

WittyPhotos

- 더 똑똑한 갤러리

24조

메모리즈

20133259

정찬영

20133256

장지은

20123371

서민호

20123376

손민지

20153203

이가빈

목차

1. 프로젝트 소개

2. 수행 내용 및 중간 결과

3. 변동사항

4. 향후 계획

프로젝트 소개

WittyPhotos란?

이미지 자동 태그 기능을 제공하고 사용자가 직접 수동으로 태그를 추가/삭제 등 관리할 수 있게 함으로써 이미지 데이터 관리를 용이하게 하고 태그 데이터를 분석하여 사용자의 라이프 스타일과 휴먼 네트워크정보를 재치있게 전달하는 안드로이드 애플리케이션입니다.

수행 내용 및 중간결과

- 이미지 태그 기능

● 자동 객체 태그 - 이미지에서 객체 인식

Azure Computer vision API, Faster R-CNN, SDD, YOLO등의 다양한 딥러닝 모델 비교
빠른 속도와 좋은 정확도를 보여주는 YOLO v3 (Python OpenCV)를 선정하여 입출력 코드 수정

● 자동 인물 태그 - 이미지에서 얼굴 인식

Python의 face_recognition package를 이용하여 128개의 얼굴 특징 벡터 추출하여 저장
저장된 모든 얼굴의 벡터 값을 DBSCAN 이용하여 클러스터링한 후 라벨 추출 (Face Clustering)
적절한 값이 추출되었는지 테스트하기 위해 t-sne를 사용하여 시각화

● 수동 태그

Python에서 SQLite3 데이터베이스에 태그를 추가, 삭제하는 코드 작성

수행 내용 및 중간결과

- 태그 카테고리화

● 태그 카테고리화

카테고리 : 인물, 사물, 동물, 식물, 음식, 지역, 장소, 색깔, 기타

자동 태그들은 먼저 카테고리를 지정하여 데이터베이스에 저장하였다.

사용자가 태그를 추가할 때 카테고리를 반드시 설정하도록 코드를 작성하였다.

테이블(T): Category

	categoryID	category_name
	필터	필터
1	1	인물
2	2	사물
3	3	동물
4	4	식물
5	5	음식
6	6	지역
7	7	장소
8	8	색깔
9	9	기타

[그림 00 -Category 테이블]

테이블(T): Object_Auto_tag

	tagID	EnglishTag	KoreanTag	tagCount	tagState	categoryID
	필터	필터	필터	필터	필터	필터
1	n00000001	person	사람	0	1	1
2	n00000002	bicycle	자전거	0	1	2
3	n00000003	car	자동차	0	1	2
4	n00000004	motorbike	오토바이	0	1	2
5	n00000005	aeroplane	비행기	0	1	2
6	n00000006	bus	버스	0	1	2
7	n00000007	train	기차	0	1	2
8	n00000008	truck	트럭	0	1	2
9	n00000009	boat	보트	0	1	2
10	n00000010	traffic light	신호등	0	1	2

[그림 00 - Object_Auto_tag 테이블]

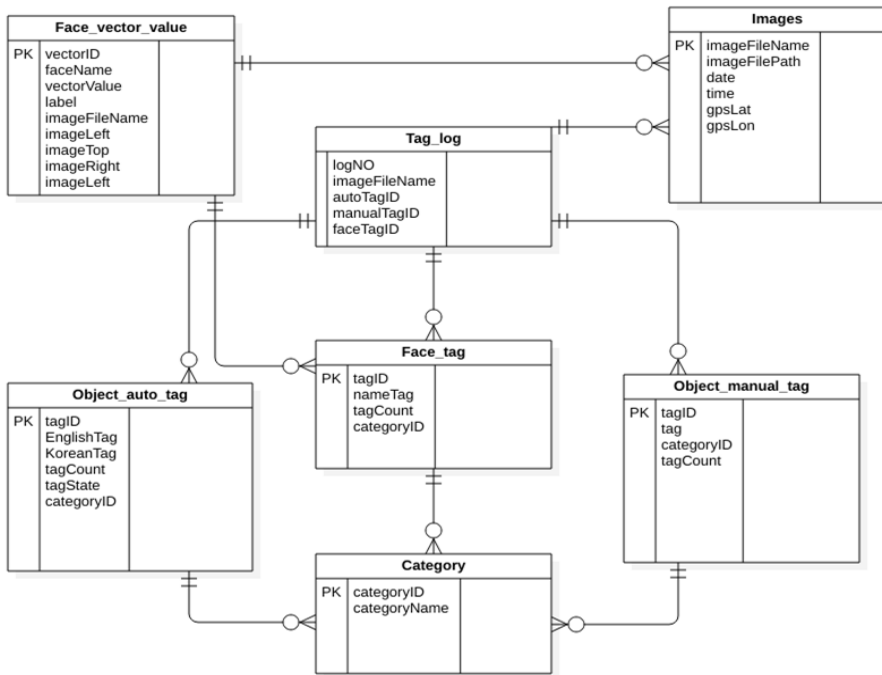
수행 내용 및 중간결과

- 데이터베이스

● 데이터베이스

데이터베이스 설계 완료

Python을 사용하여 SQLite3 데이터베이스
타 기능들을 연동하여 테스트 및 설계 수정



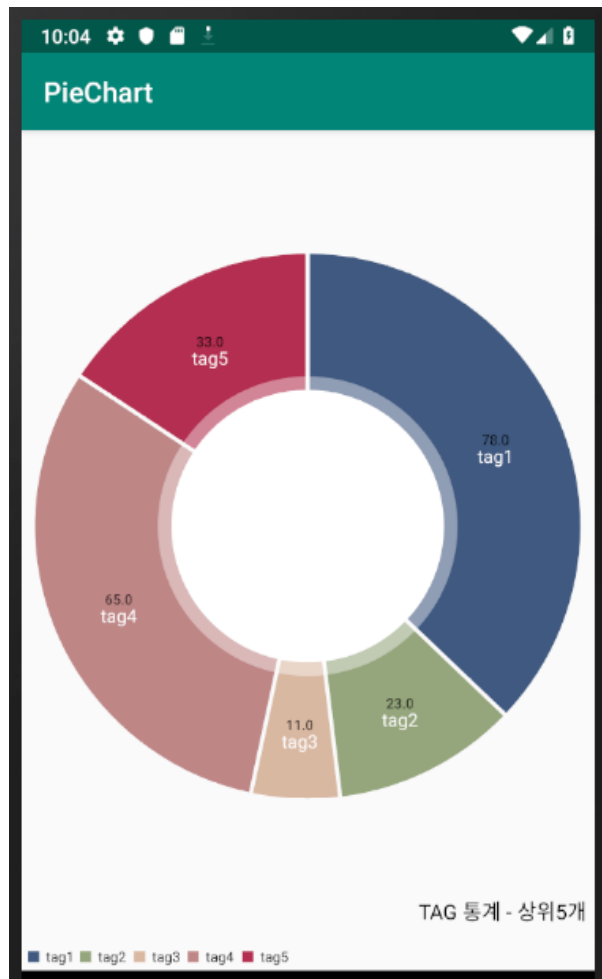
[그림00 - ER 다이어그램]

수행 내용 및 중간결과

- 통계 및 네트워크 분석

● 태그 통계

자주 사용된 태그의 통계를 파이 그래프로
시각화하기 위해 MPAndroidChart 라이브러리를
이용하여 코드작성



수행 내용 및 중간결과

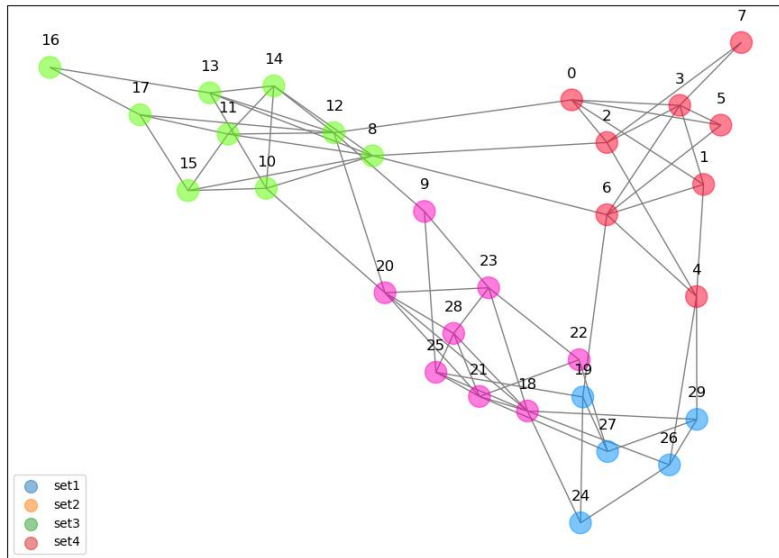
- 통계 및 네트워크 분석

- 전체 네트워크 군집화

네트워크에서 커뮤니티를 추출하는 알고리즘 조사
(Girvan_newman, Link Community, Louvain)
Python의 Girvan_newman 라이브러리를 이용해서
커뮤니티 추출, 그래프화 테스트

- Centrality 계산

노드의 중요도에 따라 노드의 크기를 다르게 시각화하기 위해
Python으로 degree, closeness, betweenness Centrality를
계산, 그래프화 테스트



수행 내용 및 중간결과

- UX/UI 디자인 설계 및 안드로이드 어플리케이션 구현

● 스플래쉬 액티비티 + 메인 액티비티 구현

- 자체 디자인 로고와 선별된 컬러 팔레트를 기반으로 1500ms 의 스플래쉬 화면을 구성
- 스플래쉬 후 앨범, 그래프 카테고리를 선택할 수 있는 메인 액티비티 화면으로 이동

● 앨범 메뉴

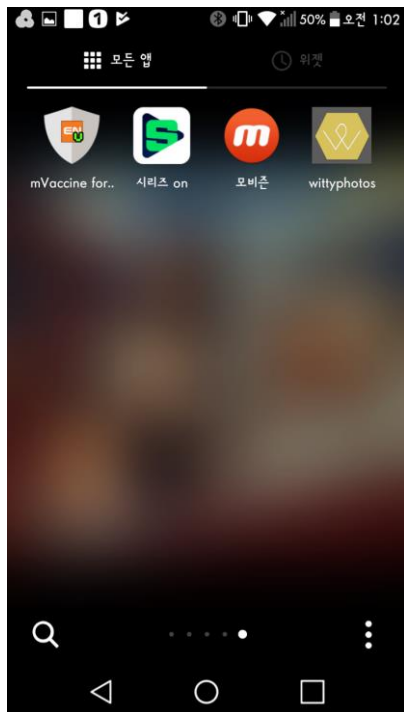
- 앨범 메뉴 선택과 동시에 유저의 기기에 저장되어 있는 앨범을 불러오는 기능 구현
- 이미지의 개수가 많을 경우를 대비해 AsyncTask - doInBackground 를 사용해 레이아웃 뒤에서 계속 이미지 작업을 하는 기능 구현
 - 불러진 앨범 카테고리에서 구체적 카테고리를 선택할 시 카테고리에 있는 이미지를 그리드 형식으로 출력하는 기능 구현

● 그래프 메뉴

- 그래프 메뉴 선택과 동시에 그래프 디렉토리로 이동하는 화면 전환
- 화면 전환 후 태그 빈도수 데이터를 기반으로 생성된 파이 그래프를 출력

수행 내용 및 중간결과

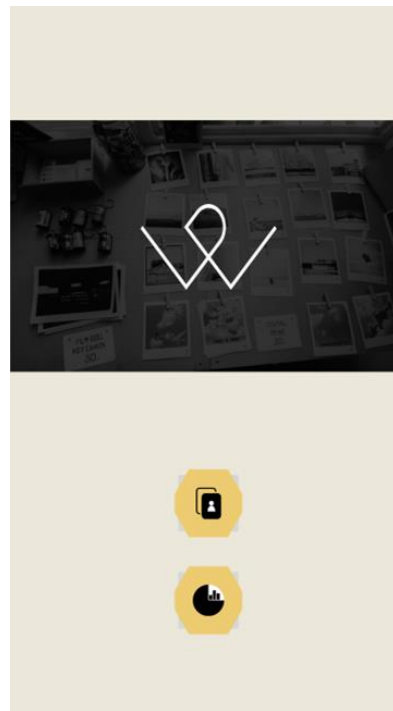
- UX/UI 디자인 설계 및 안드로이드 어플리케이션 구현



앱 아이콘



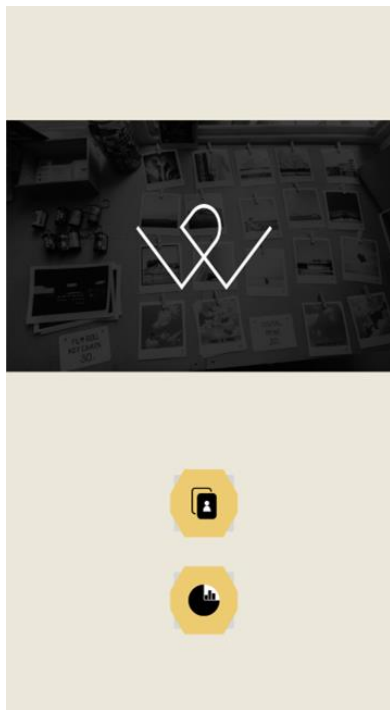
스플래쉬 화면



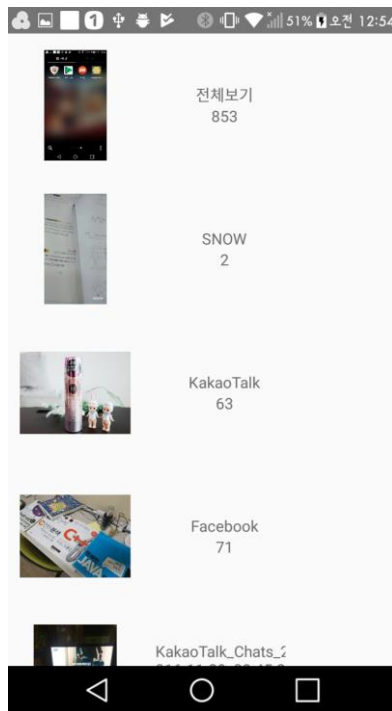
메인 화면

수행 내용 및 중간결과

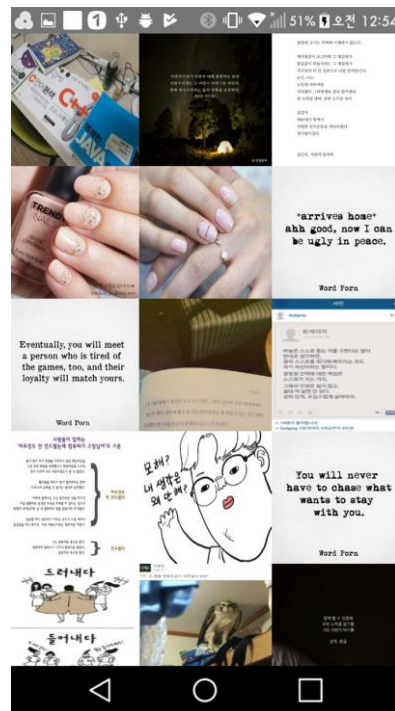
- UX/UI 디자인 설계 및 안드로이드 어플리케이션 구현



메인 화면



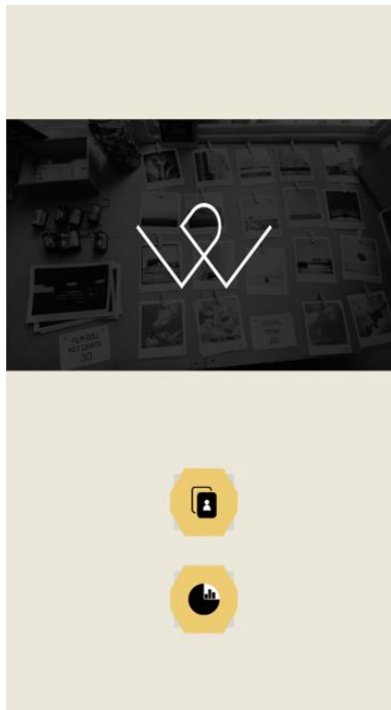
이미지 디렉토리



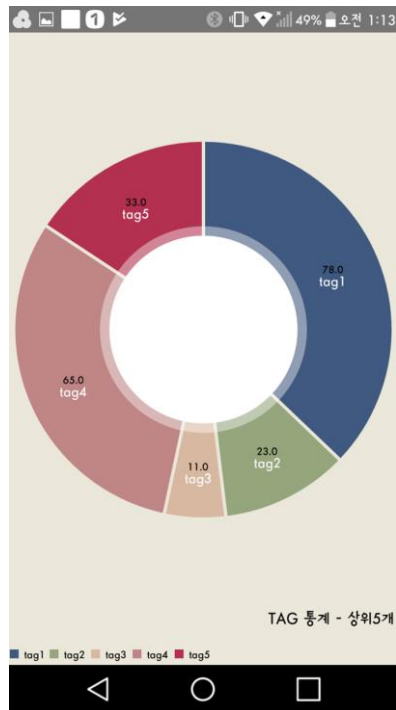
이미지 디테일 디렉토리

수행 내용 및 중간결과

- UX/UI 디자인 설계 및 안드로이드 어플리케이션 구현



메인 화면



그래프 디렉토리_(1)파이그래프

변동사항

- 이미지 태그
 - [보완 사항] Python OpenCV로 YOLO v3 구동
- 태그 카테고리화와 데이터 베이스
 - [추가 사항] 지역 카테고리 추가
 - [보완 사항] 안드로이드 앱과 연동 전 Python으로 SQLite3를 사용하여 데이터베이스 구축 및 테스트
- 통계 및 네트워크 분석
 - [보완 사항] 서버 의존도를 낮추기 위해 자바로 코드 작성
- UI/UX 디자인 설계 및 안드로이드 어플리케이션 구현
 - [보완 사항] Viewpager 사용
 - [추가 사항] 최하단에 태그 삭제 버튼 추가
- 웹 서버와 웹 클라이언트 구축
 - [보완 사항] Django를 사용하여 웹서버 개발
 - [추가 사항] PythonAnywhere 사용하여 테스트용 서버 구축

향후 계획

- 이미지 태그

- 자동 객체 태그

단어 데이터셋을 증가시켜 더 많은 자동 태그(약 100개 이상)를 제공 (현 80개)
Pascal VOC를 사용하여 객체 인식 정확도를 체크

- 수동 태그

Python으로 작성한 태그 추가, 삭제 코드를 안드로이드 어플리케이션에서
구동하기 위해 Java로 수정

- 데이터베이스

- Python. SQLite로 구축된 데이터베이스를 참고하여 안드로이드 어플리케이션 내부 데이터베이스 구축

- Network 테이블(태그의 동시 출현 빈도수 저장)을 만들어 네트워크 분석에 사용될 수 있게 함

향후 계획

- 통계 및 네트워크

- 통계
데이터베이스의 Network 테이블에서 태그 출현 빈도를 입력받는 코드 추가
- 네트워크 그래프
커뮤니티 추출 알고리즘 추가 조사
속도, 모듈성 등을 비교 분석하여 최종적으로 사용할 알고리즘 선택
군집을 찾아내 노드가 속해있는 군집 ID를 데이터베이스로 넘기는 코드
노드의 Centrality 계산 코드 작성
GraphStream 라이브러리를 이용하여 그래프 생성
(weight, centrality에 따라 edge의 색상, 노드의 크기의 변화를 준다)
- 네트워크 분석을 통한 코멘트 제공
사용자에게 추천이나 정보를 제공할 수 있는 분석 코드 작성
정해진 형식으로 코멘트를 만들어 보여주고 push 알림을 준다.

향후 계획

● UI/UX 디자인 설계 및 안드로이드 어플리케이션 구현

안드로이드 어플리케이션 완성

- 이미지를 서버에 전송하여 자동 태그를 수신할 웹 클라이언트 구축
- 네트워크 및 태그 통계 데이터 베이스 그래프 시각화
- 보다 편리한 UX 추가 리서치
- 사용자 인터페이스 디자인 완료
- 유저를 위한 푸시 알림 서비스 구축

● 웹 서버와 클라이언트 구축

웹서버

- 이미지 데이터를 클라이언트에서 전달받아 객체인식과 얼굴인식 프로그램에 전달
- 이미지에서 추출된 태그 데이터와 얼굴 벡터 값을 클라이언트에 전송
- 통신이 제대로 작동하는지 테스트

웹클라이언트

- 새롭게 동기화 된 이미지들을 서버에 전달
- 서버로부터 전송받은 데이터를 안드로이드 내부 데이터베이스에 저장