

사물인식 기반의 웹툰 AR 플레이어 만들기

2017/5/18

Team H

강민지 원종훈 품언





Contents

Overview

Problem/Goal

Approach

Development Environment

Basic Spec/Architecture

Implementation Spec

Current Status

Further plan

Demo Plan

Division and Assignment of work

Schedule

Overview

What 사물인식 기반의 웹툰 AR 플레이어

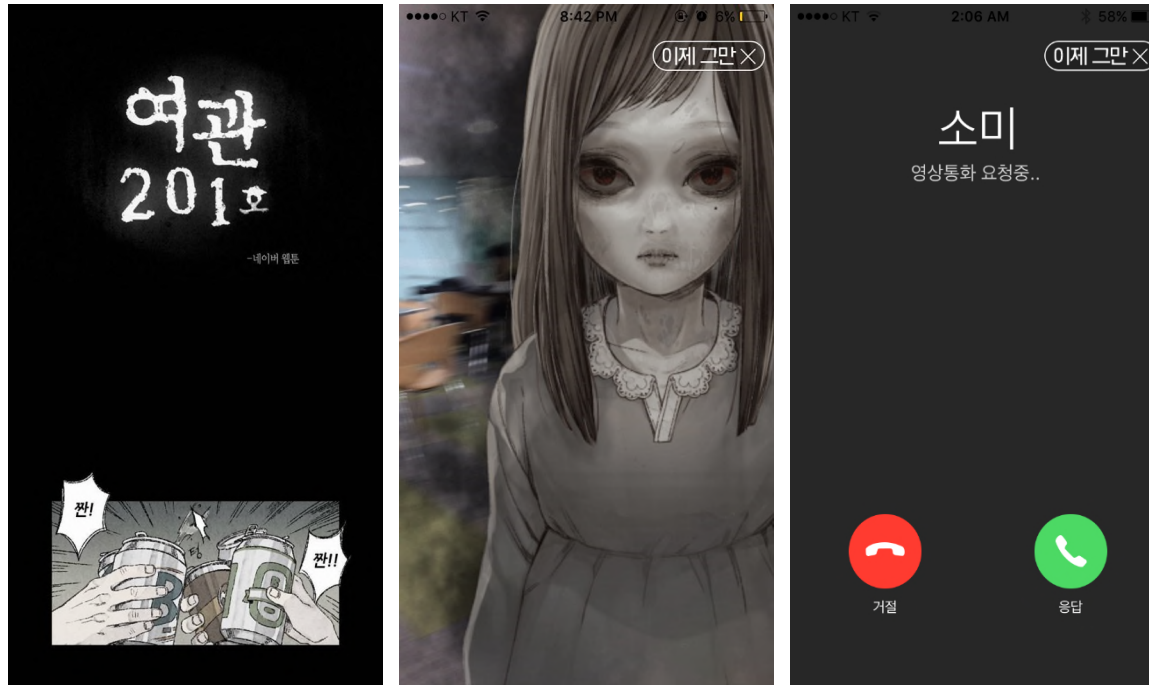
Where **NAVER**

When 3개월 동안

Why - 사용자 경험 향상
 - 다양한 스토리 표현

How - 머신 러닝
 - 안드로이드 앱 개발

Problem

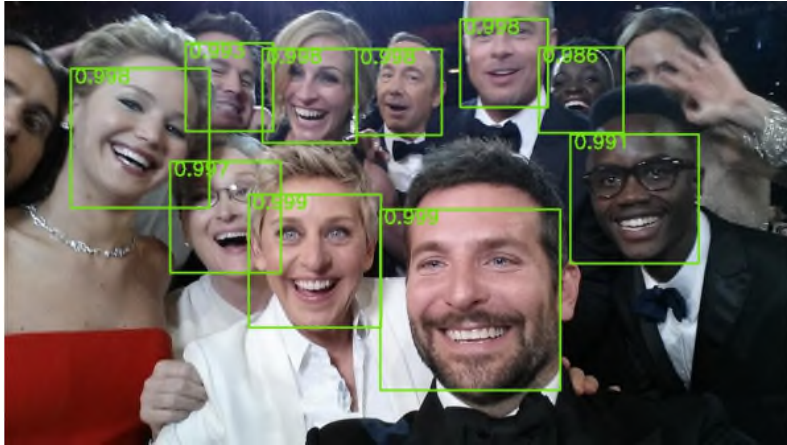


네이버 AR툰 (2016)

보는 콘텐츠에서 **경험하는 콘텐츠**로 발전하였으나

Object가 단순히 한 **고정된 지점에** 위치하여 현실성이 떨어지고 연출에 제약이 있었음

Goal & Requirement



얼굴 인식/반응



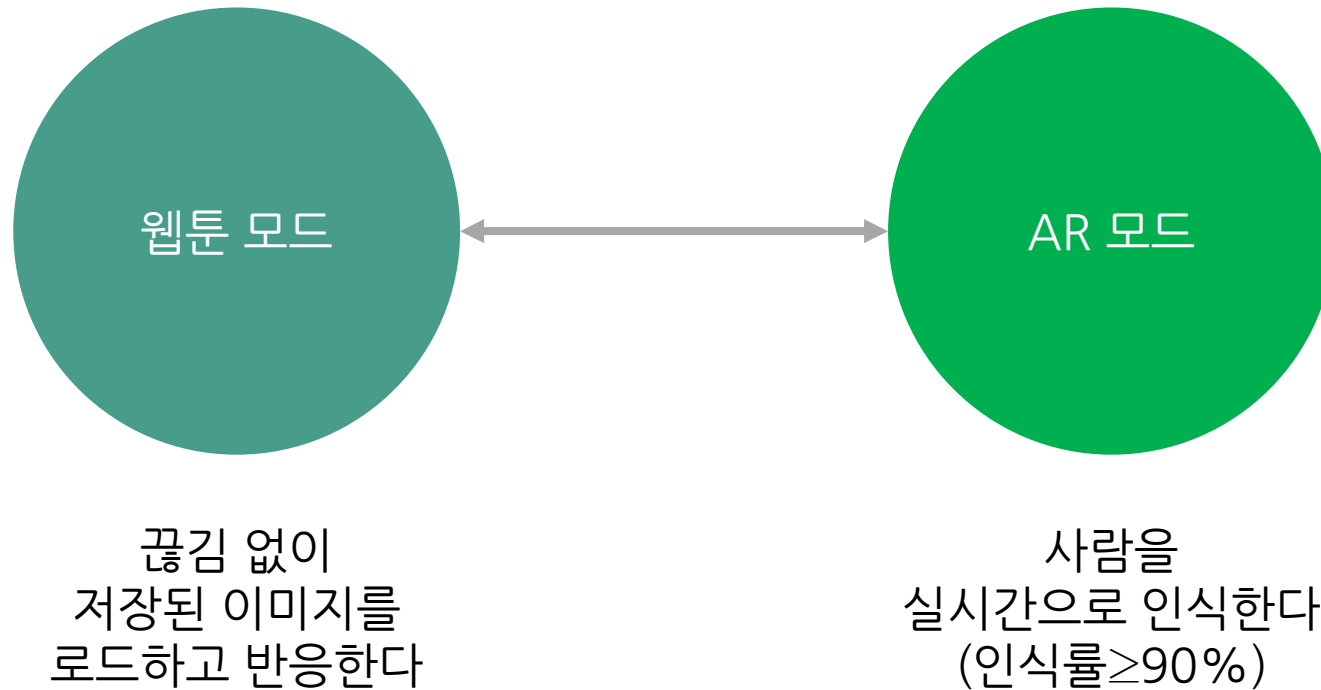
사용자
몰입감 증가



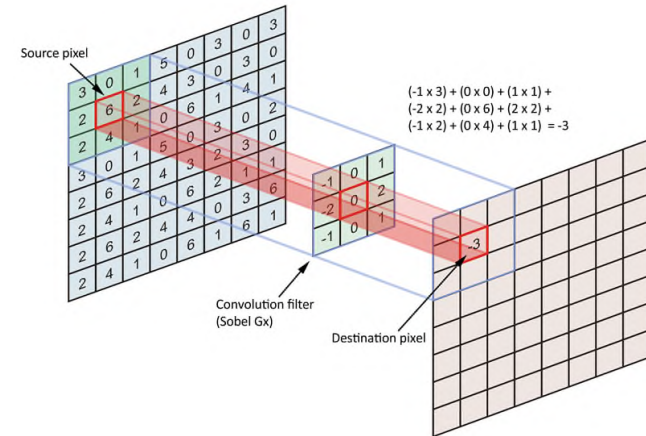
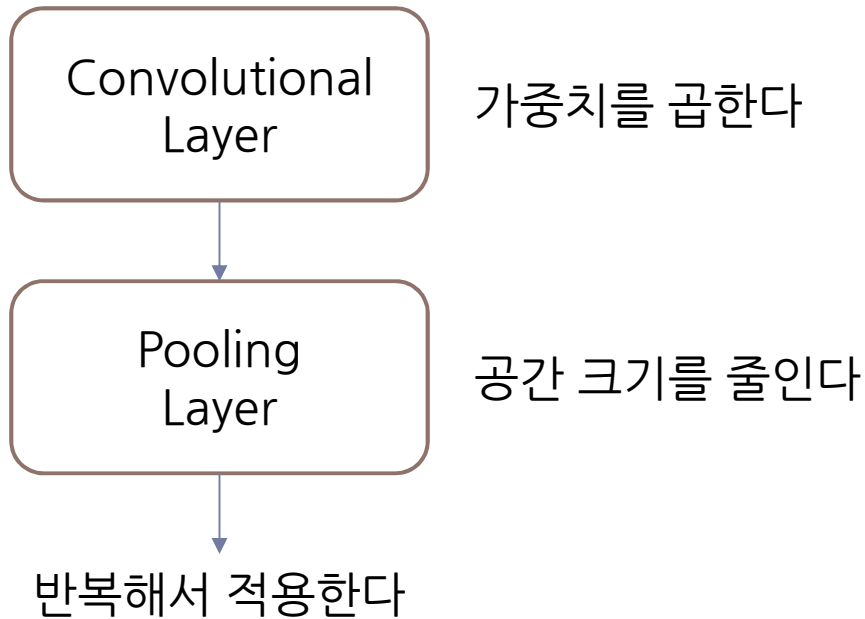
작가
다양한 스토리 전개에 활용

Goal & Requirement

인식한 사람을 기반으로 웹툰 모드/AR 모드 수행



Approach - Convolutional Neural Network



고차원 데이터의 형태 유지
-> 이미지 인식에 효과적

Approach - Transfer Learning

CNN에서 초기 가중치를 random하게 설정할 경우,
많은 dataset과 학습 시간이 요구됨.

1. 선행 연구의 **Network**를 가져와
2. 초기 가중치 또는 Feature extractor로 사용
3. 과제에 맞게 미세 조정



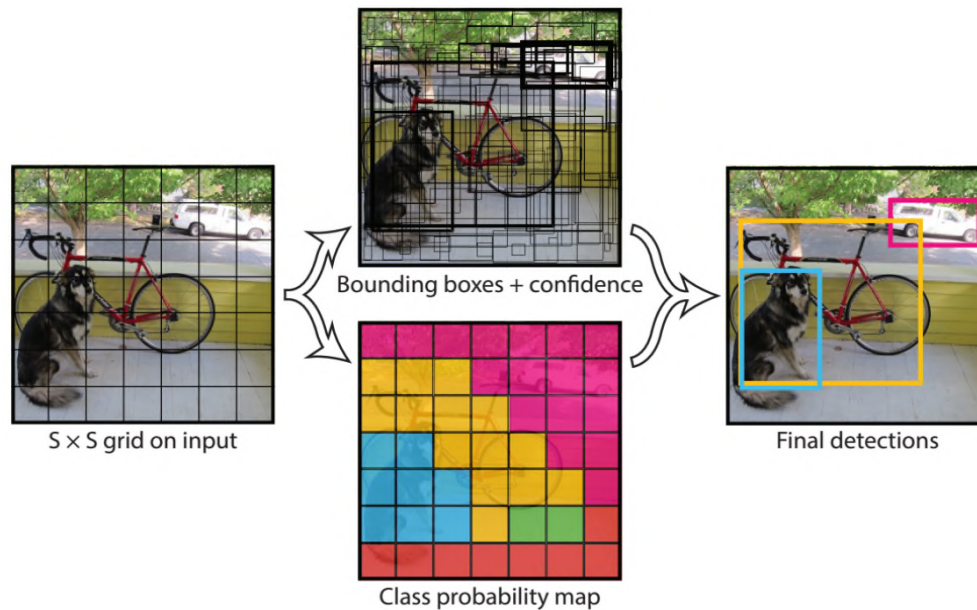
Approach - YOLO: Real-Time Detection

You Only Look Once!

이미지 전체를 한 번만 신경망에 넣는 방법

빠른 인식: 45 frame/sec

-> 실시간으로 정보를 주고 받는 앱 구현에 적합



Approach - YOLO: Real-Time Detection

Class-specific confidence score

$$= \Pr(Class_i | \text{Object}) \cdot \Pr(\text{Object}) \cdot IOU_{pred}^{truth}$$

$$= \Pr(Class_i) \cdot IOU_{pred}^{truth}$$

$\Pr(Class_i)$: box에 특정 class의 object가 있을 확률

IOU_{pred}^{truth} : box가 실제 영역과 정확하게 맞는지

Approach - YOLO: Real-Time Detection

Class-specific confidence score

$$= \Pr(Class_i | \text{Object}) \cdot \Pr(\text{Object}) \cdot \text{IOU}_{pred}^{truth}$$

$$= \Pr(Class_i) \cdot \text{IOU}_{pred}^{truth}$$

$\Pr(Class_i)$: box에 특정 class의 object가 있을 확률

$\text{IOU}_{pred}^{truth}$: box가 실제 영역과 정확하게 맞는지

Approach - Android

ANR

Application Not Responding: 애플리케이션이 반응하지 않음

메모리 낭비 제거

- RecyclerView

Multithreading

- 서버 통신
- Object Detection

Development Environment

Language

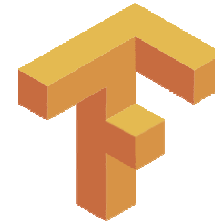


Java
Version 1.8.0



Python
Version 3.6.0

API



TensorFlow
Version 1.0.0

IDE



Android Studio
Version 2.3



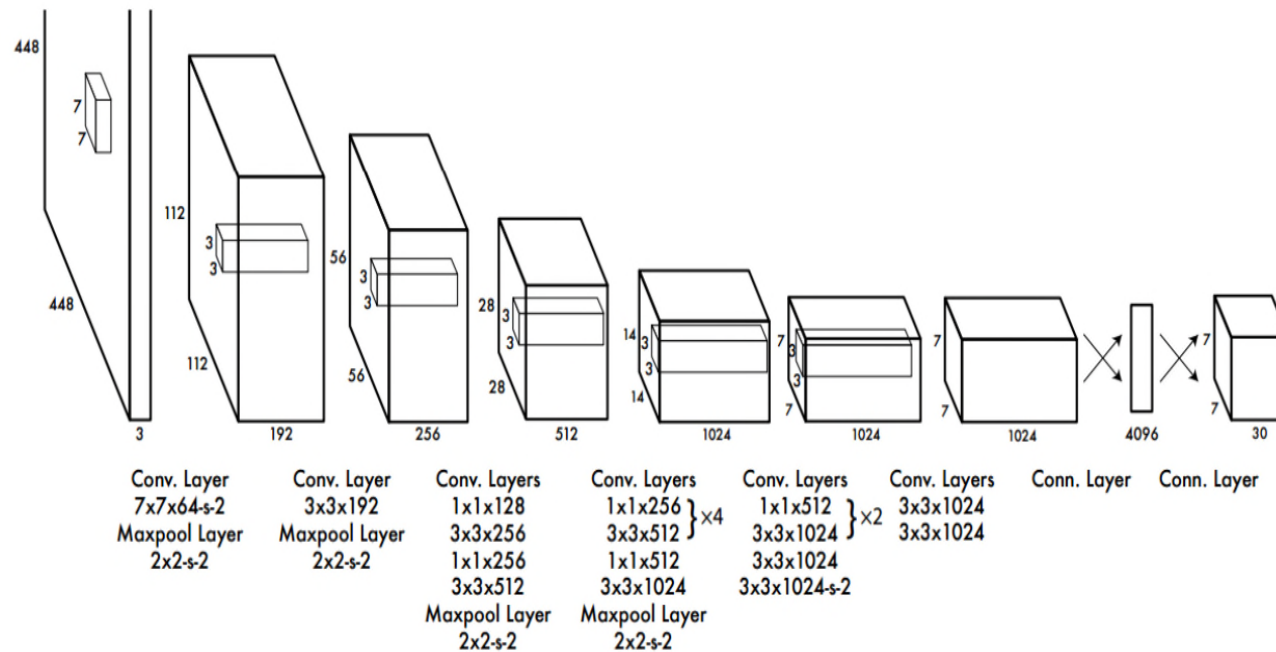
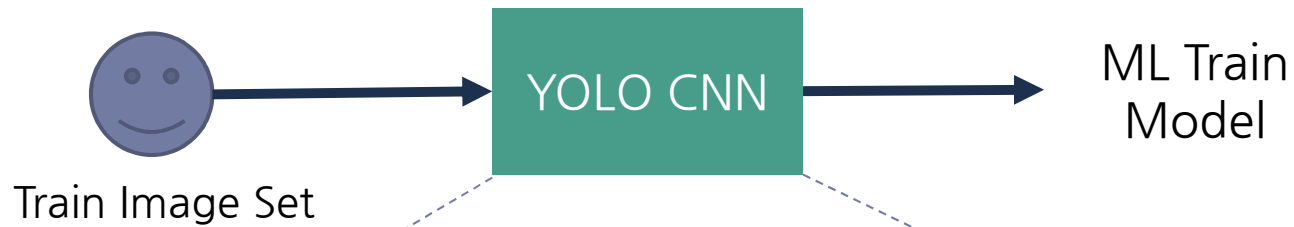
PyCharm
Version 2017.1

OS

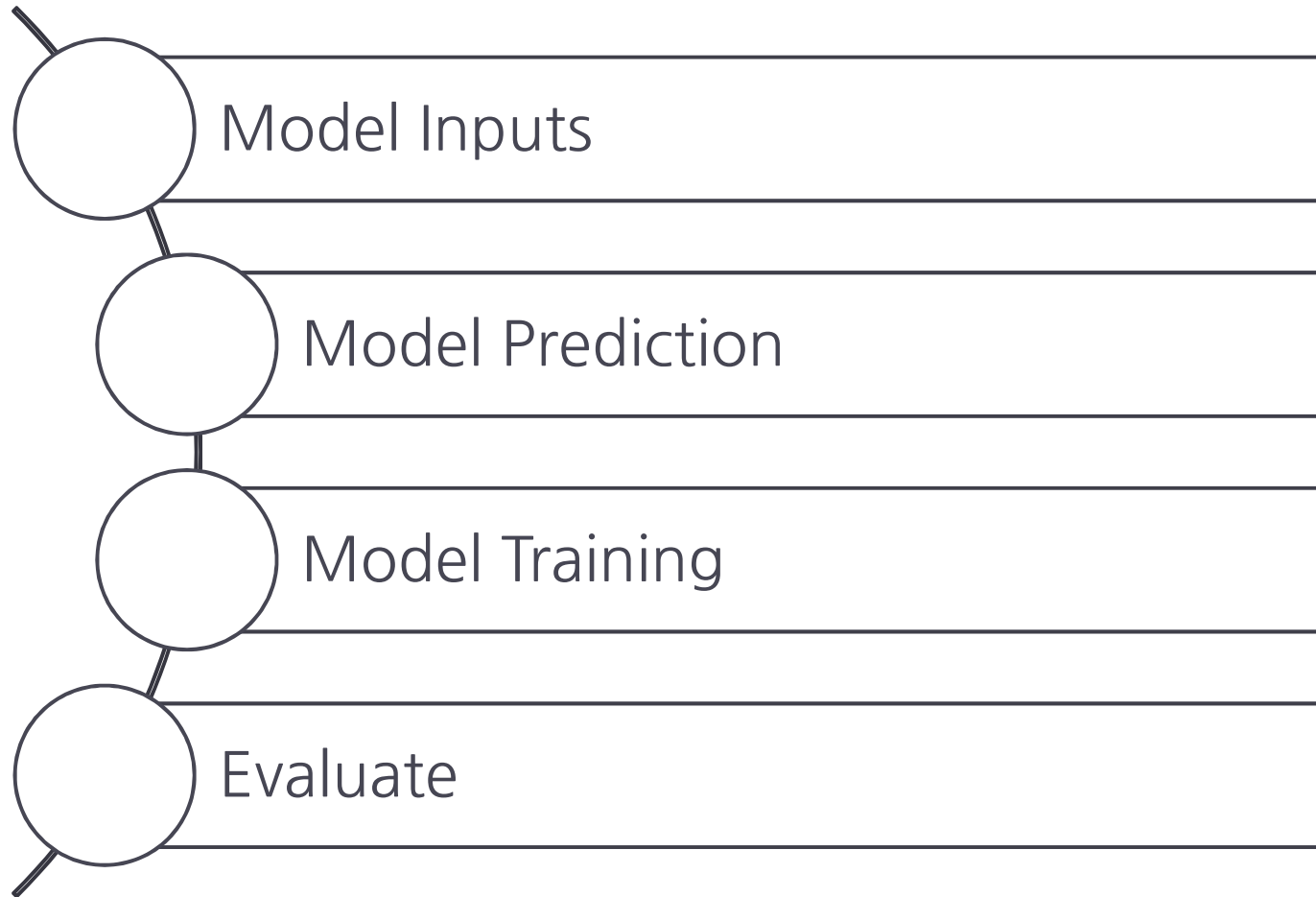


Windows 10

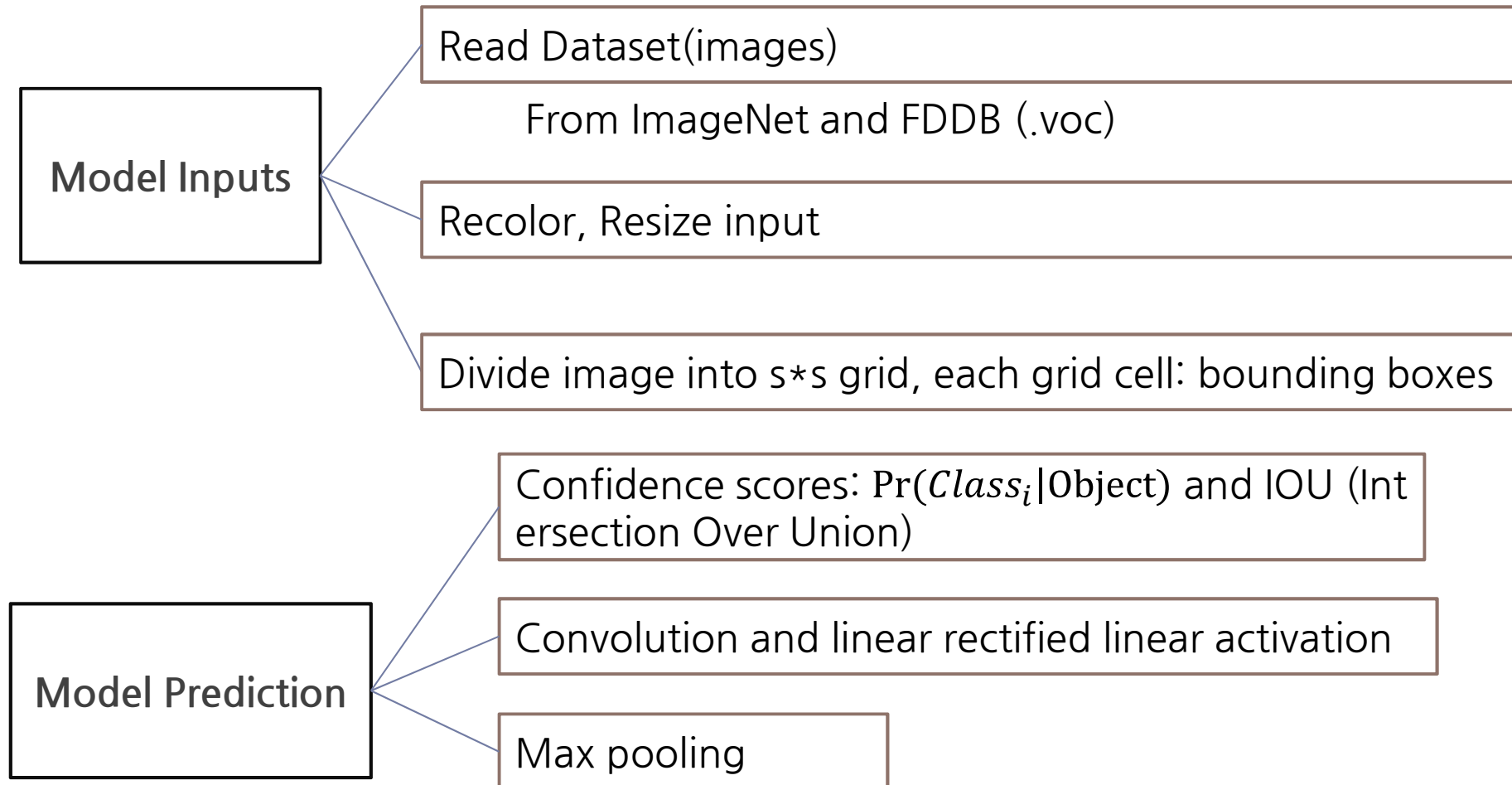
Architecture



Machine Learning Procedure



Machine Learning Procedure



Machine Learning Procedure

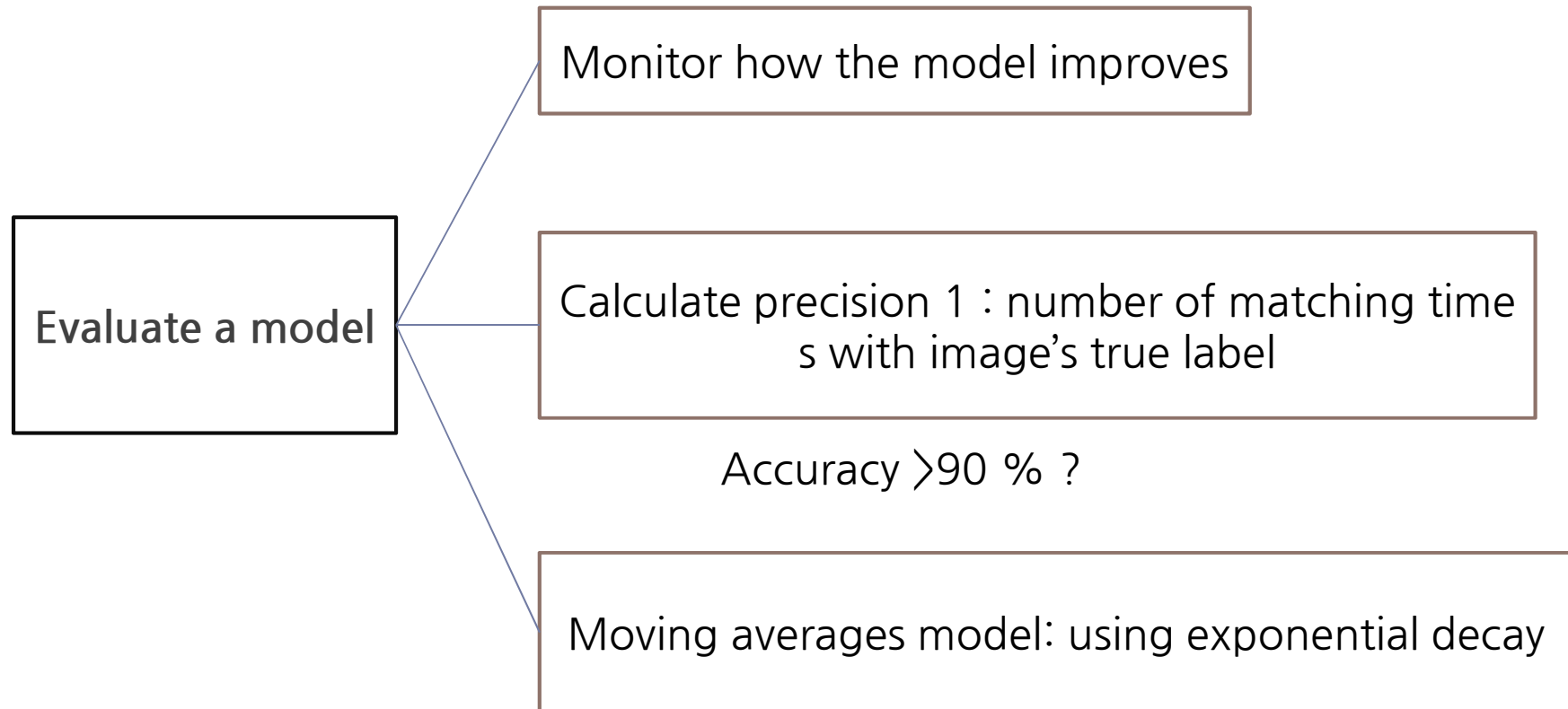
Model training

Apply a multi-part loss function

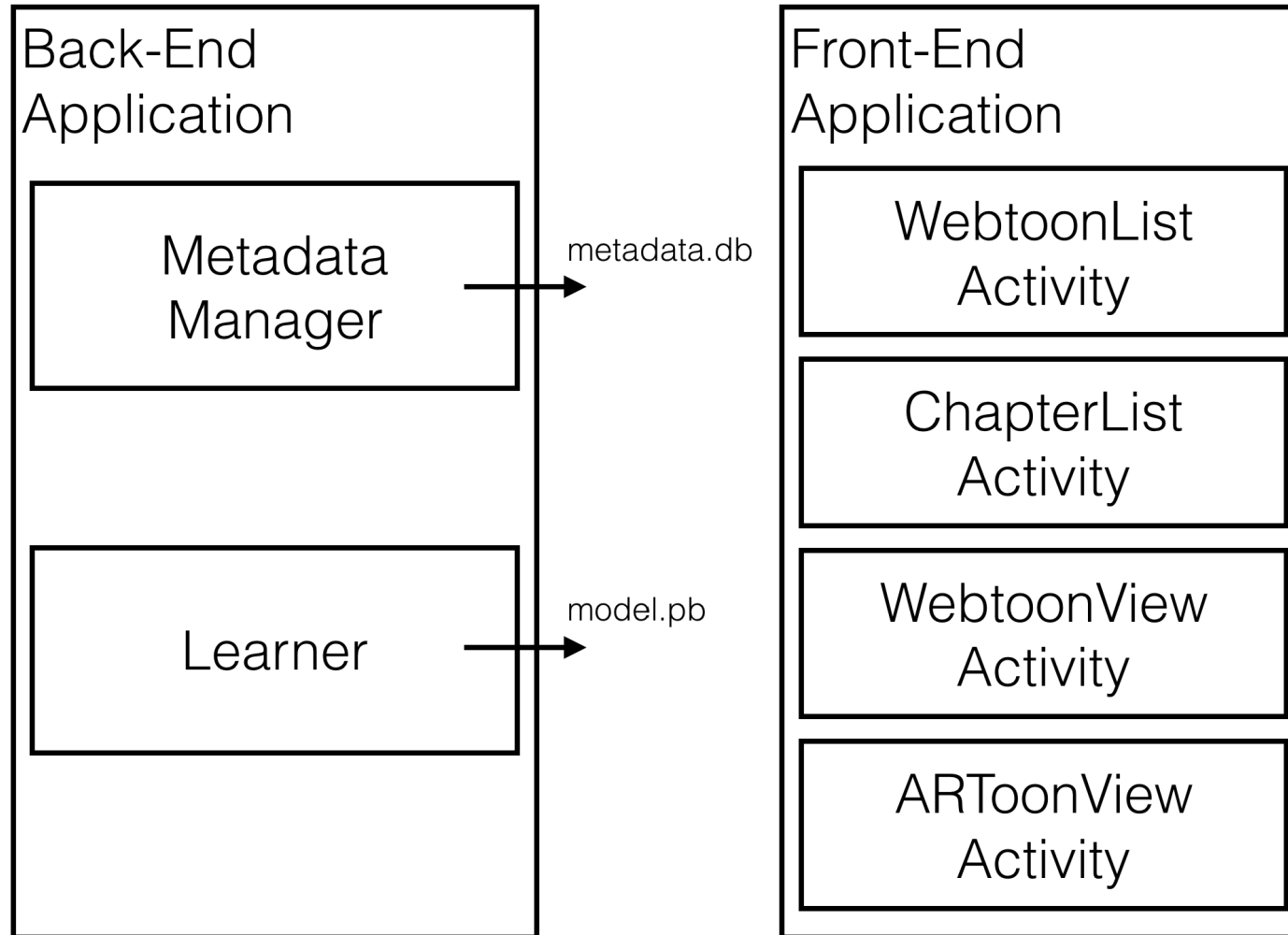
Employ Gradient descent algorithm from Tensorflow

$$\begin{aligned} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \\ & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\ & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \end{aligned}$$

Machine Learning Procedure



Architecture



Architecture

WebtoonListDB

- WebtoonName
- AuthorName
- ThumbnailImage

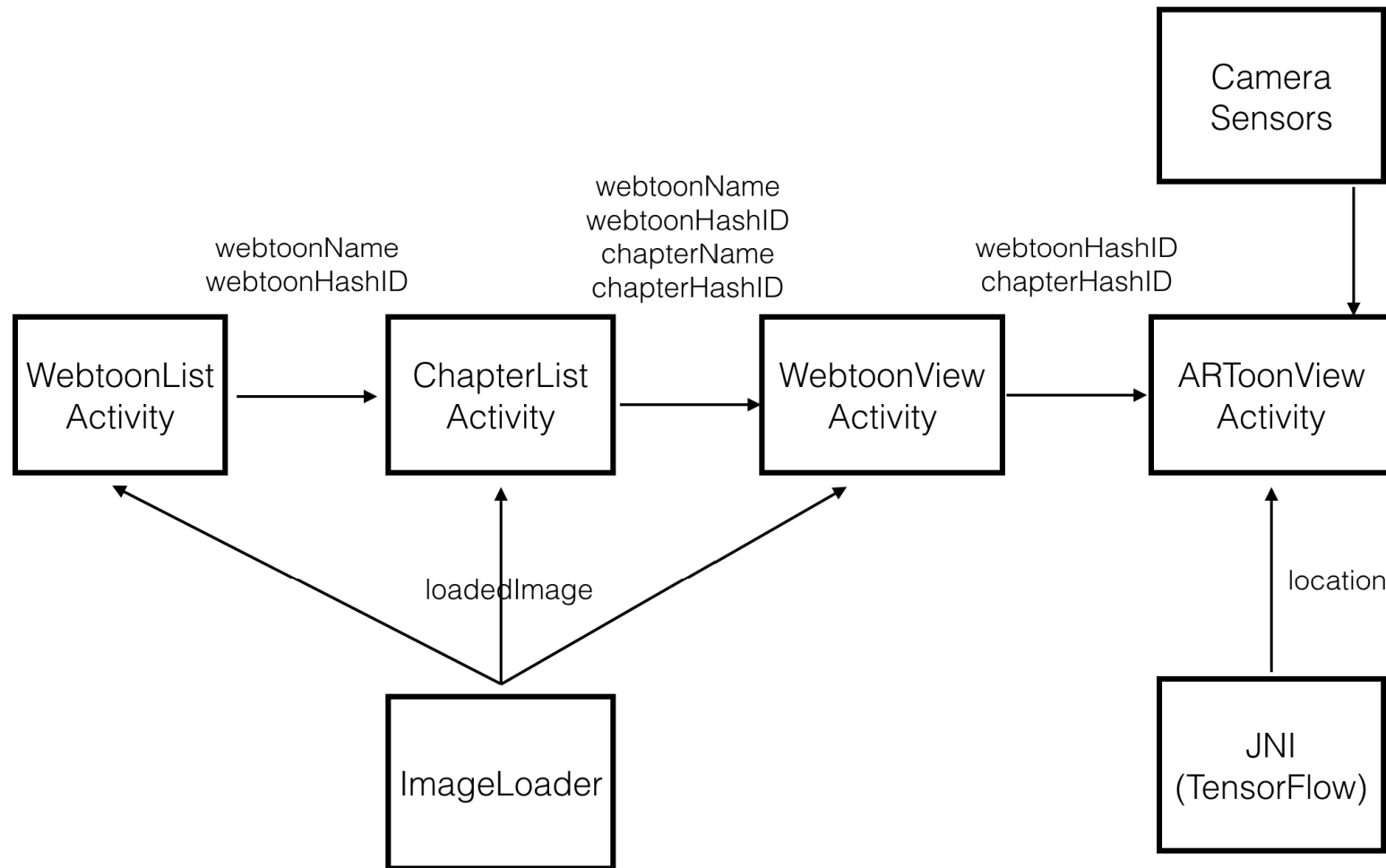
ChapterListDB

- ChapterNumber
- ChapterName
- UploadedDate
- LikeNumber
- DislikeNumber
- ThumbnailImage

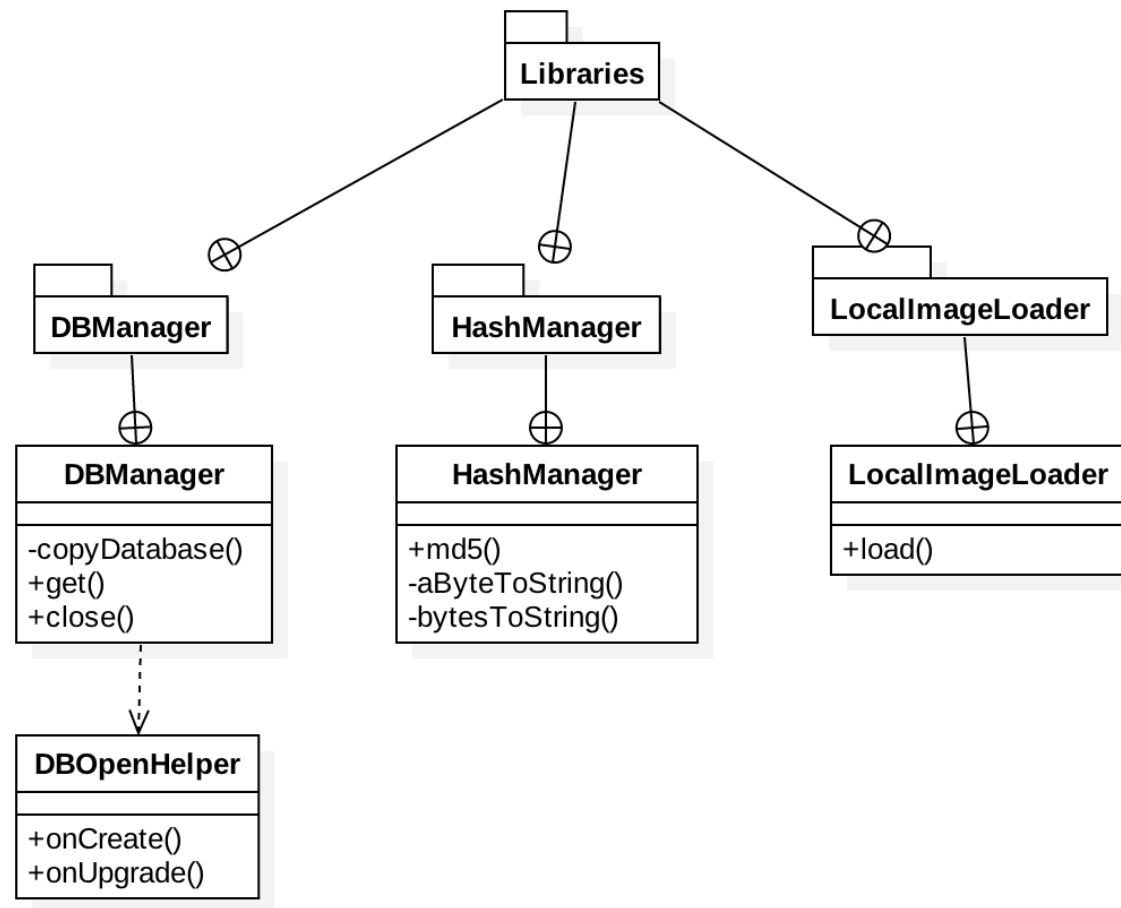
ImageDB

- Image

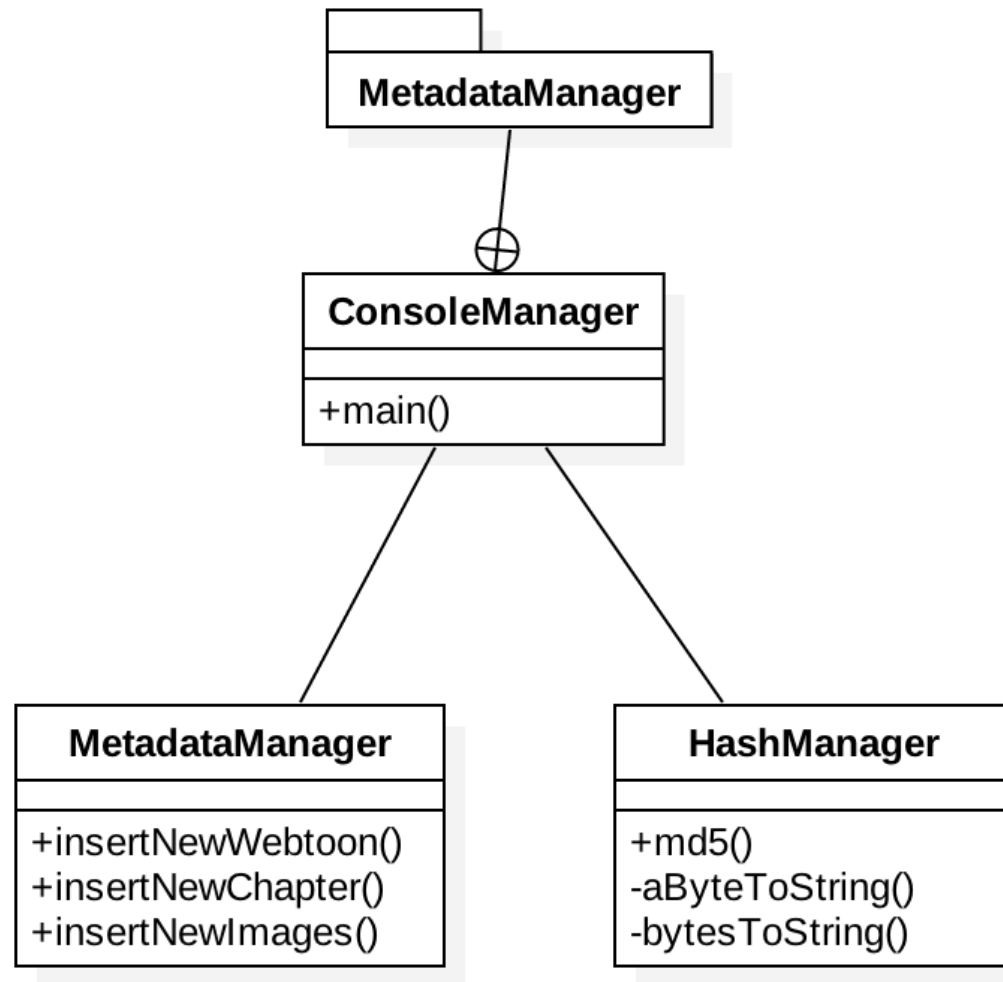
Architecture



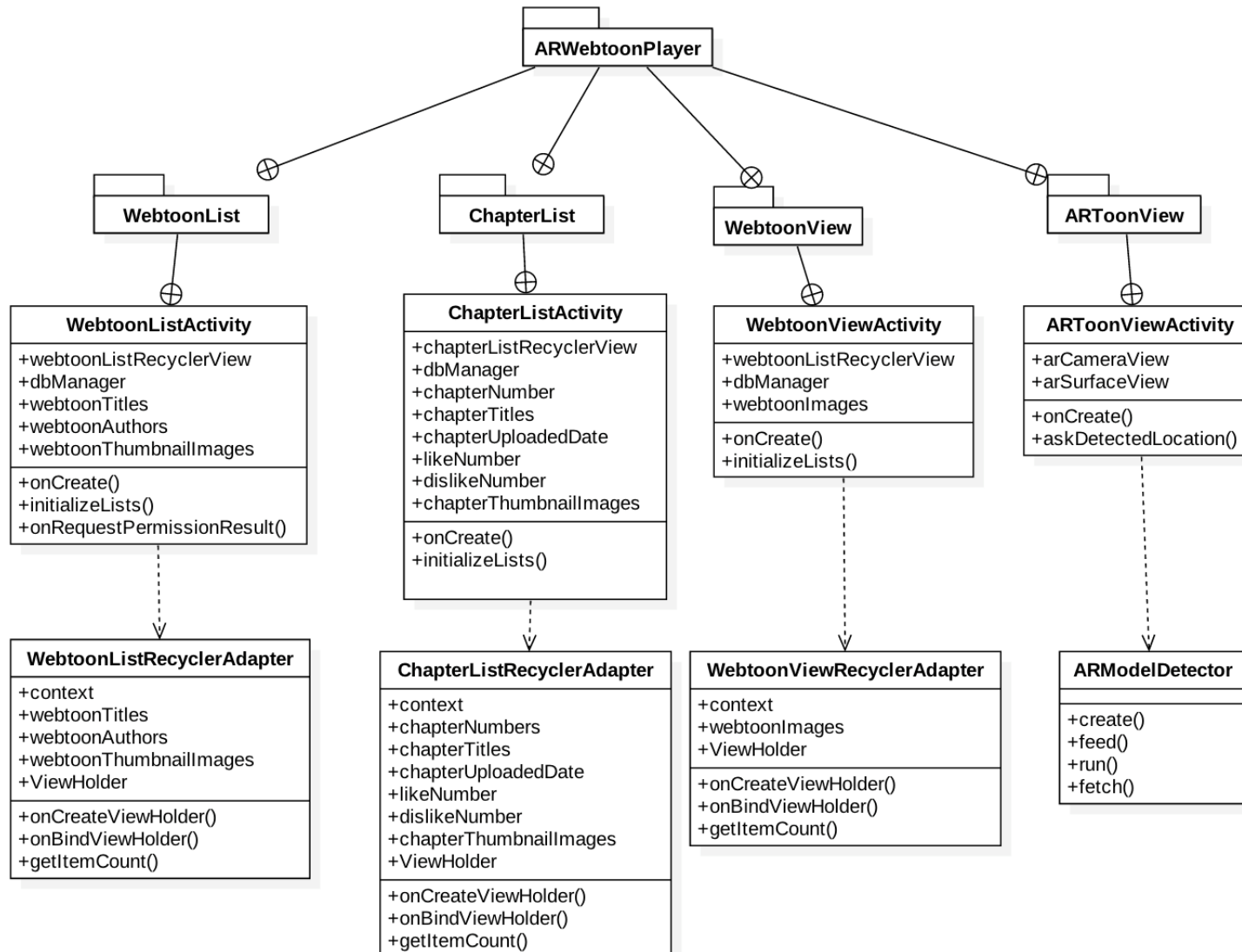
Interface



Interface



Interface



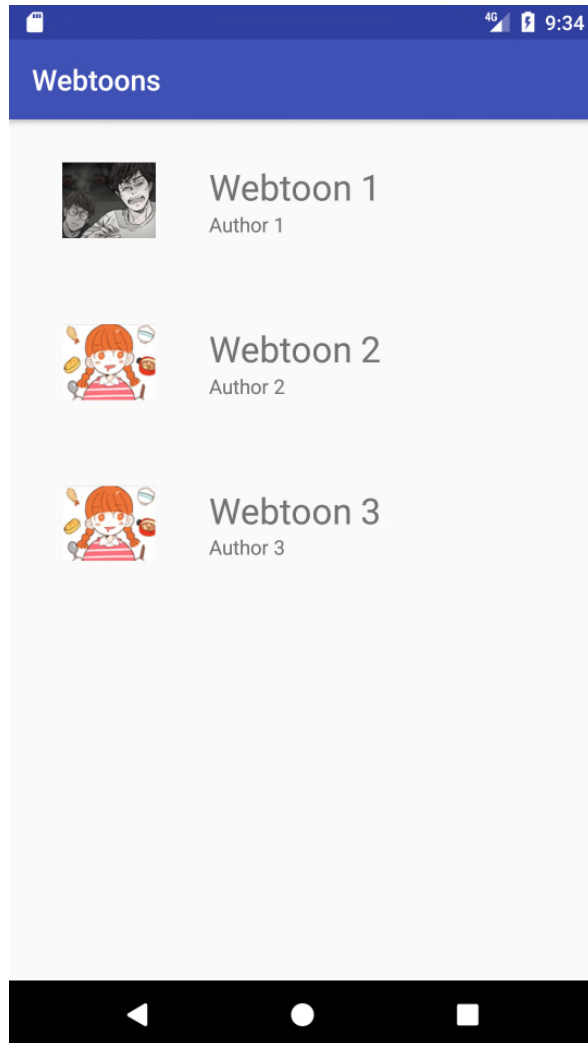
Current Status

Metadata Manager

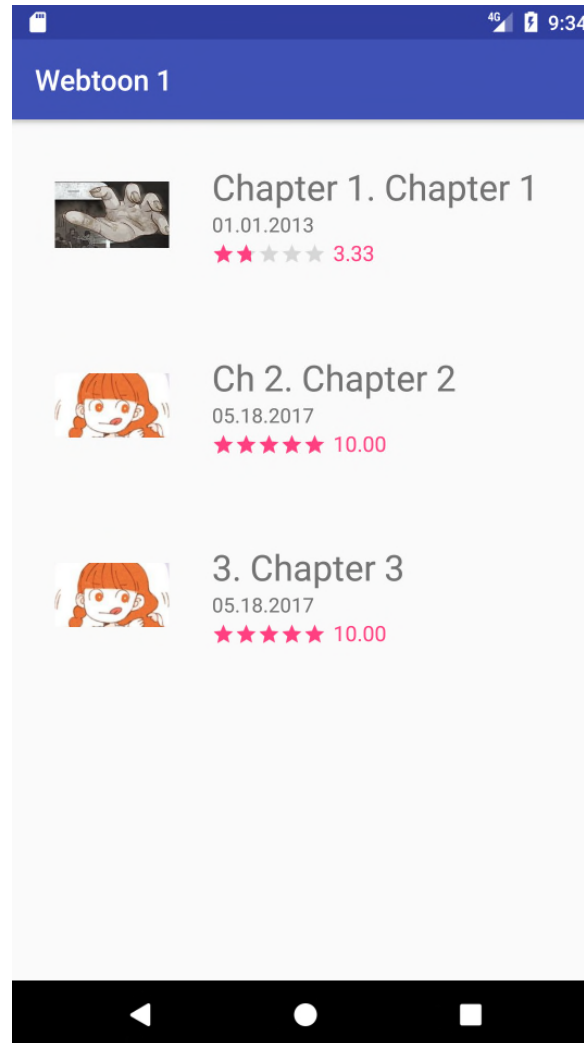
```
1 - Webtoon, 2 - Chapter, 3 - Image, 4 - quit :1
Webtoon name : Webtoon1
Author name : Author1
Thumbnail filename : thumbnail1
1 - Webtoon, 2 - Chapter, 3 - Image, 4 - quit :2
Webtoon name : Webtoon1
Author name : Author1
Chapter number : Ch. 1
Chapter name : Chapter 1
Uploaded date : 05.18.2017
Like Number : 100
Dislike Number : 0
Thumbnail filename : thumbnail
1 - Webtoon, 2 - Chapter, 3 - Image, 4 - quit :4
```

Current Status

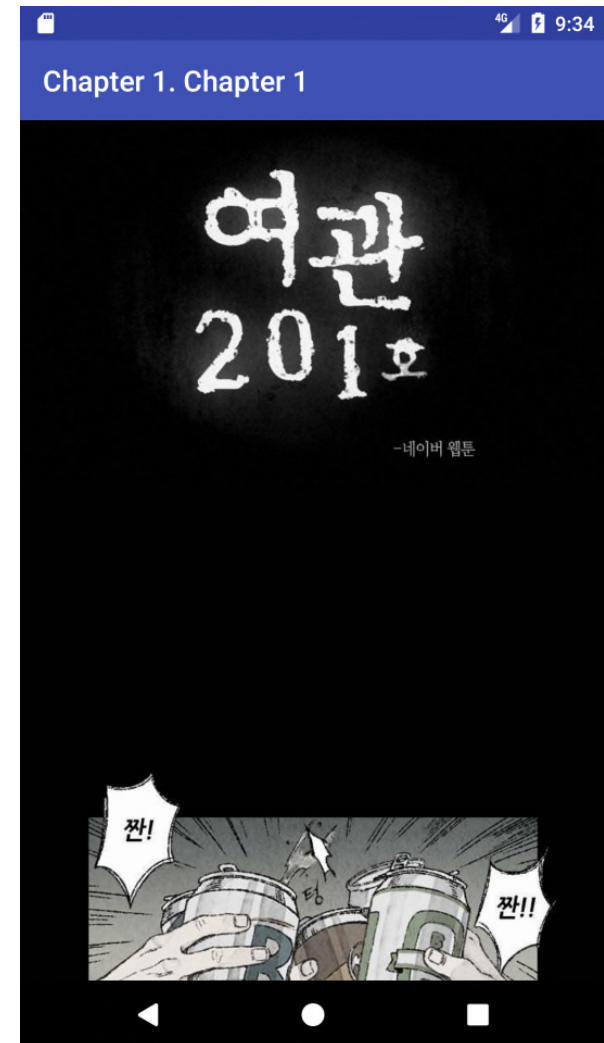
WebtoonList



ChapterList



WebtoonView



Further Plan

1. CNN 모델의 학습을 진행하고 평가한다.

- 학습: 사람 사진과 사진에서 사람이 차지하는 영역을 담은 dataset을 입력 받아 학습한다.
- 평가: 미니 배치를 통해 정확도를 평가한다.

2. 모델의 정확도를 높이기 위해 모델을 개선한다.

- dataset을 늘려서 학습량을 늘린다.
- 드롭아웃: 임의의 연결을 삭제한다. 복잡도를 낮춤으로써 overfitting을 피할 수 있어, 학습되지 않은 데이터에도 잘 적용될 수 있는 모델을 만든다.

Further Plan

3. ARToonView Activity를 구현한다

- OpenGL, SurfaceView, Camera, Sensors 사용
- 평가: 미니 배치를 통해 정확도를 평가한다.

4. 데이터를 로컬에서 서버로 옮긴다.

- HTTP 프로토콜 사용
- Multithreading

5. 애플리케이션 기능 구현

- 임시저장
- 캡처
- 화면 회전 잠금

Demo Plan (항목 별)

- ▶ 프로젝트의 apk가 설치되어 있는 안드로이드 디바이스를 준비한다. 디바이스가 서버에서 웹툰을 가져올 수 있도록 인터넷을 연결해 놓는다.
- ▶ 안드로이드의 화면을 컴퓨터 화면에 미러링하는 모비즌이나 크롬캐스트를 세팅한다.
- ▶ 웹툰 모드에서의 기능들(로딩, 스크롤링 등)을 시연한다.
- ▶ AR 모드에서 얼굴 인식을 시연한다.



Demo Plan (절차 순)

1. 앱을 실행한다.
2. 캡처나 화면 회전 잠금 등 웹툰 앱의 기능을 수행한다.
3. 서버에서의 웹툰 로딩 및 스크롤링이 끊기지 않고 이루어짐을 시연한다.
4. AR 모드로 진입하여 발표자의 얼굴을 촬영했을 때 화면에서 발표자가 잘 인식됨을 보인다.
5. 인식한 발표자 주변에 그래픽이 위치함을 확인한다.
6. 다른 사람에게도 시연하여, 모든 사람을 인식할 수 있는 범용성을 확인한다.

Division and Assignment of Work

항목	담당자
머신 러닝 및 TensorFlow - 머신 러닝 모델 개발 - Back-end Learner 개발 - Detector Library 개발 - TensorFlow Interface 개발	강민지 핍언
안드로이드 및 데이터베이스, 그래픽스 - Front-end Application 개발 - Back-end 데이터베이스, 라이브러리 개발	원종훈

각자의 분야를 중심으로 하나의 앱을 구현

Schedule

내용	3월			4월				5월				6월		
	2	3	4	1	2	3	4	1	2	3	4	1	2	
Tensorflow basics														
Android app knowledge														
CNN basics														
Database knowledge														
Graphics														
Debugging														
CNN Model Improvement														

감사합니다

강민지	kyang@snu.ac.kr
원종훈	barber@snu.ac.kr
핍언	pimonink_ja@hotmail.com

