

사진 인식을 이용한 레시피 추천 애플리케이션

김예진⁺, 김은영⁺⁺, 나현영⁺⁺⁺, 안현선⁺⁺⁺⁺

Recommending Recipe Application By Using Photo Recognition

Yejin Kim, Eunyoung Kim, Hyunyoung Na, Hyunsun Ahn

요 약

본 프로젝트는 요리하는 모든 사용자를 대상으로, 사진 인식 기술을 이용하여 요리할 때 이용하고자 하는 재료를 인식하면, 그 재료로 만들 수 있는 레시피들을 추천하여 요리에 도움을 주는 하이브리드 애플리케이션이다. 설계 제품 명칭은 ‘레시피고’로, 사진 인식 기술을 구현하기 위해 Google cloud-vision API를 사용한다. 일상생활에서 잘 쓰이지 않는 사진인식 기술을 사용하여 독창적인 방식으로 사용자들이 좀 더 편리하게 요리 레시피를 얻을 수 있게 한다. 본 제품은 사진으로 식재료 인식 및 식재료 선택 기능, 선택된 식재료들을 통한 레시피 추천 기능, 원하는 레시피에 좋아요 하는 기능을 제공한다.

1. 서 론

본 프로젝트는 사진인식 기술을 이용하여 요리할 때 이용하고자 하는 재료를 인식하면, 그 재료로 만들 수 있는 레시피들을 추천하는 하이브리드 애플리케이션이다.

기존의 다른 요리 애플리케이션들과 차별화 되는 가장 특별한 기능은 ‘사진인식’이다. ‘사진인식’은 재료를 일일이 입력하지 않고 사진으로 편리하게 입력을 할 수 있게 한다.

현재 우리나라에서는 일상생활에 사용하고 있는 여러 애플리케이션들 중 아직 많이 쓰이지 않고 있는 사진인식 기술이 이 애플리케이션에서 유용하게 사용될 것이고 이 기술의 편리함을 많은 사람들이 인식하게 되면, 더

많은 사진인식 기술을 이용한 애플리케이션이 개발될 것이라는 기대효과가 있다.

2. 시스템 설계

본 논문에서는 사진 인식 기술과 재료 기반 추천 알고리즘을 사용한 레시피 추천 애플리케이션인 ‘레시피고’를 개인 맞춤형으로 설계한다. 이는 ‘레시피고’가 기존의 단순한 레시피 추천 애플리케이션에서 벗어난, 보다 독창적이고 편리한 애플리케이션이라는 것을 의미한다.

‘레시피고’는 Ionic 프레임워크를 사용하여 MEAN 스택으로 개발되는 하이브리드 애플리케이션이다.

Ionic 프레임워크는 Cordova 기반으로 만들어졌다. 또한 AngularJS를 사용하여 MVC로 개발할 수 있는 환경을 제공한다. 여기서 Cordova는 JavaScript를 통해 카메라 같은 Native Device Function에 접근할 수 있는 API를 제공하며 다양한 종류의 플랫폼을 지원해 하이브리드 애플리케이션의 개발이 가능하다.

MEAN 스택은 MongoDB^[1], Express, AngularJS, NodeJS 이 네 가지 스택을 사용하여 개발하는 것을 의미한다. 이를 사용하게 되면 클라이언트와 서버를 분리할 필요없이 FullStack JavaScript 개발이 가능하다. NodeJS는 기본 처리가 비동기 방식이기 때문에 서버의 멀티쓰레드 문제에서 해방될 수 있으며 좋은 성능을 낸다. MongoDB는 가볍고 빠르며 규모조절성이 뛰어난 NoSQL

⁺ 명지대학교 컴퓨터공학과

(Myoungji University Computer engineering)

(h_jyo@naver.com)

⁺⁺ 명지대학교 컴퓨터공학과

(Myoungji University Computer engineering)

(dmsdud8266@naver.com)

⁺⁺⁺ 명지대학교 컴퓨터공학과

(Myoungji University Computer engineering)

(sell19402@gmail.com)

⁺⁺⁺⁺ 명지대학교 컴퓨터공학과

(Myoungji University Computer engineering)

(hyunsunWkd@gamil.com)

데이터베이스로 문서 지향적이기 때문에 데이터를 레코드에서 오브젝트로 변환할 필요가 없다.

2.1 시스템 구성 요소

앱, 서버, 데이터베이스 서버 시스템으로 구성된다. 사용자는 폰에서 원하는 재료를 사진 인식 또는 목록에서 선택을 통해 입력하면 추천 알고리즘을 통해 레시피를 추천 받는다. 이러한 재료 입력 및 레시피 검색을 위해서는 구글 클라우드 비전 api와 필터 입력을 위한 사용자 인터페이스 프로그램이 포함되어야 한다. 서버 시스템은 클라이언트 사용자와 데이터베이스 서버와의 데이터를 공유하기 위하여 REST 웹 서비스 방법을 사용한다. 즉, 클라이언트에서 URL로 서비스를 요청하면 서버에서 데이터베이스에 있는 데이터를 JSON형태로 데이터를 반환한다.

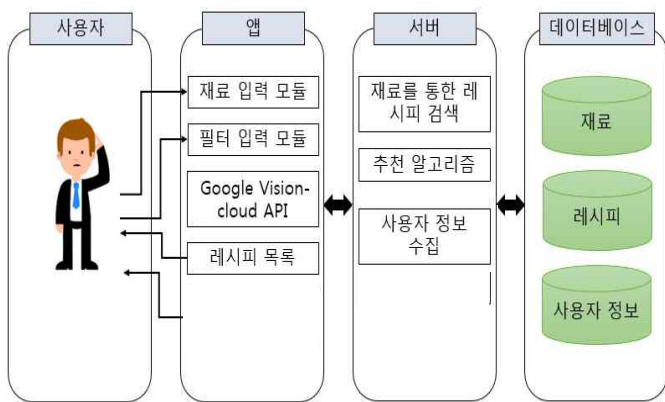


그림1. 전체 시스템 구성

이러한 레시피 추천을 위한 프로그램은 재료 관리, 레시피 관리, 추천 알고리즘을 필요로 한다. 데이터베이스 서버는 재료 및 레시피 정보, 사용자 정보를 보관한다.

2.2 데이터베이스 설계

‘레시피고’ 시스템을 구축하기 위해 설계한 스키마는 아래와 같다.

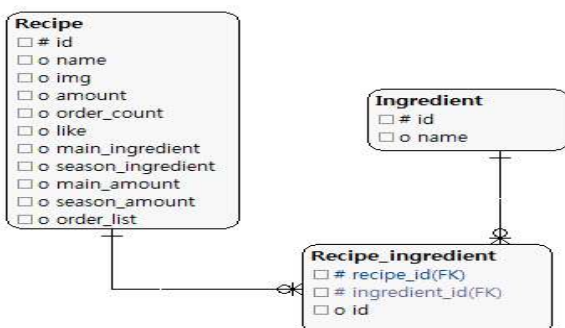


그림2. 데이터베이스 설계

이 컬렉션들은 크게 레시피 정보와 재료를 저장하는 부분으로 나뉜다. Recipes와 Ingredients 컬렉션은 Recipe_ingredient 컬렉션을 통해 연결된다.

3. 적용된 기술

3.1 추천 알고리즘[2]

사용자들에게 레시피를 추천해줄 때, 어떤 레시피를 먼저 보여줄 것인지에 대한 문제가 있다. ‘레시피고’ 프로젝트에서는 레시피 추천을 위해 아래 3가지 알고리즘을 순차적으로 적용한다.

3.1.1. 선택한 재료 포함 여부

사용자 선택한 재료를 모두 포함하고 있는 레시피들을 우선 적으로 보여준다. 그 후 일부의 재료만을 포함하고 있는 레시피들을 보여준다.

3.1.2 복잡도 필터

레시피 재료의 개수와 레시피 과정 개수에 가중치를 부여하여 복잡도를 계산한다. 재료 개수보다는 레시피 과정 개수가 복잡도에 훨씬 더 큰 영향을 미치기 때문에 AHP(Analytic Hierarchy Process)를 활용하여 다음과 같은 표를 작성한다.

	재료	과정
재료 갯수	1	1/5
레시피과정 갯수	5	1

복잡도 계산식은 아래와 같다.

$$\begin{pmatrix} 1/5 & 11/5 \\ 5 & 1 \end{pmatrix} \begin{pmatrix} 2 & 2/5 \\ 10 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 12/5 \\ 60/5 \end{pmatrix} \rightarrow \begin{pmatrix} 12/72 \\ 60/72 \end{pmatrix} = \begin{pmatrix} 0.166 \\ 0.183 \end{pmatrix}$$

복잡도 필터에는 ‘간단한 순으로’와 ‘복잡한 순으로’ 두 가지가 있으며 복잡도 수치에 따라 선택 정렬을 사용하여 레시피들을 정렬한다.

3.1.2. 좋아요 필터

사용자는 확인한 레시피가 마음에 들 경우 ‘좋아요’ 버튼을 누를 수 있다. 이 좋아요 필터를 통해 사용자는 본인이 ‘좋아요’를 누른 레시피만 확인할 수도 있고 ‘좋아요’가 많은 레시피들을 우선적으로 추천 받을 수도 있다.

3.2 웹 크롤러[3]

웹 크롤러란 웹 페이지를 방문해서 자료를 수집하는 프로그램을 말한다. 사람이 엄청난 분량의 웹 문서를 하나씩 보고 정보를 모으는 일은 불가능에 가깝다. 때문에 웹 크롤러가 이를 자동으로 수행하도록 한다. 이때 크롤러는 한 페이지만 방문하는 것이 아니라 작성한 코드가 요구하는 모든 페이지를 차례대로 방문한다.

‘레시피고’ 프로젝트를 위해 레시피를 가져올 사이트의

소스코드를 분석하여 웹 크롤러를 개발한다. 웹 크롤러 프레임워크인 Scrapy 프레임워크를 사용하여 Python언어로 개발한 ‘레시피고’ 크롤러는 지정한 범위의 웹 사이트들을 돌아다니며 수많은 레시피 정보를 긁어와 json 파일 형태로 저장한다. 서버는 이 파일을 정리하여 데이터베이스에 저장하고 요청에 알맞은 정보를 탐색하여 클라이언트로 전달해준다.

3.3 Google cloud-vision API

Google cloud-vision API는 Google에서 제공하는 오픈API로 사진을 인식하여 사진에 담긴 정보와 유사한 단어들을 출력해준다. ‘레시피고’ 프로젝트에서 사진으로 식재료 인식 및 식재료 선택 기능을 구현하기 위해 Google cloud-vision API를 사용하였다.

Front-end에서 Google cloud-vision API를 연동하여 사진을 통해 인식된 결과값들을 받아 서버로 전송하고 서버에서 이 결과값들을 처리해 다시 Front-end로 보내준다.

4. 구현과정

Scrapy 프레임워크를 사용하여 작성한 웹 크롤러를 이용해 원하는 사이트에서 레시피 정보를 긁어와 JSON형태로 저장한다.

저장된 JSON 데이터를 파싱하고 원하는 형태로 가공하여 데이터베이스에 저장해주는 코드를 작성한다. Html5와 CSS, JavaScript언어를 사용하여 Front-end코드를 작성한다.

클라이언트에서 서버로 요청을 보내면 처리하여 응답해주는 Route를 작성한다. Route는 각 기능별로 모듈화한다. Route의 종류에는 ingredients, recipes, vision, likes 총 4가지가 있다. 추천 알고리즘은 모두 이 Route 안에 구현한다.

서버의 응답은 AngularJS를 이용해 구현한 서비스를 거쳐 알맞는 컨트롤러로 보내진다. 컨트롤러는 데이터를 처리하여 뷰로 보내준다.

Front-end에서 Google cloud-vision API를 연동한다. 스마트폰에서 사진 촬영 또는 사진 선택을 하면 이미지 파일을 base64로 인코딩하고 인식 타입을 label detection 로 설정하여 Google cloud-vision 서버로 보내 결과를 받는다. 그 결과를 다시 서버로 보내 원하는 형태로 가공한다.

4.1 구현결과

재료 리스트에서 재료를 선택해 추가하거나 사진으로 식재료를 인식해 재료를 선택할 수 있다. 선택된 재료들을 통해 레시피를 추천한다. 레시피는 앞서 설명한 추천 알고리즘을 통해 정렬되어 추천한다. 사용자는 마음에 드는 레시피를 발견하면 ‘좋아요’를 할 수 있다. 이 ‘좋아요’ 정보를 활용하여 사용자 개인 맞춤형 레시피 추천이 가능하다.

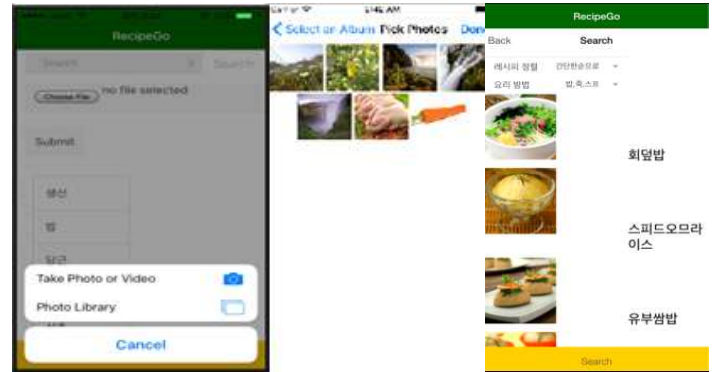


그림3. 실행 결과

5.결론

사용자가 원하는 재료를 입력하면 그 재료를 사용하는 레시피를 추천해주는 하이브리드 애플리케이션을 Ionic 프레임워크를 활용하여 MEAN 스택으로 개발한다. 재료는 사진 인식을 통해서도 선택 가능하다. 추천 알고리즘은 선택한 재료 포함 여부, 복잡도, 좋아요 필터 3가지 요소를 고려하여 작성한다. 사진 인식을 위해 Google cloud-vision API를 사용한다.

애플리케이션은 1)이 모든 기능을 위한 사용자 인터페이스 2)재료, 레시피, 사용자 정보를 보관하기 위한 데이터베이스 3)사용자 인터페이스와 데이터베이스간의 데이터 공유를 위한 REST 웹 서비스로 구성한다.

참고문헌

- [1] 브래드 데일리, Node.js, MongoDB, and AngularJS Web Development, 732, 2014
- [2] Huiying Gao, Susu Wang, Bofei Yang, Hangzhou Yang, User Preference-oriented Collaborative Recommendation Algorithm in e-commerce Journal of Software Vol. 9, No. 7, 1886p ~ 1893p 1796-217X Open Access, 2014
- [3] 서동민 정한민, 빅데이터 분석 서비스 지원을 위한 지능형 웹 크롤러, 한국콘텐츠학회논문지 제13권, 제12호 (2013년 12월) pp.575-584 1598-4877 KCI, 2013