

머신러닝 이미지 프로세싱과 협업 필터링을 이용한 와인 인식 및 추천 시스템

채희상⁰ 심인영 추연철 한익주

한국산업기술대학교 컴퓨터공학과
{ds3569⁰, siy1177, Choo-YC, ijhan}@kpu.ac.kr

Wine recognition & recommend system by machine learning image processing and collaborative filtering

HuiSang Chea⁰ InYoung Sim YeunChul Choo IkJoo Han
Dept. of Computer Engineering, Korea Polytechnic University

요 약

와인을 처음 접하는 사람들에게 있어 영어, 이탈리아어, 프랑스어, 스페인어 등의 외국어로 표기되어 있는 와인의 라벨, 다양한 와인의 종류로 인해 와인에 접근하고 그 정보를 얻는 데 어려움이 있어, 이러한 어려움을 해결하기 위해 사용자가 와인의 정보에 손쉽게 접근할 수 있는 애플리케이션을 개발하였다. 기본적으로 이미지를 통한 와인 인식, 와인 정보제공, 와인 맞춤 추천 제공 등의 기능을 지원하고, 회원가입을 통해 사용자 맞춤형 기능을 이용할 수 있다. MySQL을 사용하여 사용자 정보 및 리뷰 데이터, 와인 데이터 등을 관리하고, TCP/IP 프로토콜과 IO Stream을 통해 모바일 네트워크상에서 정상적인 통신이 가능하도록 하였다.

1. 서론

최근 국내 주류 시장에서 와인의 대중화가 활발하게 진행되고 있다. 우리 주변의 많은 변화가에서 와인바를 이제는 쉽게 찾아볼 수 있고, 대형마트, 편의점 등지에서 손쉽게 와인을 접할 수 있다. 한경닷컴 기사에 따르면, 2021년 1월 기준, 와인이 이마트 주류매출의 27.7%, 롯데마트 주류매출의 19.8%를 차지하고, 이마트24 편의점의 경우 하루 평균 와인 판매량이 4,300병을 달성하는 등 이제 와인은 국내 주류 시장에서 손쉽게 접해 볼 수 있는 주류가 되었다.[1] 그러나 와인을 처음 접하는 사람들에게 있어, 영어, 이탈리아어, 프랑스어, 스페인어 등의 외국어로 표기되어있는 와인의 라벨, 셀 수도 없을 정도로 다양한 와인의 종류는 와인에 대한 진입장벽이 된다. 이처럼 처음 와인을 접하는 사람의 와인에 대한 진입장벽을 허물기 위해 우리는 와인을 처음 접하는 사용자를 대상으로 와인의 정보를 손쉽게 얻을 수 있는 안드로이드 애플리케이션, WineWhat을 개발하였다. 애플리케이션을 이용하여 와인을 인식하고 정보를 제공하여 와인의 라벨에 대한 언어적인 장벽을 완화할 수 있도록 머신러닝 이미지 프로세싱을 개발하였고, 사용자에게 와인의 리뷰를 요청해 해당 리뷰를 기반으로 사용자에게 맞춤 추천 와인을 제공하도록 하였다. 또한, 사용자는 리뷰 기록을 바탕으로 작성된 그래프를 통해 자신의 와인 취향을 시각적으로 확인할 수 있도록 하였다. 이러한 기능들을 통해 언제 어디서나 사용자가 손쉽게 와인의 정보를 얻고 자신에게 맞는 와인을 접할 수 있도록 하여 와인에 대한 진입장벽을 완화했다. 이를 통해 국내 주류 시장에서의 와인의 대중화가 더욱 확대되는 효과를 기대할 수 있다. 본 논문에서는 WineWhat에서 사용된 서버와 클라이언트 및 세부 모듈의 작동 방식을 살펴본다.

2. 관련 연구

2.1 기존 유사 앱과의 비교

	Delectable Wine - scan & Rate	CellWine	Vivino : By the right Wine	Our Project: Winewhat
와인 라벨 인식	O	X	O	O
와인 리스트 인식	X	X	O	X
와인 정보 제공	O	O	O	O
개인 와인 기록	O	O	O	O
수집 와인 관리	X	O	X	X
와인 추천	△	△	O	O
와인 판매 및 배달	O	X	O	X

<표1> 유사 애플리케이션과의 비교

위의 표 1에서는 유사 애플리케이션과 어떠한 차이가 나는지를 보여주고 있다. Delectable Wine의 경우에는 나름의 와인 추천 기능을 제공하고 있지만, 이는 사용자 맞춤 추천이 아니라 단순히 평가가 좋은 와인을 추천한다는 점에서 WineWhat과의 차이가 있다. Cellwine의 경우 와인 인식을 지원하지 않고 와인 셀러에 등록해, 사용자가 수집한 와인의 수량 및 가격을 관리하도록 하는 애플리케이션으로 처음 와인을 접하는 사용자를 대상으로 하는 애플리케이션이 아니다. Vivino 또한 맞춤 추천 와인 추천을 제공하고 있지만, 리뷰 등의 데이터를 기반으로 제공되는 맞춤 추천이 아니고, 결정적으로 세 개의 애플리케이션 모두 영어로 서비스가 제공되고 있는 애플리케이션이므로 언어적 장벽을 해소하고자 하는 본 연구의 개발 배경에 모순된다. WineWhat은 와인을 처음 접하는 사용자에게 필요한 와인의 인식과 정보제공을 주된 목적으로 하는 애플리케이션이다.

기존에 있는 여러 유사 애플리케이션과 WineWhat이 다른 점은 WineWhat은 사용자의 리뷰 기록을 기반으로 알고리즘을 통해 사용자에게 맞춤 추천 와인을 제공한다는 점이다. 또한, 사용자 리뷰 기록을 바탕으로 작성된 그래프를 통해 자신이 고평가한 와인들의 속성을 시각적으로 한눈에 파악할 수 있다는 점도 WineWhat만의 장점이다.

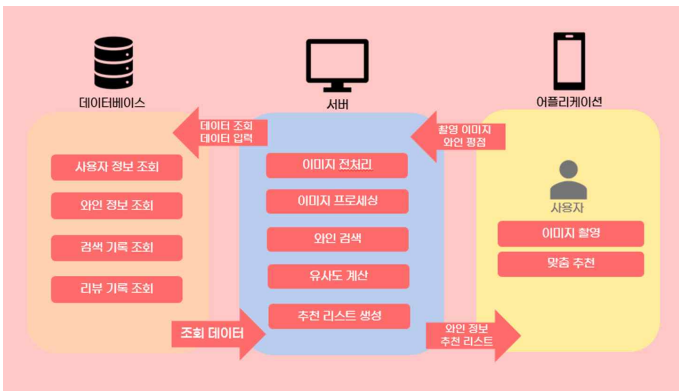
3. 세부 설계 및 구현

3.1 개발환경

본 논문의 애플리케이션은 윈도우 10에서 개발되었으며, Python 3.7, JDK 16.0.1과 Android SDK를 사용하였다. 테스트에 사용한 단말기는 삼성전자의 A50(Android 11)와 Google의 Pixel 3a(Android 11)이다. 개발을 위한 툴로 Pycharm, AndroidStudio, Eclipse를 사용하였고, 머신러닝을 위해 Tensorflow와 Keras API를 사용하였다. DB는 MySQL을 사용하였다.

3.2 시스템 구성

WineWhat의 기본적인 동작 방식은 TCP/IP 프로토콜을 통해 서로 연결된 클라이언트와 서버 사이에서 모듈의 동작을 요청하고, 모듈의 동작 결과를 돌려주는 방식이다. 서버에는 머신러닝 이미지 프로세싱 모듈, 와인 맞춤 추천 모듈, 리뷰 기반 그래프 모듈 등이 포함되어 있으며, 이를 통해 유저의 정보나 리뷰 기록이 담겨있는 사용자 DB와 와인에 대한 정보가 담겨있는 와인 DB에 접근할 수 있다.[2] 기본적으로 클라이언트로부터 서버로 송수신되는 모듈의 동작 요청은 간단한 String 형식이지만, 머신러닝 이미지 프로세싱을 위해 클라이언트의 안드로이드 스마트폰에서 촬영한 이미지를 서버로 송신할 경우, 이미지를 송수신하기 위한 이미지 서버를 열어 Java IOStream을 통해 서버로 이미지를 전송한다. 모듈의 동작 결과에 대해 서버가 수신하는 데이터 또한 String 형식으로 되어있다. 아래 그림1은 WineWhat의 전체적인 동작 흐름을 파악할 수 있는 시스템 구성도이다.



(그림 1) 시스템 구성도

3.3 세부 설계

3.3.1 TCP/IP 프로토콜 기반의 서버와 클라이언트

서버와 클라이언트 간의 통신을 위해 TCP/IP 프로토콜을 사용해 서버와 클라이언트를 구현하였다. 기본적으로 String 형식을 통해 통신하며, 서버는 Python, 클라이언트는 Android Java로 구현하였다.[3]

클라이언트의 모듈 동작 요청을 서버가 수신하고, 해당 모듈을 실행한 후 모듈의 동작 결과를 클라이언트에게 송신한다. 클라이언트는 서버로부터 수신한 데이터를 기반으로 화면의 출력 결과를 바꾸게 된다. 아래 그림2와 그림3은 서버와 클라이언트의 통신을 위한 소스코드 일부분이다.

```

def read(conn, msg):
    data = conn.recv(1000) # Should be ready
    if data:
        print('echoing', repr(data), 'to', conn)
        if "imageProcess" in data.decode("utf-8"): #이미지 프로세싱
            print("imageProcess...")
            sub.call(["java", "-jar", "\\imageServer.jar", "3100"]) #이미지 수신 스트림 실행
            wineID = imageProcess.imageProcessModule("Image Saved File Path\\wineImage.png") #작업률 받은 이미지 이미지 프로세싱 실행
            # 해당 와인 정보
            wine_data_list = wine_db.getWineInfo(wineID)
            wine_data = ''.join(str(e) for e in wine_data_list[0])
            data = bytes(wine_data + '\n', encoding='utf-8')
            conn.send(data) #와인 정보 전송
        elif "Recommend" in data.decode("utf-8"): #추천 와인 리스트
            userID = data.decode("utf-8").replace("Recommend", "")
            userID = userID.replace("\n", "") #사용자 ID 분리
            print("Recommend try...")
            print(userID)
            recommendList = recommend.recommendModule(userID) #추천 와인 리스트 생성
            #추천 와인 리스트
            recommendResult = ''.join(str(e) for e in recommendList)
            data = bytes(recommendResult + '\n', encoding='utf-8')
            conn.send(data) # Hope it won't block #추천 와인 리스트 전송

```

(그림 2) 클라이언트의 요청에 따라 모듈을 호출하는 서버의 소스코드

```

//동작 제어 변수에 따라 서버에 요청
@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    if (intent == null) {
        return Service.START_STICKY;
    }
    else {
        if (client == null) {
            ChatOperator chatOperator = new ChatOperator();
            chatOperator.execute();
        }
        else {
            if (orderCode == 1) { //로그인
                String text = "login" + intent.getStringExtra("name", "user_info");
                final Sender messageSender = new Sender(); // Initialize chat sender AsyncTask.
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
                    messageSender.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR, text);
                } else {
                    messageSender.execute(text);
                }
            }
            else if (orderCode == 2) { //회원가입
                String text = "Register" + intent.getStringExtra("name", "user_info");
                final Sender messageSender = new Sender(); // Initialize chat sender AsyncTask.
                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
                    messageSender.executeOnExecutor(AsyncTask.THREAD_POOL_EXECUTOR, text);
                } else {
                    messageSender.execute(text);
                }
            }
        }
    }
}

```

(그림 3) 동작 제어 변수에 따라 서버에 다른 데이터를 수신하는 클라이언트의 소스코드

3.3.2 이미지 송수신을 위한 이미지 서버와 클라이언트

클라이언트의 안드로이드 스마트폰에서 촬영한 이미지를 서버로 송신하기 위해 기존 서버에서 추가로 소켓을 열어 이미지 서버를 생성한다. 클라이언트가 이미지 서버에 접속하고, Java IOStream을 통해 이미지를 전송한 후 이미지 서버는 종료된다. 이미지 서버는 Java로 구현하였다. 아래 그림 4는 이미지 서버의 소스코드 일부분이다.

```

public class imageServer {
    public static void main(String[] args) throws IOException {
        if (args.length > 0) {
            for (int i = 0; i < args.length; i++) {
                String dirPath = "wineImage.png";
                int port = Integer.parseInt(args[i]);
                ServerSocket imageServer = new ServerSocket(port);
                System.out.println("image server on");
                Socket csock = imageServer.accept();

                try {
                    System.out.println("image server ready");
                    ObjectInputStream ois = new ObjectInputStream(csock.getInputStream());
                    byte[] ba = (byte[]) ois.readObject();
                    ois.close();
                    FileOutputStream fos = new FileOutputStream(dirPath);
                    fos.write(ba);
                    fos.flush();
                    System.out.println("image server got image");
                    fos.close();
                } catch (ClassNotFoundException | IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
            }
        }
    }
}

```

(그림 4) 이미지 서버의 소스코드

3.3.3 CNN 이미지 프로세싱

합성곱 신경망(CNN, Convolution Neural Network) 모델을 사용하여 클라이언트로부터 전송받은 와인의 종류를 판별하는 머신러닝 이미지 프로세싱 모듈을 구현하였고, Tensorflow와 Keras API를 사용하였다.[3] 인공지능 모델의 학습을 위한 와인 이미지는 크롤링을 통해 확보하였다. 이렇게 수집한 이미지를 RGB 모드로 변경하고, 50X50 크기로 변경한 후, 더 많은 이미지를 위해 Pillow 라이브러리를 통해 이미지를 회전하고 좌우로 흔들어 이미지의 다양성을 추가로 확보하였다. 이렇게 전처리한 이미지는 랜덤으로 학습 데이터와 테스트 데이터를 구분해 Numpy 라이브러리를 통해 벡터 형식으로 정규화하여 사용한다. CNN 모델의 입력은 50X50 픽셀의 RGB 24비트 컬러이고, 최종 데모 버전 기준 총 106개의 와인을 판별하므로 임의로 부여한 와인 ID 1부터 106까지의 출력을 가진다.

3.3.4 사용자 기반 협업 필터링 알고리즘을 이용한 와인 맞춤 추천 알고리즘

와인 맞춤 추천을 위해 사용자 기반 협업 필터링 알고리즘(User Based Collaborative Filtering Algorithm)을 사용해 와인 맞춤 추천 알고리즘을 구현하였다. 추천 시스템에서 협업 필터링 알고리즘은 많은 유저들로부터 취향 정보들을 모아 스스로 예측하는 기술을 말한다. 협업 필터링의 종류에는 Memory-Based, Model-Based, Hybrid가 있으며, 사용자 기반 협업 필터링 알고리즘은 사용자와 사용자 간의 유사도를 계산하여 동작하는 Memory-Based 협업 필터링 중 하나이다. Memory-Based 협업 필터링의 추천 시스템은 유사도를 기반으로 동작한다. 이때 주로 사용되는 것은 코사인 유사도와 피어슨 유사도이다.[5] WineWhat에서는 코사인 유사도를 사용하였다. 아래는 코사인 유사도를 계산하는 공식이다.

similarity = cos(θ) = (A · B) / (||A|| ||B||) = (Σ(A_i × B_i)) / (√(Σ(A_i)^2) × √(Σ(B_i)^2))

와인 맞춤 추천 알고리즘은 사용자의 리뷰를 기반으로 동작한다. 리뷰의 형식은 5점 만점의 별점 형식이다. 사용자의 리뷰는 리뷰 DB에 “와인 ID : 별점” 형식으로 JSON으로 저장되어 있다. 아래 그림 5은 테스트 중 사용한 테스트 데이터의 JSON 형식이다.

```
insert into reviewtest(id, review) values('userA', json_object(
1, 5,
2, 4,
3, 4,
4, 3,
5, 0));
```

(그림 5) 테스트 데이터 JSON

리뷰 DB에서 받아온 사용자의 리뷰 기록과 다른 사용자들의 리뷰 기록을 벡터로 나타내어 코사인 유사도를 연산하고, 일정 수준 이상의 높은 유사도를 보인 다른 사용자들을 모아 유사한 사용자 리스트를 작성한다. 만약 일정 수준 이상의 유사도를 보인 사용자가 없거나 사용자의 리뷰 기록이 없다면, 전체 사용자를 유사한 사용자로 판단한다. 이후 유사한 사용자들이 리뷰한 와인의 별점의 평균을 기준으로 추천 와인 리스트를 작성하여 사용자에게 제공한다. 이때 사용자가 이미 리뷰했던 와인은 추천 와인 리스트에서 제거된다.

3.3.5 리뷰 데이터 기반 취향 분석 그래프

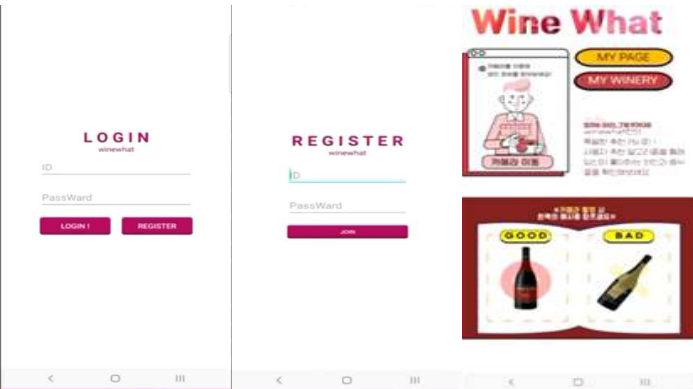
사용자의 리뷰기록과 와인 맞춤 추천 알고리즘에서 얻은 유사한 사용자 리스트를 사용하여 사용자의 취향과 유사한 사용자들의 취향을 시각적으로 비교하여 파악할 수 있는 그래프를 애플리케이션 화면에 출력하였다. 그래프를 출력하기 위해 MPAndroidChart API를 사용하였다. 와인은 당도, 산도, 바디 감, 타닌 감의 총 4가지의 기준으로 맛을 평가하였다. 이 평가 기준은 와인마다 와인 DB에 5점 만점으로 저장되어 있다. 사용자의 리뷰 기록을 기반으로 높은 평점을 받은 와인은 평가 기준에 높은 가중치를 곱하여 계산하고, 낮은 평점을 받은 와인은 평가 기준에 낮은 가중치를 곱하여 사용자가 어떠한 맛이 강조되는 와인을 선호하는지 통계적으로 파악할 수 있다. 유사한 사용자들도 같은 방식으로 계산한다. 이 데이터를 클라이언트에게 송신하면 MPAndroidChart API를 통해 사용자의 스마트폰 화면에 그래프를 표시한다.[6] 아래 그림 6은 애플리케이션 화면에 표시된 그래프이다.



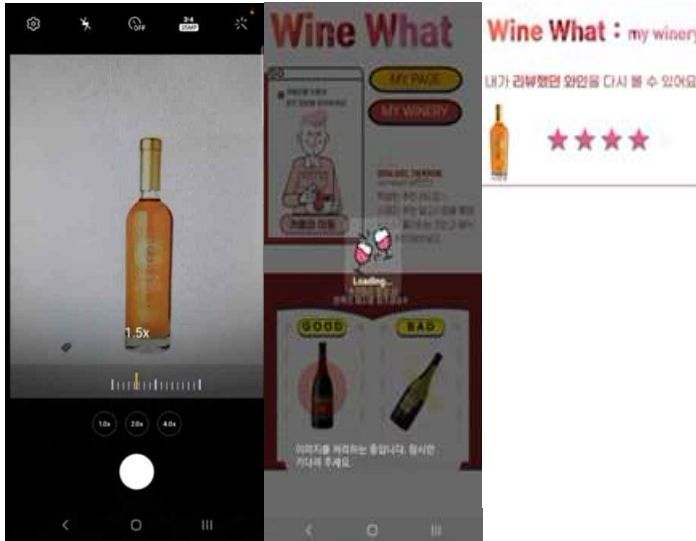
(그림 6) 취향 분석 그래프

3.4 구현결과

본 절은 WineWhat의 구현된 모습을 나타낸다. 시각적으로 구현된 애플리케이션 부분만을 포함하며,[7] 테스트를 위해 임의로 새로 회원 가입하고 화면상의 와인 이미지를 촬영하여 이미지 프로세싱을 진행한 화면임을 알린다. 그림 8의 첫 번째 화면상의 와인 이미지를 촬영하는 애플리케이션 화면이다.



(그림 7) 로그인 화면, 회원가입 화면, 메인 페이지 화면



(그림 8) 카메라 화면, 로딩 화면, 마이 와인너리 화면



(그림 9) 와인 정보 화면, 와인 리뷰 화면, 마이 페이지 화면

4. 결론 및 향후 연구과제

우리는 와인을 처음 접하는 사람들을 주요한 대상으로 와인 라벨의 언어적인 장벽이나 종류의 다양성, 와인의 정보를 쉽게 얻을 수 있는 매체의 부족으로 인한 와인의 진입장벽 문제를 해결하고자 하였다. 와인의 이미지를 통해 와인의 종류를 식별하는 머신러닝 이미지 프로세싱 모듈, 리뷰 데이터를 분석하여 추천 와인을 제공하는 와인 맞춤 추천 알고리즘, 사용자의 취향을 시각적으로 파악할 수 있는 취향 분석 그래프 모듈 등을 통해 와인을 처음 접하는 사람들이 쉽게 와인의 정보를 접하고 자신의 취향에 맞는 와인에 대한 정보를 추가로 얻을 수 있는 머신러닝 이미지 프로세싱과 사용자 기반 협업 필터링을 이용한 와인 인식 및 추천 시스템을 개발 및 구현함으로써 해당 문제를 해결할 수 있었다.

본 연구는 기존의 유사한 서비스가 대부분 해외에서 개발되어 영문으로 제공되는 서비스인 점, 사용자의 리뷰 기록과 사용자 기반 협업 필터링 알고리즘을 사용하여 사용자에게 맞춤 추천 와인을 제공하고 사용자의 취향을 시각적으로 표현하고 있다는 점에서 차별점을 가지고 있다. 빠르게 확장되고 있는 국내 와인 시장에 있어 새로운 소비자의 유입을 가능케 하고, 이들이 계속해서 자신의 입맛에 잘 맞는 와인을 선택할 수 있도록 도와주는 부분에 있어 의의가 있다.

향후 더 발전된 서비스를 위해서는 연구 중 가장 큰 문제였던 머신러닝 이미지 프로세싱의 와인의 인식률 문제와 와인의 다양성에 대한

문제를 해결하기 위한 연구가 진행되어야 한다. 또한, 기존의 유사한 서비스들이 제공하고 있는 와인 판매 기능을 추가한다면 하나의 와인 판매 플랫폼으로써 부가적인 수익 가능성도 기대할 수 있을 것이다.

참고문헌

- [1] 이미경, 와인, '국민 주류'로 자리할까...맥주 제치고 매출 1위, 한경 경제, 2020, <https://www.hankyung.com/economy/article/2020122192827>
- [2] 홍의경, 「오라클을 기반으로 하는 데이터베이스 배움터 - 개정3판」, 생능, 2014
- [3] 윤성우, 「열혈 TCP/IP 소켓 프로그래밍」, 오렌지 미디어, 2020
- [4] 쿠지라 하코우즈쿠에, 「파이썬을 이용한 머신러닝, 딥러닝 실전 개발 입문」, 위키북스, 2019
- [5] scvgoe, 협업 필터링 추천 시스템 (Collaborative Filtering Recommendation System) 세상 간단한 추천 시스템, 2017, [https://scvgoe.github.io/2017-02-01-%ED%98%91%EC%97%85-%ED%95%84%ED%84%B0%EB%A7%81-%EC%B6%94%EC%B2%9C-%EC%8B%9C%EC%8A%A4%ED%85%9C-\(Collaborative-Filtering-Recommendation-System\)/](https://scvgoe.github.io/2017-02-01-%ED%98%91%EC%97%85-%ED%95%84%ED%84%B0%EB%A7%81-%EC%B6%94%EC%B2%9C-%EC%8B%9C%EC%8A%A4%ED%85%9C-(Collaborative-Filtering-Recommendation-System)/)
- [6] MPAndroidchart, <https://github.com/PhilJay/MPAndroidChart>
- [7] 우재남, 박남길, 「Android Studio를 활용한 안드로이드 프로그래밍 5판」, 한빛아카데미, 2020