

GPS 기반의 실시간 정보 제공 시스템 개발

김충일, 정승원, 김형준, 황인준
고려대학교 전기전자공학과

e-mail:(cilkim1, jsw161, hyungjun89, ehwang04)@korea.ac.kr

Development of GPS-Based Real-Time Information Service System

Chung-Il Kim, Seungwon Jung, Hyungjoon Kim and Eenjun Hwang
School of Electrical Engineering, Korea University

요 약

최근 사용자 편의성 제공을 위한 위치 기반의 데이터 마이닝 활용 서비스가 다양하게 개발되고 있다. 대부분의 이러한 응용 서비스들은 지도 API를 기반으로 한 정적인 정보를 주로 이용하여 사용자에게 서비스를 제공한다. 하지만, 사용자는 그에 비해 해당 GPS 좌표 위치 기반의 실시간 정보 서비스 요구한다. 이에 본 논문에서는 해당 GPS 좌표 위치의 공간 인식, 밀도 및 움직임 포착 등을 파악하고 이를 기반으로 새로운 서비스를 제공하는 기법을 제안한다.

1. 서론

IT 기술이 발달함에 따라, 사람들이 식사 시간에 음식점이나 메뉴 등을 선택할 때, 미디어 매체나, SNS 및 인터넷 등을 통해 타인의 조언이나 추천 등을 참고하여 결정하는 것이 흔한 풍경이 되었다[1]. 특히 미디어 매체나 인터넷상에서 음식점을 방문한 경험이나 의견을 글이나 사진으로 작성하여 여러 사람과 공유함으로써 좀 더 사실적이고 객관적인 자료를 제공할 수 있게 되었다. 한편, 컴퓨터 기술의 발달로 인터넷상에서 생성되는 데이터의 양이 방대해짐에 따라 그러한 데이터에서 유용한 정보를 찾아내기 위한 데이터 마이닝 기술이 크게 활용되고 있다 [2]. 이와 함께, 데이터 마이닝을 이용한 관련 서비스로 사용자 맞춤형 서비스도 빠르게 성장하고 있다. 이러한 여건들이 맞물려 인터넷상의 음식점이나 메뉴와 관련된 정보를 이용하거나 음식에 특화된 서비스를 사용자에게 효과적으로 제공하기 위한 연구들이 늘어나고 있다.

그러나, 이러한 서비스들을 살펴보면, 실제 길거리에서 손쉽게 사용하기에는 아직 개선의 여지가 많다. 예를 들어, '배달의 민족'이나 '요기요'는 개발부터 배달에 중점을 두었기 때문에 직접 가서 먹을 때 참고할만한 정보가 부족하다. 반면, '다이닝 코드'의 경우, 현재 식당의 상태를 알기 어렵다. 다시 말해서, 다이닝 코드를 참고하여, 순위가 높은 음식점을 직접 방문하면 사람이 많아 바로 식사하지 못하고 기다리거나 다른 음식점을 찾아야만 하는 번거로움이 발생할 수 있다. 또한, 다이닝 코드뿐만 아니라 네이버, 구글 같은 포털 사이트의 경우에도 주변에 정보를 알고 싶은 음식점이 있더라도 직접 검색을 통해서 관련된 식당의 정보와 리뷰를 찾아봐야 한다.

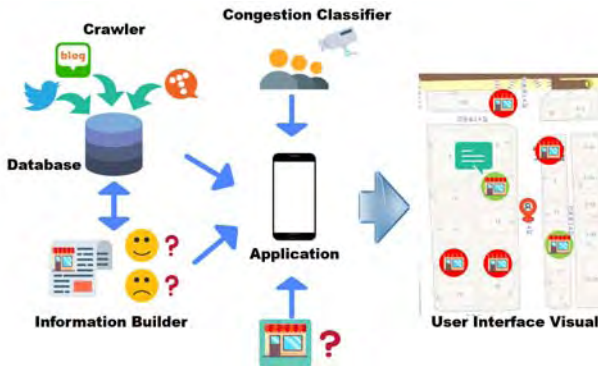
이러한 불편함과 문제점을 고려하여, 본 연구에서는 실제 사용자가 알고 싶어 하는 음식점에 대한 정보를 제공하고, 원하는 음식점을 빠르게 찾을 수 있게 해주는 시스템을 제안하고자 한다. 특히, 다른 애플리케이션에 비해 실제 길거리에서 더 유용하고 편리하게 사용할 수 있게 하고자 한다. 이를 위해, 지도 API를 기반으로 사용자가 주변을 쉽게 인식할 수 있도록 하며, 사용자에게 필요한 정보를 제공하기 위해 음식점 정보와 블로그, SNS에서 해당 음식점에 대한 리뷰를 데이터베이스에 저장하고 정리한다. 다음으로 길거리에서 식사할 음식점을 찾는 경우, 사용자가 근처 음식점의 간판 사진만으로 쉽게 음식점 정보를 얻을 수 있는 기능을 제공하여, 사용자가 편리하게 정보를 찾아볼 수 있도록 한다.

2. 시스템 구성도

본 논문에서 제안하는 시스템 구성도는 그림 1과 같다. 구성 요소로는 기능에 따라 크롤러(Crawler), 감정 인식기, 데이터베이스, 식당 인식기(Restaurant Recognizer), 혼잡도 분류기(Congestion Classifier)로 구성한다. 2장에서는 각 요소의 구체적인 기능과 구현 방법과 사용자 인터페이스에 관해 설명한다.

2.1 크롤러(Crawler)

본 논문에서 제안하는 응용 시스템을 구축하기 위해서는 사전에 관련 데이터의 수집이 필수적이다(필요하다). 이를 위해, 응용프로그램 인터페이스(API)의 종류인 네이버 검색 API와 Google Place API로 총 2개의 API를 이용하여 응용 시스템 구축한다. 네이버 검색 API[3]와



(그림 1) 시스템 구상도

Google Place API[4]에서 수집한 정보는 음식점 이름, 위치, 영업시간, 메뉴, 리뷰와 별점 등이다.

2.2 감정 인식기

사용자에게 유의한 정보를 제공하기 위해서는, 비정형 데이터의 수집 후 정보 추출을 통해 객관적인 사실을 토대로 이를 제공해야 한다. 음식점의 평가는 사람마다 주관성이 개입되어 있으므로, 이러한 정보들에서 객관성 있는 정보를 추출할 필요가 있다. 본 논문은 앞서 2.1장에서 수집한 데이터를 토대로 평가 점수를 나타낼 수 있는 감정 인식기를 구축한다. 감정 인식기는 합성곱 신경망(Convolutional Neural Networks)을 기반으로 구현하고, 음식점 리뷰와 별점을 통해 평가 점수를 산출한다. 감정 인식기를 학습하기 위해 형태소 기반 주석 도구를 활용하여[5], 음식점의 리뷰를 형태소 기반으로 나눈다. 이 형태소와 별점은 평가기를 학습시키는 데이터 셋이 된다. 이를 통해 검색된 음식점에 대해 평가 점수를 표시한다.

2.3 데이터베이스(Database)

앞서, 2.1장과 2.2장에서 기술한 것과 같이, 데이터는 사용자에게 제공할 수 있는 형태로 전처리 과정 후 데이터베이스에 저장된다. 제안한 응용 시스템은 구글 파이어베이스(Firebase)를 이용하여[6], 데이터베이스를 구축한다. 파이어베이스는 모바일 앱 및 웹 애플리케이션 개발 플랫폼으로, JSON 형태로 데이터를 저장, 삭제가 가능한 실시간 데이터베이스, 파일을 저장할 수 있는 스토리지 등을 제공한다. 실시간 데이터베이스는 크롤러와 감정 인식기 학습 데이터 두 분류로 나누어 데이터를 관리한다. 스토리지(Storage)는 음식점에 대한 사진과 사용자가 간판 사진 인식을 위해 보낸 사진을 임시로 저장하는 위치로 사용된다.

2.4 음식점 인식기(Restaurant Recognizer)

사용자가 그들의 고려 대상으로 음식점을 발견하였을 때, 음식점에 관한 정보를 검색하기 위해서는 포털 사이트에 입력해야 하는 번거로움이 존재한다. 사용자에게 편의성을 제공하기 위해, 음식점의 간판 사진을 찍었을 때 해당 음식점을 인식하고 정보를 사용자에게 보여주는 기능

을 구현한다. 음식점 인식기는 CNN을 기반으로 한 간소화된 GoogleNet 알고리즘을 활용한다[7]. 분류기 학습을 위해 먼저 음식점마다 입구, 간판을 찍은 사진을 수집한다. 이 데이터는 분류기를 학습시키는 데이터가 되며, 분류기의 결과 클래스는 각 음식점의 이름이 된다. 따라서 분류기의 결과 클래스는 정보가 존재하는 음식점만큼 늘어난다.

2.5 혼잡도 분류기(Congestion Classifier)

일상생활에 인터넷이나 대중 매체를 통해 음식점에 대한 정보를 얻은 사용자가 그 음식점에 방문하였을 때, 음식점에 사람이 많아 기다리거나 다른 정보를 찾는 번거로움이 존재한다. 사용자에게 음식점의 실시간 정보를 제공하기 위해 가게 안의 CCTV 화면을 이용하여 음식점 내 혼잡도를 분류하는 기능을 구현한다. 하지만, 실제 개인정보 문제 등으로 인해 CCTV의 데이터를 사용하는 것은 여러 문제가 발생할 수 있다. 본 논문은 칼텍 보행자 데이터 셋을 활용하여 구현한다[8]. 이 분류기는 언급한 데이터 셋을 사용하여, 해당 공간이 사람으로 인해 혼잡한지 한산한지 2가지 결과로 분류한다.

2.6 UI(User Interface)

본 논문에서 제안하는 사용자 인터페이스는 Google Maps API[9]를 기반으로, 사용자와 음식점 간의 거리를 시각적으로 표현하여 사용자가 많은 움직임을 요구하지 않도록 실질적인 도움을 제공한다. 이를 기반으로 지도에 마커, 이미지, 경로를 시각적으로 제공한다.

3. 애플리케이션

이 장에서는 2장에서 설명했던 기능들을 기능별 UI 단계별로 설명한다. 구현된 응용 시스템은 안드로이드 기반 모바일 애플리케이션이며, Minimum SDK 버전은 21 이상으로 설정하였다.

3.1 기본 사용자 인터페이스

그림 2는 기본적인 사용자 인터페이스를 캡처한 그림이다. (a)는 애플리케이션의 로딩화면을 보여준다. 이 UI 단계에서는 파이어베이스로부터 음식점 정보를 로드하고 애플리케이션을 실행하는데 필요한 퍼미션(Permission)을 관리하는 작업을 수행한다. (b)는 이 본 논문에서 제안하는 애플리케이션의 기본 화면이자 메인 액티비티(Main Activity)로 2.6장에서 설명한 Google Maps API와 2.5장에서 활용한 혼잡도 분류기를 활용하여 인근 음식점 정보를 표시한다. 여기서 음식점 마크에 초록색 배경은 손님이 많지 않아 자리가 여유로움을, 붉은색 배경은 손님이 많아 자리가 별로 없음을 나타낸다. 이는 음식점 선택의 폭을 넓혀준다. (c)는 기본 화면에서 음식점을 선택하였을 때 표시되는 정보를 나타낸다. 음식점 이름, 음식점 범주, 가



(a) 로딩 화면



(b) 기본 화면



(c) 음식점 선택 화면

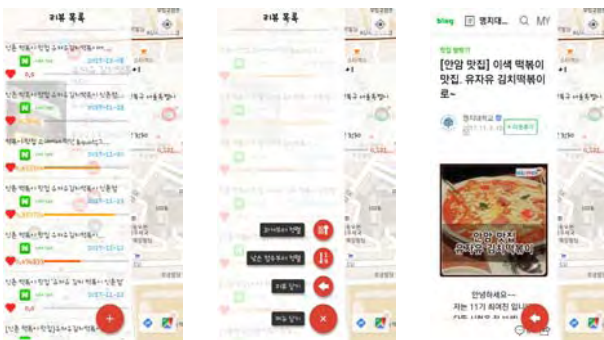


(d) 메뉴

(그림 2) 기본적인 사용자 인터페이스

계 주소, 전화번호 및 해당 요일에 맞는 영업시간이 표시된다. 가장 하단에 나타나는 프로그레스 바(Progress Bar)는 2.2장에서 제시한 긍정/부정 평가기를 통해 도출해 낸 평가 점수의 평균을 나타낸다. 쉽게 알아볼 수 있도록 이 막대와 감정 점수의 색은 수치가 따라 변화하게 된다. (d)는 기본 화면의 오른쪽 아래에 있는 ‘+’ 버튼을 눌렀을 때 나타나는 메뉴이다. 새로 고침은 기본 화면의 모든 마커, 즉 음식점을 나타내는 아이콘들을 초기 상태로 변환한다. 나머지 메뉴와 리뷰에 대해서는 3.2장에서 자세히 다룬다.

3.2 리뷰



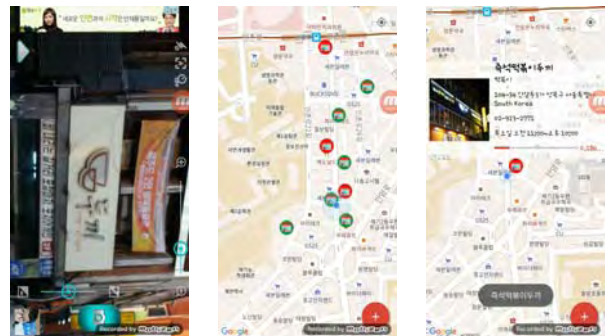
(a) 리뷰 목록 (b) 리뷰 정렬 (c) 리뷰 확인

(그림 3) 리뷰 예시 인터페이스

기본 화면에 표시된 음식점을 누르면 음식점 정보가 나타난다. 이때 음식점 정보를 한 번 더 누르면 리뷰 메뉴가 나타난다. 그림 3은 리뷰를 보여주는 인터페이스의 한 예시이다. (a)는 리뷰 목록 인터페이스에 대한 예시이다. 데이터베이스에 존재하는 음식점에 대한 리뷰들을 보여주고 있

으며, 게시물의 제목, 출처, 업로드 날짜가 표시된다. 각 리뷰의 하단 부분 프로그레스 바는 2.2장에서 제시한 감정 인식기를 통해 얻은 평가 결과를 나타내고 있다. 수치가 높을수록 게시 리뷰가 긍정적인 의견을 포함하고, 수치가 낮을수록 부정적인 의견을 포함하고 있다. 이 프로그레스 바 역시 음식점 선택 화면과 마찬가지로 수치에 따라 색이 달라져 사용자가 쉽게 확인할 수 있다. (b)는 리뷰 목록 오른쪽 아래에 있는 ‘+’ 버튼을 눌렀을 때 나오는 메뉴로, 정렬기준을 나타내고 있다. 정렬기준은 기본적으로 데이터베이스에 저장된 리뷰 중 가장 최근에 업로드된 리뷰가 먼저 나타나고, 사용자의 의도에 따라 정렬할 수 있도록 기준을 4개 추가하여 기능을 구현했다. ‘과거부터 정렬’은 리뷰의 작성 일자리를 기준으로 가장 오래된 리뷰부터 나오게 되며, ‘낮은 점수부터 정렬’은 감정 점수를 기준으로 가장 낮은 점수부터 나오게 정렬한다. 반대의 기능인 ‘최신부터 정렬’, ‘높은 점수부터 정렬’도 포함한다. 처음의 리뷰 목록은 리뷰 10개로 이루어져 있지만, 리뷰 목록이 나오는 리스트뷰(Listview)를 내리면 다음 목록을 읽어올 수 있다. (c)는 리뷰 목록 중 보길 원하는 리뷰를 눌렀을 때 나타나는 인터페이스에 대한 예시이다. 웹뷰를 이용하여 인터넷상에 작성된 원본 링크의 내용을 그대로 보여준다. 이는 데이터베이스의 저장 공간을 아낄 뿐만 아니라, 처리 과정에서 생길 수 있는 해석의 왜곡 없이 리뷰를 보여줄 수 있다.

3.3 음식점 인식기



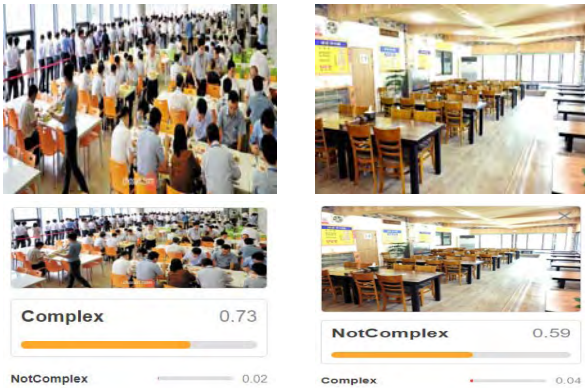
(a) 음식점 사진 촬영 (b) 대기 중 (c) 음식점 인식기 결과

(그림 4) 음식점 인식기 활용 인터페이스

제한한 응용 시스템의 음식점 인식기는 2.4장에서 설명한 음식점 인식기를 이용하여 사용자가 음식점 사진을 찍었을 때 어떤 음식점인지 쉽게 찾을 수 있도록 돕는다. 그림 4는 실제로 음식점 인식기를 활용하는 장면들이다. (a)는 애플리케이션 내에서 사용자가 스마트폰의 카메라를 이용하여 음식점 인식기에 보낼 사진을 찍는 장면이다. (a)처럼 사용자가 사진을 찍게 되면 사진은 사전에 학습한 음식점 인식기로 보내진다. 이후 애플리케이션은 (b)와 같이 음식점 인식기로 판별한 결과를 기다린다. 결과가 도착

하면 (c)와 같이 인식된 음식점만을 지도에 표시하고 해당 음식점의 정보를 보여준다.

3.4 혼잡도 분류기



(a) 혼잡한 상태 (b) 한산한 상태

(그림 5) 혼잡도 분류기 테스트 결과

혼잡도 분류기는 2.5장에서 언급했던 사진 학습한 분류기를 사용한다. 그림 5는 트레이닝 셋에 포함되지 않은 이미지를 이용하여 혼잡도 분류기를 테스트한 결과다. (a)는 혼잡한 상태의 이미지고 (b)는 한산한 상태의 이미지이다. (a), (b) 모두 제대로 판별된 것을 확인할 수 있다. 향후 연구로 가게의 CCTV 사진을 받을 수 있다면 혼잡도 분류기를 통해 ‘혼잡도 확인’ 버튼을 누르거나 애플리케이션을 실행할 때마다 인근에 있는 음식점의 CCTV 이미지를 전송받고 혼잡도 분류기를 이용하여 실시간으로 음식점 내부 상황이 어떤지 파악이 가능할 것으로 본다. 또한, 이를 그림 9의 (b)처럼 혼잡할 때는 빨간색 배경의 마커로, 한산할 때는 초록색 배경의 마커로 표시하여 사용자가 쉽게 확인할 수 있다.

4. 결론

본 논문은 기존의 음식점 관련 애플리케이션처럼 음식점에 대한 기본적인 정보의 제공뿐만 아니라 해당 음식점을 이용했던 리뷰를 제공하여, 실제 길거리에 더 적합한 응용 시스템을 제안하였다. 이를 위해, Google Maps API 기반으로 음식점 정보를 사용자에게 제공하며, 음식점 인식기로 사진으로 편리하게 음식점에 관한 정보를 얻을 수 있도록 제안하였다. 더불어, 리뷰에 담긴 감정을 분석하고 사용자에게 제공하고 혼잡도 분류기를 활용하여 사용자가 불편함을 갖지 않도록 음식점 내부 상황을 알 수 있도록 했다.

그러나, 애플리케이션을 구현하면서 몇 가지 아쉬운 점이 있었다. 대표적으로 혼잡도 분류기의 데이터 셋을 CCTV 데이터로 구현하지 못했다. 음식점의 CCTV 화면을 받을 수 없어 인터넷상의 사진만으로 혼잡도 분류기를 테스트해야 했다. 만약 CCTV 화면을 원할 때마다 받을 수 있다면 사용자가 요청할 때마다 음식점 내부 상황을 혼잡도 분류기를 통해 알 수 있었을 것이다.

Acknowledgement

본 결과물은 환경부의 재원으로 한국환경산업기술원의 환경정책기반공공기술개발사업의 지원을 받아 연구되었습니다(2017000210001).

참고문헌

- [1] Susskind, Alex M. "I told you so! Restaurant customers' word-of-mouth communication patterns." *Cornell Hotel and Restaurant Administration Quarterly* 43.2 (2002): 75-85.
- [2] Hill, Thomas, Pawel Lewicki, and Paweł Lewicki. *Statistics: methods and applications: a comprehensive reference for science, industry, and data mining*. StatSoft, Inc., 2006.
- [3] 네이버 검색 API, <https://developers.naver.com/products/search/>
- [4] Google Place API, <https://developers.google.com/places/web-service/search>
- [5] Shin, Hyopil, Munhyong Kim, Yu-Mi Jo, Hayeon Jang, and Andrew Cattle. 2013. KOSAC(Korean Sentiment Analysis Corpus): 한국어 감정 및 의견 분석 코퍼스, *Information and Computation*, pages 181-190
- [6] Firebase, <https://firebase.google.com/?hl=en>
- [7] Kim, Yeon-gyu, and Eui-young Cha. "Streamlined GoogLeNet algorithm based on CNN for Korean character recognition." *J. Korea Inst. Inf. Commun. Eng* 20.9 (2016): 1657-1665.
- [8] Caltech Pedestrian Dataset, http://www.vision.caltech.edu/Image_Datasets/Caltech_Pedestrians/
- [9] Google Maps API, <https://cloud.google.com/maps-platform/?hl=en>