



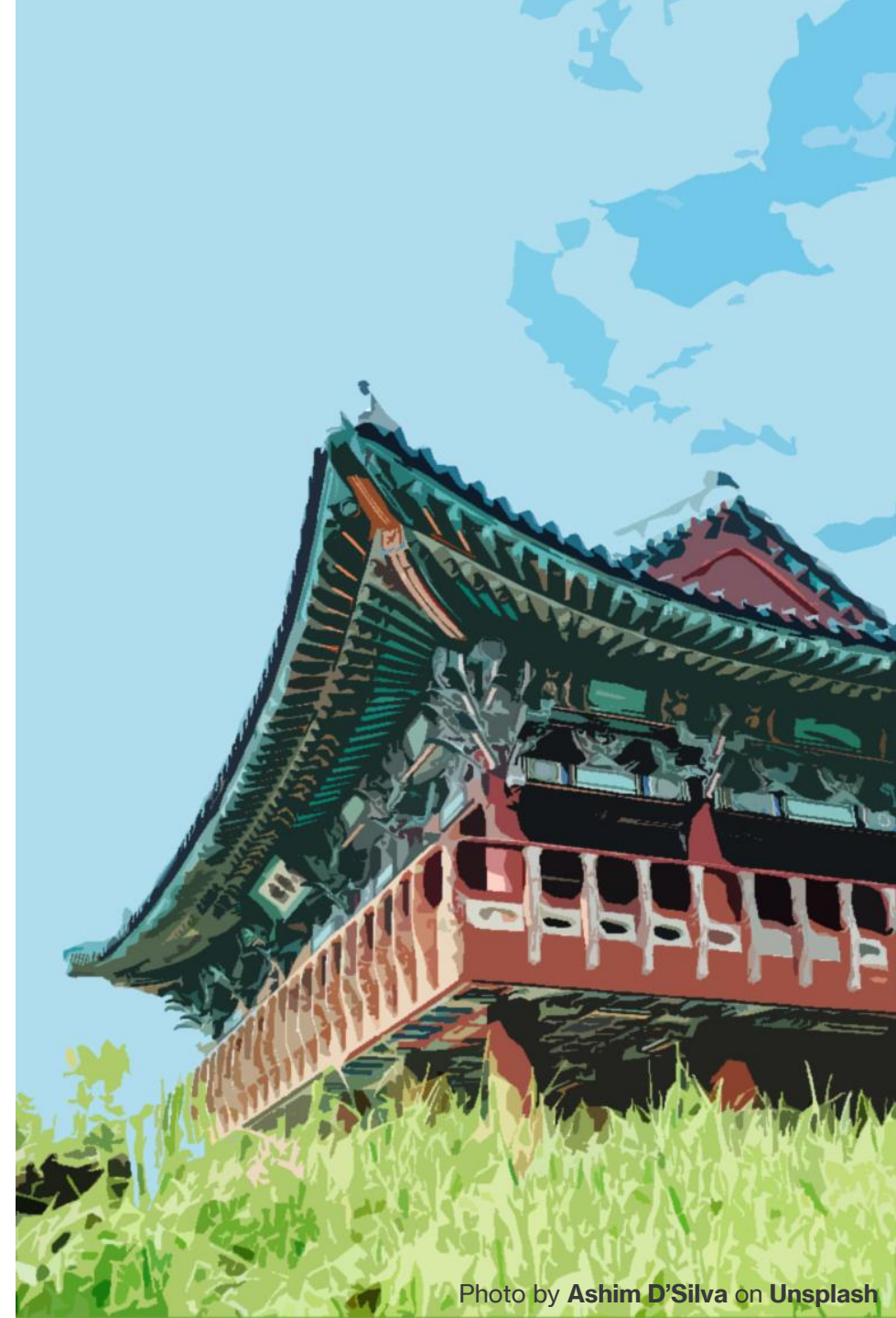
# MOJO

프로젝트형 AI 서비스 개발  
인공지능 활용 프로젝트 3조



# ***Table of Content***

1. 프로젝트 배경
2. 팀 구성 및 역할
3. 수행 절차 및 방법
4. 수행 결과 및 시연
5. 프로젝트를 진행하며 느낀 점

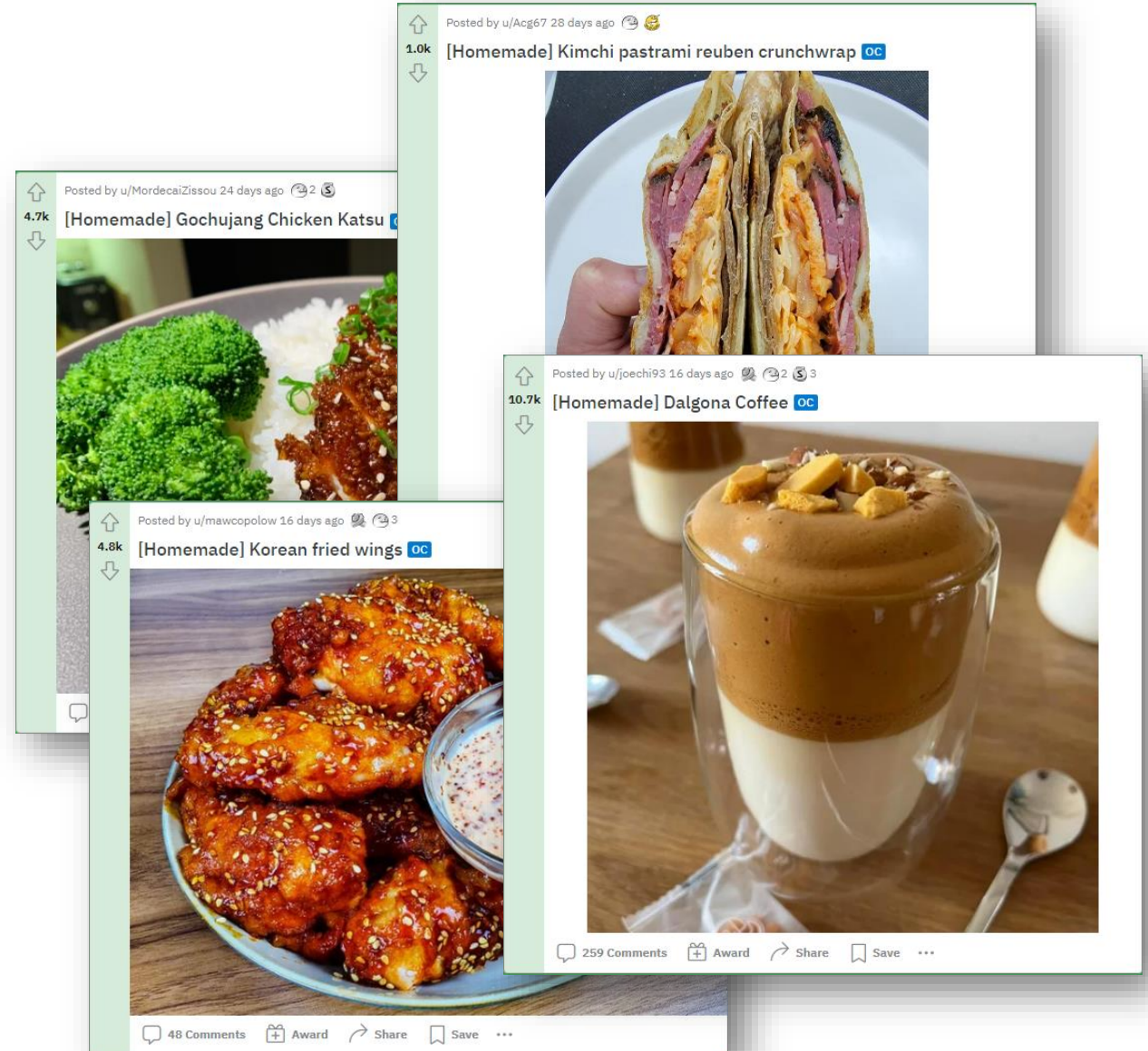


# 프로젝트 배경



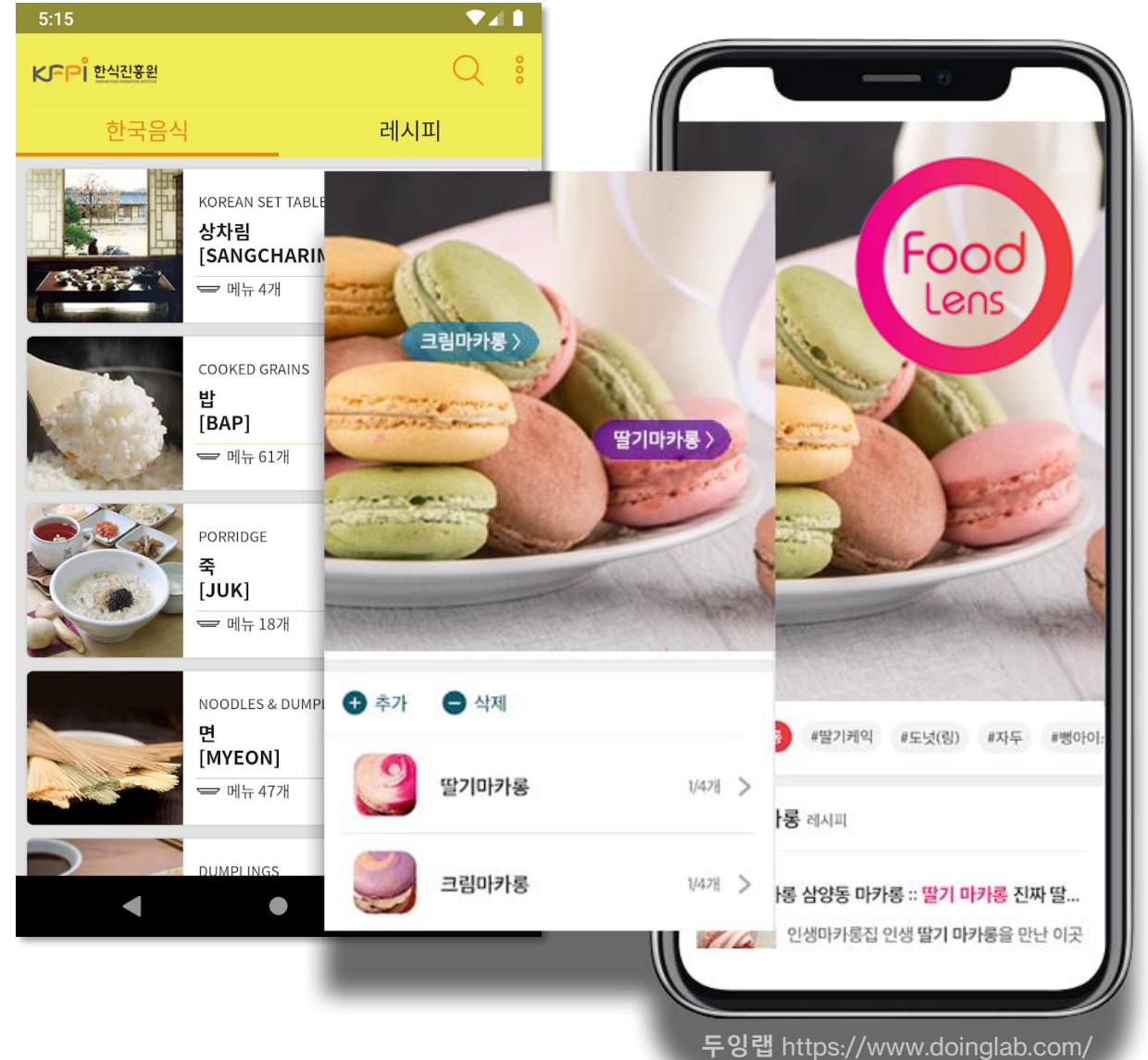
## 외국인 대상 한국 음식 인식 및 정보 제공 애플리케이션 **MOJO**

- 대한민국의 소프트파워는 현재 성장 중
- 국제사회에서 한국 음식을 접할 기회가 많아진 시대적 흐름에 발맞추어, 한국 음식에 대한 접근성을 높이고 관심을 유지시킬 수 있는 서비스로서 기획
- 맛, 영양성분, 재료, 조리법 등 관련 정보를 제공하여 자신과 완전히 유리된 분야가 아니라는 사실을 주지시켜 일회성으로 휘발될 가능성이 있는 관심을 유지시키는 효과가 있을 것으로 기대
- MOJO는 “이게 뭐죠?”에서 유래한 말로, 이것이 한국 음식인지, 어떤 음식인지 언제든지 물어볼 수 있다는 의미를 담음



# Background

- **한식진흥원**(舊 한식재단)은 한식 및 한식산업의 진흥과 관련 산업의 경쟁력 강화를 위해 설립된 공공기관으로, 한식 세계화의 수단으로 한국 음식의 이름과 정보를 알려주는 앱을 개발해 활용함.
- 또한, AI를 통해 사진에 있는 음식이 무엇인지 분석해 주는 음식 인식 솔루션을 주력으로 한 기업 '**두잉랩**'의 등장은 딥 러닝을 활용한 이미지 인식 및 분류 서비스가 어느 정도 시장성을 갖추었다는 방증으로 볼 수 있음.



# Structure

CNN 및 파생 신경망을 응용한 이미지 분류 모델

신경망

- TensorFlow Keras

데이터

- AI Hub 비전 데이터 (KIST 제공)

학습 환경

- Google Colaboratory
- Amazon Web Service

서비스 배포

- Django
- Swing2App





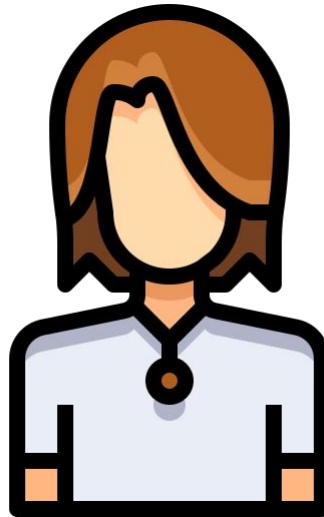
# 팀 구성 및 역할

# *Proud Members of MOJO!*



윤상민

- 프로젝트 수행 기획
- 데이터 수집/전처리
- 모델 설계



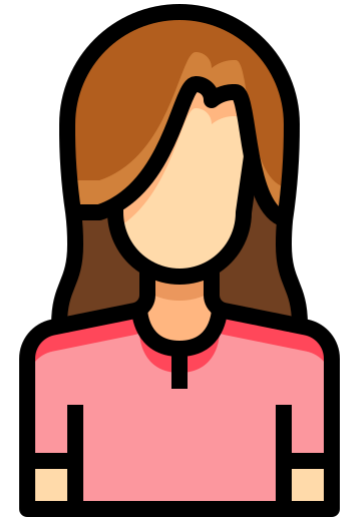
이수민

- 데이터 수집
- 모델 설계
- 프로젝트 보고 및 발표



이승환

- 도메인 리서치
- 모델 설계
- 웹 사이트 제작



윤수진

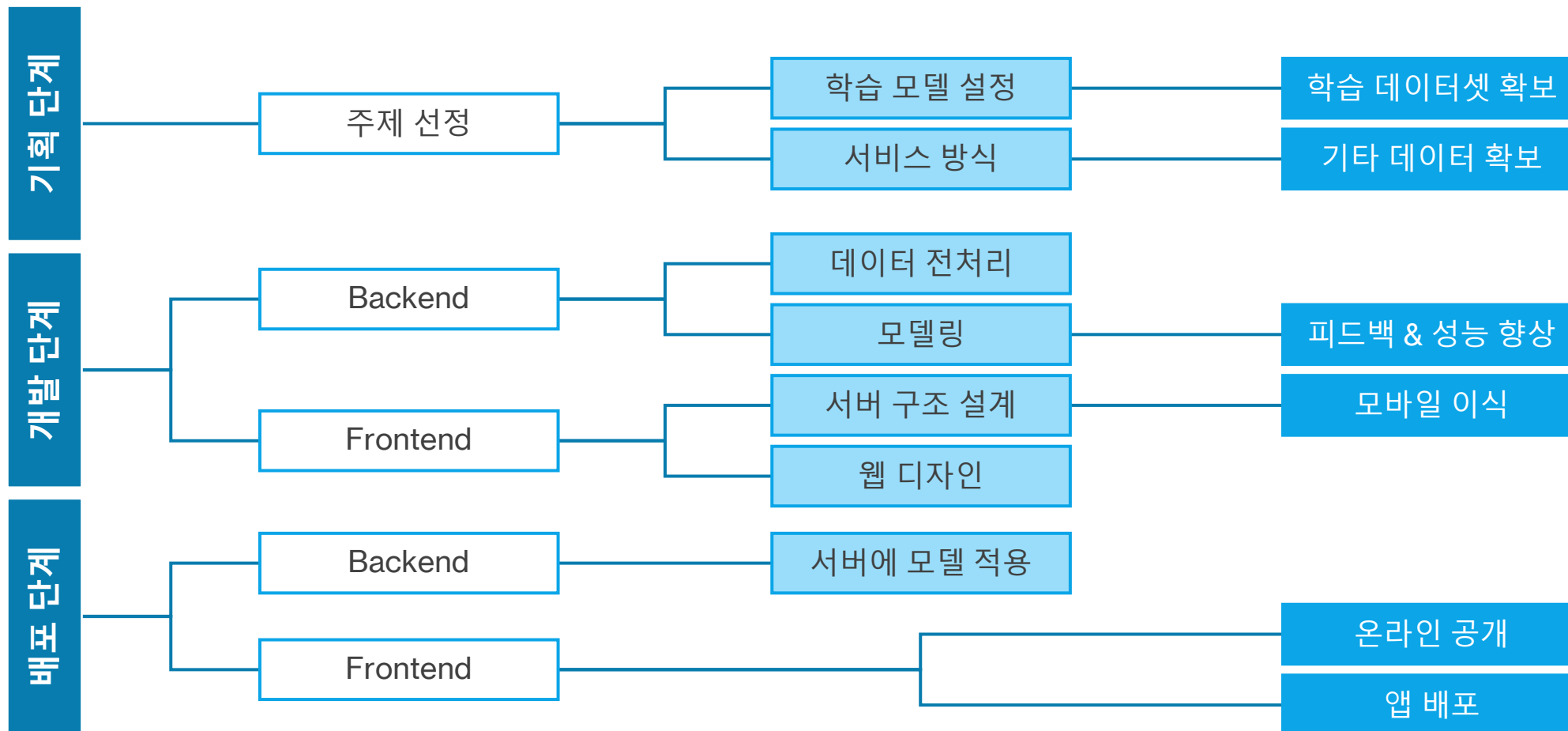
- 데이터 전처리
- 프로젝트 관리
- 프로젝트 보고 및 발표



# 수행 절차



# Process





# Timetable

Process	Period	Detail	Tool
기획	4/11 (월) ~ 4/14 (목)	프로젝트 기획 & 주제 선정	
	4/13 (수) ~ 4/15 (금)	데이터셋 선정	
개발	4/14 (목) ~ 4/27 (수)	모델 설계 및 시험	<ul style="list-style-type: none"> <li>TensorFlow (Keras)</li> <li>AWS, Colab</li> </ul>
	4/14 (월) ~ 4/20 (수)	서버 개발	<ul style="list-style-type: none"> <li>Django</li> <li>HTML, CSS, JavaScript</li> </ul>
	4/20 (수) ~ 4/25 (월)	앱 이식	<ul style="list-style-type: none"> <li>Swing2App</li> </ul>
발표	4/26 (화) ~ 4/29 (금)	발표 자료 제작	<ul style="list-style-type: none"> <li>MS PowerPoint</li> </ul>
	5/2 (월)	발표	



# 수행 결과 및 시연



# Feature Study

## Input Size

Model	Class	Input Size	Top Layer	Dropout	Trainable	Activation	Optimizer	Learning Rate	Epoch	Runtime	Accuracy
ResNet50V2	5	300*300	32, 64, 128, 256	-	30,117,733	Relu	Adam	5e-7	200	1h 19m 17s	0.91
ResNet50V2	5	400*400	32, 64, 128, 256	-	34,639,717	Relu	Adam	5e-7	200	1h 38m 53s	0.92

- Input size가 클수록 이미지 특성을 더 많이 반영할 수 있으므로 학습 정확도를 확보하기 위해서는 input size가 큰 것이 바람직함
- 다만 제한된 프로젝트 리소스와 대다수 사용자가 제공할 이미지의 크기를 고려했을 때 400~500 정도가 적당한 것으로 보임

## Number of Class

Model	Class	Input Size	Top Layer	Dropout	Trainable	Activation	Optimizer	Learning Rate	Epoch	Runtime	Accuracy
ResNet50V2	3	200*200	20	0.5	25,526,483	Relu	Adam	5e-7	100	1h 5m 0s	0.52
ResNet50V2	10	200*200	40	-	27,533,890	Relu	Adam	5e-7	100	1h 56m 30s	0.72
ResNet50V2	30	224*224	40	0.5	27,534,710	Softplus	Nadam	1e-6	100	1h 51m 32s	0.90

- 최초 수집한 데이터셋의 클래스 수는 150개였으나, 클래스 수가 지나치게 많을 경우 학습률 둔화를 우려하여 3, 5, 10, 30으로 나누어 진행함
- 30개 클래스로 학습을 진행한 경우에도 프로젝트 진행 기간 안에 전체 정확도 자체는 90% 수준으로 향상시킬 수 있는 것으로 보임
- 그러나 학습 평가 시 overall accuracy는 mean 형식으로 나타나므로 클래스 수를 늘릴 경우 개별 클래스 정확도를 판정할 evaluation 방식을 새로 마련해야 함

# Feature Study

## Learning Rate

Model	Class	Input Size	Top Layer	Dropout	Trainable	Activation	Optimizer	Learning Rate	Epoch	Runtime	Accuracy
VGG19	10	200*200	32, 64, 128	-	20,625,962	Relu	Adam	<b>5e-8</b>	300	1h 27m 39s	0.24
VGG19	10	300*300	32, 64, 128	-	21,363,242	Relu	Adam	<b>5e-7</b>	300	2h 58m 20s	0.78

- 학습 초기에 local minima에 영향을 받지 않고 원활한 학습이 진행될 수 있는 learning rate의 하한은 1e-7 수준인 것으로 생각됨

## Dropout

Model	Class	Input Size	Top Layer	Dropout	Trainable	Activation	Optimizer	Learning Rate	Epoch	Runtime	Accuracy
NASNetMobile	30	224*224	20, 20	<b>0.4</b>	5,268,508	Relu	Adam	5e-6	100	1h 5m 0s	0.52
NASNetMobile	30	224*224	20	-	5,268,508	Relu	Adam	5e-6	100	1h 56m 30s	0.72

- Dropout 적용시 training accuracy/loss보다 validation accuracy/loss 결과가 현저히 높은 현상이 관찰됨
- Dropout은 과적합을 해결하기 위한 방법이므로 과적합이 나타나지 않는 현재 모델에서는 사용하지 않는 것이 좋다는 결론을 내림



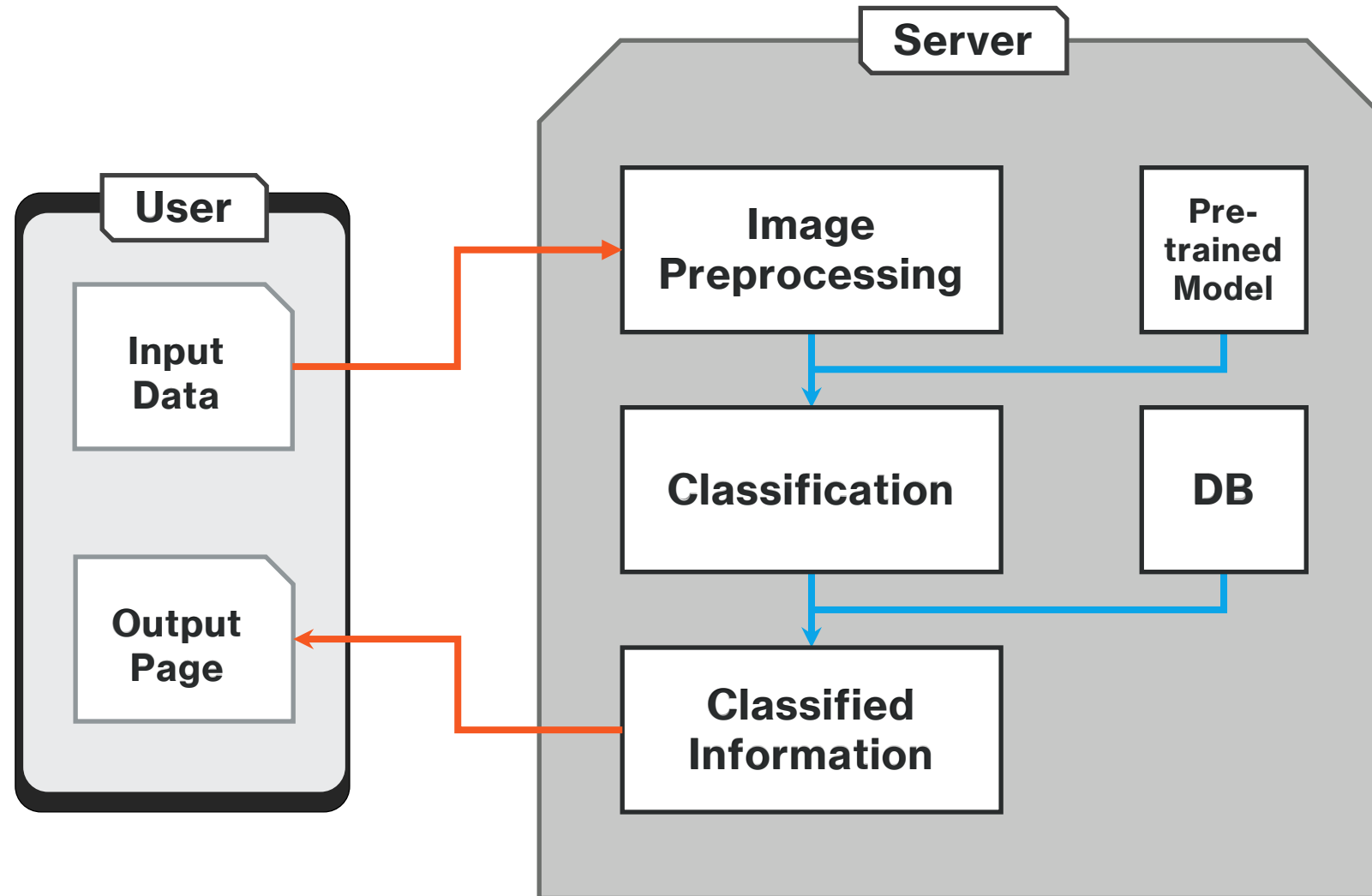
# Feature Study

## Base Model

Model	Class	Input Size	Top Layer	Dropout	Trainable	Activation	Optimizer	Learning Rate	Epoch	Runtime	Accuracy
NASNetMobile	3	224*224	20	0.5	5,267,941	Softplus	Nadam	1e-6	100	1h 34m 6s	0.90
ResNet50V2	3	224*224	20	0.5	25,526,483	Softplus	Nadam	1e-6	100	1h 51m 32s	0.90
VGG19	3	224*224	20	0.5	20,526,227	Softplus	Nadam	1e-6	100	1h 45m 57s	0.94
EfficientNetB4	3	224*224	20	0.5	19,304,859	Softplus	Nadam	1e-6	100	4h 13m 53s	0.89

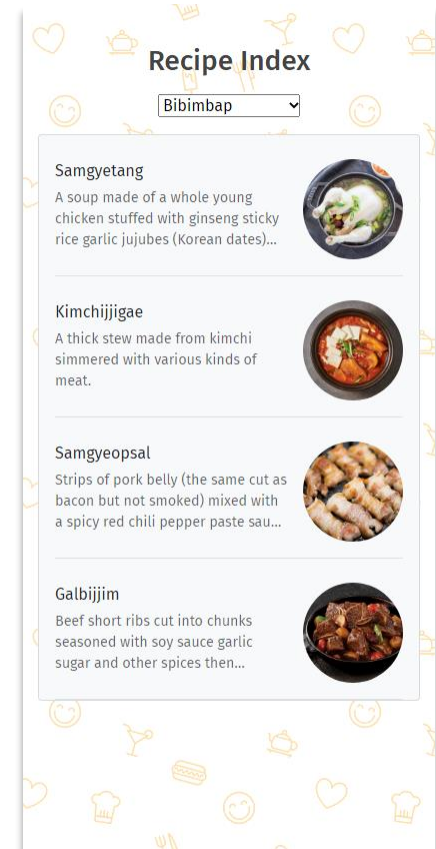
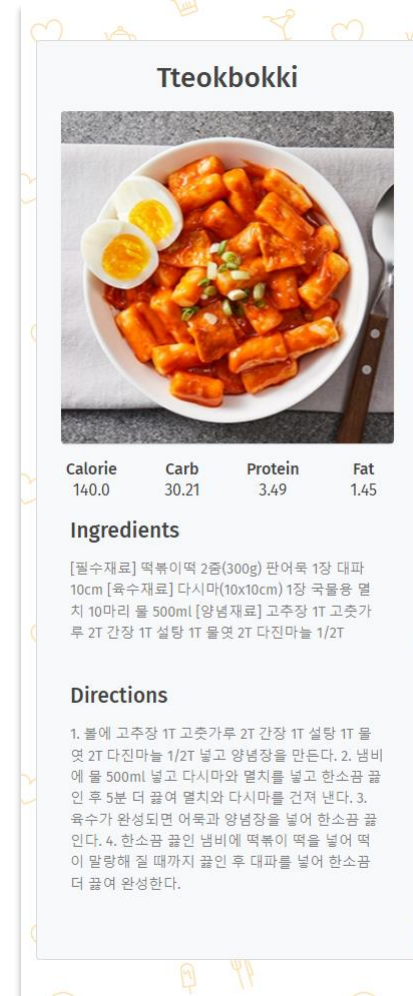
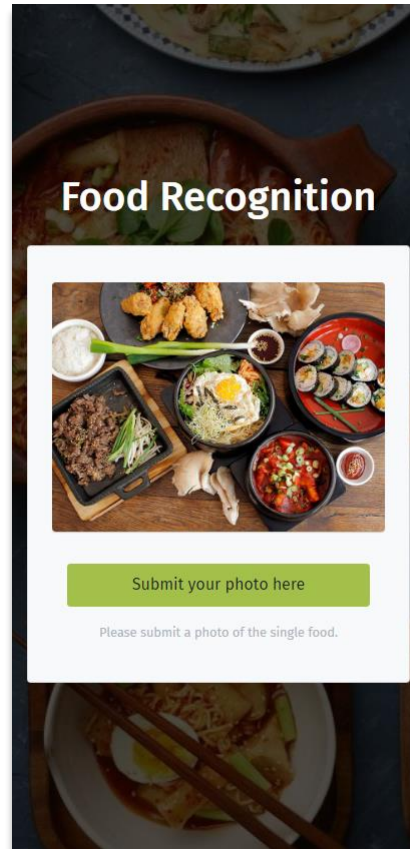
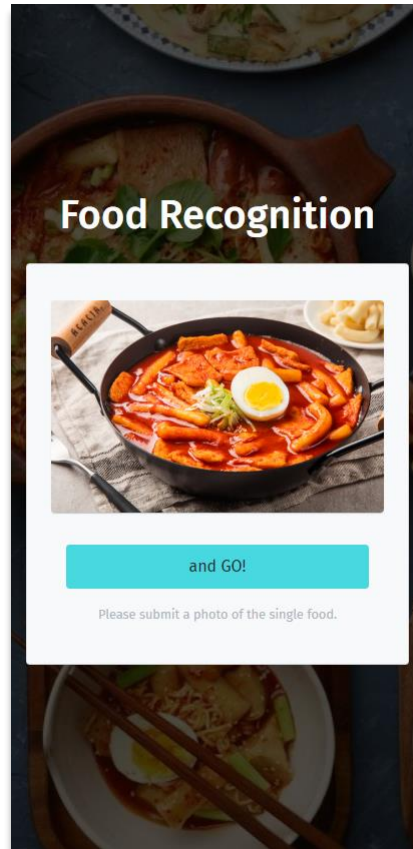
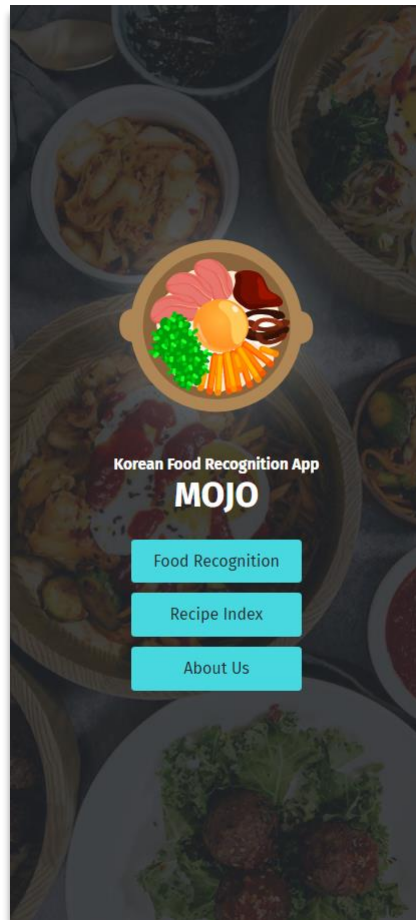
- NASNetMobile, NASNetLarge, InceptionV3, VGG16, VGG19, ResNet50V2, EfficientNetB4, EfficientNetB7 등 모델을 기반으로 실험하였음
- InceptionNetV3의 경우 타 모델에 비해 학습 효율이 크게 떨어져 초기에 배제하였음
- EfficientNetB7의 경우 학습에 사용한 GPU 자원 문제로 학습이 불가능해 trainable parameter 규모를 낮춘 이전 버전 B4를 사용함
- EfficientNetB4의 경우 타 모델과 비교하여 epoch당 학습 소요 시간이 4배 이상이며 최종 정확도가 타 모델에 비해 부족함
- NASNetMobile과 ResNet50V2, VGG19는 학습 소요 시간과 최종 성능이 유사하나 VGG19가 초기에 성능 안정화가 빠르며 최종 성능도 유효한 수준으로 높음

# Process Structure





# Process Structure



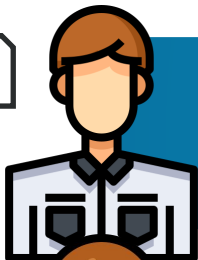


# 느낀 점



# Takeaways

윤상민



프로젝트를 수행하는 동안 생각지 못한 변수들에 의해 중간중간 되돌아가거나 계획을 바꾸어야 하는 일도 많았지만, 목표를 이루기 위해 모두 열심히 해 줘서 고마움을 느꼈습니다. 여태까지 배운 내용을 십분 활용하기 위해 다시 천천히 살펴보는 과정을 거치면서 좀 더 이해가 되었습니다.

이수민



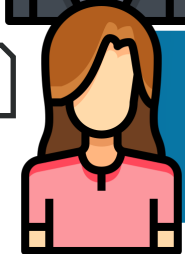
밑바닥부터 시작해서 예상치 못한 시행착오를 겪어 쉽지 않았지만 포기하지 않고 오히려 발판으로 삼아 팀원들과 문제를 해결하는 과정을 통해 뿌듯함을 느꼈습니다. 또한 수업 과정에서 배운 내용을 직접 찾아보고 적용해보고 고민해 보면서 모델 뿐만 아니라 나 자신도 학습되어 효율적이라 재미있고 좋았습니다.

이승환



다수의 인원이 참여하는 프로젝트에서 워크플로우 관리의 중요성을 느낄 수 있었습니다. MLOps란 이러한 것이구나 하고 이해할 수 있는 계기가 되었다고 생각합니다. 모델 개발 파이프라인을 바닥에서부터 새로 구축하는 것도 좋은 경험이었지만 이미 구현되어 운용되고 있는 MLOps를 살펴보면서 지금의 경험과 비교하며 어떻게 적용하고 개선할 수 있을지 알아보는 것도 재미있을 것 같습니다.

윤수진



목표를 위해 프로젝트를 수행하며 배웠던 내용을 복습하니 훨씬 머리에 잘 들어왔고, 프로젝트 진행의 전체적인 흐름과 구조를 볼 수 있어서 너무 유익한 시간이었습니다. 시간과 자원이 좀 더 주어졌더라면 더 좋은 결과를 낼 수 있지 않았을까 하는 아쉬움이 남습니다.

The background is a dense, abstract pattern of small, overlapping circles in various colors including blue, green, orange, red, and purple. On the left side, there is a vertical bar with a gradient from light blue at the top to deep purple at the bottom. The text "END OF PRESENTATION" is centered in the middle of the image in a large, white, sans-serif font.

# END OF PRESENTATION