

# 사진 자동 분류 및 랭킹 서비스 개발



캡스톤 다반 - 사연있는 사람들

팀장 : 박민준

팀원 : 김상엽 / 박정준 / 임규빈 / 정창도

# 문제 인식

간편한 사진 촬영



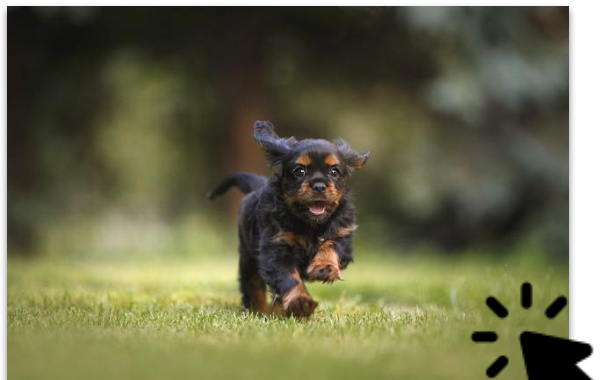
내 갤러리 속  
무수히 많은  
사진들



원하는 사진을  
찾는데 오래 걸림



# 목표



사람들과의 사진  
공유 및 평가



자동 분류



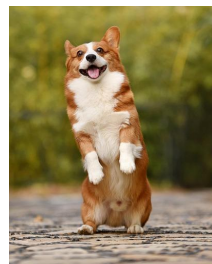
♥ 4.5



♥ 4.4



♥ 4.1



같은 분류끼리의 평가를 통한 순위 매기기 3

# 설계

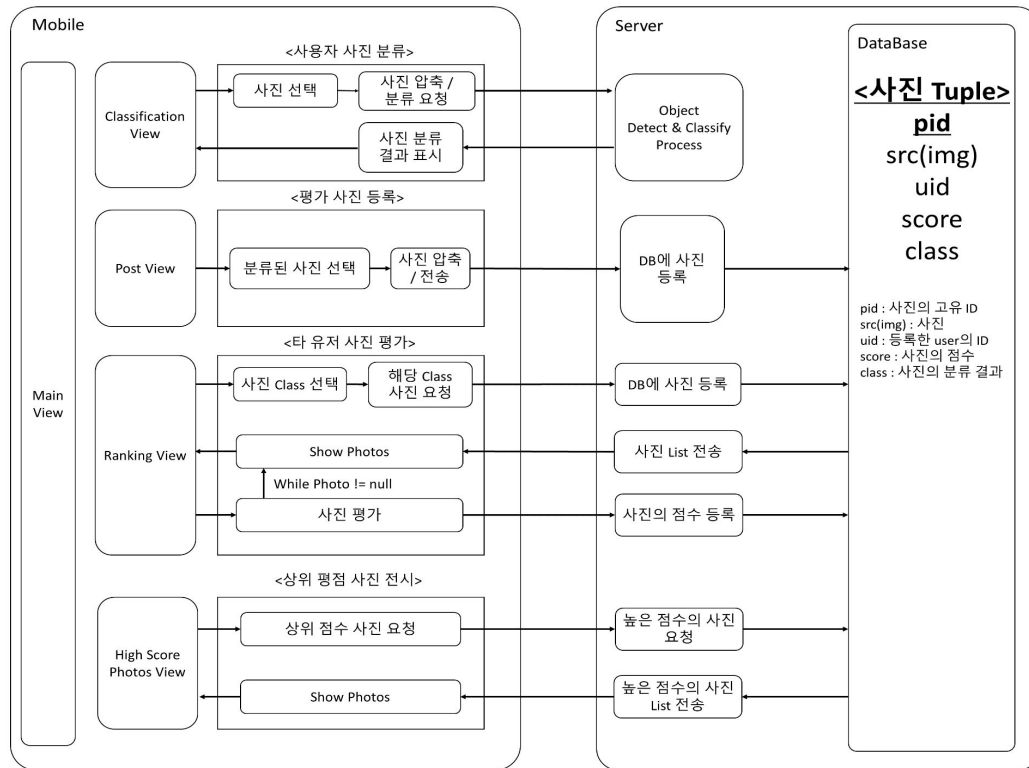
---

## 프로세스 설계 - 주요 기능

- 01 사용자 사진 분류
- 02 평가 사진 등록
- 03 타 유저 사진 평가
- 04 상위 평점 사진 전시

# 설계

## 프로세스 설계 - 구조도



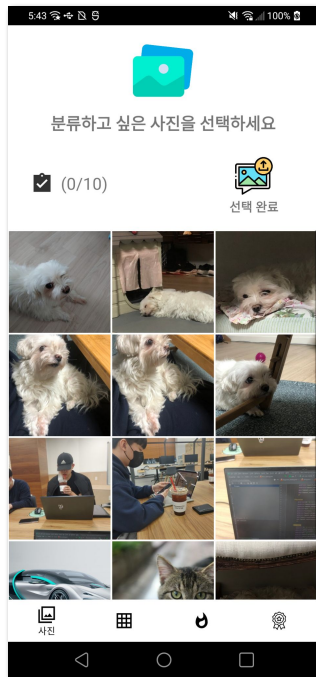
# 설계

## 프로세스 설계 - API

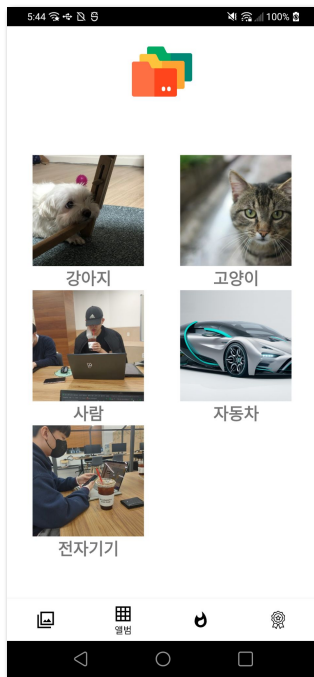
			KEY	VALUE	RESPONSE
<사용자 사진 분류>	사진 압축 / 분류 요청	POST	List<file>	사진	List<lable>
			KEY	VALUE	RESPONSE
<평가 사진 등록>	사진 압축 / 전송	POST	file	사진	200,500,404
			uid	""	
			label	"강아지"	
			REQUEST		RESPONSE
<타 유저 사진 평가>	해당 Label 사진 요청	GET	label	"강아지"	file
			uid	""	score_average
			page	???	pid
			KEY	VALUE	RESPONSE
	사진 평가	POST	pid	""	200,500,404
			uid	""	
			score_evaluate		4
			REQUEST		RESPONSE
<상위 평점 사진 전시>	상위 점수 사진 요청	GET	label	"강아지"	file
					score_average

# 인터페이스 설계

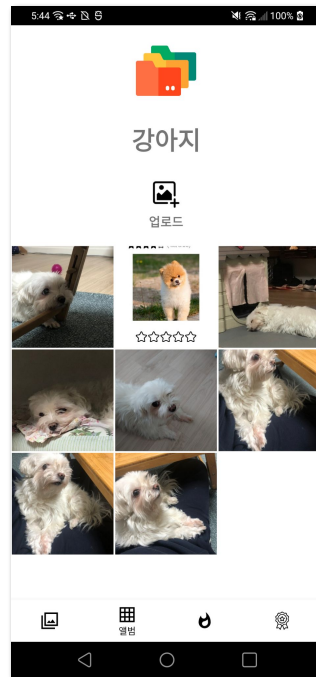
사용자 갤러리  
화면



분류된 사진 확인,  
평가 사진 등록  
화면

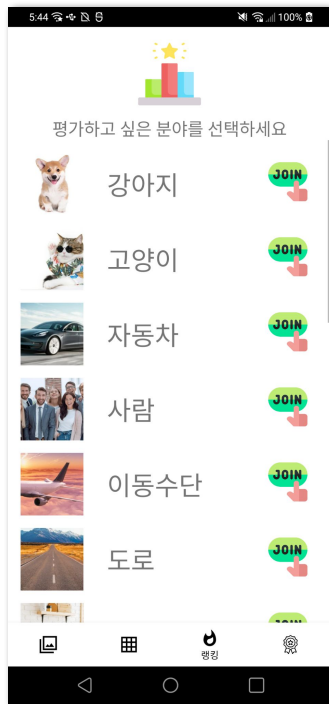


분류된 사진  
갤러리

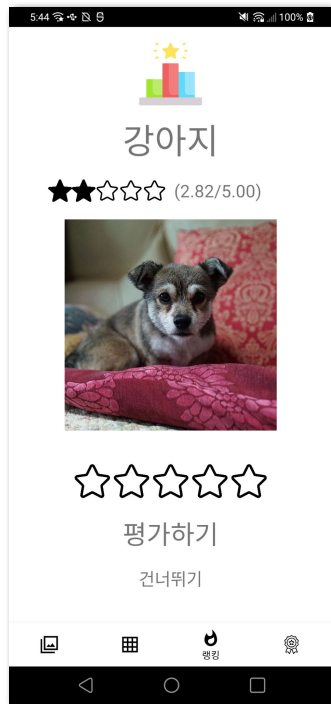


# 인터페이스 설계

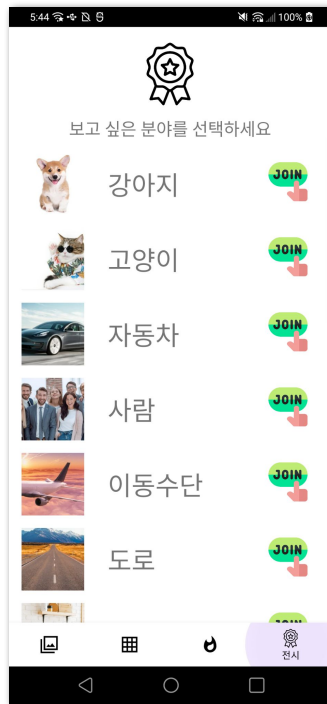
사진 평가  
라벨 화면



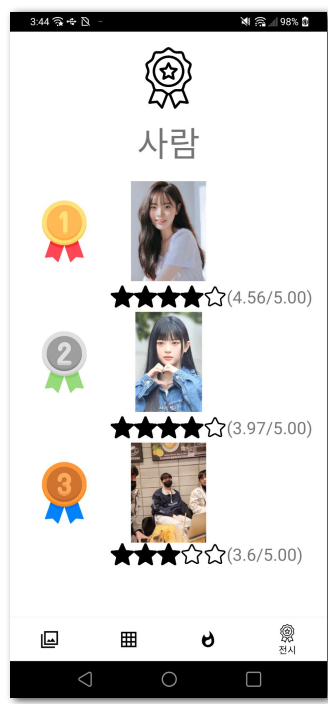
타 유저 사진  
평가 화면



상위 평점사진  
전시 화면



상위 평점사진  
전시 화면





# 세부 기술사항 - 기술 구현

## Object Detect & Classify Process

- Detect Process

Yolo V3 model

학습을 진행하지 않고,  
weight 파일을 활용한 객체 검출

OpenCV를 활용해 GPU가  
지원되지 않는 AWS EC2  
환경에서도 Object Detection 가능

### <Yolo V3 model 로 검출 가능한 객체 리스트>

사람	이동수단	도로	주방도구	가방	도구	식품	운동기구	가구	전자기기	동물	강아지	고양이	자동차	새	기타
person	bicycle	traffic light	bottle	backpack	umbrella	banana	frisbee	chair	laptop	horse	dog	cat	car	bird	none
		fire hydrant	wine glass	handbag	clock	apple	skis	sofa	mouse	sheep					
	motorbike	stop sign	cup	suitcase	vase	sandwich	snowboard	pottedplant	remote	cow					
	aeroplane	parking meter	fork		scissors	orange	sports ball	bed	keyboard	elephant					
	bus	bench	knife		teddy bear	broccoli	baseball bat	diningtable	cell phone						
	train		spoon		hair drier	carrot	baseball glove	toilet		zebra					
	truck		bowl		toothbrush	hot dog	skateboard	tvmonitor		giraffe					
	boat				tie	pizza	surfboard	microwave							
					kite	donut	tennis racket	oven							
					book	cake		toaster							
								sink							
								refrigerator							

총 80개의 Object 를 16개의 Main Classes 로 분류

# 세부 기술사항 - 기술 구현

## Object Detect & Classify Process

- Classify Process

다양한 객체가 검출된 경우 Main Class를 알아낼 필요가 있음.

직관 : **가운데** 있을 수록 Main Class가 될 확률이 높을것이다.

### <Centerness weight 적용>

$$\text{centerness}^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}}$$

검출된 객체의 정중앙을 기준으로 잡고,

왼쪽테두리 - 기준점 = l

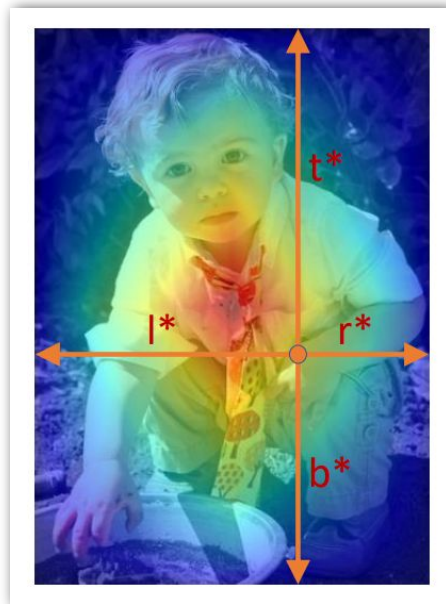
오른쪽테두리 - 기준점 = r

상단테두리 - 기준점 = t

하단테두리 - 기준점 = b

로 변수를 지정하여 centerness 를 계산

Centerness 가 최대인 객체가 해당 사진의 Main Class



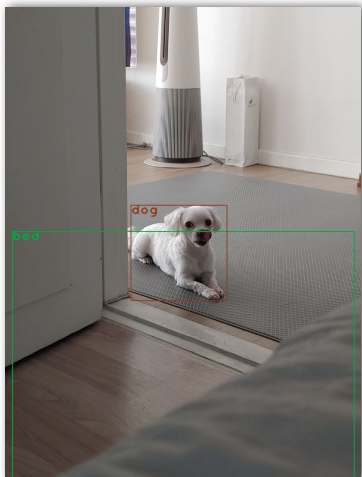
# 세부 기술사항 - 기술 구현

## Object Detect & Classify Process



PhotoCup

AI를 통한 자동분류



객체를 Detect 한 결과물

$$\text{centerness}^* = \sqrt{\frac{\min(l^*, r^*)}{\max(l^*, r^*)} \times \frac{\min(t^*, b^*)}{\max(t^*, b^*)}}$$

Centerness 산출 수식

```
ubuntu@ip-172-31-61-100:~/opencv$ python3 yolo.py
0.9644796368481325
this centerness is best : 0.9644796368481325
centerness : [0.9644796368481325]
0.9644796368481325
centerness : [0.9644796368481325, 0.5950146734150564]
ubuntu@ip-172-31-61-100:~/opencv$
```

Centerness 결과



Classify한 결과물

# 세부 기술사항 - 성능 평가

## Detect 과정 정확도 측정

- 셔터스톡 사이트 내 15가지의 카테고리로 분류된 사진을 각 카테고리별 50장 씩 가져와 Detect 성능 평가를 진행

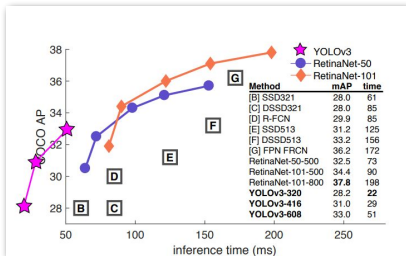
### <테스트 과정>

테스트 사진 : 셔터스톡 사이트 내 카테고리 검색

표본 : 카테고리별 50장 + 속하지 않은 50장

결과 : mAP = 0.123

### <YoloV3 성능 비교>



```
ubuntu@ip-172-31-61-100:~/TestImg$ python3 mAP_animal.py
-----confidence threshold : 0.1-----
precision : 1.0
recall : 0.76
-----confidence threshold : 0.2-----
precision : 1.0
recall : 0.76
-----confidence threshold : 0.3-----
precision : 1.0
recall : 0.76
-----confidence threshold : 0.4-----
precision : 1.0
recall : 0.76
-----confidence threshold : 0.5-----
precision : 1.0
recall : 0.76
-----confidence threshold : 0.6-----
precision : 1.0
recall : 0.76
-----confidence threshold : 0.7-----
precision : 1.0
recall : 0.76
-----confidence threshold : 0.8-----
precision : 1.0
recall : 0.62
-----confidence threshold : 0.9-----
precision : 1.0
recall : 0.52
ubuntu@ip-172-31-61-100:~/TestImg$
```

```
YoloTest
> animal
> bag
> bear
> bird
> car
> cat
> dog
> electronic
> food
> furniture
> gym
> kitchen
> people
> road
> tools
> vehicle
```

# 세부 기술사항 - 성능 평가

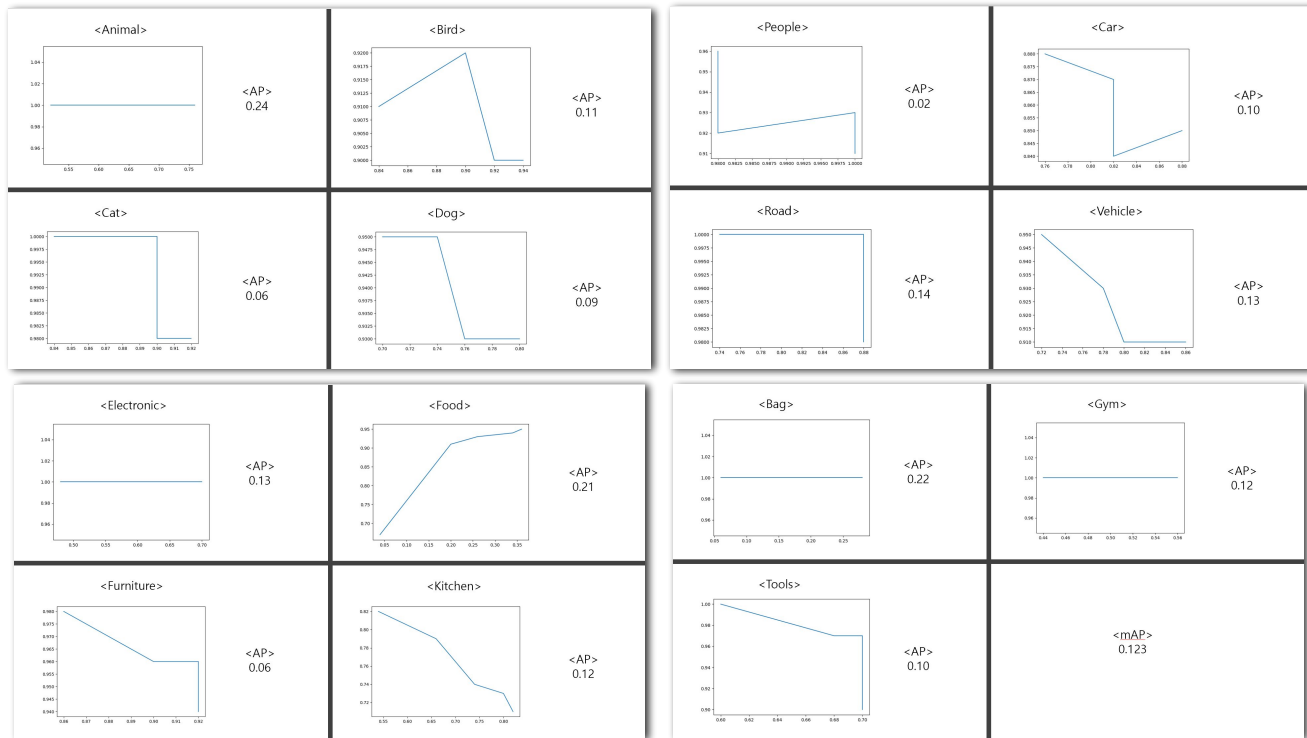
## Detect 과정 카테고리별 성능 평가

### <성능 평가 분석 및 피드백>

YoloV3 자체 성능보다  
분류 성능 하락

각 객체의 카테고리화,  
성능 평가 사진 등의  
문제들로 예상

더 나은 Object Detection  
성능을 보유한 모델로  
교체 함에 따라 해결할  
수 있을것으로 보임



# 세부 기술사항 - 성능 평가

## Classify 과정 성능 평가

### <테스트 과정>

테스트 사진 : 셔터스톡 사이트 내 복수 객체 사진 검색

정답 : 외부 인원의 메인 객체 평가

결과 : 72% Accuracy

---

### <성능 평가 분석 및 피드백>

오검출한 사진 분석

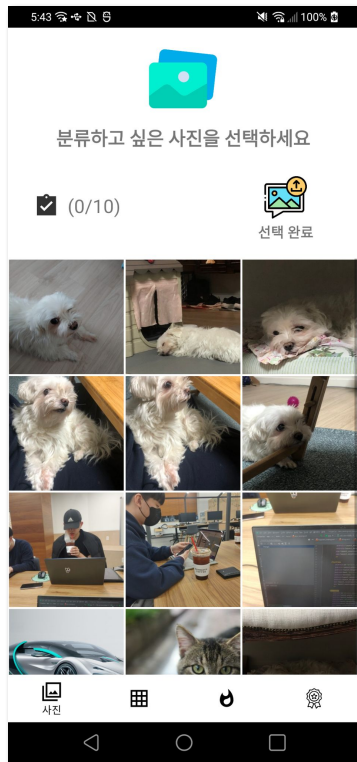
타 객체가 중앙에 위치한 경우 오류 발생

centerness 뿐만 아니라 검출된 객체의 크기 비율에 따른  
메인 객체 점수 증가폭을 달리하여 해결 가능

```
ubuntu@ip-172-31-61-100:~/TestImg$ python3 test_centerness.py
2 : furniture :: people
7 : gym :: people
11 : vehicle :: people
31 : people :: dog
33 : people :: dog
36 : people :: dog
41 : people :: dog
42 : kitchen :: dog
47 : people :: dog
54 : people :: dog
61 : car :: cat
65 : car :: cat
66 : furniture :: cat
69 : furniture :: cat
70 : bag :: cat
72 : tools :: cat
73 : people :: cat
75 : bird :: cat
76 : people :: cat
79 : kitchen :: cat
85 : kitchen :: tools
87 : people :: tools
88 : people :: tools
95 : people :: bag
96 : people :: bag
97 : people :: bag
98 : people :: bag
99 : electronic :: bag
accuracy : 0.72
ubuntu@ip-172-31-61-100:~/TestImg$
```



# 시연 영상



# Q&A

