



국민대학교  
전자정보통신대학  
컴퓨터공학부

# 캡스톤 디자인 I

## 종합설계 프로젝트


프로젝트 명	WittyPhotos - 더 똑똑한 갤러리
팀 명	메모리즈
문서 제목	24조_결과보고서

Version	1.4
Date	2019-05-28

팀원	정찬영 (조장)
	손민지
	이가빈
	장지은
	서민호

### CONFIDENTIALITY/SECURITY WARNING


이 문서에 포함되어 있는 정보는 국민대학교 전자정보통신대학 컴퓨터공학부 및 컴퓨터공학부 개설 교과목 캡스톤 디자인I 수강 학생 중 프로젝트 “WittyPhotos”를 수행하는 팀 “메모리즈(24조)”의 팀원들의 자산입니다. 국민대학교 컴퓨터공학부 및 팀 “메모리즈(24조)”의 팀원들의 서면 허락없이 사용되거나, 재가공 될 수 없습니다.

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

## 문서 정보 / 수정 내역


<b>Filename</b>	결과보고서-WittyPhotos.doc
<b>원안작성자</b>	정찬영, 손민지, 이가빈, 장지은
<b>수정작업자</b>	정찬영, 손민지, 이가빈, 장지은

수정날짜	대표수정 자	Revisio n	추가/수정 항목	내 용
2019-05-22	전원	1.0	최초 작성	
2019-05-24	전원	1.1	내용 수정	수정된 연구 및 개발 내용 추가
2019-05-26	전원	1.2	내용 수정	수정된 연구 및 개발 내용 추가
2019-05-27	전원	1.3	내용 수정	수정된 연구 및 개발 내용 추가
2019-05-28	전원	1.4	내용 수정	수정된 연구 및 개발 내용 추가

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

## 목 차

1	개요	4
1.1	프로젝트 개요	4
1.2	추진 배경 및 필요성	4
2	개발 내용 및 결과물	5
2.1	목표	5
2.2	연구/개발 내용 및 결과물	6
2.2.1	연구/개발 내용	6
2.2.2	시스템 기능 및 구조 설계도	6
2.2.3	활용/개발된 기술	6
2.2.4	현실적 제한 요소 및 그 해결 방안	6
2.2.5	결과물 목록	6
2.3	기대효과 및 활용방안	6
3	자기평가	7
4	참고 문헌	7
5	부록	7
5.1	사용자 매뉴얼	7
5.2	운영자 매뉴얼	7

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

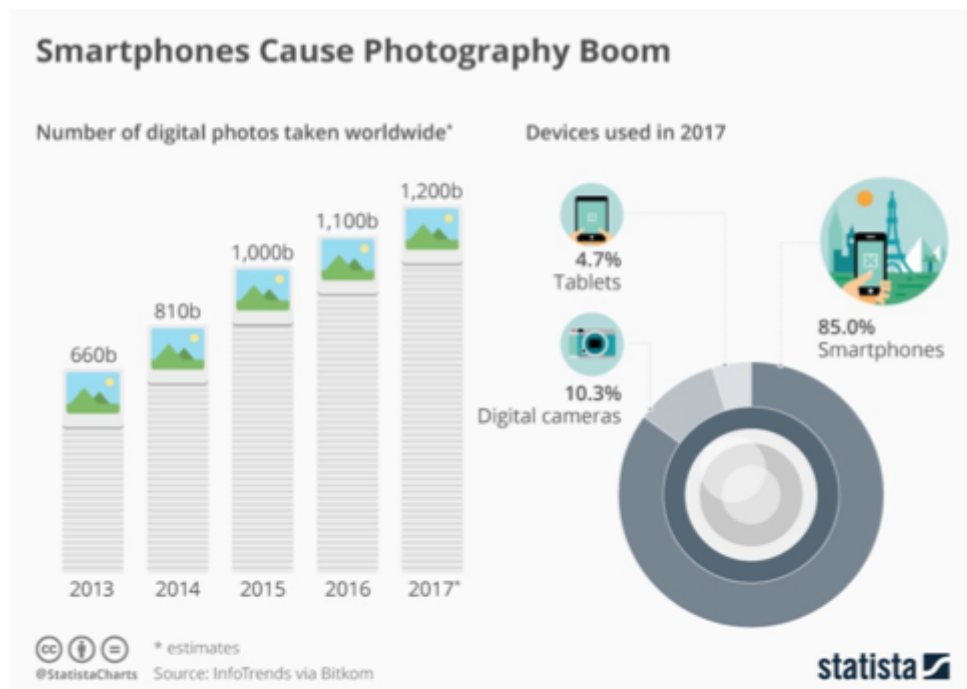
## 1 개요

### 1.1 프로젝트 개요

현대인들은 이전에 비해 훨씬 방대한 양의 이미지를 생산해 내고 있으며, 때때로 개인이 갖는 이미지가 지나치게 많아지기도 합니다. 개인이 소장하고 있는 이미지의 양이 많아짐에 따라 이미지 관리가 필요하지만 개인적으로 이미지 데이터 관리를 하는 것에 어려움을 겪고 있습니다.


본 프로젝트는 이러한 문제를 해결하기 위해 딥러닝을 이용한 **이미지 자동 태그 기능**을 제공하고 사용자가 직접 수동적으로 태그를 추가, 삭제 등 관리할 수 있게 함으로써 이미지 데이터 관리를 용이하게 합니다. 또한 태그 데이터를 분석하여 **사용자의 라이프 스타일과 휴먼 네트워크정보**를 재치있게 전달하는 안드로이드 어플리케이션을 제작합니다.

### 1.2 추진 배경 및 필요성



[그림 1 - 2013~2017년 전세계 이미지 생성 통계]

현대사회에서 개개인은 주변에서 발생하는 모든 일들을 이미지로 기록한다. 단순한 기록 용도를 넘어서 이미지를 찍는 행위가 하나의 문화로 자리를 잡았다. 스마트폰 보급이 완성시되었던 2013년 부터 꾸준히 스마트기기를 통해 생성된 이미지의 데이터가 꾸준히 증가하고 있는 수치를 위의 도표를 통해서 확인 할수 있다.

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	결과보고서		
	프로젝트 명	WittyPhotos - 더 똑똑한 갤러리	
	팀 명	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

이미지 관련 기술과 스마트 기기의 보급화로 저장되는 이미지의 양이 꾸준히 증가하면서 점차 이미지 데이터 관리가 어려워졌다. 사진 데이터 관리가 어려워지면서 사진이 포함하고 있는 다양한 정보들의 활용도 이루어지지 못하고 있다.

이러한 사회의 니즈(needs)에 따라 다수의 사진 관리 어플리케이션이 출시되었고 대표적인 사진 관리 어플리케이션을 비교 분석해 본 결과 다음과 같은 한계점을 발견했다.




[그림2, 3 - Google포토 Application & result]

위의 예시인 '구글 포토'의 경우, 별도의 태그를 지정하지 않아도 이미지에 등장 하는 인물, 장소, 사물로 이미지를 검색할 수 있으나 정확도가 부족하고 모든 이미지를 인식하지 못한다는 단점이 있다. 또한 개인 사용자는 각 이미지에 구체적으로 어떤 태그가 지정되어있는지 알 수 없기 때문에 원하는 이미지를 찾기 위해 검색어를 고민해야 하며, 상세한 태그가 지정되지 않아 위의 그림과 같이 포괄적인 의미의 단어를 통하여 태그를 검색해야 한다는 점에서 한계가 있다.




[그림 4 - HashPhotos Application]

위 의 'HashPhotos'의 이미지 검색 기능은 '구글 포토'와 반대로 이미지에 자동으로 태그를 지정해 줄 수 없다. 따라서 개인 사용자가 이미지마다 개별적으로 태그를 지정해 주지 않는 한 검색기능을 이용할 수 없기 때문에 대량의 이미지를 소유한 사용자의 경우 이미지들을 관리하기 어렵다는 단점이 있다.

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

예시의 두 애플리케이션의 단점을 해결하기 위해 본 프로젝트에서는 자동으로 이미지에서 태그를 추출하고 사용자가 태그를 수정할 수 있는 애플리케이션을 만들고자 한다. 노출된 태그 리스트 중 사용자가 보기에 부정확하거나 사용하고 싶지 않은 태그가 존재하는 경우, 사용자가 직접 원하는 태그를 삭제 또는 추가 할 수 있는 기능은 '구글 포토'가 가지는 부정확성의 문제를 보완하며, 자동 태그 추출 기능은 'HashPhotos'가 가지는 대량 이미지 관리 문제를 완화 시킬 것으로 기대한다.


 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

## 2 개발 내용 및 결과물

### 2.1 목표

본 프로젝트는 대량의 이미지를 쉽게 열람, 검색, 분석하는 이미지 관리 애플리케이션 개발을 목표로 한다. 상기 목적을 달성하기 위해, 제안하는 애플리케이션은 다음의 기능을 갖는다.

- 이미지에서 자동으로 단어를 추출하여 태그 삽입
- 사용자가 직접 이미지에 태그 삽입
- 태그를 통해 원하는 이미지 검색 및 삭제 - 다중 태그 검색 가능
- 등록된 인물에 대한 지속적인 자동 태그 삽입
- 태그를 분석하여 통계 그래프 제공
- 이미지 태그를 기반으로 한 네트워크 분석을 통해 사용자에게 시각적 관계 그래프 제공
- 사용자의 라이프스타일 패턴 분석
- 사용자의 휴먼네트워크(Human Network) 분석

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28


## 2.2 연구/개발 내용 및 결과물

### 2.2.1 연구/개발 내용

#### 1. 안드로이드 어플리케이션 기본 기능 구축 및 UI/UX 디자인

- 안드로이드 스튜디오 베이스로 네이티브 앱을 개발한다. 사용자에게 보다 직관적이고 깔끔한 어플리케이션을 제공한다. 앱 내 기능은 크게 두 가지로 세분화된다.
- 안드로이드 기본 기능
  - 모바일 User Interface와 작동 순서
    1. 구글플레이에서 어플리케이션 다운로드
    2. WittyPhotos - 더 똑똑한 갤러리 어플리케이션 실행
    3. 메인 페이지에서 사용자가 원하는 카테고리 선택 (album or graph)
    4. 사용자 기기 이미지 갤러리와 동기화 (연동 후 이미지마다 자동적으로 태그 생성)
  - 이미지 갤러리 리스트 뷰
    1. [앨범]버튼 클릭 후 동기화된 유저의 이미지 갤러리 디렉토리 리스트 뷰(디렉토리id, thumbnail image, 이미지 counting 확인)
    2. [전체보기]버튼 클릭 후 전체 이미지 그리드 뷰
  - 이미지 태그 자동 생성 확인, 수동 생성 및 삭제
    1. 이미지 클릭 후 이미지 상세보기와 하단의 태그 리스트에서 자동으로 부여된 태그 확인
    2. 최하단의 ['+']버튼 클릭 후 다이얼로그에 추가를 원하는 태그 직접 입력
    3. [ADD] 버튼 클릭 후 실시간으로 생성된 태그 리스트에 노출
    4. 삭제를 원하는 태그 톱클릭 액션 후 삭제 확인
  - 태그 검색
    1. 전체보기 레이아웃 상단 Search Bar 확인
    2. [돋보기] 아이콘 클릭 후 원하는 태그 검색
    3. 다중 태그 검색 기능을 지원하며 최대 5개까지의 다중 태그 검색이 가능
    4. 태그와 태그는 스페이스로 구분
    5. 다중 태그 검색 결과는 입력한 모든 태그를 내포한 이미지 소팅



 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

- Graph 메뉴 선택

1. 메인 레이아웃에서 [graph]버튼 클릭 시 태그 통계 기반 파이그래프 레이아웃으로 전환
2. 파이그래프 레이아웃 하단 [>]버튼 클릭 시 네트워크 그래프로 이동

- 안드로이드 동작 구현(Java)

- Splash Activity -> Main Activity

1. onCreate() 메소드 내 run() 함수를 구현하여 activity\_splash화면을 시작하고 splash화면이 다시 호출되지 않도록 finish()한다.

- 세부 이미지 클릭 후 태그 리스트 노출

1. RecyclerView를 통해 itemList 생성 후 txttag와 연결

- 태그 추가 동작

1. [+]버튼 클릭 후 AlertDialog()를 이용해 태그 입력 창 생성
2. 태그 입력 후 [add]버튼 클릭 후 editText.getTag.getText()를 이용해 입력 받은 데이터를 ImageTagInfo에 fileName, filePath, id, tagType 저장
3. 태그 string을 ArrayList<> tags 저장
4. 태그를 db [ImageInfo] 테이블에 저장

- 태그 삭제 동작

1. 태그 리스트에서 삭제하고자 하는 태그 데이터를 setOnLongClickListener()를 통해 삭제
2. db [deleteImageInfo]를 통해 db 에서 삭제

- 태그 검색

1. MenuItem 을 통한 searchMenu 구성
2. 태그 텍스트를 검색하면 onQueryTextChange()을 통해 태그 검색
3. tags.length <= 5 이면 db 내 selectMultiTag(tag)로 이동 후 검색
4. tags.length > 5 이면 Toast 문구 throw

2. 안드로이드 어플리케이션 내부 데이터베이스 구축


- sqLite를 사용해 안드로이드 내부 데이터베이스를 구축한다.

**1. [Image\_Info]** : 사용자가 가지고 있는 이미지의 모든 정보를 다룬다. 속도를 고려하여 로컬 파일 시스템에 저장한 후 이미지 절대 경로를 db에 저장한다.

Image\_info (id integer primary key, ImageNo integer, imageFileName text, imgFilePath text, created text, time text, gpsLat text, gpsLon text)

**2. [Face\_Tag]** : 자동으로 이미지 속에서 얼굴을 찾은 후 사용자가 입력한 사람의 이름 태그를 다룬다.

Face\_Tag (tagID text primary key, name string)

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

**3. [Face\_Vector\_Value]** : Face Recognize 프로세스를 거쳐 추출된 128개의 vector 값을 다룬다.

Face\_Vector\_Value (vectorId integer primary key, imgFileName text, imgLeft integer, imgTop integer, imgRight integer, imgBottom integer, vectorValue text, faceName text, label integer)

**4. [Auto\_Tag]** : 이미지 속의 사물을 자동 인식하여 추출한 태그를 다룬다.

Auto\_Tag (tagID text primary key, name\_end text, name\_kor text, category text, tagState boolean)

**5. [Manual\_Tag]** : 사용자가 직접 입력한 태그를 다룬다.

Manual\_Tag (id integer primary key, tagID text, tagName text)

**6. [Tag\_Log]** : 각 이미지에 부여된 태그 정보를 관리한다.

Tag\_Log (tagID text, tag text, imgFilePath text, ImageNo integer, created text)

**7. [Tag\_Network]** : 하나의 사진에 동시에 부여된 태그들 사이의 동시 출현 빈도를 관리한다.


Tag\_Network (\_id integer primary key, tag\_1\_id text, tag\_2\_id text, weight integer)

**8. [Total\_Tag\_Info]** : 중복되지 않은 전체 태그의 정보를 관리한다.

Tag\_Info (\_id integer primary key, tag\_id text, tag\_name text, group\_num integer, tag\_freq integer)

### 3. 이미지 동기화

- WITTYPHOTOS 앱이 안드로이드 기기 내부 시스템에 접근하여 허용권한을 받아오는 다이얼로그를 throw하고 mediastore 를 통해 필요한 이미지 정보를 가져온다.
- 앨범 메뉴가 실행되면 기기내의 이미지를 동기화하는 동작
  - ImagePickerDirectoryActivity.class : 실제 앨범 카테고리가 보여지는 기능을 구현한 클래스이다.
  - ImagePickerAdapter 를 이용해 이미지 데이터 값을 ArrayList로 받아온다. onCreateViewHolder 와 onBindViewHolder 를 이용해 앨범 카테고리 버킷을 받아오고 레이아웃에 표시한다. onBindViewHolder를 사용해서 버킷화 되어있는 카테고리의 dirName과 카테고리별 이미지 개수를 보여주는 dirCounter를 표시한다.
  - PickerDir.class : 기기의 이미지를 가져 오기 위한 element들을 정의하고 받아오는 기능을 구현한 클래스로 dirName, dirId, imgResId, imgPath, count로 구성

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

- ImgPickerDirController.class : ImagePickerDirectoryActivity 클래스가 작동하기 위해서 화면 background에서 돌아가는 동작을 구현한 클래스이다.  
내부 LoadImageDirList 클래스  
AsyncTask - doInBackground 를 이용해 동작한다. MediaStore.Images.Media 에서 DATA, \_ID, BUCKET\_DISPLAY\_NAME, BUCKET\_ID를 받아와서 실질적인 이미지 경로를 저장한다.  
  
Cursor를 이용해 이미지 데이터를 움직이면서 이미지 세부의 카테고리화를 시작한다. 이 부분에서 각 이미지의 개수를 명시하는 dirCounter의 개수가 저장된다. 이미지를 HashMap으로 받아 이미지 서치에 용이하게 배열한다.
- ImagePickerDetailActivity.class : 위의 ImgPickerDirectoryActivity의 함수와 기능을 같이 한다. ImagePickerDetailActivity는 디렉토리가 아닌 이미지 한 장 한 장을 보여줄 수 있는 동작을 구현한 클래스이다.  
init()을 이용해 실질적인 이미지 아이템 데이터 값을 표시한다.
- PickerImage.class : 이미지 마다 가지고 있는 imgPath를 Uri값으로 받고 저장하는 동작을 한다. boolean을 이용해 imgPath가 올바른지 더블 체크한다.
- ImgPickerDetailController.class : ImgPickerDirectoryController와 동일한 동작을 하고 추가적으로 각 이미지의 path값을 사용한다.

#### 4. 클라이언트 구축

#### 5. pythonanywhere 서버 구축

#### 6. 이미지를 인식하여 자동으로 태그 지정


- 객체태그 - YOLO v3 (Python OpenCV) (객체 인식)  
서버에서 YOLO v3(Python OpenCV)를 이용하여 이미지에 추출된 영어 태그 값을 안드로이드에서 Json으로 받아온다.  
DB를 참조하여 미리 저장되어 있는 한글 태그와 태그ID를 받아와서 DB에 정보를 저장한다.
- 인물태그 - Face\_recognition (얼굴 인식)  
서버에서 python의 face\_recognition 라이브러리를 이용하여 얼굴의 위치 박스 값과 얼굴에 대한 벡터값을 받아온다.  
안드로이드 내에서 DBSCAN 알고리즘을 이용하여 클러스터링 한다.  
클러스터링 라벨에 따라 태그ID를 부여하여 DB에 저장한다.

#### 7. 태그 네트워크 커뮤니티 추출

비교적 빠르고 높은 모듈성을 가진 커뮤니티 추출을 보이는 Louvain Algorithm을 선택하여 사용하였다.

새로운 사진이 입력될 때 이미지에서 자동으로 추출된 태그와 사용자가 수동으로 입력한 태그들 사이의 동시 출현빈도를 함께 DB(Tag\_network)에 저장한다.

태그ID와 태그 동시 출현 빈도 정보를 DB로부터 받아와 Lovain 알고리즘을 이용하여 추출된 커뮤니티 Group number를 DB에 저장한다.

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	결과보고서		
	프로젝트 명	WittyPhotos - 더 똑똑한 갤러리	
	팀 명	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

## 8. 태그 통계 및 네트워크 시각화

- 태그 통계

DB(Total\_Tag\_Info)에 저장된 태그의 출현 빈도를 받아온다. 상위 5개의 태그와 출현 빈도를 불러와서 자주 사용된 태그의 이름, 빈도를 PieChart로 나타낸다.

(MPAndroidChart 라이브러리 이용 - PieChart)

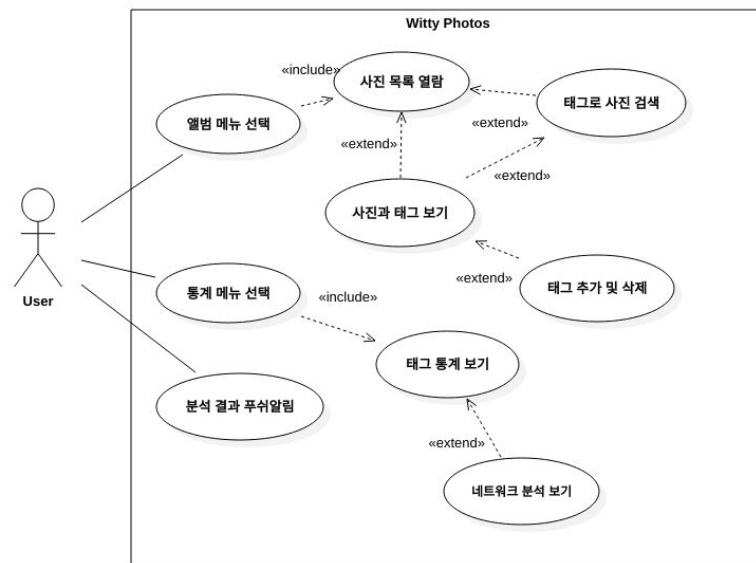
- 네트워크 시각화


네트워크 그래프 화면을 띄울 때 즉시 Louvain 알고리즘을 커뮤니티를 number를 DB(Total\_Tag\_Info)에 저장되어 있는 Group number에 따라 노드 색상을 지정하여 연관도가 높은 태그끼리 같은 색상으로 표시한다. 한 이미지에 동시 출현한 적이 있는 태그들 사이는 선으로 연결하여 그래프화한다.

(GraphStream 라이브러리 이용)

## 2.2.2 시스템 기능 요구사항 시스템 기능 요구사항

[Use Case Diagram]



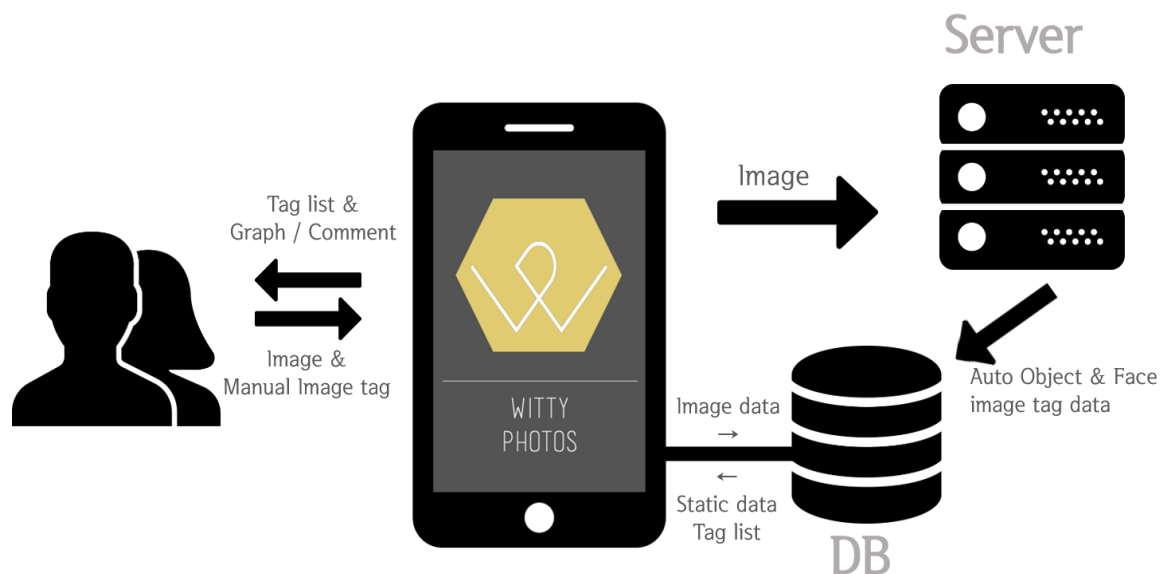
 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

### 2.2.3 시스템 비기능(품질) 요구사항


1. 모바일 환경에서 사용자가 기능을 쉽게 파악할 수 있는 직관적인 UI/UX를 제공한다. - 달성
2. 이미지 인식 기술을 적용하여 자동 태그 기능의 정확도를 높인다. - 미달성
3. 사용자의 데이터가 안전하게 유지되도록 데이터베이스 관리를한다. - 달성
4. 어플리케이션이 사용자의 스마트폰 내의 이미지 경로에 접근할 때 사용자에게 접근 허가 요청 메시지를 보낸다. - 달성
5. 사용자가 각 페이지를 넘어갈 때 1초 미만의 시간이 걸린다. - 미달성
6. 새로운 이미지들이 동기화 될 때 이미지 인식 후 데이터베이스에 자동 태그 지정까지 10장당 1분 미만의 시간이 걸린다. - 미달성

### 2.2.4 시스템 구조 및 설계도

[시스템 구조도]

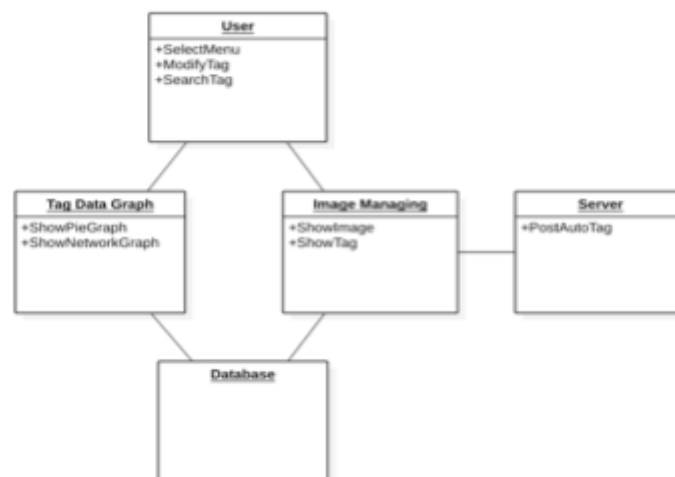


[시스템 환경]

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

1. 운영체제 : Android, Linux
2. 네트워크 : TCP/IP, WIFI
3. 응용 환경 : Mobile(Android), Server(pythonanywhere)
4. GUI 환경 : Android Application
5. 프레임워크
  - 서버 : Django
  - 클라이언트 : Retrofit2
  - 이미지 객체 인식 : openCV, YOLO v3
  - 이미지 얼굴 인식 : python face recognition API, DBSCAN
  - 지오코딩(GeoCoding) : EXIF API, google Map API
  - 데이터베이스 : SQLite3


#### [객체 설계도]

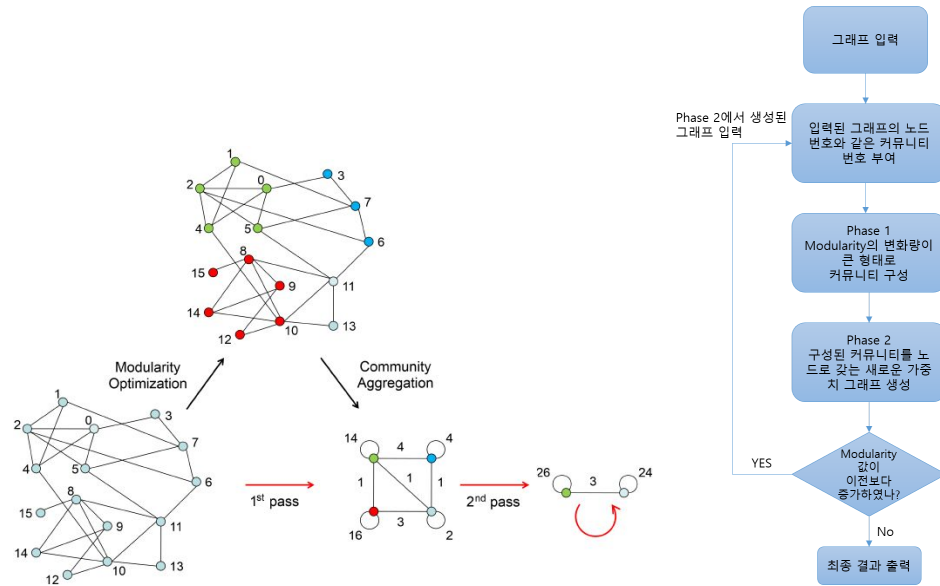


## 2.2.5 활용/개발된 기술

### 1. Louvain Algorithm

: 두 태그의 동시 출현 빈도를 이용하여 커뮤니티 number를 계산한다. 그래프화할때 커뮤니티별로 색상을 구분을 하여 시각화하기 위해 사용하였다.

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	결과보고서		
	프로젝트 명	WittyPhotos - 더 똑똑한 갤러리	
	팀 명	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28



## 2. YOLO v3 (Python OpenCV)

: 사진에서 객체를 인식해 내어 자동으로 태그를 부여하기 위하여 사용하였다.

이미지 또는 비디오 상의 객체를 식별하는 컴퓨터 비전 기술로 딥러닝 혹은 머신러닝 알고리즘을 통해 산출되는 기술이다.

OpenCV로 구현된 YOLO는 기본 darknet 모델보다 10배 이상 빠른 속도를 보인다.

이미지 한 장당 객체 인식하는데 약 2초가 소요되었다.


정확도는 YOLO3의 weight를 가져와 사용하므로 정확도는 기본 모델과 큰 차이는 없다.

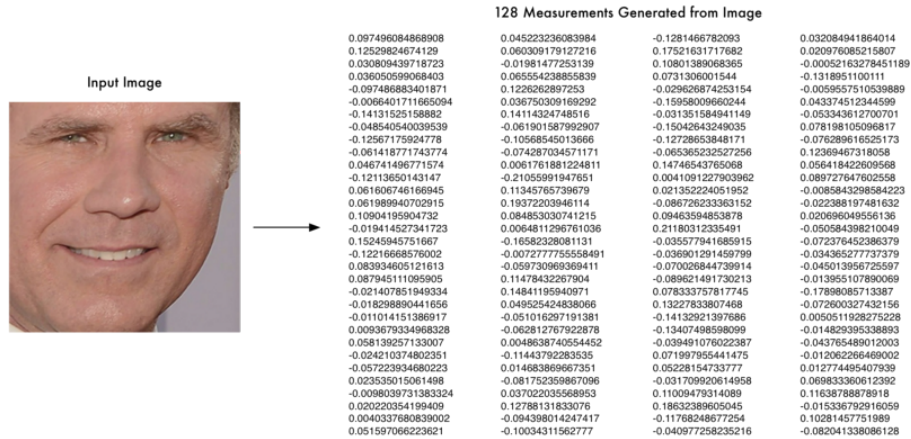
## 3. face\_recognition

: 사진에서 얼굴을 찾아내어 얼굴의 위치와 각 얼굴의 벡터값을 추출하기 위해 사용하였다.

dlib 라이브러리에 포함되어 있다.

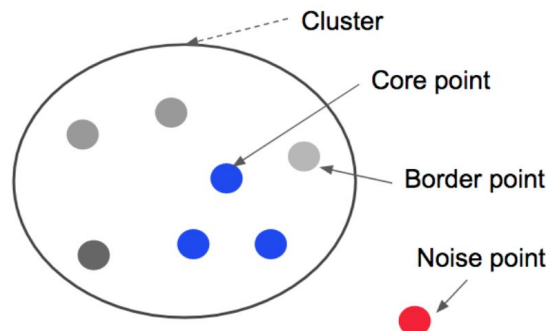
사람의 얼굴에 대해 128개의 측정값(임베딩 - embedding)을 생성할 수 있도록 훈련된 신경망이다.

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28



#### 4. DBSCAN Algorithm

: face\_recognition을 이용하여 얻어낸 벡터값을 클러스터링하여 같은 군집으로 묶인 얼굴에는 동일한 이름의 태그를 부여하기 위해 사용하였다.



밀도 기반 클러스터링 알고리즘

점이 세밀하게 몰려 있어서 밀도가 높은 부분을 클러스터링 하는 방식


한 점을 기준으로 반경 eps 이내에 점이 n개 이상 있으면 하나의 군집으로 인식한다.

#### 2.2.6 현실적 제한 요소 및 그 해결 방안

- 안드로이드 기기가 WI-FI 혹은 데이터 연결이 안되어있는 경우 서버 연결이 끊어져 새로운 이미지 대하여 객체 인식과 얼굴 인식이 불가능하다.
- 기기에서 어플을 삭제했을 때 DB데이터 또한 삭제 된다.

#### 2.2.7 결과물 목록



 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

대분류	소분류	기능	기술문서
안드로이드	태그 검색	최대 5개까지 태그 입력을 통해 이미지 검색한다. 태그 구분 스페이스로 한다.	무
	태그 리스트 확인	하나의 사진을 누르면 사진과 함께 입력된 태그 리스트를 노출한다.	무
	태그 삽입	[+]버튼 클릭 후 수동으로 원하는 태그를 입력하고 삽입 후 실시간으로 리스트에 업데이트 한다.	무
	통계 그래프	자주 사용된 최대 상위 5개의 태그 이름과 빈도수를 PieChart로 나타낸다.	무
알고리즘	네트워크 그래프	사용자의 태그를 분석하여 분류된 커뮤니티를 그래프로 보여준다.	무
	얼굴 인식	사진에 입력된 얼굴들의 벡터값을 추출해 얼굴을 분류한다.	무
	객체 인식	사진 속의 객체를 자동 인식하여 태그를 부여한다.	무
서버	서버		무
DB	DB	태그 삽입, 입력, 커뮤니티 추출에 필요한 데이터를 저장한다.	무


## 2.3 기대효과 및 활용방안


### 2.3.1 기대효과

사용자의 갤러리 내의 이미지들에 태그를 부여하고 태그를 다중 검색 기능을 이용하여 이미지의 특징을 몇가지 기억하는 것만으로도 사용자가 원하는 이미지의 양을 줄여 빠른 검색이 가능하다. 또한 부여된 태그 분석을 통해 사용자가 연결되어있는 커뮤니티 분류를 시각적으로 확인 할 수 있다.

### 2.3.2 활용방안

사람들의 만남 속에서 이야기 중에 보여줄 사진이 있다며 핸드폰의 사진첩을 찾을 때가 있다. 이 경우에 사진이 정확히 어디에 위치해 있는지 알지 못하여 오랜시간이 걸리는 경우가 생긴다. WittyPhotos 앱을 이용하면 검색 범위를 줄여 빠르게 사진을 찾을 수 있다.

 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

### 3 자기평가

- 객체 인식 YOLO v3(Python OpenCV)

python3 openCV를 이용하여 YOLO v3기능을 구현하였다. DNN, YOLO의 darknet 모델 등 타 딥러닝 모델보다 우수한 속도를 보였다.  
이미지 한 장당 객체 인식하는데 약 2초가 소요되었다.

- 얼굴 인식, 클러스터링

python의 face\_recognition을 통해 얼굴의 128개 벡터 값을 받아내어 DBSCAN을 이용하여 클러스터링을 할때 적절한 eps값과 min\_samples 값을 찾아내기 위해 고민을 하던 중 교수님의 조언을 얻어 t-SNE를 이용하였다. t-SNE는 고차원의 데이터를 저차원의 데이터로 매핑하여 시각화하는 것으로 클러스터링 된 값에 따라 점의 색상을 다르게 하여 그래프를 확인하며 가장 적절한 값을 찾았다.

최선의 eps와 min\_samples 값을 찾았음에도 클러스터링 되었을 때 적절히 묶이지 않는 값과 라벨이 붙지 않는 값이 생긴다. 미처 찾아내지 못한 더 나은 군집을 형성하는 더 나은 알고리즘을 찾아 낼 수 있다면 얼굴 분류가 더욱 세밀하게 이루어질 수 있을 것으로 보인다.


- 서버 / 클라이언트

Django rest framework 를 사용하여 pythonanywhere 클라우드 서버에 이미지 업로드, 객체 인식, 얼굴 인식 기능을 배포하였다.  
안드로이드 클라이언트에서는 okhttpClient 와 retrofit2를 사용하여 구축하였다.

- 네트워크 커뮤니티 분석


처음에는 betweenness centrality기반으로 커뮤니티를 찾는 Girvan-Newman 알고리즘을 선택하였으나 서버 의존도를 낮추기 위해 java로 코드를 변경하는 과정에서 더 나은 커뮤니티 추출 알고리즘을 찾아 보게되었다. 이 과정에서 Louvain Algorithm이 더 빠르고 높은 모듈성을 갖는 커뮤니티를 추출한다는 것을 알게되어 이 알고리즘을 최종 선택하였다.

추후에 더 나은 커뮤니티 추출 알고리즘을 찾아볼 기회가 생긴다면 그래프 생성시 더욱 짧은 시간이 걸릴 것이라고 본다.

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

## 4 참고 문헌

번호	종류	제목	출처	발행년도	저자	기타
1	서적	Do it! 안드로이드 앱 프로그래밍	이지스퍼블리싱	2018	정재곤	
2	사이트	Smartphones Cause Photography Boom	<a href="https://stactista.com">https://stactista.com</a>	2017	Flex Richer	
3	논문	ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software.	PLOS ONE 9(6): e98679.	2014	Jacomy M, Venturini T, Heymann S, Bastian M	
4	논문	You Only Look Once: Unified, Real-Time Object Detection	CVPR	2016	Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi	
5	논문	A density-based algorithm for discovering clusters in large spatial databases with noise (DBSCAN)	KDD	1996	Ester, M., Kriegel, H. P., Sander, J., & Xu, X.	
6	사이트	[네트워크이론] Louvain algorithm for community detection	<a href="https://monsieur1220.tistory.com/129">https://monsieur1220.tistory.com/129</a>	2018		
7	사이트	클러스터링 #3 - DBSCAN (밀도 기반 클러스터링)	<a href="https://bcho.tistory.com/1205">https://bcho.tistory.com/1205</a>	2017		
8	사이트	Python Face Recognition in Real Time	<a href="https://ukayzm.github.io/python-face-recognition/">https://ukayzm.github.io/python-face-recognition/</a>	2018		
9	사이트	Python Face Clustering	<a href="https://ukayzm.github.io/face-clustering/">https://ukayzm.github.io/face-clustering/</a>	2018		

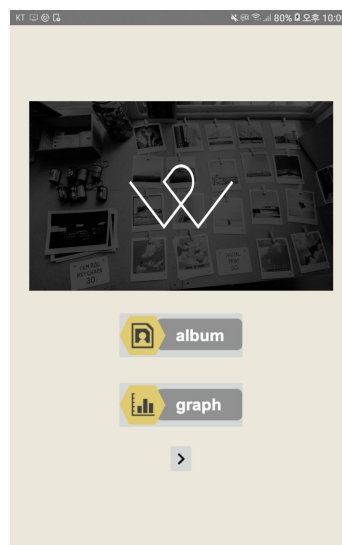
 <b>국민대학교</b> <b>컴퓨터공학부</b> <b>캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

## 5 부록

### 5.1 사용자 매뉴얼


- 메인 화면

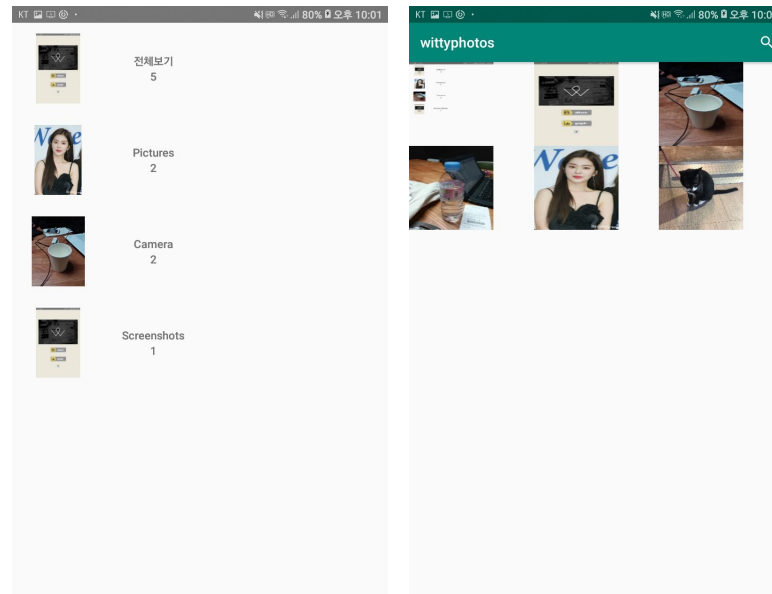
[album] 과 [graph] 버튼을 확인할 수 있다.



- 사진 목록

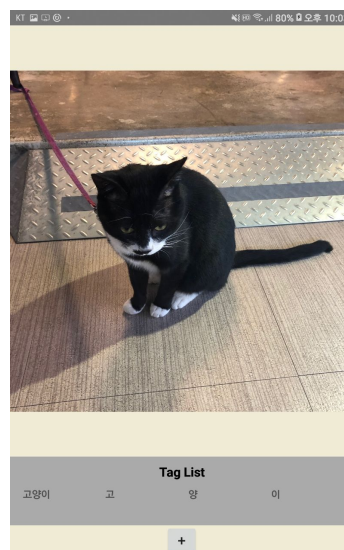
메인 화면에서 [album]버튼 클릭하면 핸드폰에 저장되어있는 모든 사진 디렉토리 별로 확인 할 수있다.

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	결과보고서		
	프로젝트 명	WittyPhotos - 더 똑똑한 갤러리	
	팀 명	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28




- 사진 선택

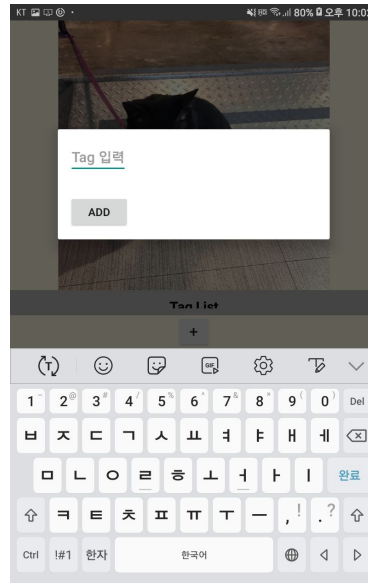
사진 목록에서 원하는 사진을 선택하면 해당 사진에 부여되어있는 태그 목록을 확인 할 수 있다.



- 태그 추가/삭제

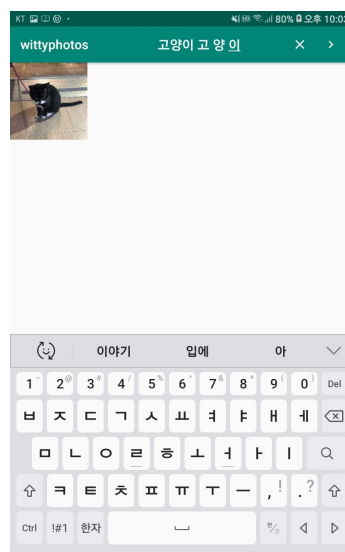
태그 리스트 하단의 [+] 버튼 클릭 시 원하는 태그를 직접 입력 가능하다. 태그 삭제의 경우 삭제를 원하는 태그를 길게 누르면 삭제된다.

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	결과보고서		
	프로젝트 명	WittyPhotos - 더 똑똑한 갤러리	
	팀 명	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28



- 태그 검색

앨범 내의 전체보기 메뉴로 이동 후 화면 최상단에 search bar를 이용해 태그를 검색할 수 있다. 다중 태그 검색이 가능하며 태그 검색은 최대 5개의 태그까지 가능하다.



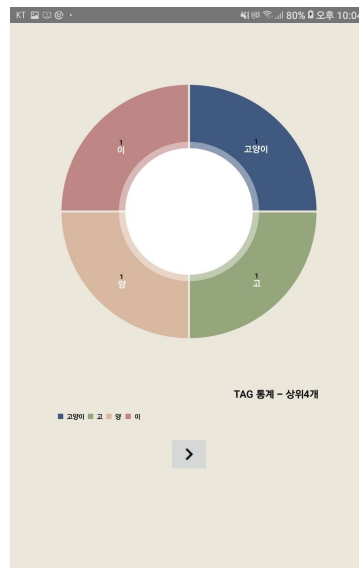
- 태그 그래프

메인 화면에서 [graph]버튼 클릭 시 PieChart 화면으로 전환되며 화면 하단 [>]버튼을 누르면 Network 그래프를 볼 수 있다.

결과보고서	
프로젝트 명	WittyPhotos - 더 똑똑한 갤러리
팀 명	메모리즈
Confidential Restricted	Version 1.4
	2019-May-28



국민대학교  
컴퓨터공학부  
캡스톤 디자인 I



## 5.2 운영자 매뉴얼

- 서버 설치

Python3.6설치

Django rest framework 설치

- Face\_recognition 설치

- Linux

```
$ sudo apt-get install python3 python3-dev python3-venv
```

```
$ python3 -m venv py3
```

```
$ source py3/bin/activate
```

```
(py3) $ pip install --upgrade pip
```

```
(py3) $ pip install opencv-python
```

```
(py3) $ pip install opencv-contrib-python
```

```
(py3) $ pip install dlib
```

```
(py3) $ pip install face_recognition
```


-Window

<https://www.python.org/downloads/> Python 3 설치

<https://cmake.org/download/> cmake설치

```
C:\> pip install opencv-python
```



 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

C:\> pip install opencv-contrib-python

C:\> pip install dlib

C:\> pip install face\_recognition

C:\> pip install dlib-19.8.1-cp36-cp36m-win\_amd64.whl

- 통계 그래프 - MPAndroidChart  
- Android studio gradle 에 추가

repositories {

maven { url 'https://jitpack.io' }

}

dependencies {

implementation 'com.github.PhilJay:MPAndroidChart:v3.1.0-alpha'

}

- 네트워크 그래프 - Graphstream  
- java 8로 실행  
- Android studio gradle 에 추가

repositories {

maven { url 'https://jitpack.io' }


}

dependencies {

api 'com.github.graphstream:gs-ui-android:2.0-alpha'

api 'com.github.graphstream:gs-core:1.3'

}

 <b>국민대학교 컴퓨터공학부 캡스톤 디자인 I</b>	<b>결과보고서</b>		
	<b>프로젝트 명</b>	WittyPhotos - 더 똑똑한 갤러리	
	<b>팀 명</b>	메모리즈	
	Confidential Restricted	Version 1.4	2019-May-28

### 5.3 테스트 케이스

분류	기능	테스트 방법	기대 결과	테스트 결과
앱	앨범 동기화	스플래쉬화면 후 메인 화면에서 [앨범] 버튼을 누른다.	안드로이드 기기에 있는 이미지 디렉토리가 리스트뷰로 보인다.	성공
	자동 태그 확인	메인 화면에서 [앨범] 버튼을 누르고 사진을 선택한다.	자동으로 부여된 태그를 확인할 수 있다.	성공
	수동 태그 삽입	메인 화면에서 [앨범] 버튼을 누르고 사진을 선택한 후 [+]버튼을 누른 후 원하는 태그를 입력한다.	이미지의 수동 태그가 부여된다.	성공
	태그 삭제	메인 화면에서 [앨범] 버튼을 누르고 사진을 선택한 후 지우고자 하는 태그를 오래 누른다.	이미지에 부여되었던 태그가 삭제되며 리스트에서도 삭제된다.	성공
	태그 검색	메인화면에서 [앨범] 버튼을 누르고 상단의 검색 아이콘을 누르고 검색하고자 하는 태그를 입력한다.	입력된 태그가 부여되어있는 사진들이 보여진다.	성공
	PieChart	메인 화면에서 [그래프] 버튼을 누른다.	사용된 태그의 통계 그래프가 보여진다.	성공
	Network Graph	PieChart 화면에서 하단의 화살표를 누른다.	사용된 태그의 네트워크 그래프가 보여진다.	성공