

ADG

Automatic Detection Gallery

Contents



프로젝트 소개



시스템 구성



시연



프로젝트 리뷰

1

팀원 소개

팀장, 데이터셋 구성 : 박노준

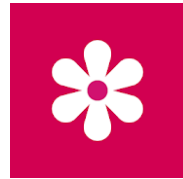
메인 앱 개발 : 박건우

데이터셋 구성 및 학습 : 이현민

데이터셋 구성 및 앱 개발 도움 : 정은진

1

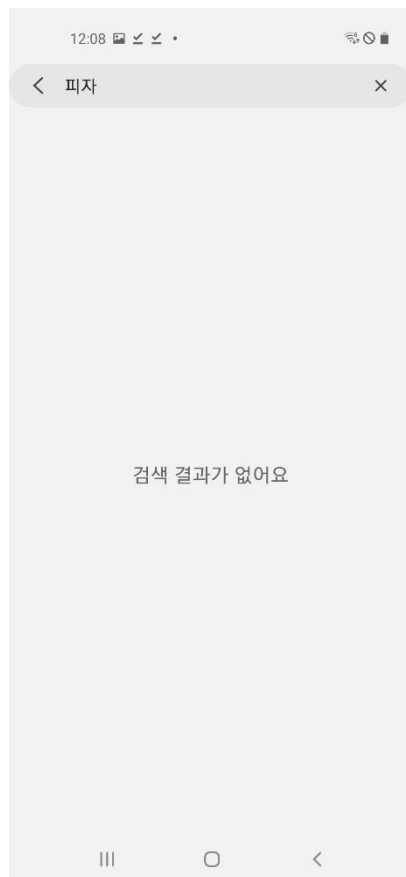
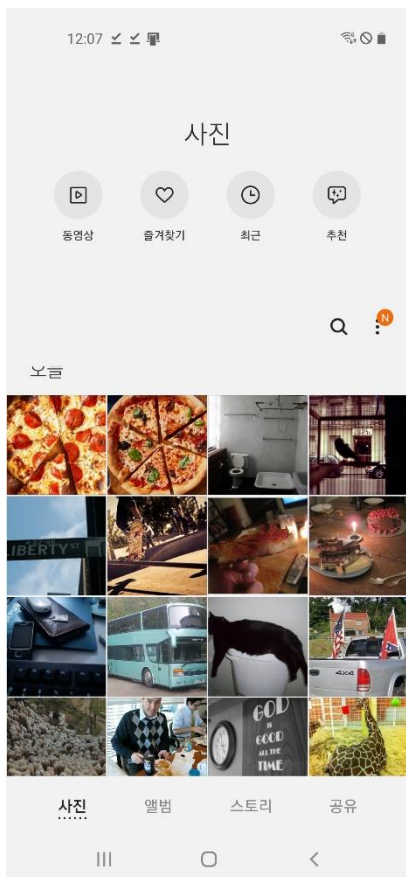
프로젝트 소개



GO



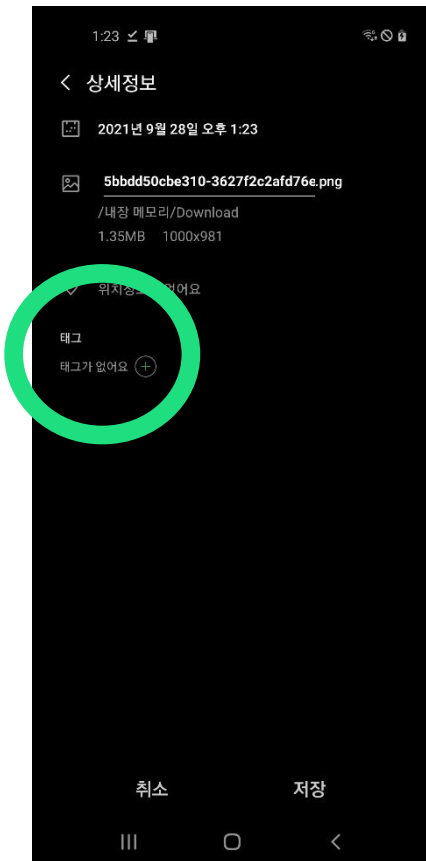
1 개발 동기



삼성 갤러리 앱에서 피자를 검색했으나
검색되지 않는 모습

1

개발 동기



태그의 가시성 낮음
 태그의 종류가 적음
 태깅 방식을 알 수 없음

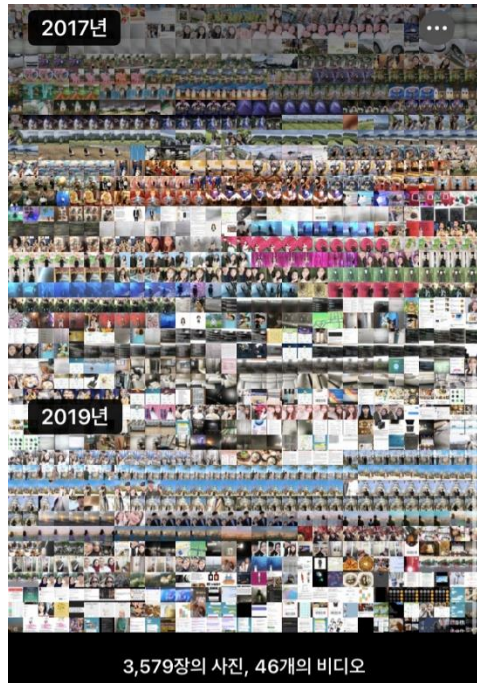
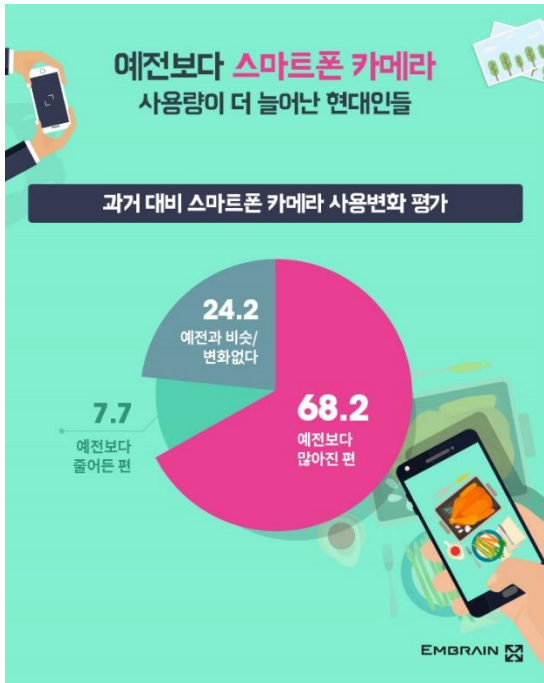
-
-
-

기존 갤러리 앱은 기본 기능은 매우 훌륭하지만
 태그 활용이 부족함

-> 태그를 활용하여 다양한 편의 기능을 제공
 할 수 있다면?

1

개발 동기



1. 스마트폰에 탑재되는 카메라의 스펙이 매우 상승함
2. 틱톡, 인스타그램 등 다양한 SNS서비스의 이용이 늘어나면서 스마트폰 카메라의 사용량이 증가

-> 늘어나는 사진의 수로 인해 원하는 사진을 찾는 것이 어려워짐

1 그래서 ADG는 ?

Auto Tagging

Fast searching

Tag customizing

Real-time Detection



프로젝트 소개



시스템 구성



시연



프로젝트 리뷰

2

시스템 구성 - Dataset

Dataset

- LVIS Dataset
 - Place365 Dataset
 - Food Dataset
- +
- Google Image web Crawling

그 결과 330개의 classes + 100000장 이상의 train images

2 시스템 구성 - Dataset

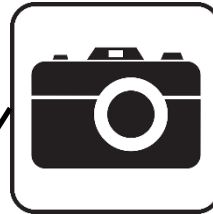
Dataset Classification

- Dataset에 포함되어 있는 annotations 파일을 기준으로 분류 시작
- [Google flare dendrogram](#)을 기반으로 1차 분류 진행
- 2차 분류는 회의를 통해 자체 분류

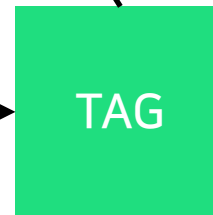
2 시스템 구성



- Model : YOLO V4
- Tensorflow 2.0
- CUDA >= 10.2, cuDNN >= 8.0.2
- Colab Pro 환경 사용



실시간



android

- IDE : Android Studio
- Lang : JAVA

2

시스템 구성 - Model

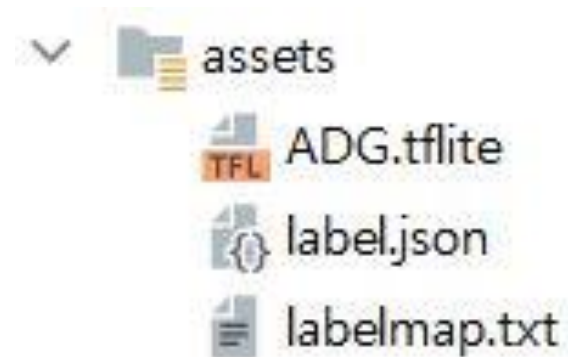


android

2 시스템 구성 - Model

학습시킨 모델 파일은 안드로이드 앱 자체에 탑재

(assets 폴더에 label이 적힌 파일과 같이 탑재)



* 현재 탑재된 (21-10-31 기준) model의 학습 진행도
Iteration : 203000, avg loss : 7-10

2

시스템 구성(안드로이드 스튜디오)

```
public class TFHelper {
    public TFHelper(Context context) {
        this.context = context;
        try {
            detector = YoloV4Classifier.create(
                context.getAssets(),
                TF_OD_API_MODEL_FILE,
                TF_OD_API_LABELS_FILE,
                TF_OD_API_IS_QUANTIZED);
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    private float minScore = 0.7f;

    Context context;
    public static final float MINIMUM_CONFIDENCE_TF_OD_API = 0.2f;

    public static final int TF_OD_API_INPUT_SIZE = 608;
    private static final boolean TF_OD_API_IS_QUANTIZED = false;
    private static final String TF_OD_API_MODEL_FILE = "yolov4-tiny-608.tflite";
    private static final String TF_OD_API_LABELS_FILE = "file:///android_asset/labelmap.txt";

    // Minimum detection confidence to track a detection.
    private static final boolean MAINTAIN_ASPECT = false;
    private Integer sensorOrientation = 90;

    private Classifier detector;

    private Matrix frameToCropTransform;
    private Matrix cropToFrameTransform;
    private MultiBoxTracker tracker;
    private OverlayView trackingOverlay;

    protected int previewWidth = 0;
    protected int previewHeight = 0;
}
```

- TFHelper class

Tflite model을 다루기 위한 class

모델이름과 태그리스트가 담긴 파일을 정적변수로 저장

minscore 변수로 최소 검출 정확도 지정 가능



프로젝트 소개



시스템 구성



시연



프로젝트 리뷰

3

시스템 구성 - 메인 화면

검색하길 원하는 단어 입력



태깅 진행도를 알려주는 부분



원하는 정렬 방식 선택

첫 태깅 이후 새로 추가된 사진들은 앱 재실행시 자동으로 태깅 진행

3 시스템 구성 - 사진 선택 화면

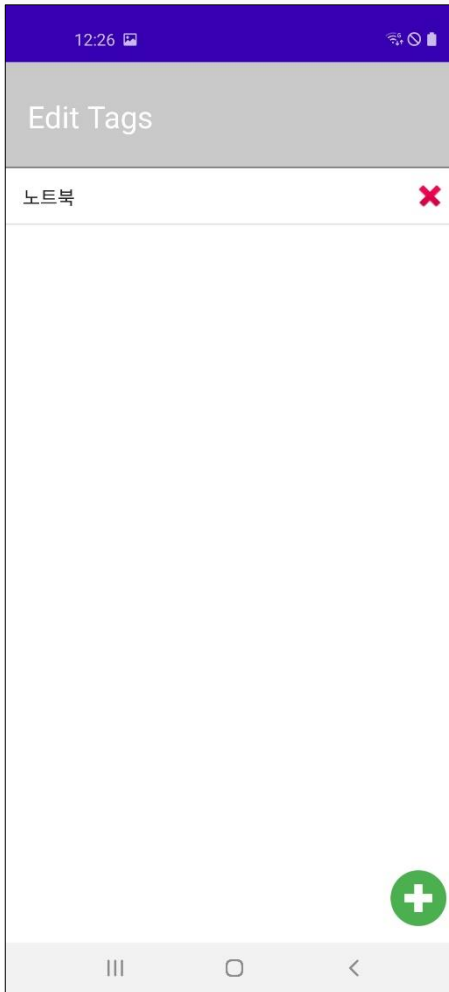
사진이 출력되는 부분 →

현재 사진에 달린 태그들을 확인할 수 있음 →



터치 시 태그 수정 가능 →

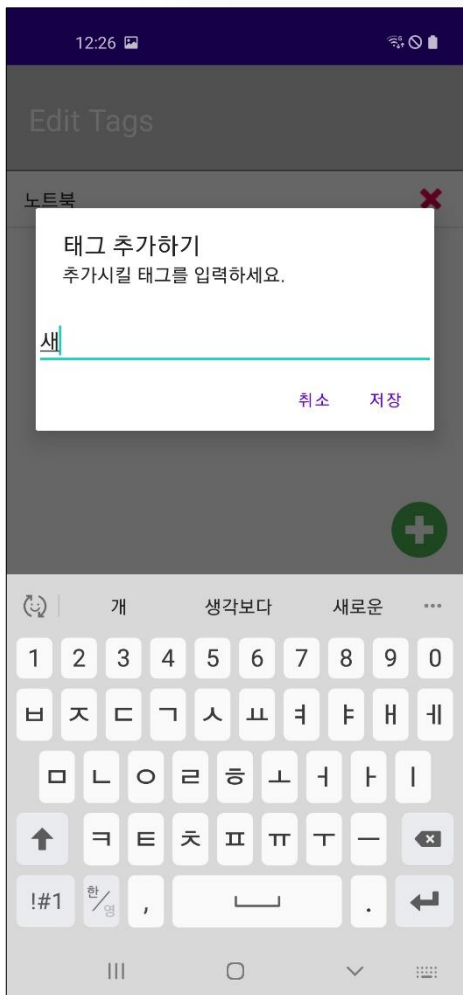
수정 버튼을 누르면 태그를 편집할 수 있는 화면으로 이동 ←



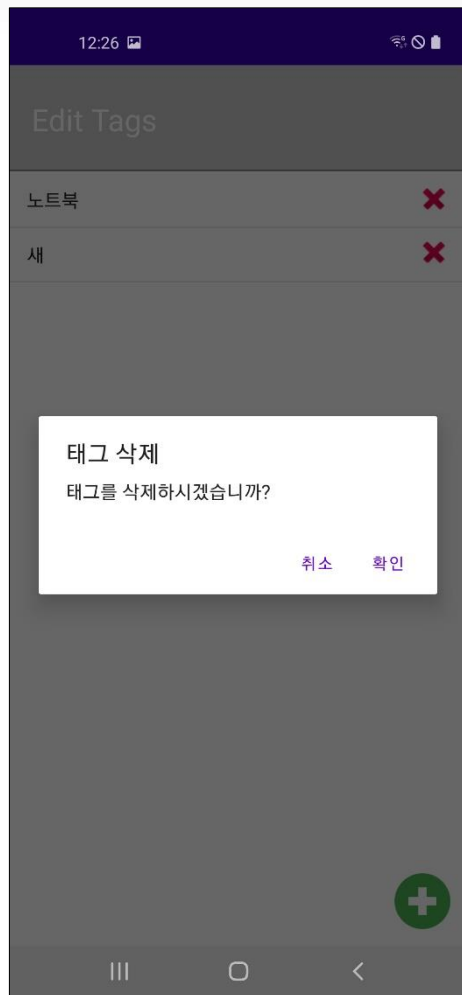
← 터치 시 태그 삭제 가능

← 터치 시 새로운 태그 추가 가능

3 시스템 구성 - 사진 선택 화면



← 저장버튼을 누르면
입력한 태그를 추가

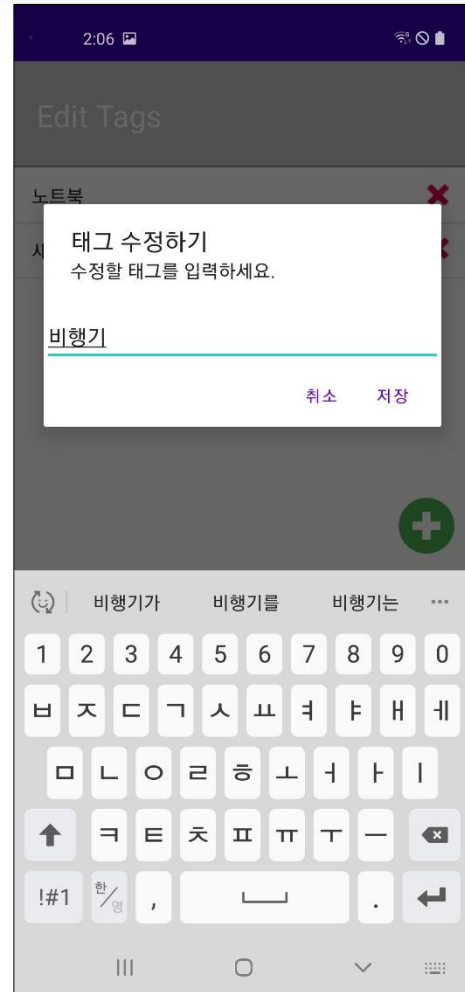


← 확인버튼을 누르면
선택한 태그를 삭제
(DB)

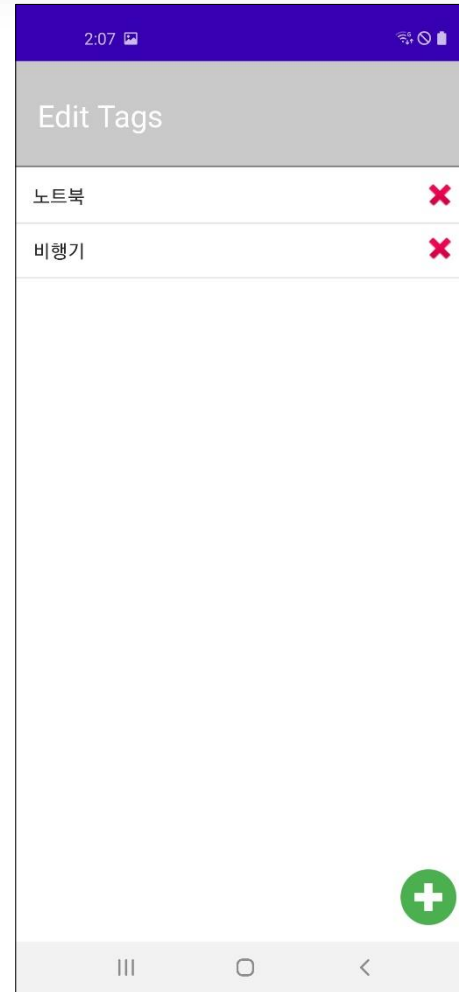
3

시스템 구성 - 사진 선택 화면

원하는 태그를 터치
하면 수정할 수 있음



태그가 올바르게 수
정된 것을 확인할 수
있음



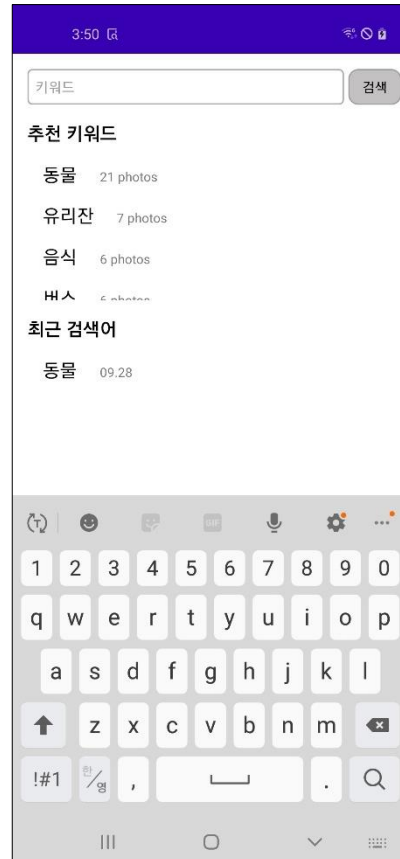
3

시스템 구성 - 검색 화면

사진들의 태그 수를 바탕으로 추천 키워드 제공

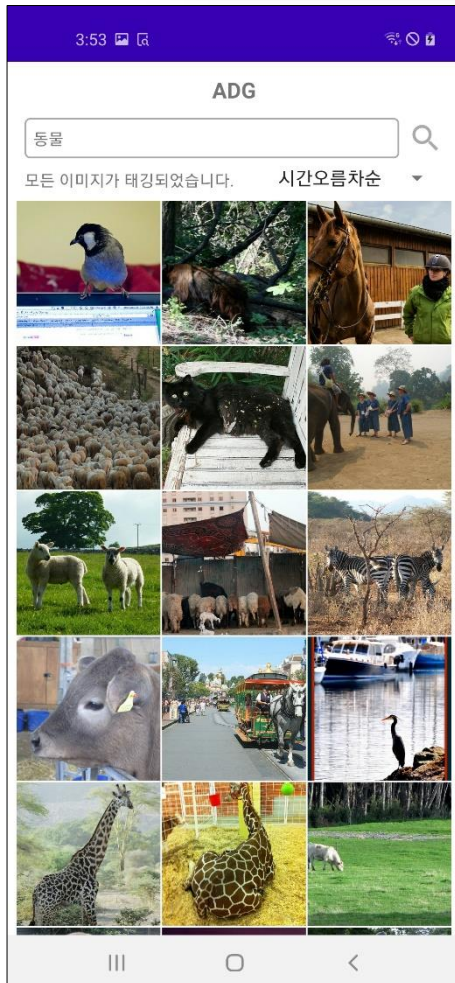


사용자의 검색기록을 DB에 저장해 최근 검색어를 알려줌



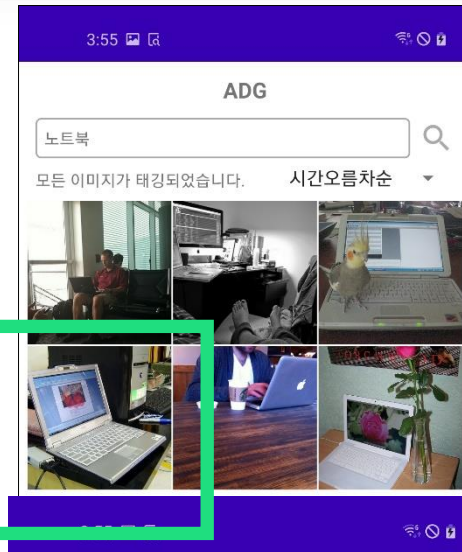
혹은 사용자가 직접 검색어를 입력할 때 해당 글자를 포함한 태그들을 알려줌

3 시스템 구성 - 검색 결과 화면

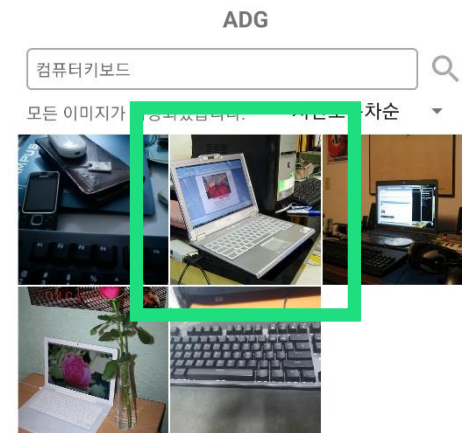


검색어에 해당되는 사진 출력

메인 화면의 뷰 이용으로 정렬기능과 검색기능을 똑같이 이용 가능



두 개 이상의 태그를 가지고 있는 사진도 각각의 결과에 맞게 출력



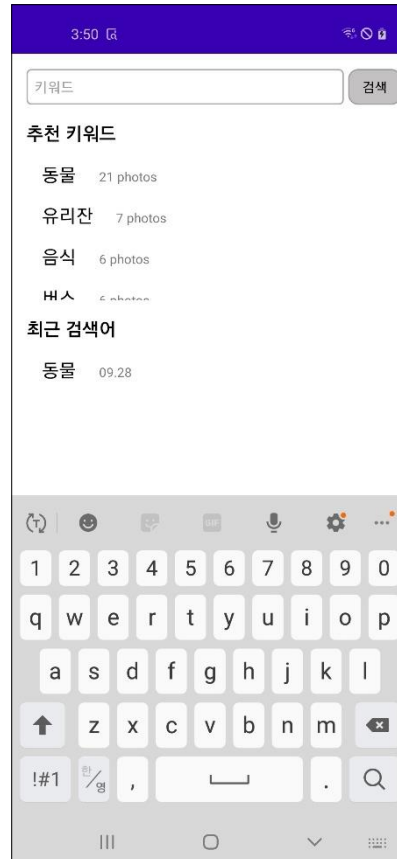


3

시연 영상

3

시연 영상





프로젝트 소개



시스템 구성



시연



프로젝트 리뷰

4 앱 리뷰

추가 될 수 있는 기능

1. SNS 글 작성시 사진 태그 자동 본문 추가
2. 태그 데이터를 활용한 다양한 정보 제공
3. 앨범 기능

배운 점

1. 형상 관리 프로그램의 중요성
2. 딥러닝 학습 환경 구축
3. 안드로이드 앱의 구성
4. 학습 데이터 셋의 중요성



ADG

프로젝트 리뷰

100%

「Q&A」



ADG

프로젝트 리뷰

100%

감사합니다.