



Journal of Knowledge Information Technology and Systems

ISSN 1975-7700 (Print), ISSN 2734-0570 (Online)

<http://www.kkits.or.kr>

Development of iOS English Vocabulary App Using Vision System

Myeong-Cheol Lee, Kang-Hee Lee*

Global School of Media, Soongsil University

ABSTRACT

Before smartphones, English learners used to organize words in their notebooks. As smartphones became more popular over time, more people started to study by storing words in word book apps than notebooks. Users will be able to study words anytime, anywhere via their smartphones. These word-field apps are a way for users to enter data themselves. However, this paper designed and implemented an app that automatically enters words using vision system and letter recognition. The app recognizes areas painted with highlighter first, then recognizes words within the highlighter area and organizes them into tables. It also helps to understand how certain words are used in contexts in certain English bodies with simple actions. Because the data of words can have multiple words in a particular body, the database schema was designed in a one-to-many relationship between the images taken with the camera and the words contained in that body. The differentiated advantages of this app are as follows: First, it increases the time efficiency of studying. Second, it improves learners' vocabulary by learning based on context. These advantages are described in detail in this paper through practical research materials and techniques. The app's platform used Apple Inc.'s iOS. Character recognition, a key technology in the app, utilizes OpenCV's library and Apple Inc.'s CoreML.

© 2020 KKITS All rights reserved

KEYWORDS: Vocabulary app, iOS smartphones, Vision, OpenCV, Text recognition

ARTICLE INFO: Received 14 October 2020, Revised 11 December 2020, Accepted 11 December 2020.

*Corresponding author, Kang-Hee Lee, is with the Global School of Media, Soongsil University, 369 Sando-Ro Dongjak-Gu, Seoul, 06978, KOREA.

E-mail address: kanghee.lee@ssu.ac.kr

1. 서론

H. Nicod, H. Felipe, A. Mayutan, A. Mohamed, P. Konstantina, and N. Saverio[1]의 연구에서 스마트폰의 대부분은 사용자에게 응용 프로그램을 위한 기반이 된다. 단순히 “앱”이라고도 부르는 이러한 응용 프로그램은 소위 응용 프로그램 시장(application markets)을 통해 제공된다. 이러한 시장은 모바일 경제의 추세를 결정하는데 막대한 정보의 원천임을 보여준다. 다시 말해, 스마트폰 앱 시장은 이제 경제에서도 아주 중요한 영역이 되었다.

앱 시장이 커짐에 따라 앱으로 공부하는 모바일 러닝(mobile-learning) 앱도 등장했다. 이에 따라 영어공부를 도와주는 앱들도 생겨나기 시작했다. F. Martin, & J. Ertzberger의 연구는 모바일 러닝으로 영어를 공부하면 학습자로 하여금 흥미를 유발함으로써 주의를 집중하게 만들어 효과적인 학습이 가능함을 보여준다 [2].

앱 시장에는 회화, 독해, 토익, 듣기 등의 영어 공부를 위한 앱들이 계속해서 출시가 되었다. 마찬가지로 영어 단어장 앱도 출시가 되었다. 하지만 영어 단어장 앱은 사용자가 일일이 직접 데이터를 입력하는 불편함이 있었다. 이로 인한 문제점을 해결하기 위해 컴퓨터 비전과 글자 인식을 활용하여 자동으로 단어 데이터를 입력해주는 앱을 설계 및 구현하였다. 앱의 이름은 앱의 특성을 잘 나타내기 위해 ‘찍어보카’로 정했다.

본 논문은 단어 데이터 입력과정에 집중하여 앱 개발을 하였다. 따라서 단어장 앱의 시험보기, 그림 연상법, 어원 분석, 발음 듣기 등의 부가적인 기능은 제외하고, 사용자의 데이터 입력과정만 본 논문에서 다루게 됨을 분명히 한다.

‘찍어보카’ 앱은 특별히 영어 학습자들이 독해 공부를 할 때 모르는 단어를 형광펜으로 강조하는

점에 집중했다. 그래서 독해 지문 속 형광펜 영역 안에 있는 단어만 자동으로 등록시켜 주는 방식으로 앱을 설계하였다.

앱 구현에는 컴퓨터 비전을 위한 OpenCV 라이브러리와 글자 인식을 위한 Apple Inc.에서 제공하는 라이브러리들을 활용하였다. 플랫폼은 아이폰의 운영체제인 Apple Inc.의 iOS를 사용하였고, 프로그래밍 언어는 전체적인 앱의 기능은 swift를 사용하였으나, OpenCV의 영상처리 부분은 C++를 사용하였다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구를 소개한다. 관련 연구에서는 현재 시장에 나와 있는 앱을 분석한다. 제 3장에서는 앱 설계 과정을 살펴본다. 이 장에서는 전반적인 개발 환경, 앱을 사용하는 방법, 사용자 인터페이스를 소개한다. 제 4장에서는 구현을 통해 앱이 만들어지는 과정을 서술한다. 제 5장에서는 실제 실험을 통해 여러 조건 속에서 앱의 동작을 살펴본다. 마지막 제 6장에서는 결론을 기술한다.

2. 관련 연구

2.1. 기존의 단어장 앱 관련 조사

최근 스마트폰 보급률이 높아지면서, 영어 단어를 정리해주는 앱도 많아졌다. 대표적인 앱으로는 Onevoca[16], 영단무시(영어 단어 무한 시험)[17], 암기고래, 워드뱅크, 암기고래 등이 있다. 단어장 앱의 종류는 크게 두 가지로 나누어진다. 토익이나 수능을 위해 기본적으로 단어들이 내장되어 있는 앱과 사용자가 자신이 외워야할 단어들을 추가하는 앱들이다. 본 논문에서 소개하는 앱은 후자에 속하는 앱이다. 그러므로 후자에 해당하는 앱들 중에서 대표적인 ‘Onevoca’와 ‘영단무시’ 앱을 살펴본다. 서론에서 언급했듯이 앱 관련 조사도 다른

기능은 제외하고, 데이터 입력과정만 제한하여 조사 하였다.

1) Onevoca

‘Onevoca’ 는 <그림 1>의 (b)화면에서 새로운 단어를 등록할 수 있다. 단어와 뜻은 필수로 적어야 하며, 발음과 추가 설명은 선택사항이다. 사용자가 단어 하나를 추가하려면 해당 단어에 대한 모든 데이터를 사용자가 직접 입력해야 한다. 단어 입력이 끝나면 (a)화면으로 넘어가서 정리된 단어들을 볼 수 있다.

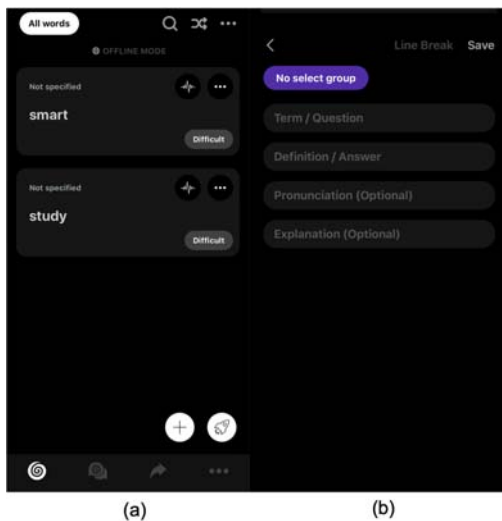


그림 1. ‘Onevoca’ 화면 : (a) 단어 목록 (b) 단어 추가
Figure 1. Screen of ‘Onevoca’ : (a) list of words (b) adding word

2) 영단무시

‘영단무시’ 는 <그림 2>의 (a)화면에서 새로운 단어를 등록하게 된다. (a)화면에서 사용자가 영어 단어를 입력하는 순간 (a)화면의 ‘해석’ 밑에 단어의

대표적인 뜻들이 자동으로 생성된다. 연구를 위해 화면 (a)처럼 ‘apple’ 이라는 단어를 입력하였다. 입력한 후에 자동으로 ‘애플, 사과, 사과나무, 뉴욕’ 이라는 뜻이 생성되었다. 이 앱은 앞서 살펴본 ‘Onevoca’ 앱과 달리 사용자가 영어 단어만 직접 입력하게 된다. 입력이 완료되면 (b)화면으로 넘어와 지금까지 입력한 단어들을 볼 수 있다.

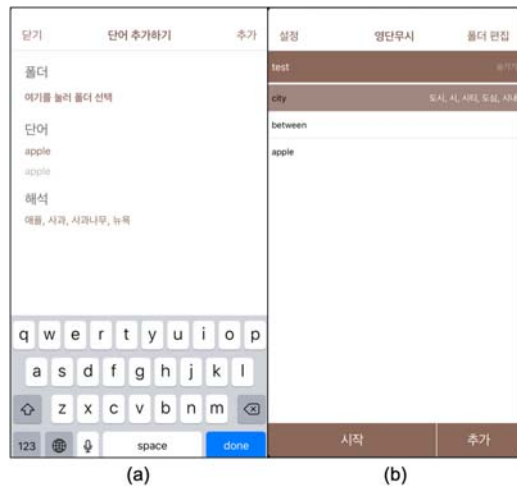


그림 2. ‘영단무시’화면 : (a) 단어 추가 (b) 단어 목록
Figure 2. Screen of ‘English Word Infinite Test’ : (a) adding word (b) list of words

2.2. 기존의 단어장 앱 분석

2.1장에서 살펴본 것처럼 기존의 사용자가 자신이 외워야할 단어들을 추가하는 앱들은 사용자가 하나하나 직접 손으로 입력해야 한다는 불편함이 있었다. ‘영단무시’ 앱 같은 경우 영어 단어만 입력하면 자동으로 뜻이 생성되긴 하지만, 여전히 영어 단어는 직접 입력해야한다. 이러한 방식은 한 번에 여러 개의 데이터를 입력할 수 없어서 시간이 오래 걸리는 문제점이 있다. ‘Onevoca’ 앱은 모든 데이터를 하나씩 직접 손으로 입력해야 하므로 더욱

시간이 오래 걸린다. 반면에 ‘영단무시’ 앱은 뜻은 자동으로 완성해 주어서 시간이 절약된다고는 할 수 있다. 그러나 문맥에 따라 단어의 뜻이 달라지는 경우나 자동완성이 된 뜻들 중에서 사용자가 바라는 뜻이 없을 경우의 문제들이 발생할 수 있다. 왜냐하면 자동완성이 되어서 자동으로 입력되는 뜻들이 앱 안에 사용자가 바꿀 수 없는 기본값으로 설정되어있기 때문이다.

기존에 출시 되어있는 단어장 앱들의 대표적인 문제는 두 가지로 요약할 수 있다. 첫째, 직접 손으로 입력해야 한다. 둘째, 한 번에 많은 데이터를 입력하지 못하고 하나씩 밖에 입력하지 못한다. 이러한 문제들로 인해 시간이 늘어나는 추가적인 문제도 발생하게 된다. 이러한 문제들을 보완하기 위해 본 논문에서는 사용자가 형광펜을 칠한 영어 단어들을 한 번에 인식하여 단어장에 추가해주는 서비스를 제공하려 한다. 이를 통해 사용자가 직접 데이터를 하나씩 입력해야하는 불편함을 해소시킨다. <표 1>은 기존 단어장 앱들과 본 논문에서 제안하는 앱의 차이점을 보여준다.

표 1. 기존 앱과 제안하는 앱의 차이점
Table 1. Differences of apps between existing and proposed

기존의 앱들	제안하는 앱
1. 단어를 직접 손으로 입력해야 한다. 2. 하나씩 입력한다.	1. 단어가 쓰여진 문서를 카메라로 스캔하면 자동입력된다. 2. 스캔된 단어를 한꺼번에 등록한다.

추가로 본 논문에서 제안하는 앱은 문맥과 단어 뜻의 다양성을 고려하여 뜻은 사용자가 직접 입력하게끔 설계하였다.

3. ‘찍어보카’ 앱 설계

3.1. 플랫폼, 언어, 디자인 패턴

개발 플랫폼은 Apple Inc.의 iOS를 사용하였다. 개발 언어는 Swift와 C++를 사용하였다. 디자인 패턴은 MVC(Model-View-Controller)[3]을 사용하였다.

Apple Inc. 개발자 사이트[3]에 따르면 MVC 디자인 패턴은 앱 안에 있는 객체들(objects)을 3가지 역할로 나누어서 배정하는 방식으로 정의된다. 하나의 객체는 model, view, 또는 controller가 될 수 있다.

model객체는 앱 안에 있는 데이터 처리, 연산을 담당한다. 단어장 앱의 model객체는 전반적인 이미지 처리를 담당한다.

view객체는 사용자에게 보이는 화면 객체이다. 이 객체는 사용자의 행동에 따라 화면을 바꿔준다. 또한 model객체에서 처리된 데이터를 화면에 보여주는 객체이다. 단어장 앱에서는 문서 스캔을 시작하는 버튼이나 완료된 결과를 저장하는 확인버튼 등을 보여주는 ‘ButtonView’, 입력한 단어를 순서대로 보여주는 ‘TableView’, 스캔된 이미지와, 형광펜 영역의 글자만 추출된 이미지를 보여주는 ‘ImageView’, 인식된 단어들의 목록을 보여주는 ‘TextView’ 등을 사용하였다.

controller객체는 view객체에서 일어난 사용자의 명령을 model객체에 전달해준다. 쉽게 말해 controller객체는 이름 그대로 view객체와 model객체를 서로 의사소통하게 해주는 중재자라고 볼 수 있다. 단어장 앱에서는 스캔 화면, 단어장 화면, 형광펜으로 칠해진 특정 단어를 검출하는 화면 등을 서로 전환하는데 각종 controller객체가 사용되었다. 그리고 model객체에 이미지 처리 명령을 내리는 것 또한 controller객체를 활용하였다.

3.2. 앱 사용법

사용자가 앱을 사용하는 순서는 다음과 같다. (1) 앱 단어장에 저장하고 싶은 단어들을 형광펜으로 강조한다. (2) 앱을 실행하여 단어가 적혀있는 종이를 스캔한다. (3) 스캔된 사진에서 추출하기 원하는 색을 터치한다. (4) 사용자가 선택한 형광펜 색에 해당하는 단어들만 검색이 된다. (5) 최종적으로 사용자가 단어가 잘 인식되었는지 확인한다. (6) 확인이 끝나면 사용자가 원하는 뜻을 직접 입력한 후 저장한다.

3.3. 사용자 인터페이스

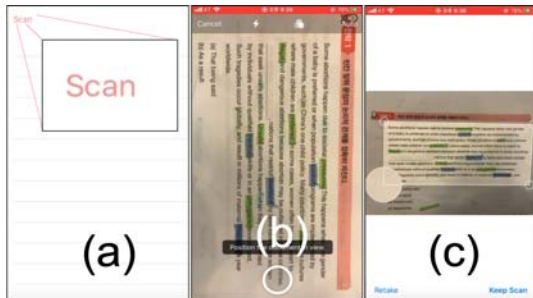


그림 3. 앱의 사용자 인터페이스(스캔 영역)
Figure 3. User interface of app(Scanning area)

앱을 처음 실행하면 <그림 3>의 (a)와 같이 사용자는 앱의 “Scan” 버튼을 먼저 누르게 된다. “Scan” 버튼을 누르게 되면 (b)화면으로 이동하게 된다. (b)화면은 문서를 스캔하는 화면이다. 이 화면에서 사용자가 등록하기 원하는 독해 지문을 촬영하게 된다. 카메라로 촬영이 완료되면 바로 (c)화면으로 이동하게 된다. (c)화면에서는 스캔된 화면에서 원하는 부분을 선택해서 잘라낸다. 직사각형의 꼭짓점 4개를 이용해 잘라낼 수 있다. (c)화면에서는 불필요한 부분을 제외한 본문의 영역만 남겼다. 그 다음 “Keep Scan” 버튼을 누르면 최종적으로 필요한 영역만 <그림 4>의 (d-1)화면으로 넘어온다.



그림 4. 앱의 사용자 인터페이스(단어 추출 영역)
Figure 4. User interface of app(word extraction area)

화면 (d-1)에서 추출하기 원하는 색상을 터치하면, (d-2)에 터치한 색의 텍스트 영역만 보여주게 된다. 동시에 (d-3)에 사용자가 선택한 색상의 단어 들만 인식하여 보여주게 된다. 화면 (d)의 왼쪽은 초록색 형광펜 영역을 터치한 결과이고, 오른쪽은 파란색 형광펜 영역을 터치한 결과이다. 최종적으로 화면 맨 아래 “OK” 버튼을 누르면 인식된 단어 들이 (e)화면으로 넘어와서 사용자가 직접 원하는 뜻을 입력할 수 있다. 사용자가 뜻 입력을 마치면 최종적으로 저장한다.

4. ‘찍어보카’ 앱 구현

‘찍어보카’ 앱의 핵심은 형광펜으로 칠해진 단어만 추출하는 것이다. 이를 세 과정으로 나누면 다음과 같다. (1) 형광펜으로 칠해진 영역을 검출한다. (2) 형광펜 영역을 제외한 부분은 지운다. (3) 남겨진 형광펜 영역의 글자만 인식한다. (1)과 (2) 과정은 OpenCV 라이브러리를 사용하여 알고리즘을 직접 만들었다. (3) 과정은 Apple Inc.에서 제공하는 Vision 라이브러리 안에 좋은 글자 인식 기능이 이미 있어서 상대적으로 쉽게 해결할 수 있었다. 본 장에서는 이미지 처리 과정인 (1)과 (2)과정을 중심으로 다룬다. 다음 <그림 5>는 ‘찍어보카’ 앱의 순서

도(flowchart)이다.

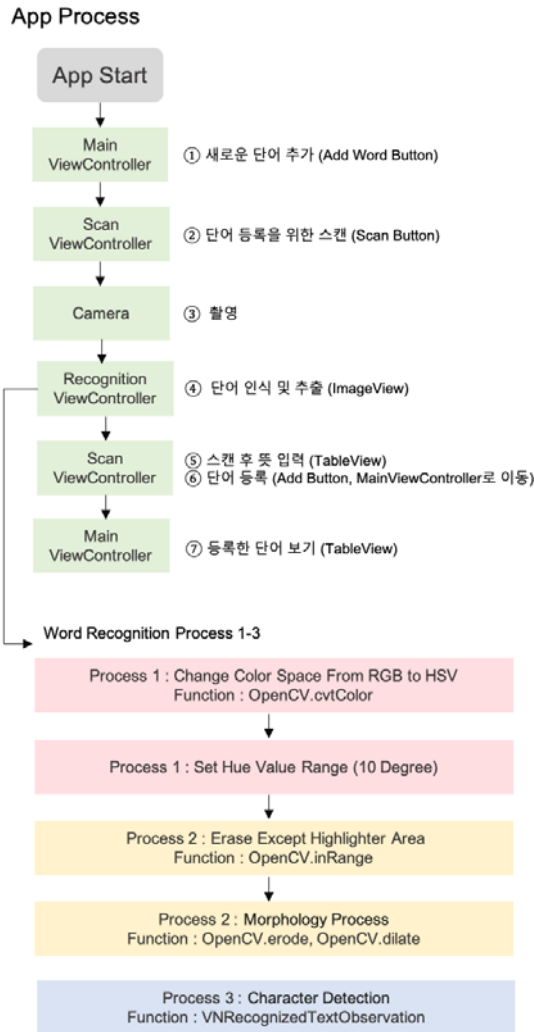


그림 5. 앱의 순서도(단어 추출 영역)
Figure 5. Flowchart of App

4.1. 과정 (1): 형광펜으로 칠해진 영역 검출

1) RGB 색공간을 HSV 색 공간으로 변환

iOS플랫폼에서 모든 이미지 정보는 ‘UIImage’ 라는 클래스로 표현된다. 이때 이미지는 기본적으로 RGB 색 공간을 사용한다. 색의 검출을 위해 이미지의 RGB 색 공간을 먼저 HSV 색 공간으로 변환하는 과정이 필요하였다. RGB 색 공간은 빨강, 초록, 파랑이라는 세 가지 색의 적절한 조합으로 한 가지 색을 표현한다. 하지만 세 가지 색의 조합으로는 색상, 채도, 명도를 알 수 없기 때문에 특정 색상만 추출하기에는 적절하지 않기 때문이다. 반면에 HSV 색 공간은 색상(Hue), 채도(Saturation), 명도(Value)의 조합으로 한 가지 색을 표현한다. 그래서 특정 색상(Hue)의 범위를 정하기가 쉽게 때문에 HSV 색 공간을 사용하였다. 색 공간을 변환하기 위해 OpenCV함수인 ‘cvtColor’ 함수를 사용하였다.

2) 색상(Hue) 범위 정하기

카메라로 피사체를 촬영하면 주변 조도, 카메라의 위치, 카메라의 각도에 따라 같은 피사체라도 다른 사진이 나오게 된다. 이러한 특성이 ‘찍어보카’ 앱에서도 존재한다. 만일 빨간색 형광펜으로 칠해진 영역을 촬영한다면, 앞서 말한 여러 조건들로 인해 매번 다른 색깔의 결과가 나올 것이다. 사람의 육안으로 보면 같은 빨간색으로 판단할 수 있을지 모르나, 컴퓨터는 아니다. 컴퓨터는 조금이라도 빛의 양이 달라지면, 색상(Hue)값이 달라진다. 이러한 특성을 고려하여 HSV 색 공간에서 적절한 범위 설정이 필요하였다.

OpenCV에서 색상은 0~180, 채도와 명도는 0~255의 값을 갖는다. HSV 색 공간에서 채도나 명도 둘 중에 하나가 너무 낮으면 <그림 6>과 같이 색 구분이 어려워지기 때문에 채도와 명도 값을 둘 다 최솟값은 30으로 하고 최댓값은 255로 설정하였다.

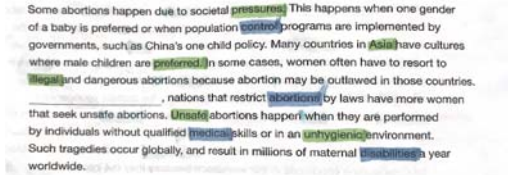


그림 6. 명도와 채도를 매우 낮게 했을 경우의 이미지
Figure 6. Image with very low brightness and saturation

‘찍어보카’ 앱은 사용자가 이미지 속 형광펜 영역을 터치하여 기준색을 정할 수 있다. 인식된 기준색은 <그림 6>가운데에 있는 직사각형 영역에 보이게 된다.

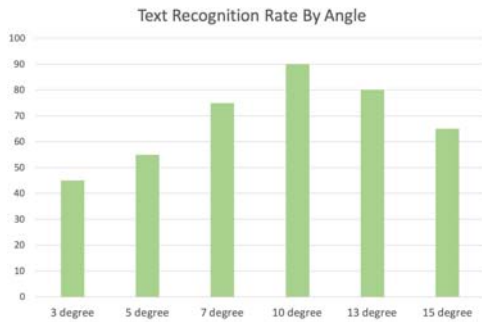


그림 7. 각도에 따른 글자 인식률
Figure 7. Text recognition rate by angle

최적의 색상 값의 범위를 정하기 위해 기준이 되는 색을 기준으로 좌우 각도를 각각 3도, 5도, 7도, 10도, 13도, 15도로 설정하여 실험(각각 20회 시행)

을 진행하였다. 실험 결과는 <그림 7>과 같이 10도가 가장 인식률(90%)이 좋았다. 따라서 <그림 8>과 같이 기준이 되는 색을 기준으로 좌우 10도를 포함하는 영역을 색상 값의 범위로 정하였다. <그림 8>에서 흰색 선은 기준 색이고, 검은 선들은 흰색 선을 기준으로 좌우 10도 벌어져 있는 선들이다.

4.2. 과정 (2): 형광펜으로 칠해진 영역을 제외하고 지우기

카메라로 촬영된 수많은 단어 중에서 사용자가 원하는 단어는 형광펜으로 칠해진 단어들이다. 이러한 단어들만 인식하기 위해서는 과정 (1)에서 검출되지 않은 영역은 지워줘야 한다. 이를 구현하기 위해 OpenCV의 ‘inRange’ 함수를 사용하였다. 이 함수는 색 범위를 인자 값으로 넘겨주면 해당 색 범위만 남기고 <그림 9>처럼 나머지는 지워져 흰색으로 된다.

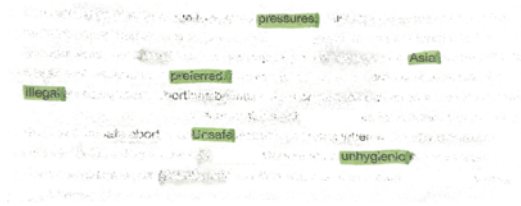


그림 8. OpenCV의 색상(Hue) 원
Figure 8. Hue circle in OpenCV

그러나 ‘inRange’ 함수는 모든 픽셀을 검사하여 인자 값으로 넘겨진 범위에 들어가는 모든 픽셀을 찾아내기 때문에 <그림 10>처럼 불필요한 점, 면, 글자까지 인식되는 문제가 생긴다. 따라서 <그림 9>의 ‘Recognized Text’ 목록의 2번 단어처럼

원하는 단어를 정확하게 인식하기가 어려워진다.

할 수 있게 되었다.



Recognized Text

Total

- 1.pressures
- 2.preferred J
- 3.illegal
- 4.Unsafe
- 5.unhygienic

그림 9. 'inRange' 함수를 적용한 이미지
Figure 9. Image with the 'inRange' function applied

<그림 10>과 같은 문제를 해결하기 위해 모폴로지(Morphology) 연산 기법을 사용하였다. 모폴로지 연산은 간단한 연산의 집합으로 이루어져 있다. 이 연산은 복잡한 모양을 처리하는 문제를 해결할 수 있다[4]. 모폴로지 기법은 미리 특정한 필터를 만들고, 이 필터를 이미지의 각 픽셀에 적용하여 원하는 결과를 얻는다.

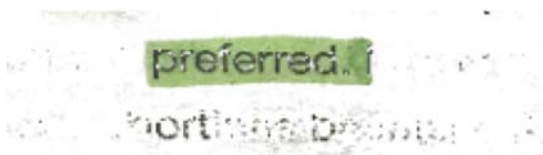


그림 10. <그림 9>을 부분 확대한 이미지
Figure 10. Enlarged image of <figure 9>

모폴로지 기법을 사용하여 <그림 10>와 같은 이미지에서 작은 점들을 모폴로지 기법을 사용하여 <그림 10>와 같은 이미지에서 작은 점들을 제거하고, 색상이 비슷한 영역은 합쳐주었다. 이를 위해 OpenCV의 'erode' 와 'dilate' 함수를 사용하였다. 그 결과 <그림 11>과 같이 원하는 형광펜 영역만 검출



그림 11. 모폴로지 연산 기법을 적용한 이미지
Figure 11. Image with morphology algorithm

하지만 사용자에 따라 처리되는 이미지의 크기가 다르기 때문에 이미지의 사이즈에 따라 정확도가 달라지는 문제점이 남아있었다. 따라서 이미지의 비율을 유지하면서 이미지의 가로 길이를 800으로 정하였다. <그림 12>의 먼저 (1)처럼 입력된 이미지(inputImage)의 비율(Ratio)을 구한다. 다음으로 (2)와 같이 입력된 이미지의 높이 값에 비율을 곱해준다. 마지막으로 가로 길이가 800으로 고정된 출력 이미지를 만들어 준다.

- (1) $Ratio = 800 / inputImage.Width$
- (2) $outputHeight = inputImage.Height \times Ratio$
- (3) $outputImage = (width: 800, height: outputHeight)$

그림 12. 이미지 크기 변환 코드
Figure 12. Image size conversion code

4.3. 과정 (3): 남겨진 형광펜 영역의 글자 인식

마지막으로 Apple Inc.에서 제공하는 'Vision' 라이브러리에 있는 'VNRecognizeTextRequest' 함수를 이용하여 글자 인식을 요청하고, 같은 라이브러리에 있는 'VNRecognizedTextObservation' 함수를 이용하여 글자를 검출하였다.

글자 인식에 사용되는 'Vision' 라이브러리는 'Core ML' 이라는 Apple Inc.의 머신러닝 프레임

워크를 사용한다. Core ML은 일반 머신러닝과 마찬가지로 컴퓨터에 다양한 이미지를 학습시키고, 학습된 결과를 바탕으로 특정 이미지를 판단할 수 있다.

본 논문에서 사용한 글자 인식의 원리는 다음과 같다. 먼저 <그림 13>과 같이 여러 글자 이미지를 컴퓨터에게 학습시킨다. 그림 12의 (a)는 글자 이미지 집합이고, (b)는 글자가 없는 이미지 집합이다. ‘Core ML’은 각 집합을 구분해서 학습시킨다. 예를 들어 (a)의 첫 번째 열, 네 번째 행의 이미지를 학습시킨다고 했을 때, 컴퓨터에게 이 이미지는 ‘D’라고 알려준다. 컴퓨터는 (a)에 있는 이미지 외에도 수많은 다른 ‘D’ 문자의 이미지를 학습한다. 학습된 결과를 바탕으로 컴퓨터는 특정 이미지 속에서 ‘D’ 문자를 인식하게 된다.



그림 13. 기계학습 이미지 집합[15]
Figure13. Image set for machine learning[15]

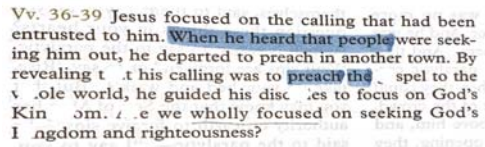
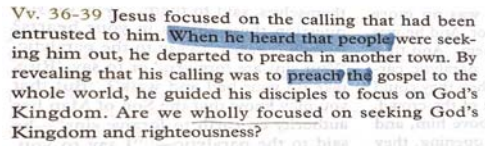
‘Vision’ 라이브러리를 사용하여 글자 인식에 성공했지만, <그림 11>에 ‘preferred’ 끝에 찍혀있는 ‘.’와 같은 문자와 띄어쓰기 뒤에 있는 ‘l’ 문자도 같이 인식되는 문제가 남아있었다. 따라서 특정 문자와 띄어쓰기 뒤에 있는 문자를 제거해주는 함수를 만들어서 최종적으로 앱을 구현하였다.

5. ‘찍어보카’ 앱 실험

‘찍어보카’ 앱 실험은 4가지 경우로 나누어서 진행하였다. 1) 형광펜 색이 너무 어두운 경우, 2)

형광펜이 글자 영역을 일부만 칠한 경우, 3) 이미지에 형광펜과 비슷한 색상이 있을 경우, 4) 기대되는 경우로 나누었다. (여기서 기대되는 경우란 사용자가 앱을 사용목적에 따라 정상적으로 사용하게 되는 경우를 의미한다. 본 논문에서는 글자 인식의 오류 없이 정확하게 인식되는 경우를 뜻한다.) 1), 2), 3)의 경우는 글자 인식의 오류가 생기는 경우이고, 4)의 경우는 사용자가 기대하는 정확한 인식의 경우이다.

1) 형광펜 색이 너무 어두운 경우



Recognized Text	Total
1.Vv Jesus focused on the calling that had been	
2.entrusted to him When he heard that people were	
seek	
3.ing him out he departed to preach in another town	
By	
4.revealing	
5.his calling was to preach the ospel to the	
6.ole world he guided his disc	
7.es to focus on Gods	
8.Kin	
9.om	
10.we wholly focused on seeking Gods	
11. Kingdom and righteousness?	

그림 14. 형광펜 색이 너무 어두운 경우
Figure14. Case of that highlighter's color is too dark

<그림 14>는 어두운 색상의 형광펜을 사용하였다. 그 결과, 형광펜으로 칠하지 않은 영역까지 인

식되는 오류가 발생하였다. 형광펜이 어두울수록 밝기 값은 낮아지게 된다. HSV색 공간에서 밝기가 낮아지면 색상은 검정색에 가까워진다. 따라서 글자도 검정색이기 때문에 <그림 14>와 같이 주변에 있는 단어까지 전부 인식이 되었다.

2) 형광펜이 글자 영역을 일부만 칠한 경우



그림 15. 형광펜으로 일부만 칠한 경우
Figure15. Case of that highlighter is painted with partially

<그림 15>는 형광펜이 글자 전부를 칠하지 않고 일부만 칠한 결과를 보여준다.

일부만 칠하게 되면 기대하는 글자가 인식되지 않는다는 것을 알 수 있다. <그림 15>에서 주황색으로 칠한 문장은 ‘they brought to him’ 이다. 하지

만 ‘they’ 단어의 위쪽 부분이 칠해지지 않아 인식에 오류가 발생했다. <그림 15>에서 알 수 있듯이 ‘they’ 가 ‘mey’ 로 잘못 인식되었다.

3) 이미지에 형광펜과 비슷한 색상이 있을 경우

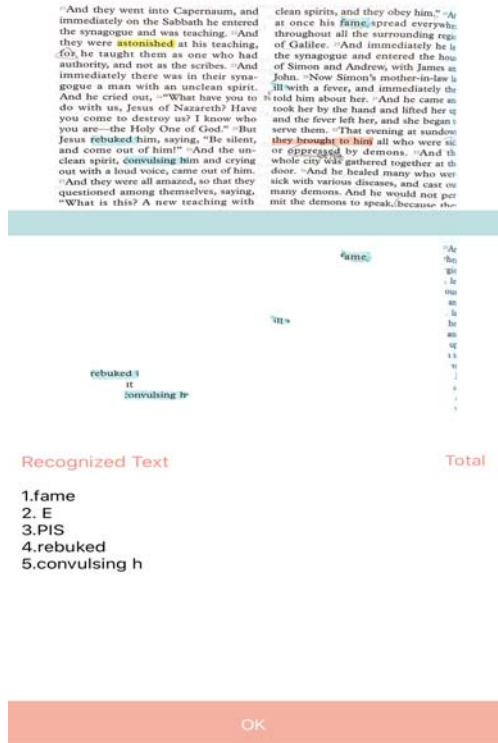


그림 16. 이미지에 비슷한 색상이 있을 경우
Figure16. Case of that similar colors exist in the image

4.2장에서 언급했듯이 ‘찍어보카’ 앱은 색상(Hue)값을 기준으로 형광펜이 칠해진 영역을 감지한다. 하지만 만일 형광펜과 비슷한 색상값을 가진 픽셀이 이미지 내에 존재한다면, 그 픽셀 또한 감지가 될 것이다. 4.3장에서 소개한 모폴리지 연산을 사용하면 작은 픽셀들은 제거가 되겠지만, 영역이 큰 색상 영역은 제거가 되지 않아 <그림 16>과 같이 불필요한 영역까지 감지가 되어 인식에 오류가 발생한다.

4) 기대되는 경우

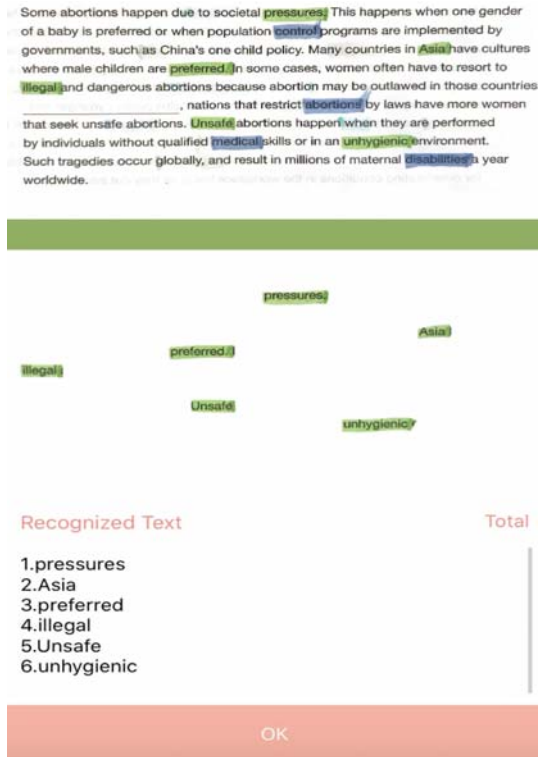


그림 17. 기대되는 경우
Figure17. Expected Case

<그림 17>은 앞서 실험한 3가지의 오류가 나는 경우를 모두 제외하였다. 그 결과 원하는 단어들 정확하게 인식되었다.

실험 결과를 정리하면 글자 인식에 영향을 미치는 요인은 형광펜의 색상, 칠해진 형광펜의 영역, 형광펜과 비슷한 픽셀이라는 것을 알 수 있다. 따라서 보다 정확한 글자 인식을 위해서는 앞서 실험한 3가지 오류의 경우를 피해야 한다는 결론이 도출된다. 이 3가지 오류를 모두 피한다면, 정확성이 높은 결과를 얻을 수 있다.

6. 결론

스마트폰 앱 시장에 영어 단어 암기를 도와주는 앱이 많지만, 직접 사용자가 하나씩 등록해야 하는 문제가 있었다. 본 논문에서는 비전 시스템을 활용한 ‘찍어보카’ 앱을 제안함으로써 사용자가 쉽고, 편하고, 빠르게 단어를 등록해서 사용하는 서비스를 제안했다. 이 앱을 사용했을 때 예상되는 기대 효과는 다음과 같다.

1) 공부의 시간적 효율성 증가

기존에 출시되어 있는 앱들은 학습자가 단어를 하나씩 입력해야 했다. 따라서 많은 단어들을 입력할 경우에 그만큼 시간이 오래 걸리게 된다. 하지만 본 논문이 제안하는 ‘찍어보카’ 앱은 여러 개의 단어를 한 번의 사진 촬영으로 입력이 가능하다. 그러므로 공부에 불필요한 시간이 단축된다. 그래서 학습자가 일일이 단어를 입력하는 시간을 절약하여 그 시간을 사용하여 공부를 더 할 수 있게 된다.

2) 학습자의 어휘능력 향상 도움

학습자는 시간이 지난 뒤에 다시 단어를 학습할 때 자신이 직접 촬영했던 사진도 같이 볼 수 있다. 모든 단어는 같은 단어라도 문맥에 따라 뜻이 달라진다. 만약 학습자가 어떤 단어를 학습 했다고 했을 때, 그 단어가 사용된 문장 까지 같이 볼 수 있다면 학습자는 그 단어가 어떤 문맥에서 사용되었는지 다시 한 번 기억할 수 있게 된다.

김익환[18]에 따르면, 어휘능력 향상을 위해 널리 쓰이는 것은 문맥 속에서 단어를 이해하게 되는 부수적 학습(incidental learning)이다. 또한 언어의 능력이 우수한 학생일수록 문맥 속에 있는 단어의 의미를 잘 유추한다.

따라서 ‘찍어보카’ 앱으로 단어를 공부하게 되

면 공부를 할 때 시간이 절약되고, 문맥기반의 어휘학습이 가능하기 때문에 학습자의 어휘능력 향상에 도움을 줄 수 있을 것이라 기대된다.

References

- [1] H. Nicod, H. Felipe, A. Mayutan, A. Mohamed, P. Konstantina, and N. Saverio, *What's app? A wide-scale measuremonet study of smart phone markets*, ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 16, No. 2. pp. 1, 2012.
- [2] F. Martin, and J. Ertzberger, *Here and now mobile learning: An experimental study on the use of mobile technology*, Computers & Education, Vol. 68, pp. 76-85, 2013.
- [3] Model-View-Controller, <https://developer.apple.com/library/archive/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>, Feb. 2020.
- [4] R. O. Janderson, A. S. Edson, and A. F. Roseli, *mathematical morphology filters applied to an image skeletonization method to generation of topological maps*, ICIS '09: Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, Seoul Korea, pp. 1312-1317, 2009.
- [5] K. Adrian, and B. Gary, *Learning OpenCV 3*, O'Reilly Media, Inc., 2016.
- [6] H. Jon, *Mastering swift 4 - fourth edition*, Packt Publishing, 2017.
- [7] K. Kirill, and S. Alexander, *Instant OpenCV for iOS*, Packt Publishing, 2013.
- [8] W. Qun, *Designing a smartphone app to teach English (L2) vocabulary*, Computers & Education, Vol. 85, pp. 170-179, 2015.
- [9] S. S. So, and H. Y. Hur, *iPhone app programming (for iOS beginners): iOS 7 edition*, Easy Publishing, Seoul, 2014.
- [10] D. Michel, <https://github.com/MichelDeiman>, Feb. 2020.
- [11] OpenCV Tutorials, https://docs.opencv.org/master/d9/df8/tutorial_root.html, Feb. 2020.
- [12] Framework UIKit, <https://developer.apple.com/documentation/uikit>, Feb. 2020.
- [13] VNRecognizeTextRequest, <https://developer.apple.com/documentation/vision/vnrecognizetextrequest>, Feb. 2020.
- [14] Human Interface Guidelines(iOS), <https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/>, Feb. 2020.
- [15] photo OCR, <https://daeson.tistory.com/228>, Mar. 2020.
- [16] Onevoca, <https://apps.apple.com/kr/app/%EC%9B%90EB%B3%B4%EC%B9%B4-%EB%82%98%EB%A7%8C%EC%9D%98-%EB%8B%A8%EC%96%B4%EC%9E%A5/id1242173441>, Mar, 2020.
- [17] English word infinite exam, <https://apps.apple.com/kr/app/%EC%98%81%EB%8B%A8%EB%AC%B4%EC%8B%9C-%EC%98%81%EC%96%B4-%EB%8B%A8%EC%96%B4-%EB%AC%B4%ED%95%9C-%EC%8B%9C%ED%97%98/id1138211987>, Mar, 2020.
- [18] I. H. Kim, *English vocabulary learning and contextual analogy*, East-West Culture, Vol. 31, pp.43-62, 1998.
- [19] J. H. Jo, and S. H. Lee, *PTZ camera based real-time object tracking System using OpenCV*, KKITS, Vol. 7, pp. 59-65, 2012.
- [20] M. S. Sang, *Effective implementation for fast deep learning algorithm*, KKITS, Vol. 14, pp. 553-561, 2019.

비전시스템을 이용한 iOS 영어단어장 앱 개발

이명철¹, 이강희²

¹승실대학교 글로벌미디어학부 학사과정

²승실대학교 글로벌미디어학부 교수

요 약

스마트폰이 나오기 전에는 영어 공부를 하는 사람들은 단어를 공책에 정리를 하곤 했다. 시간이 지나 스마트폰이 보급이 되면서 공책보다 단어장 앱에 단어를 저장하여 공부하는 사람들이 생겨났다. 사용자들은 언제 어디서든 스마트폰을 통해 단어를 공부할 수 있게 되었다. 이러한 단어장 앱들은 사용자가 직접 데이터를 입력하는 방식이다. 하지만 본 논문에서는 비전시스템과 글자인식을 이용하여 자동으로 단어를 입력해주는 앱을 설계 및 구현했다. 이 앱은 형광펜으로 칠해진 영역을 먼저 인식한 후, 형광펜 영역 안에 있는 단어를 인식하여 테이블로 정리해 준다. 또한 정리된 단어를 간단한 동작으로 특정 단어가 어떤 영어 본문에서 어떻게 문맥에 따라 의미가 사용되는지 알 수 있도록 도와준다. 단어의 데이터는 특정 본문에 여러개의 단어가 존재할 수 있으므로, 카메라로 본문을 찍은 이미지와 해당 본문에 포함되는 단어들을 일대다의 관계로 데이터베이스 스키마를 설계하였다. 이 앱의 차별화된 장점은 다음과 같다. 첫째, 공부의 시간적 효율성을 높여준다. 둘째, 문맥 기반으로 학습하여 학습자의 어휘능력을 향상 시켜준다. 본 논문에서는 이러한 장점들을 실제적인 연구 자료와 기술을 통해 자세히 설명한다. 앱의 플랫폼은 Apple Inc.의 iOS를 사용하였다. 앱의 핵심 기술인 글자 인식은 OpenCV의 라이브러리와 Apple Inc.의 CoreML을 활용하였다.

감사의 글

이 논문은 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2017R1D1A1B05036028). 이 논문은 2019년 대

한민국 교육부와 한국연구재단의 인문사회분야 중견연구자지원사업의 지원을 받아 수행된 연구임(NRF-2019S1A5A2A01043252)



Myeong-Cheol Lee is in the bachelor's degree in Global School of Media from Soongsil University, Korea. He belongs to the Media Robotics Laboratory. He is interested in physical computing and artificial intelligence.

E-mail address: understanding_light@icloud.com



Kang-Hee Lee received the B.S., M.S. and Ph.D. degrees in EECS from KAIST, Korea, in 1999, 2001 and 2006, respectively. Dr Lee is currently an Associate Professor in Global School of Media, Soongsil University. He is interested in robotic intelligence & emotion and media robotics.

E-mail address: kanghee.lee@ssu.ac.kr