어플리케이션의 성능을 측정한다. 4

장에서는 결론을 맺는다.

II. 제안하는 기법

제안하는 시스템에서는 스마트폰에서 촬영한 메뉴판을 OCR 기술을 통해 글자로 인식한다. 글자 인식 단계는 크게 전처리 과정, 인식 과정 그리고 후처리 과정으로 이루어지며, 각각의 과정을 통해 최종적으로 인식한 글자를 출력하게 된다. 그림 1은 제안하는 시스템의 흐름도이다.

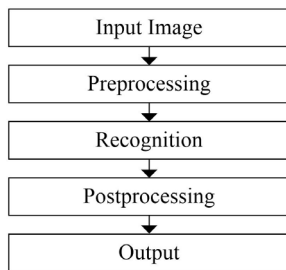


Fig. 1 Flowchart of the proposed system

입력 영상의 경우 스마트폰으로 촬영한 영상 전체가 아닌 사용자가 Region Of Interest (ROI)에 인식하려는 음식 메뉴를 위치시켜 놓고 촬영한 영상으로 한다. 영상 전체를 사용하면 스마트폰의 하드웨어적인 한계로 인하여 실시간으로 작동되기 어렵기 때문이다.

스마트폰으로 촬영한 영상에서는 손 떨림으로 인한 블러 현상, 불균일한 조명에 의한 영상의 품질 저하, 복잡한 배경, 노이즈 등이 발생할 수 있다[17]. 이러한 현상들은 글자 인식의 성능을 떨어뜨리는 요소가 된다. 따라서 이러한 요소들을 제거하고 글자 인식 성능을 향상시키기 위한 전처리 기술이 필요하다.

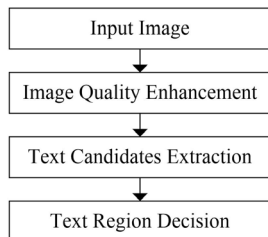


Fig. 2 Flowchart of the proposed preprocessing

제안하는 전처리 과정은 스마트폰 카메라로부터 입력받은 영상에서 노이즈를 제거하고 글자 영역만을 추출함으로써 인식 과정의 성능을 향상시킨다. 전처리 과정은 그림 2와 같이 크게 영상 품질 개선, 글자 후보 추출, 글자 영역 결정 단계로 구성된다.

2.1. 영상 품질 개선

영상 품질 개선 단계에서는 영상의 노이즈를 제거하고 글자 영역을 부각시킨다. 다운샘플링은 영상의 노이즈를 제거하는 동시에 영상의 크기를 줄여줌으로써 연산 속도를 빠르게 하는 효과를 얻을 수 있다. 제안하는 기법에서는 짝수 번째 행과 열을 제거하여 전체 크기의 1/4로 다운샘플링을 한다. 다운샘플링 시 앨리어싱을 방지하기 위하여 입력영상에 가우시안 블러를 적용한다[18]. 이 과정에서 영상의 크기가 작은 노이즈들은 제거되고 글자 영역은 그대로 남아있게 된다. 글자의 인식률을 향상시키기 위해 영상에 샤프닝 필터를 적용하여 글자의 에지 영역을 뚜렷하게 만든다. 그림 3은 영상 품질을 개선하는 과정을 보여준다.

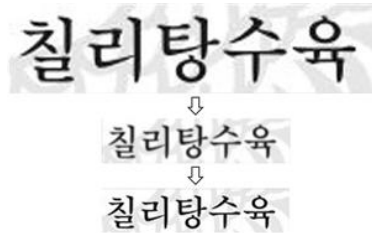


Fig. 3 The process of image quality enhancement

2.2. 글자 후보 추출

글자 후보 추출 단계에서는 영상 품질 개선 단계를 거친 영상에서 글자라고 판단되는 후보를 추출하는 과정으로서, 그림 4의 순서대로 진행된다.

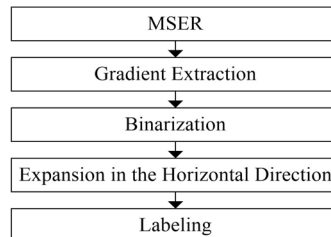


Fig. 4 The flowchart of text candidates extraction

일반적으로 영상에서 글자는 blob의 형태를 가지고 있다. 따라서 blob 검출 기법인 MSER 기법[15]을 이용하여 영상에서 blob의 형태를 가지는 성분을 추출한다. MSER 기법[15]은 영상에서 MSER[15]로 불리는 비슷한 색상들이 안정적으로 연결되어있는 영역을 추출한다. 여기서 안정적이란 영상에서 임계값을 0부터 시작하여 255까지 천천히 변화시킬 때 영역의 변화량이 최소가 되는 것을 의미한다. MSER 기법[15]을 수식으로 표현하면 식 (1)과 같다.

$$q(i) = \arg \min_i |Q_{i+\square} \setminus Q_{i-\square}| / |Q_i| \quad (1)$$

여기서 i 는 인덱스로 0부터 255까지의 값을 가지며, Q_i 는 인덱스 i 에 대한 영역을 의미한다. Q_i, Q_2, \dots, Q_n 은 i 가 커질수록 이전 영역의 부분집합을 의미한다. 즉 $Q_i \supset Q_{i+1}$ 이다. \square 는 최소가 되는 영역의 크기를 어느 정도로 제한할지 결정하는 사용자 정의 파라미터이다. 추출하려는 객체에 따라 그 크기에 제한이 있기 때문에 사용한다. 제안하는 기법에서는 글자를 추출하기 때문에 실험 결과 가장 좋은 결과를 보여주는 5로 설정한다. 식 (1)을 통해 제안하는 기법에서는 영역의 변화량이 최소가 되는, 즉 글자 영역을 획득할 수 있다.

제안하는 기법에서는 글자 영역을 추출하기 때문에 추출하는 영역의 크기의 제한을 두는 파라미터가 필요하다. MaxArea의 값을 정하여 글자보다 크다고 판단되는 영역은 제외하고 마찬가지로 MinArea 값을 정하여 글자라고 하기에는 너무 작은 영역을 제외한다. 실험에는 MaxArea는 14400, MinArea는 225로 설정하였다. 그림 5는 추출한 MSER 기법[15]을 이용하여 글자를 추출한 결과를 보여준다.



Fig. 5 MSER result

추출한 글자에 대하여 팽창과 침식 연산을 이용하여 기울기 성분을 추출한다. 3x3 크기로 팽창한 영상에서 같은 크기로 침식한 영상을 차 연산하여 기울기 성분을 획득한다. 다음으로, 영상에서 배경과 문자를 분

리하기 위해 이진화 알고리즘을 적용한다. 이진화 알고리즘은 특정 임계값 T 를 기준으로 밝기 값이 T 보다 작으면 0으로 T 보다 크면 255로 변환하는 것이다. 따라서 적절한 임계값 T 를 정하는 것이 매우 중요하다. 제안하는 기법에서는 OTSU의 기법[19]을 이용하여 이진화를 한다. OTSU의 기법[19]은 영상에서 T 를 기준으로 두 영역으로 분류할 때 클래스 간 분산을 최대로 하는 T 를 찾는 방법이다. 클래스 간 분산을 표현하면 식 (2)와 같다.

$$\sigma_{Between}^2(T) = W_B(T) [u_B(T) - u]^2 + W_F(T) [u_F(T) - u]^2 \quad (2)$$

여기서 $\sigma_{Between}^2$ 은 클래스 간 분산이며 T 는 임계값을 의미한다. W_B 는 전체 영상에서 배경에 해당하는 픽셀이 나타날 확률이며 W_F 는 전체 영상에서 전경이 나타날 확률이다. u_B, u_F 그리고 u 는 각각 배경, 전경 그리고 전체 영상의 평균을 의미한다. 식 (3)과 식 (4)를 이용하여 정리하면 식 (5)를 유도할 수 있다.

$$W_B + W_F = 1 \quad (3)$$

$$W_B u_B + W_F u_F = u \quad (4)$$

$$\sigma_{Between}^2(T) = W_B(T) W_F(T) [u_B(T) - u_F(T)]^2 \quad (5)$$

식 (3)은 W_B 와 W_F , 즉 전체 영상에서 전경과 배경이 될 확률의 합은 1이 된다는 것을 의미한다. 식 (4)는 전체 영상에서 배경 픽셀의 확률, W_B 와 배경 픽셀들의 밝기 값의 평균, u_B 의 곱과 전체 영상에서 전경 픽셀의 확률, W_F 와 전경 픽셀들의 밝기 값, u_F 의 합은 전체 영상 픽셀들의 밝기 값의 평균, u 와 같다는 것을 의미한다. 식 (5)는 식 (3)과 식 (4)를 이용하여 식 (2)에서 유도된 식으로 변수 u 를 제거하여 식을 단순하게 만든 식이다.

이진화 영상에는 글자 영역을 포함한 여러 영역들이 검출이 되는데, 글자 성분만을 추출하기 위해서 수평으로 모폴로지 연산을 적용한다. 적용할 때 마스크의 형태는 수평으로 10의 크기를 가지는 마스크를 적용한다. 이후 수직으로 연결된 영역들을 사각형 형태로 레이블링한다. 레이블링한 영역 중에 글자 영역인 경우와 아닌 경우를 구별하기 위해 조건을 정해놓고 필터링을 한

다. 글자가 차지하는 픽셀 수의 사각형 넓이에 대한 비율이 특정값 보다 크거나 사각형의 크기가 특정값 보다 큰 경우만을 글자 후보로 선정한다. 그림 6은 글자 후보를 선정하는 단계별 영상을 보여준다.



Fig. 6 The process of text candidates extraction

2.3. 글자 영역 결정

글자 영역 결정 단계는 글자 영역 후보들 중 인식할 글자 영역을 최종적으로 결정하는 단계이다. 일반적으로 사용자가 ROI에 인식할 메뉴를 위치시킬 때 ROI의 가운데에 위치시킨다. 따라서 글자 후보들 중 인식하려는 글자 영역은 영상의 가운데에 있다는 가정을 할 수 있다. 제안하는 기법에서는 이러한 사용자의 특성을 이용하여, 영상의 중앙 수평선에 위치해 있는 글자 영역 후보를 최종적으로 인식할 글자 영역으로 판단한다. 그림 7은 인식할 글자 영역을 결정하는 과정을 보여준다.

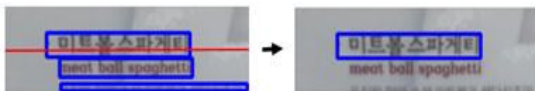


Fig. 7 The text region decision using the horizontal line

음식 메뉴 중 그림 8의 왼쪽 영상과 같이 가운데가 비워있는 구조를 가지는 메뉴가 존재한다. 앞의 방식대로 글자 영역을 결정한다면 중앙의 공백으로 인하여 중앙 수평선에 위치해 있는 글자 영역이 2개가 존재하기 때문에 에러가 발생하게 된다. 따라서 중앙 수평선에 위치해 있는 글자 영역이 2개 이상인 경우에는 각 글자 영역을 확장하여 합친 영역을 최종 글자 영역으로 결정한다.

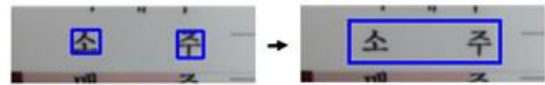


Fig. 8 The process for merging two candidates

2.4. 인식 및 후처리 과정

글자 인식은 무료 글자 인식 엔진인 Tesseract-OCR [16]을 이용하여 인식한다.

Tesseract-OCR[16]은 1985년에 개발된 이후로 현재까지 지속적인 업데이트를 통해 현재 100개 이상의 언어를 지원하고 있으며, 언어 학습 도구 또한 제공하고 있다. 하지만 입력 영상의 전처리를 아무것도 하지 않기 때문에 입력 영상 그대로 Tesseract-OCR[16]을 통해 인식한다면 인식률은 떨어지는 편이다. 제안하는 기법에서는 전처리 과정을 거친 영상을 Tesseract-OCR[16]의 입력 영상으로 사용하고 학습은 음식 메뉴 83개에 대하여 따로 학습한 데이터를 사용한다. 또한 음식 메뉴는 대부분 한 줄로 되어있는 특성을 고려하여 Tesseract-OCR[16]의 인식 옵션을 한 줄로만 인식되게끔 설정함으로써 인식률을 향상시킨다.



Fig. 9 The error correction using dictionary

Tesseract-OCR[16]의 인식 결과 촬영한 음식 메뉴의 각 글자를 정확하게 인식하는 경우가 있는 반면에 철자를 틀리게 인식하는 경우가 발생한다. 예를 들어 “해물

칼국수”를 “해무가국수”로 철자를 틀리게 인식한다. 이러한 문제가 발생한 경우 후처리 과정을 통해 잘못 인식된 결과를 수정하는 것이 필요하다. 제안하는 기법에서는 사전 DB 기반의 후처리 기법을 적용한다. N개의 문자로 구성된 단어에 대해서 1~2개 정도 문자에 대해 오인식이 발생한 경우, 사용자가 등록한 단어사전을 검색 및 참조하여 오인식 된 문자를 정정한다. 그림 9는 사전을 이용한 오류 수정 방법을 보여준다. 이러한 방법을 통해 인식률을 크게 향상시킬 수 있다.

III. 실험 결과



Fig. 10 The UI of menu recognition application

본 논문에서 제안하는 음식 메뉴 인식 알고리즘의 성능을 평가하기에 앞서 제안한 알고리즘을 스마트폰 어플리케이션으로 이식하였다. 개발환경은 Windows10에서 Android Studio를 이용하여 어플리케이션을 개발하였다. 스마트폰은 SAMSUNG Galaxy A5이며 운영체제는 Android 4.0 Ice Cream Sandwich이다. 그림 10은 개발한 어플리케이션의 UI를 보여준다. 빨간색 박스 안에 인식하려는 음식 메뉴를 위치시키고 하단의 카메라 버튼을 눌러 촬영을 하면 인식 결과가 출력된다.

일반적으로 알고리즘의 성능을 평가하기 위해서는 인종된 DB를 이용하여 실험을 진행하여야 한다. 하지만 한글 음식 메뉴에 대한 신뢰도 높은 DB가 존재하지 않는다.



Fig. 11 The menu used in the experiment

따라서 실험에서는 실제 음식점에서 사용하는 메뉴판을 DB로 사용하였다. DB는 83개의 각기 다른 메뉴로 이루어져 있다. 글자의 배경이 없는 경우에는 인식률이 상승하지만 실제 메뉴판에는 그런 경우가 없기 때문에 대부분의 DB는 글자의 배경이 존재한다. 그림 11은 실험에 사용한 DB를 보여준다.



Fig. 12 The result of the text region extraction

첫 번째 실험에서는 테스트 영상에 대해 전처리 과정에서 얼마나 정확하게 글자를 추출하는지를 진행하였다. 그림 12는 테스트 영상에 대한 전처리 과정 결과를 보여준다. 그림 12(a)의 경우 빨간색 박스 내부에서 글자 영역만을 깔끔하게 추출한 것을 확인할 수 있다. 그림 12(b)의 경우 “해물전골”이라는 글자뿐만 아니라 빨간색 박스 왼쪽 끝의 숫자 7의 일부가 같이 추출되었다. 전처리 과정에서 이진화한 후 수평으로 팽창하는 과정에서 숫자 7이 글자와 같이 같은 범주에 포함되었기 때문이다. 이러한 경우 후처리 과정에서 사전과 비교하여 가장 유사한 글자를 결과로 출력하기 때문에 숫자 7과 같은 인식하려는 글자가 아닌 것들은 결과에 영향을 미치지 않는다. 그림 12(c)와 같이 음식 메뉴에 띄어쓰기가 되어 있는 경우에도 글자가 잘 추출되는 것을 확인할 수 있다. 그림 12(d)는 띄어쓰기와 배경이 글자 뒤에 있는 경우에도 글자 영역만을 추출하는 모습을 보여준다.

두 번째 실험에서는 제안하는 기술에서 후처리 과정을 거쳤을 경우와 그렇지 않을 경우의 인식률을 비교하였다. 후처리 과정의 사전 DB는 첫 번째 실험 영상의 음식 메뉴 83개를 이용하였다. 표 1은 후처리 과정 유무에 따른 인식 결과를 보여준다. 후처리 과정이 들어갈 경우 인식률이 크게 상승하였음을 보여준다.

Table. 1 Recognition rate according to the postprocessing

Postprocessing	Recognition rate
No	68.4%
Yes	92.8%

세 번째 실험에서는 성능의 비교 측정을 위해 제안하는 기법과 구글 플레이스토어에서 현재 서비스를 제공하는 OCR Instantly Free[20], Text Scanner[21] 그리고 Text Fairy[22]의 기법의 성능을 측정하였다. Text Scanner[21]는 제안하는 기법과 같은 Tesseract-OCR [16]을 이용하며, 나머지 어플리케이션들은 알고리즘을 공개하지 않는다. 실험의 방식은 네 가지 기법 모두 사전 DB 기반의 후처리 기법을 사용하지 않을 때와 사용할 경우에 대해 각각 진행하였다. DB는 직접 촬영한 한글 메뉴판 DB를 이용하였으며 인식의 기준은 한글자라도 잘못 인식하면 오인식으로 평가하였다. 예를 들어 “부대찌개”를 “부다찌개”로 인식하였다면 오인식으로

간주하였다. 표 2는 후처리 기법 적용 유무에 대한 각 기법의 인식률을 보여준다. 후처리 기법을 적용하지 않은 경우에는 제안하는 기법이 68.4%로 다른 세 가지의 기법 중 가장 높은 Text Fairy[22]보다 8.2% 높은 인식률을 보여주었다. 제안하는 기법은 전처리 과정에서 글자 영역만을 추출하여 인식을 수행하기 때문에 배경을 제거하지 않는 다른 기법에 비해 높은 인식률을 보여준다.

사전 DB 기반의 후처리를 적용하는 경우 4가지 알고리즘 모두다 인식률이 상승하지만 제안하는 기법의 인식률은 92.8%로 다른 세 가지의 기법 중 가장 높은 인식률을 보여주는 Text Fairy[22]보다 7.2% 높은 인식률을 보여준다.

Table. 2 Recognition rate comparison for each method

Method	non-post processing	post processing
OCR Instantly Free[20]	53.5%	74.3%
Text Scanner[21]	55.6%	76.2%
Text Fairy[22]	60.2%	85.6%
Proposed method	68.4%	92.8%

IV. 결 론

본 논문에서는 스마트폰 환경에서 OCR 기술을 이용하여 한글 음식 메뉴를 인식하는 기법을 개발하였다. 제안하는 기법은 크게 전처리, 인식 그리고 후처리 과정으로 구성되어 있다. 스마트폰으로 글자를 촬영하여 ROI를 획득한 후에 전처리 과정을 진행한다.

전처리 과정은 ROI에서 글자만을 추출하는 단계이다. 먼저 카메라 촬영 시 발생하는 노이즈를 다운샘플링을 적용하여 제거한다. ROI에는 글자뿐만 아니라 배경도 존재하기 때문에 MSER 기법[15]을 이용하여 배경을 제거한다. 글자 영역만을 하나의 ROI 형태로 추출하기 위해서는 각각의 글자들을 하나로 분류하여야 할 뿐만 아니라 인식하려는 글자만을 추출하여야 한다. 제안하는 기법에서는 수평 방향 모폴로지 연산을 통해 각각의 글자 영역을 연결하여 한 줄로 분류한 후 ROI에서 인식하려는 글자는 가운데에 존재한다는 기하학적인 특징을 이용하여 여러 영역 중 정확히 인식하려는 글자

영역만을 추출하였다.

인식 과정에서는 무료 OCR 엔진인 Tesseract-OCR [16]을 이용하여 글자를 인식한다. 음식 메뉴 특성상 인식하려는 글자는 한 줄로 되어있다는 점을 고려하여 Tesseract-OCR[16]의 옵션을 한 줄만 인식되게끔 수정하여 인식률을 향상시킨다.

마지막으로, 후처리 과정에서는 인식 과정의 오류를 사전에 입력된 단어와 비교하여 가장 유사한 단어로 출력해줌으로써 인식률은 크게 향상시켰다. 실험을 통해 제안하는 시스템이 다른 기법들과 비교하여 우수한 성능의 인식률을 보여주는 것을 확인하였다.

제안 기술을 활용한 메뉴판 인식 어플리케이션을 통해 외국인 관광객에게 음식에 대한 정보를 신속하고 정확하게 전달함으로써 편의성을 제공할 수 있다. 또한 인식 언어를 한글뿐만 아니라 외국어로 확장함으로써 다른 나라에서의 서비스 제공도 가능하다. 관광객들이 불편함을 가지고 있는 표지판이나 지하철역 인식에 대해서도 추가적으로 확장성이 있을 것으로 예상된다.

ACKNOWLEDGMENTS

The present Research has been conducted by the Research Grant of Kwangwoon University in 2016

REFERENCES

- [1] H. J. Jeon and Y. M. Kim, "The achievement of 14 million people tourists in 2014," Korea Culture & Tourism Institute, Tourgo focus, 2015.
- [2] M. S. Park, D. H. Lee, J. H. Park, and J. K. Koo, "Analysis of economic impacts and implications of the increase in Chinese tourists visit," Korea Institute for Industrial Economics & Trade, Issue paper, 2014.
- [3] E. K. Choi, "Activation measures to attract Chinese tourists," Korea Culture & Tourism Institute, Basic research, 2011.
- [4] E. B. Go, Y. J. Ha, S. R. Choi, K. H. Lee, and Y. H. Park, "An implementation of an android mobile system for extracting and retrieving texts from images," *Journal of Digital Contents Society*, vol. 12, no. 1, pp. 57-67, Mar. 2011.
- [5] R. Mithe, S. Indalkar, and N. Divekar, "Optical character recognition," *International Journal of Recent Technology and Engineering*, vol. 2, no. 1, pp. 72-75, Mar. 2013.
- [6] M. H. Cho, "A study on character recognition using wavelet transformation and moment," *Journal of The Korea Society of Computer and Information*, vol. 15, no. 10, pp. 49-57, Oct. 2010.
- [7] J. H. Roh and D. S. Choi, "Character recognition using a target string," *Proceedings of Symposium of the Korean Institute of communications and Information Sciences*, Jun. 2014.
- [8] S. G. Hong, S. S. Hwang, and S. D. Kim, "A license plate recognition system robust to vehicle location and viewing angle," *Journal of the Institute of Electronics and Information Engineers*, vol. 49, no. 12, pp. 113-123, Dec. 2012.
- [9] J. H. Ju and J. S. Oh, "An adaptive binarization algorithm for degraded document images," *Journal of Korean Institute of Communications and Information Sciences*, vol. 37, no. 7, pp 581-585, Jul. 2012.
- [10] C. H. Son and H. M. Park, "Fast multiple-image-based deblurring method," *Journal of the Institute of Electronics Engineers of Korea SP*, vol. 49, no. 4, pp. 49-57, Jul. 2012.
- [11] J. W. Song, N. R. Jung, and H. S. Kang, "Container BIC-code region extraction and recognition method using multiple thresholding," *Journal of the Korea Institute of Information and Communication Engineering*, vol. 19, no. 6, pp. 1462-1470, Jun. 2015.
- [12] DioLense. Available: http://www.diotek.com/?page_id=8532.
- [13] ABBYY. Available: <https://www.abbyy.com>.
- [14] Google Translate. Available: https://translate.google.com/about/intl/en_ALL/.
- [15] M. Donoser and H. Bischof "Efficient maximally stable extremal region (MSER) tracking," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2006*, Jun. 2006.
- [16] Tesseract-OCR. Available: <https://github.com/tesseract-ocr>.
- [17] S. U. Kim, "An image denoising algorithm for the mobile phone cameras," *Journal of the Korea Institute of Electronic Communication Sciences*, vol. 9, no. 5, pp. 601-608, May 2014.
- [18] J. W. Jeong and K. K. Jun, "Noise robust document image

- binarization using text region detection and down sampling,” *Journal of Korea Multimedia Society*, vol. 18, no. 7, pp. 843-852, Jul. 2015.
- [19] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems*, vol. 9, no. 1, pp. 62-66, Jan. 1979.
- [20] TheSimplest.Net. Available: <https://play.google.com/store/apps/details?id=com.theSimplest.ocr&hl=ko%20Renard>.
- [21] Generic Co. Available: <https://play.google.com/store/apps/details?id=co.kr.generic.ocr&hl=ko%20Renard>.
- [22] Text Fairy. Available: <https://github.com/renard314/textfairy>.



이규철(Gyu-Cheol Lee)

2013년 2월 광운대학교 전자공학과 학사
2015년 2월 광운대학교 전자공학과 석사
2015년 3월 ~ 현재 광운대학교 전자공학과 박사
※관심분야 : 영상압축, 영상인식, 영상 신호처리



유지상(Jisang Yoo)

1985년 2월 서울대학교 전자공학과 학사
1987년 2월 서울대학교 전자공학과 석사
1993년 5월 Purdue Univ. EE, ph.D
1997년 9월 ~ 현재 광운대학교 전자공학과 교수
※관심분야 : 영상압축, 영상인식, 신호처리