

# PACK

보 고 싶 다 엄 강 우 조

# MAN

강창현 이지은 성열호 김성중 권하늘



# CONTENTS

---

01 AI

02 Front-End

03 Back-End



# 01 | AI

- 01. 기술 선정
- 02. 모델 선정
- 03. 학습 및 인식
- 04. 개선 사항

# 01 AI

## 01 기술 선정

02. 모델 선정

03. 학습 및 인식

04. 개선 사항

# 기술 선정



“준비한 **물품의 사진을 찍어 인식**한 후 빠진 물품을 확인”



객체 인식  
(Object Detection)

# 01 AI

01. 기술 선정

02 모델 선정

03. 학습 및 인식

04. 개선 사항

## 모델 선정

SSD

(Single Shot Detection)

비교적 빠른 속도

비교적 높은 정확도

사용하기 어려움

YOLO

(You Only Look Once)

가장 빠른 속도

비교적 낮은 정확도

Faster  
R-CNN

가장 높은 정확도

느린 속도

# 01 AI

01. 기술 선정

02. 모델 선정

03 학습 및 인식

04. 개선 사항

## 학습 및 인식 데이터 수집



# 01 AI

01. 기술 선정

02. 모델 선정

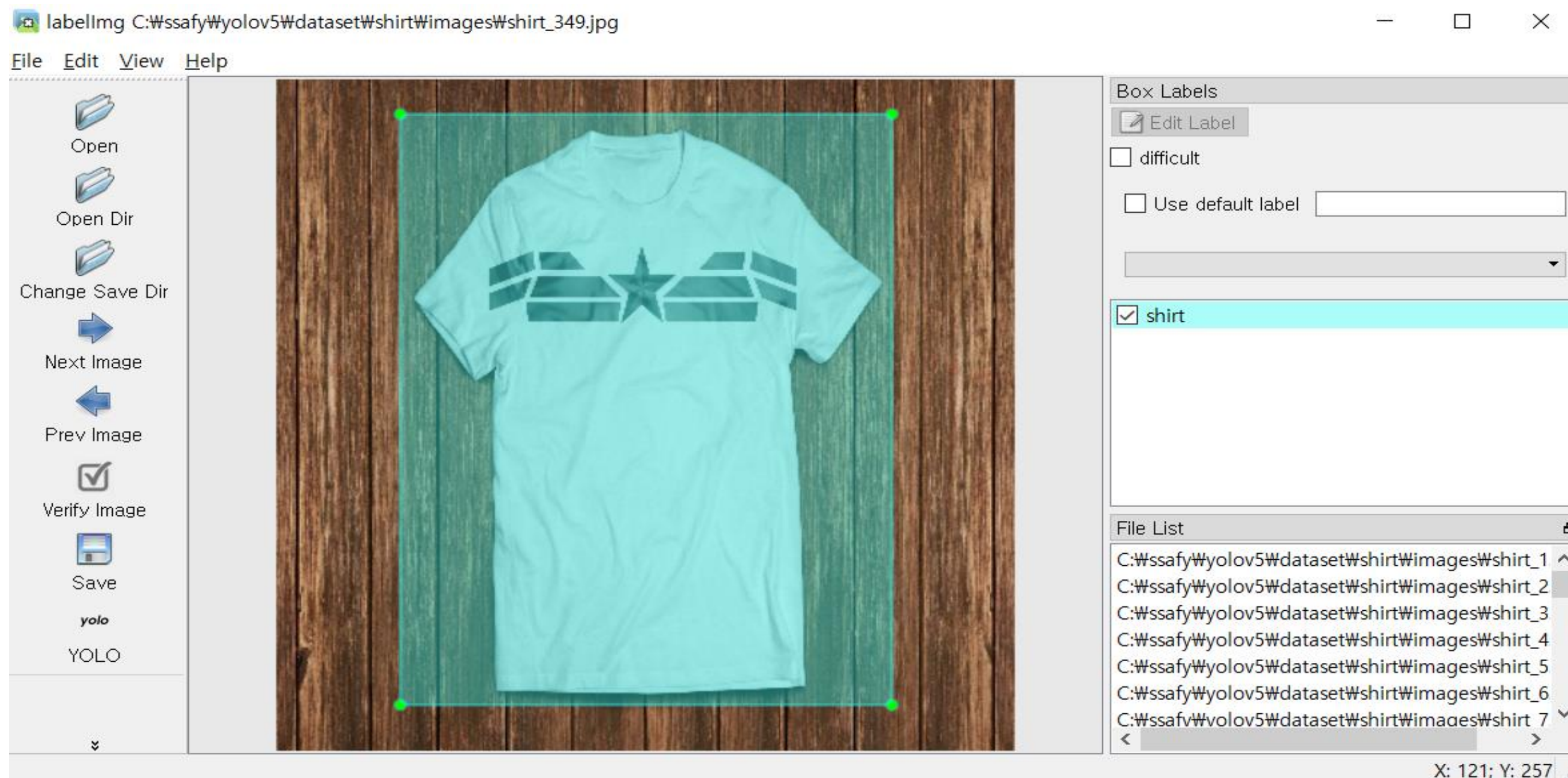
03 학습 및 인식

04. 개선 사항

# 학습 및 인식 라벨링



LabelImg





# 01 AI

01. 기술 선정

02. 모델 선정

03 학습 및 인식

04. 개선 사항

## 학습 및 인식 이미지 증폭

**imgaug** (<https://github.com/aleju/imgaug>)



Original  
Input





# 01 AI

## 01. 기술 선정

## 02. 모델 선정

## 03 학습 및 인식

## 04. 개선 사항



# 학습 및 인식

학습

- 학습 명령어

```
(first) PS C:\ssafy\darkflow> python flow --model ./cfg/my-tiny-yolo.cfg --labels ./labels.txt --trainer adam --dataset ./data/shirt/ --annotation ./data/shirt/ --train --summary ./logs --batch 5 --epoch 100 --save 50 --keep 5 --lr 1e-03 --gpu 0.5 --load -1
```

- 학습 화면

```
step 254065 - loss 1.143254041671753 - moving ave loss 1.2632310524180088
step 254066 - loss 1.2379668951034546 - moving ave loss 1.2607046366865535
step 254067 - loss 1.2787315845489502 - moving ave loss 1.2625073314727933
step 254068 - loss 1.4990596771240234 - moving ave loss 1.2861625660379163
step 254069 - loss 1.1885559558868408 - moving ave loss 1.2764019050228088
step 254070 - loss 1.622638463973999 - moving ave loss 1.311025560917928
Checkpoint at step 254070
step 254071 - loss 1.5445830821990967 - moving ave loss 1.3343813130460451
step 254072 - loss 1.308659553527832 - moving ave loss 1.3318091370942238
step 254073 - loss 1.2347103357315063 - moving ave loss 1.322099256957952
step 254074 - loss 1.2067638635635376 - moving ave loss 1.3105657176185106
step 254075 - loss 1.8284895420074463 - moving ave loss 1.362358100057404
step 254076 - loss 1.004594326019287 - moving ave loss 1.3265817226535925
step 254077 - loss 1.2778853178024292 - moving ave loss 1.3217120821684762
step 254078 - loss 1.674308180809021 - moving ave loss 1.3569716920325308
step 254079 - loss 1.6052249670028687 - moving ave loss 1.3817970195295646
step 254080 - loss 0.8349714279174805 - moving ave loss 1.3271144603683562
Checkpoint at step 254080
step 254081 - loss 1.1685837507247925 - moving ave loss 1.311261389404
step 254082 - loss 0.9722221493721008 - moving ave loss 1.2773574654008102
step 254083 - loss 0.8444846868515015 - moving ave loss 1.2340701875458793
step 254084 - loss 1.0062812566757202 - moving ave loss 1.2112912944588634
step 254085 - loss 1.1539814472198486 - moving ave loss 1.205560309734962
step 254086 - loss 1.8575661182403564 - moving ave loss 1.2707608905855017
step 254087 - loss 1.133678913116455 - moving ave loss 1.2570526928385972
step 254088 - loss 1.7376455068588257 - moving ave loss 1.3051119742406203
step 254089 - loss 1.8041881322860718 - moving ave loss 1.3550195900451656
step 254090 - loss 2.1254541873931885 - moving ave loss 1.432063049779968
Checkpoint at step 254090
```

# 01 AI

01. 기술 선정

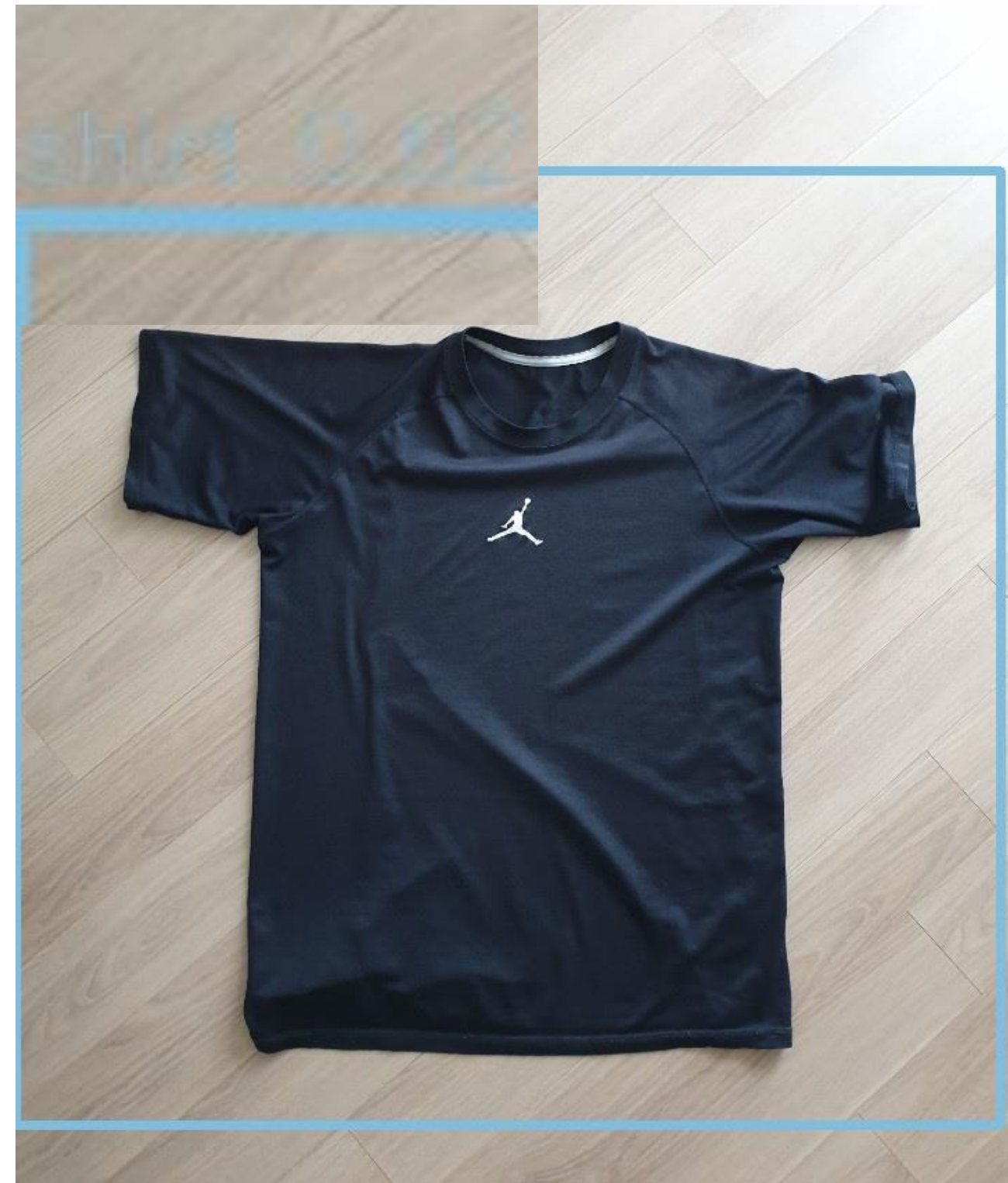
02. 모델 선정

**03 학습 및 인식**

04. 개선 사항

## 학습 및 인식

인식 결과



# 01 AI

01. 기술 선정

02. 모델 선정

03. 학습 및 인식

04 개선 사항

## 개선 사항

YOLOv2 + Tensorflow → YOLOv5 + Pytorch

- 학습 명령어 및 학습

```
[ ] !python train.py --img 416 --batch 16 --epochs 100 --data ./dataset/my_data.yaml --cfg ./models/yolov5s.yaml --weights yolov5s.pt --name ballcap_socks
```

Epoch	gpu_mem	GIoU	obj	cls	total	targets	img_size
1/99	1.53G	0.0897	0.05778	0.02809	0.1756	37	416: 100% 27/27 [00:11<00:00, 2.30it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95: 100% 7/7 [00:02<00:00, 2.51it/s]
	all	108	120	0.37	0.0838	0.214	0.0635
Epoch	gpu_mem	GIoU	obj	cls	total	targets	img_size
2/99	1.53G	0.07541	0.05837	0.02459	0.1584	36	416: 100% 27/27 [00:11<00:00, 2.30it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95: 100% 7/7 [00:02<00:00, 2.49it/s]
	all	108	120	0.123	0.455	0.281	0.105
Epoch	gpu_mem	GIoU	obj	cls	total	targets	img_size
3/99	1.53G	0.06953	0.05776	0.0244	0.1517	33	416: 100% 27/27 [00:11<00:00, 2.29it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95: 100% 7/7 [00:02<00:00, 2.49it/s]
	all	108	120	0.0867	0.699	0.342	0.161
Epoch	gpu_mem	GIoU	obj	cls	total	targets	img_size
4/99	1.53G	0.0619	0.05261	0.02291	0.1374	41	416: 100% 27/27 [00:11<00:00, 2.31it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95: 100% 7/7 [00:02<00:00, 2.60it/s]
	all	108	120	0.151	0.639	0.377	0.204
Epoch	gpu_mem	GIoU	obj	cls	total	targets	img_size
5/99	1.53G	0.05383	0.04969	0.02134	0.1249	33	416: 100% 27/27 [00:11<00:00, 2.32it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95: 100% 7/7 [00:02<00:00, 2.63it/s]
	all	108	120	0.207	0.731	0.529	0.327
Epoch	gpu_mem	GIoU	obj	cls	total	targets	img_size
6/99	1.53G	0.05231	0.04946	0.02025	0.122	35	416: 100% 27/27 [00:11<00:00, 2.35it/s]
	Class	Images	Targets	P	R	mAP@.5	mAP@.5:.95: 100% 7/7 [00:02<00:00, 3.01it/s]
	all	108	120	0.255	0.691	0.461	0.27
Epoch	gpu_mem	GIoU	obj	cls	total	targets	img_size
7/99	1.53G	0.05302	0.04405	0.01935	0.1164	33	416: 100% 27/27 [00:11<00:00, 2.35it/s]



# 01 AI

01. 기술 선정

02. 모델 선정

03. 학습 및 인식

04 개선 사항

## 개선 사항

YOLOv2 + Tensorflow → YOLOv5 + Pytorch

- 인식 결과







# 02 | Front-End

- 01. 기술 선정
- 02. 개발 환경 선정
- 03. 진행 상황

# 02 Front-End

## 01 기술 선정

02. 개발 환경 선정

03. 진행 상황

# 기술 선정



Web vs Mobile App

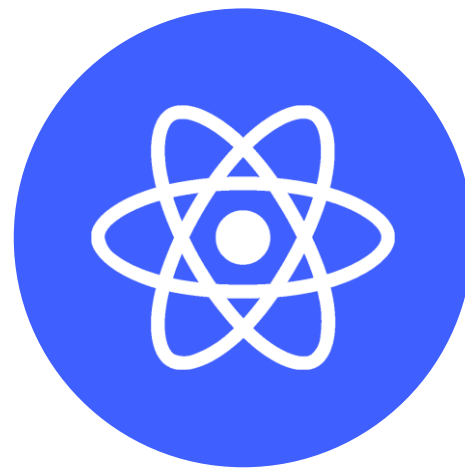
# 02 Front-End

01. 기술 선정

02 개발 환경 선정

03. 진행 상황

## 개발 환경 선정



React Native

iOS, Android 동시 개발 가능

수정된 코드를 바로 확인 가능 (Live Reload)

다양한 아키텍처 패턴 지원 (Redux, MobX)

⋮

React 를 이용하여  
개발했던 경험이 있음!



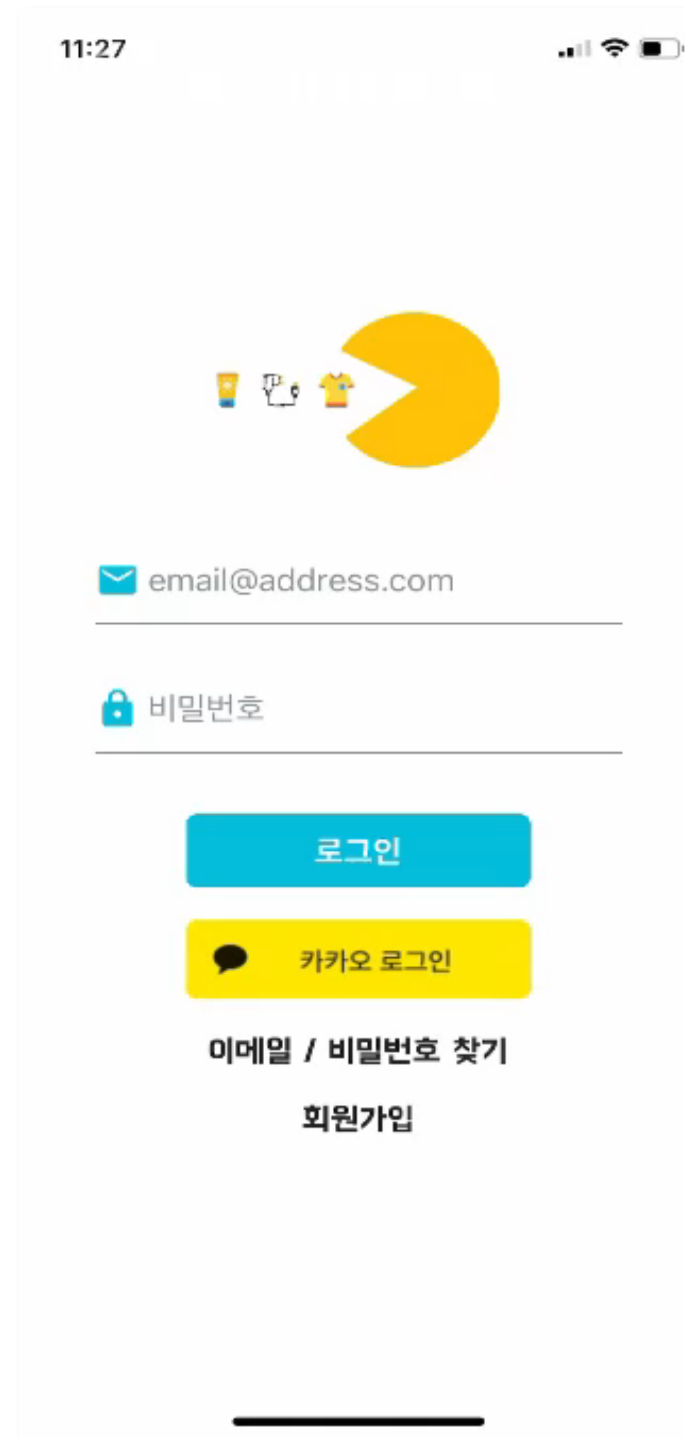
# 02 Front-End

01. 기술 선정

02. 개발 환경 선정

03 진행 상황

## 진행 상황



- ☒ 회원가입 / 로그인 / 로그아웃
- ☒ 날짜 선택을 위한 달력
- ☒ 카메라 촬영 후 이미지 전송
- ☐ 체크리스트 작성
- ☐ 체크리스트 비교 및 수정





# 03

## Back-End

01. 개발 환경 선정

02. 진행 상황

# 03 Back-End

## 01 개발 환경 선정

### 02. 진행 상황

# 개발 환경 선정



# 03 Back-End

01. 개발환경선정

02 진행 상황

## 진행 상황

▼	📁	djang-PackMan	9 requests
▼	📁	checks	
	GET	travel imformation	
	POST	create information	
	GET	read check	
	POST	create check	
	POST	update check	
	POST	delete check	
▼	📁	accounts	
	POST	login	
	POST	logout	
	POST	signup	

☒ 회원 CRUD

☒ 체크리스트 CRUD

☐ 여행 물품 추천 알고리즘

☐ 날씨 API

---

# 감사합니다

---

강창현 이지은 성열호 김성중 권하늘