

보아즈 프로젝트 최종 발표

# 음식 칼로리 자동 계산 서비스

A central image of a variety of fresh vegetables, including green leafy lettuce, a head of broccoli, several red tomatoes, white mushrooms, garlic bulbs, and green beans, arranged in a pile. The vegetables are positioned behind the main title text.

김민식, 마민정, 유신혁, 음승호



## 1. 주제 선정 배경

음식 칼로리 자동 계산 서비스



## 3. 데이터 셋

건강 관리를 위한 영양분석 및 음식 이미지  
통합 식품 영양성분 DB



## 5. 제안 방법론

음식 인지 및 분류 모델 전체 Flow  
모델 주요 Contribution

## 2. 기존 연구



Food Recognition 관련 논문  
카카오 AI LAB

## 4. 문제 정의



용어 정리 및 가정

## 6. 실험 및 결과

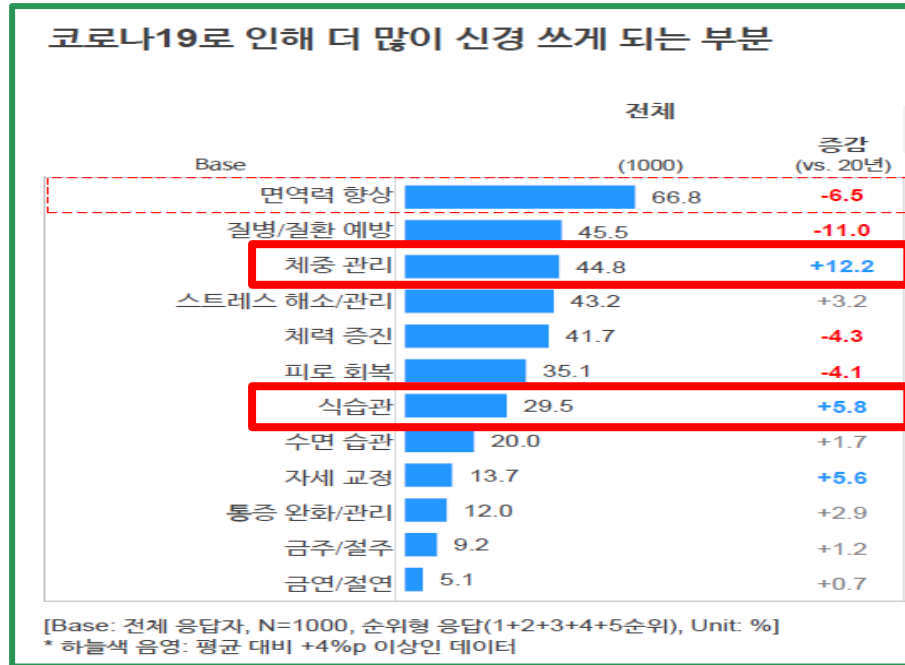


실험 시나리오(1), (2)  
실험 결과(1), (2)  
실제 음식 이미지 적용

# 1. 주제 선정 배경

## "음식 칼로리 자동 계산 서비스"

[ 체중관리와 식습관에 대한 관심 증가 ]



→ 코로나19 장기화로 인해 전년대비  
'체중관리'와 '식습관'에 대한 관심 증폭

AI를 활용한 간편하고 지속 가능한  
음식 탐지와 영양 성분 분석

- ☑ 전문가가 아닌 일반인도 음식의 종류와 영양 성분 및 칼로리를 알 수 있음
- ☑ 다이어트나 체중 관리를 하는 사람들에게 유용한 서비스 제공
- ☑ 한식 데이터셋을 통한 우리나라 맞춤형 서비스 제공

## 2. 기존 연구

# "Food Recognition 관련 논문"

- FoodTracker: A Real-time Food Detection Mobile Application by Deep Convolutional Neural Networks

인용 분야	특징	한계점 및 차이점
Preprocessing, multi-food recognition	Data Augmentation(blur, horizontal flip, Gaussian noise or color shift), 음식 검출과 분류를 YOLO로 동시에 수행	color shift의 경우 음식 이미지의 색을 임의로 바꾸면 제대로 된 인식을 못할 가능성이 큼 → 제안된 기법은 우리의 dataset과 맞지 않음 → <b>Brightness</b> 와 <b>Saturation</b> 을 통해 밝기와 채도만 조정

- Recognition of Multiple Food items in A Single Photo for Use in A Buffet-Style Restaurant

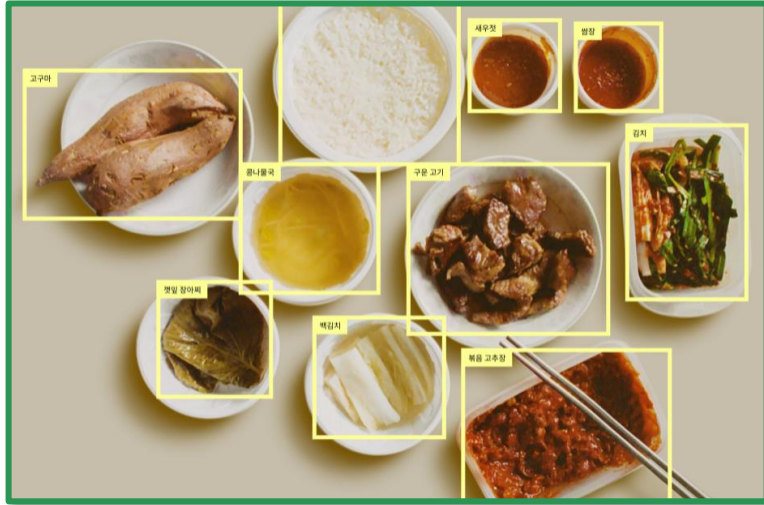
인용 분야	특징	한계점 및 차이점
Modeling	Modeling(SSD300, NN Search, hierarchical recognition)	dish area를 기준으로 bbox를 그림 → 음식의 일부만 사진에 나와있는 경우 등 dataset과 맞지 않음 → <b>food</b> 를 기준으로 bbox 그림

- Mobile Multi-Food Recognition Using Deep Learning

인용 분야	특징	한계점 및 차이점
Optimizer	SGD Optimizer	SGD optimizer를 사용해 모든 가중치에 동일한 학습률 적용 → SGD보다 Adabound의 성능이 더 좋음 → <b>SGD에 Adam</b> 방법을 적용한 Adabound 채택

## 2. 기존 연구

# “카카오 AI LAB : 스마트하게 식단을 관리하는 딥러닝 기술”



### [ 인용한 기법 ]

1. Inception V4 모델
2. Adabound Optimizer
3. Data Augmentation(crop, rotation, horizontal flip, color jittering)
4. Test-Time Augmentation(TTA)

### [ 한계점 및 차이점 ]

당시 최신 분류 모델 중 자체 실험에서 가장 높은 성능을 달성한 Inception V4를 baseline model로 사용  
→ Inception V4는 pre-train된 모델이 없으며 computer power상 Inception V3 모델로 pre-training  
→ Class별 다양한 Augmentation 기법을 사용해 성능 비교

Detector에서 Bounding Box는 잘 잡지만 Multi Food를 Classification하는 부분의 성능이 좋지 않음  
→ Classifier를 먼저 학습시킨 뒤, Detector에서 Classifier를 backbone으로 가져와 학습시킴

### 3. 데이터 셋

# "건강 관리를 위한 영양분석 및 식이관리 음식 이미지"

## I. 원천 데이터

데이터 종류	구축 수량	포함 내용	제공 방식
음식이미지 데이터베이스	500여종 X 60,000장 = 3,000,000장	원본이미지, JSON파일	파일 다운로드

음식 인식 성능의 강화를 위해 **카테고리 500여개**의 음식 데이터 확보  
ex. 한식, 중식, 일식, 수산물, 분식, 정통양식, 패스트푸드, 제과제빵케익 등등

## II. 라벨링 데이터

* 어노테이션 포맷			
내용	이름	타입	필수 요소
데이터셋명	Code Name	string	Y
음식이름	Name	string	Y
이미지 너비	W	Number	Y
이미지 높이	H	Number	Y
파일포맷	File Format	string	Y
대분류	Cat 1	Number	Y
중분류	Cat 2	Number	Y
소분류	Cat 3	Number	Y
미소분류	Cat 4	Number	N
annotation 유형	Annotation Type	string	Y
라벨링 영역	Point (x,y)	Number	Y
class 정보	Label	Number	Y
정량	Serving Size	Number	N
촬영각도	Camera Angle	Number	N
방위각도	Cardinal Angle	Number	N
용기색상	Color of Container	Number	N
용기재질	Material of Container	Number	N
조도	Illuminance	Number	N

## 12가지 Category

갈치조림, 김치볶음밥, 등심돈가스, 떡갈비,  
떡볶이, 비빔밥, 시래기된장국, 쌀밥, 잡곡밥,  
짜장면, 크림파스타, 후라이드치킨

## III. Info Table

코드	음식/식품명	에너지	탄수화물	단백질	지방
A020102	고구마케이크	326.4	39.7	4.5	16.6
A020103	고로케	245.6	23.8	4.8	14.6
A020104	과배기	242.4	28.3	4.7	12.2
A020105	북자카스테라	239.2	47.6	5.5	3.0
A020106	도넛	340.8	32.5	5.8	20.8
A020107	롤케이크	296.8	38.5	6.0	13.2
A020109	머핀	296.0	41.4	6.9	11.4
A020110	오닝빵	331.0	52.1	9.9	9.3
A020111	모카빵	292.8	46.3	7.0	8.9
A020112	바게트빵	223.2	46.0	7.5	1.0
A020113	방식빵	223.2	39.7	7.2	3.9
A020114	베이글	208.5	39.8	8.0	1.6
A020115	보리빵	152.4	26.5	2.9	3.9
A020116	불어빵	203.2	35.3	3.8	5.2
A020119	생크림빵	181.8	26.3	3.9	6.8
A020120	생크림케이크	222.5	22.1	3.1	13.5
A020121	소보로빵	323.0	48.0	7.5	11.2
A020122	소시지빵	258.4	26.4	8.5	13.2
A020123	슈크림빵	220.0	36.4	4.8	6.2
A020124	식빵	213.0	38.3	6.2	3.9
A020127	와플	232.8	26.3	6.3	11.3
A020131	편빵	158.6	33.5	4.7	0.6
A020132	찰찰도넛츠	236.6	45.7	4.0	4.2
A020133	초코머핀	355.2	49.7	8.3	13.7
A020134	초코케이크	362.4	32.2	3.8	24.3
A020137	치즈케이크	256.8	20.4	4.4	18.0
A020138	카스텔라	259.2	44.1	5.4	6.8
A020140	크로와상	358.4	35.1	6.3	21.4
A020141	크림빵	192.5	31.9	4.2	5.4
A020145	말빵	236.0	42.0	6.1	4.9
A020147	피자빵	337.9	44.1	16.4	10.6
A020148	핫도그	213.6	28.0	6.5	8.4
A020149	핫케이크	197.6	41.3	3.4	2.1
A020151	호두과자	269.7	40.7	5.6	9.4
A020152	호떡	251.4	48.6	3.6	4.7
A020153	호밀빵	250.8	50.1	8.0	2.1
A020154	호밀식빵	250.8	50.1	8.0	2.1
A020156	꽃빵	130.0	28.0	3.3	0.7
A020157	장근케이크	279.2	36.0	4.1	12.6
A020175	마늘바게트	339.2	46.0	7.8	13.8
A020176	도라야끼	165.1	28.7	3.1	4.2
A020501	마카로니과자	137.1	21.4	3.5	4.2
A020503	깨강정	41.9	6.6	0.5	1.4
A020505	두부과자	153.9	17.6	2.1	8.3
A020512	센베이	230.8	48.7	4.4	2.0
A020514	쌀강정	187.7	44.1	1.9	0.4
A020516	약과	295.4	50.2	2.7	9.3
A020517	양갱	143.0	33.9	1.6	0.2
A020519	유과	114.6	25.3	0.7	1.2
A020522	카라멜랍콘	137.7	23.3	1.4	4.3

※ 에너지(kcal) = 탄수화물(g)\*4+단백질(g)\*4+지방(g)\*9

### 3. 데이터 셋

### “통합 식품 영양성분 DB”

파일

홈

삽입

레이아웃

수식

데이터

검토

보기

도움말

원본 보기

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

가

통합 식품 영양성분 DB\_12

식품코드	식품명	에너지(kcal)	나트륨(mg)	탄수화물(g)	총당류(g)	지방(g)	트랜스 지방산(g)	총 포화 지방산(g)	콜레스테롤(mg)	단백질(g)
9011002	갈치조림	100	567.55	6.2	6.3	2.8	0	0.6	50.49	12.7
1014001	김치볶음밥	550	1549.8	51.9	45.9	30.3	0.1	4.7	232.34	17.6
10012003	등심돈가스	623.828	574.078	38.489	3.66	37.545	0.263	9.115	98.162	32.992
6012004	떡갈비	403	807.59	4.9	0.8	23.6	0.8	8.6	136.07	42.8
8014001	떡볶이	260	703.12	46.7	7.9					6.3
1014005	비빔밥	637	1041.85	84.8	0.2	19				30.9
4012008	시래기된장국	99	1523.66	11	1.3					6.7
1011001	쌀밥	130	3	28.73	81.6	0				2.36
1012001	잡곡밥	292	5.29	58.6	0.6					10.6
2011027	짜장면	529	1843.68	86.2	36.4	13				16.2
2011034	크림파스타	470	950	0	0					14
10012002	후라이드치킨	530	386.81	21.3	0	33				35.5

영양정보

총 내용량 100g  
000kcal

1일 영양성분  
기준치에 대한 비율

영양성분	100g 기준	1일 영양성분 기준치에 대한 비율(%)
총 내용량당		
나트륨	00mg	00%
탄수화물	00g	00%
당류	00g	00%
지방	00g	00%
트랜스지방	00g	
포화지방	00g	00%
콜레스테롤	00mg	00%
단백질	00g	00%

1일 영양성분 기준치에 대한 비율(%)은 2,000kcal 기준  
이므로 개인의 필요 열량에 따라 다를 수 있습니다.

통합 식품 영양성분 DB(52944 x 235)의 영양성분을 활용하여

1인 1회분량에 대한 총 8개의 영양성분 메타 정보를 계산하여 구축

식품의약품안전처(식약처) 영양성분 표시기준

나트륨, 탄수화물, 당류, 지방, 트랜스지방, 포화지방, 콜레스테롤, 단백질



### 3. 데이터 셋

## "Data Augmentation"



Original Image



Rotation



Brightness



Saturation

**Rotation**(회전), **Brightness**(밝기), **Saturation**(채도)

음식 이미지를 찍을 때, 조명이나 필터 사용으로 밝기와 채도가 다양하게 나타나는 것을 반영

각 Epoch마다 정확도가 낮은 3개 클래스에만 Data Augmentation을 진행

→ 학습시간도 줄일 수 있고 이미 충분히 잘 나오는 클래스에 불필요한 데이터가 추가되는 것을 방지

→ Epoch마다 훈련하는 데이터셋이 달라져서 여러 데이터셋을 학습한 것과 유사한 효과



## 4. 문제 정의

## “용어 정리 및 가정”

### [ 용어 정리 ]

#### ✓ UFC

Useful Food Calorie의 약자로 우리가 정의한 모델명

기존의 다른 모델 대비 한번 학습시킨 Classifier을 Backbone 모델로 사용

#### ✓ Partial Augmentation

전체 클래스가 아닌 성능이 낮은 특정 클래스에만 data augmentation을 적용한 것

### [ 문제 가정 ]

- 분류 학습한 것을 backbone으로 가져왔을 때 성능이 더 잘 나올 것이라 가정
- 다양한 증강기법을 통해 시간/메모리 상 효율적일 것이라 가정

### [ 목적함수 ]

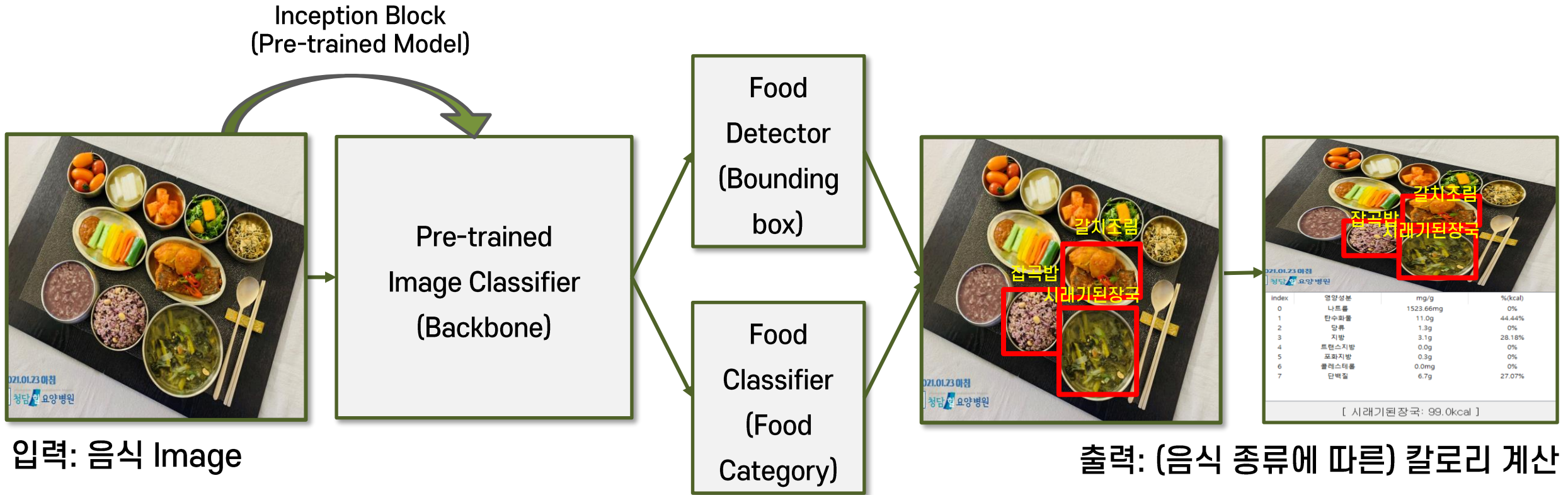
Classification ) Cross Entropy Loss

Detection ) Fast RCNN Multi-task Loss (Classifier와 Bounding Box를 동시에 학습)

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v)$$

## 5. 제안 방법론

# "음식 인지 및 분류 모델 전체 Flow"



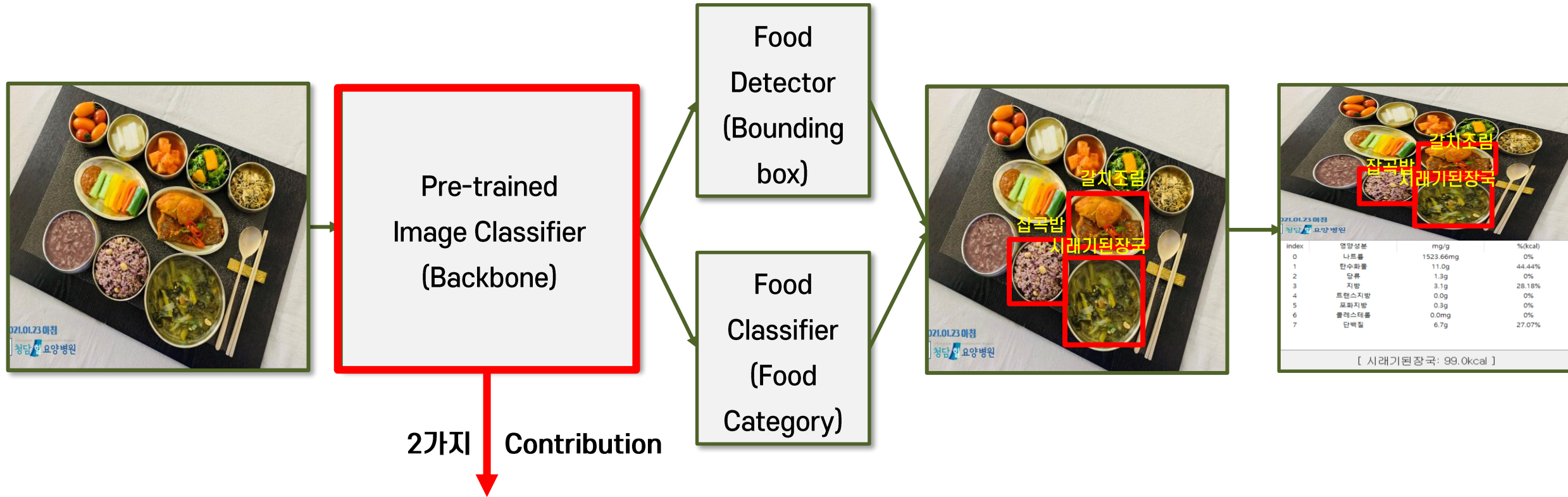
① Inception v3 model을 Pre-Trained한 Backbone 모형 개발

② Food Image Classifier에서 학습한 모형을 Pre-trained 모형으로 활용, Fast RCNN 가중치로 고정 후 분류기 학습

③ 음식 분류 이후 Data Frame 내의 칼로리를 계산해 출력

## 5. 제안 방법론

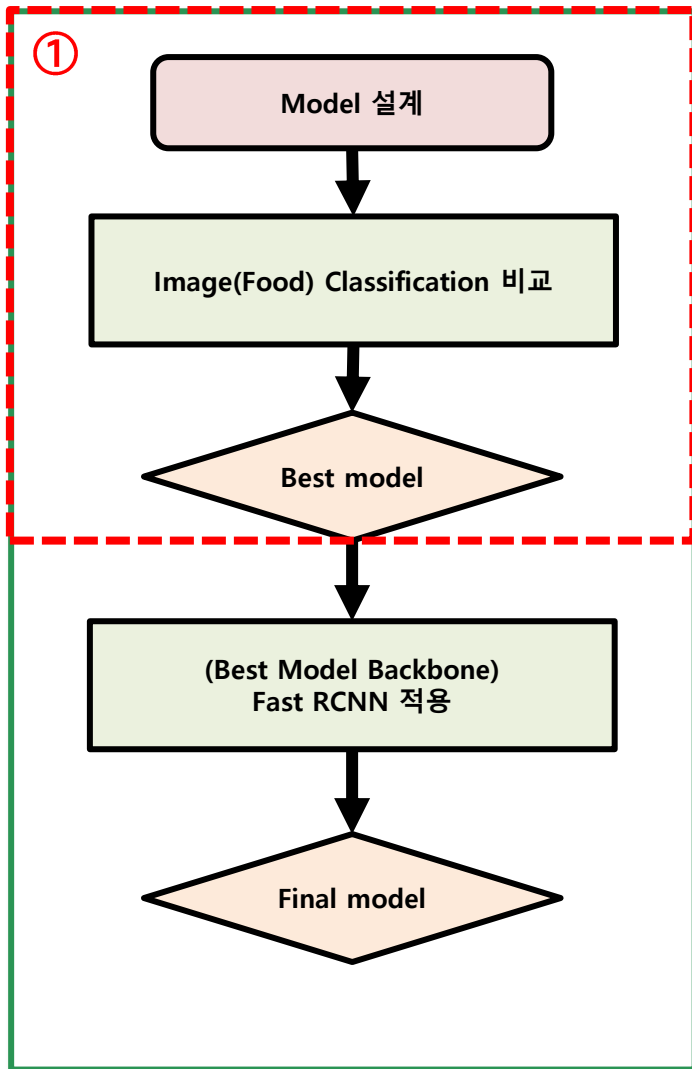
## “모델 주요 Contribution”



- Training Data로 Image Classifier를 학습
- **진행한 Model의 Parameter를 Backbone**으로 활용 Fast RCNN 성능 향상 비교
- Overfitting 방지하기 위해 Partial Augmentation(**특정 Class만 Augmentation**) 방법 제안 (Pre-trained Model 학습)

## 6. 실험 및 결과

### “실험 시나리오(1) Classification”



[실험 시나리오]

#### [ 해당 이미지에 최적화된 Pre-Trained Model 개발 ]

Pytorch 제공한 Pre-trained Inception v3 parameter 활용해 Image에 맞는 모델 구축

#### [ 실험(1) 및 비교 모델 ]

학습 Image와 최적의 Pre-trained Parameter를 찾기 위한 실험은 아래 표와 같음

실험 대상	세부사항
Fine Tunning 정도	(1) Classifier만 교체 (2) 뒤쪽 Layer $\alpha$ ( $= 15, 30, 56$ ) 만큼을 학습 진행
Optimizer	Adam vs Adabound
Augmentation	(1) 적용하지 않는 경우 (2) 전체 적용한 경우 (3) 정확도가 낮은 Class(3개)만 적용한 경우

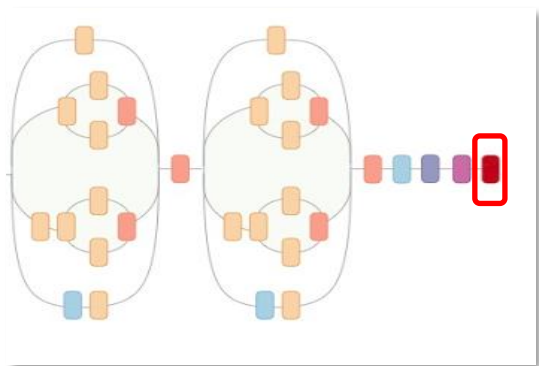
Pytorch 제공한 pre-trained Inception v3을 Baseline으로 삼아 결과 비교

평가지표 : Train/Validation의 Epoch별 Accuracy 및 Loss 비교

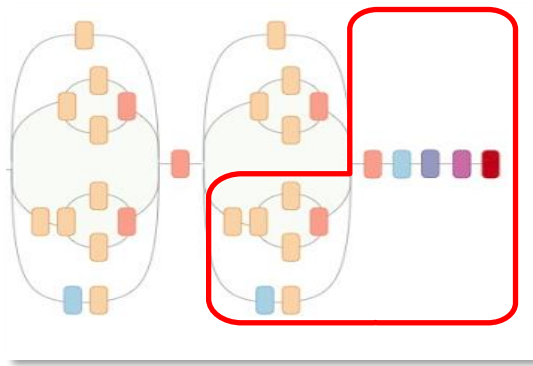
## 6. 실험 및 결과

### “실험 결과(1) Classification”

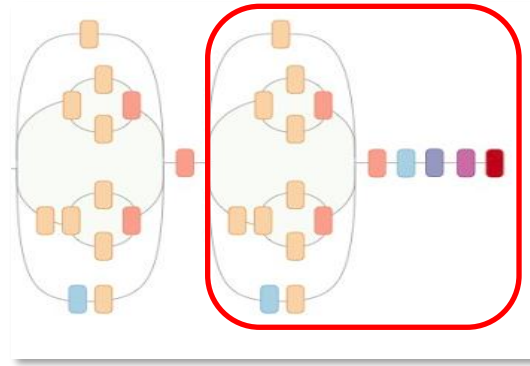
1. Freeze lambda = 0



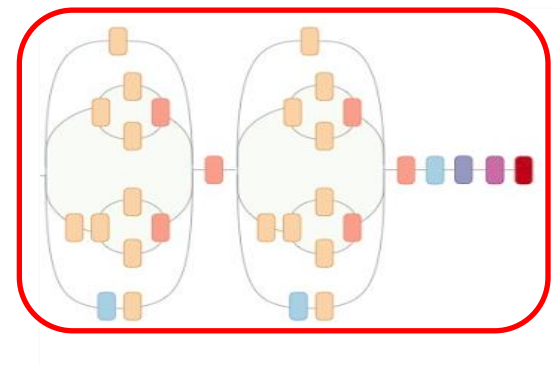
2. Freeze lambda = 15



3. Freeze lambda = 30



4. Freeze lambda = 56



#### [Inception Module C 부분]

Pytorch 제공하는 Pre-trained Model의 경우 더 많은 Parameter를 학습 시킬수록 성능이 향상 (threshold: 30)

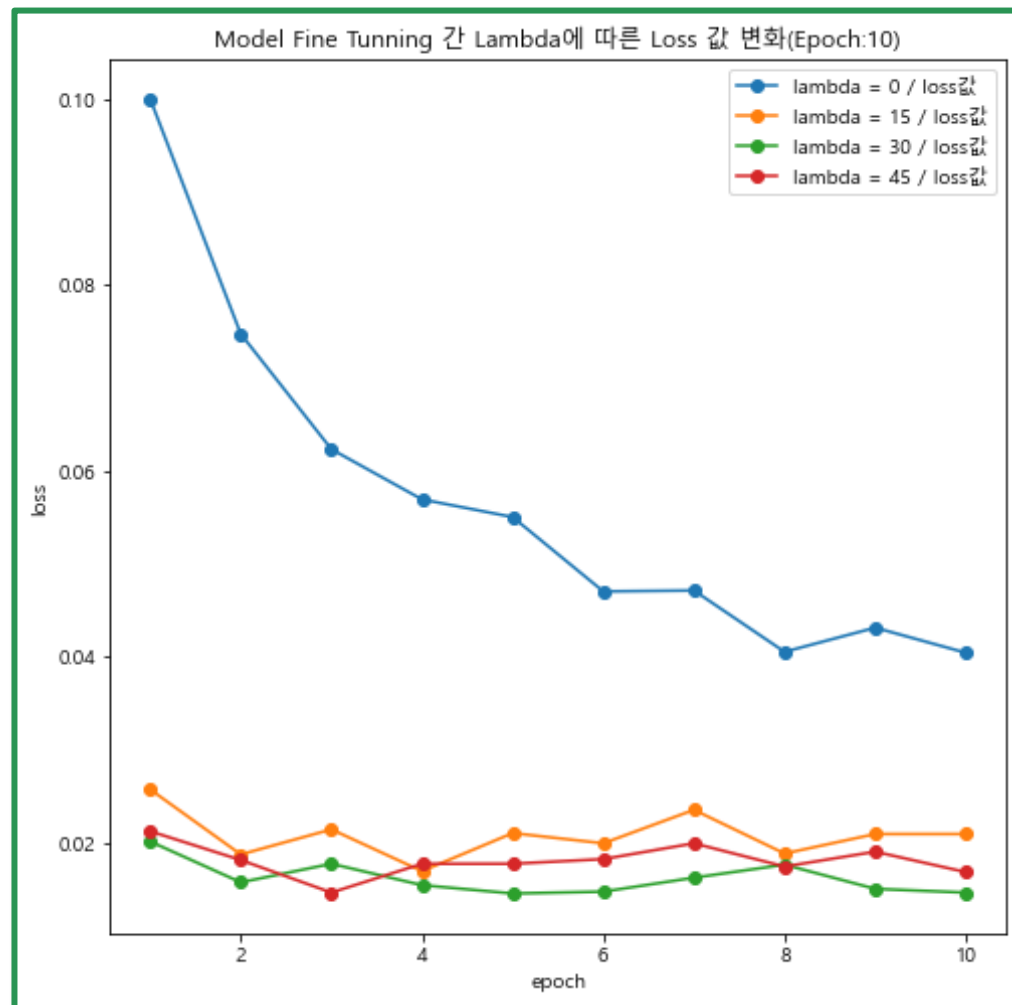
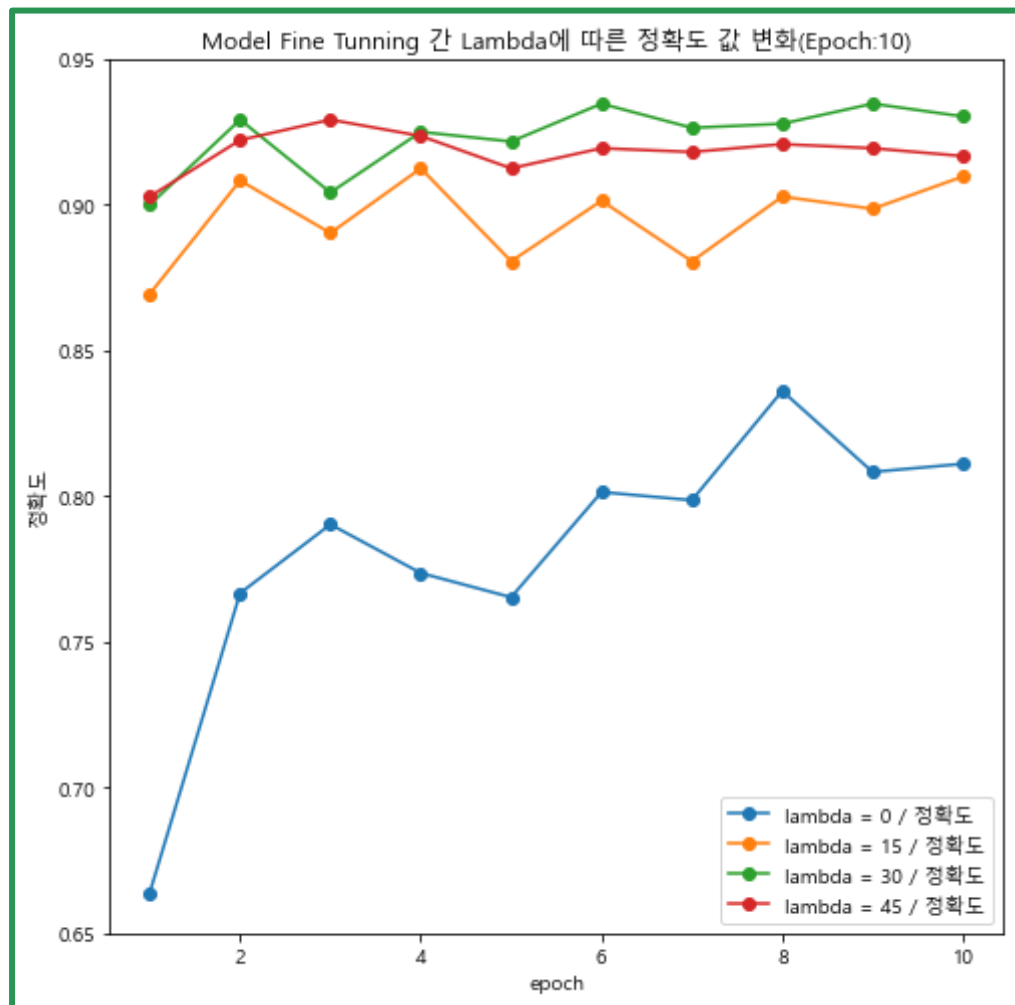
평가 척도 모형[Tuning]	Top (epoch)		Final (epoch)	
	Accuracy	loss	Accuracy	loss
$\alpha = 0$	0.8361	0.0405	0.8111	0.0404
$\alpha = 15$	0.9125	0.0169	0.9097	0.0209
$\alpha = 30$	0.9347	0.0147	0.9303	0.0146
$\alpha = 45$	0.9292	0.0146	0.9167	0.0168



## 6. 실험 및 결과

### “실험 결과(1) Classification”

각 Freezing별 정확도와 Loss 값 변화 Plot (왼쪽: 정확도, 오른쪽: Loss)



## 6. 실험 및 결과

### “실험 결과(1) Classification”

- Optimizer의 경우 Adabound의 성능이 높음
- Augmentation의 경우 적용하는 것이 상대적으로 성능이 높음
- Augmentation은 학습 속도와 정확도에 Trade-off 관계가 있음

평가 척도 모형[Tuning]	Top (epoch)		Final (epoch)	
	Accuracy	loss	Accuracy	loss
Optimizer: Adam Augmentation: 미적용	0.9347	0.0147	0.9303	0.0146
Optimizer: Adabound Augmentation: 미적용	0.9306	0.014	0.9264	0.0149
Optimizer: Adabound Augmentation: 부분 적용	0.9458	0.0128	0.9417	0.0149
Optimizer: Adabound Augmentation: 전체 적용	0.9375	0.0142	0.9292	0.0176

#### [ 최종 모델 ]

Optimizer: Adabound

Augmentation: 정확도가 낮은

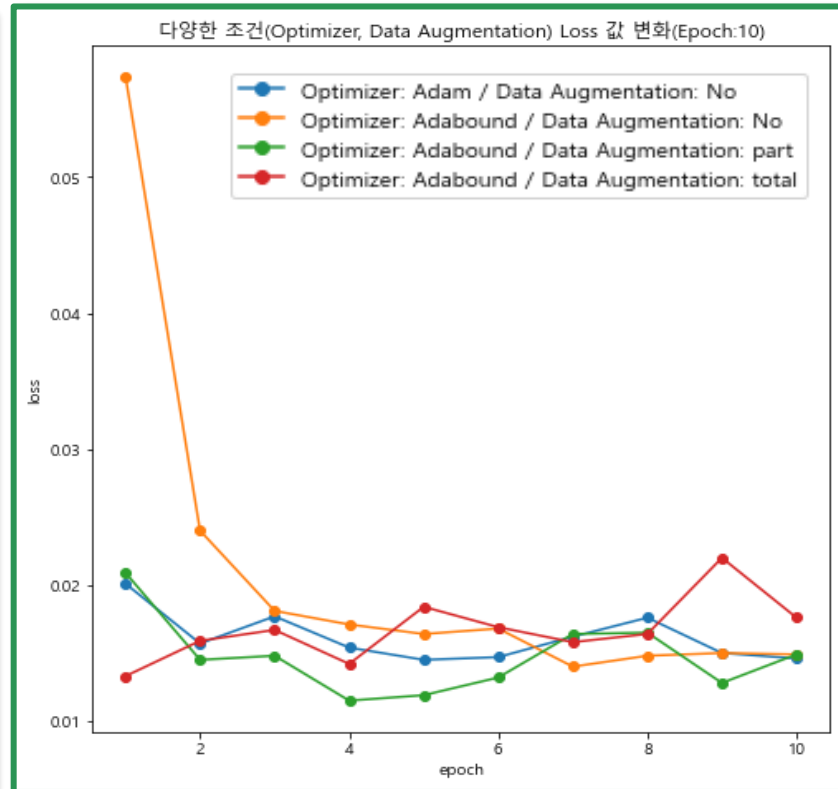
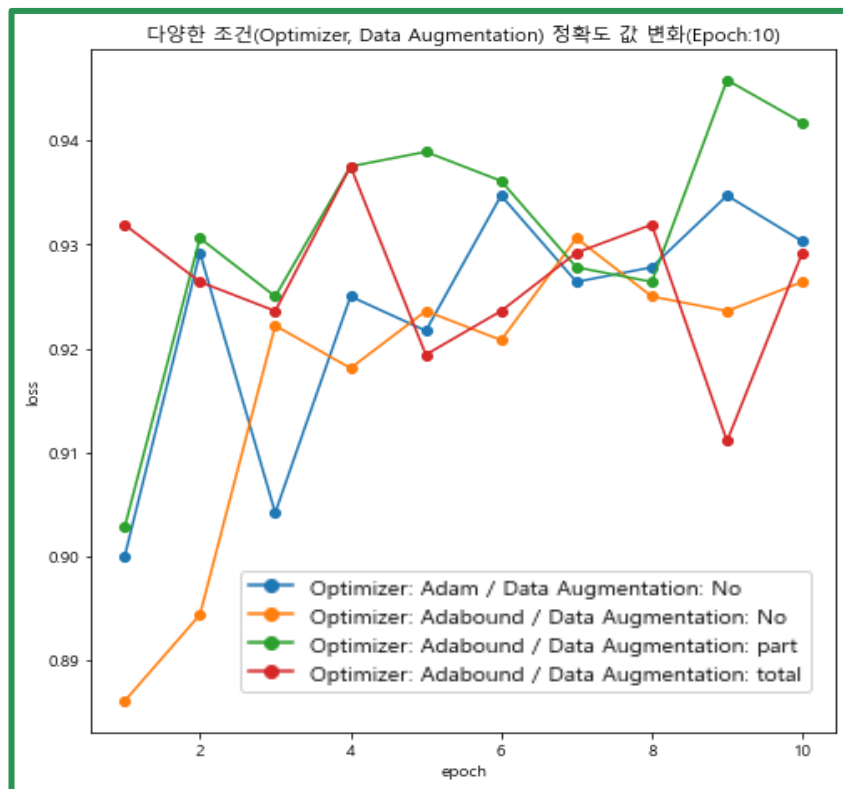
Class에 대한 부분 적용

Learning rate= 5e-4

freezing lambda: 30

## 6. 실험 및 결과

### “실험 결과(1) Classification”



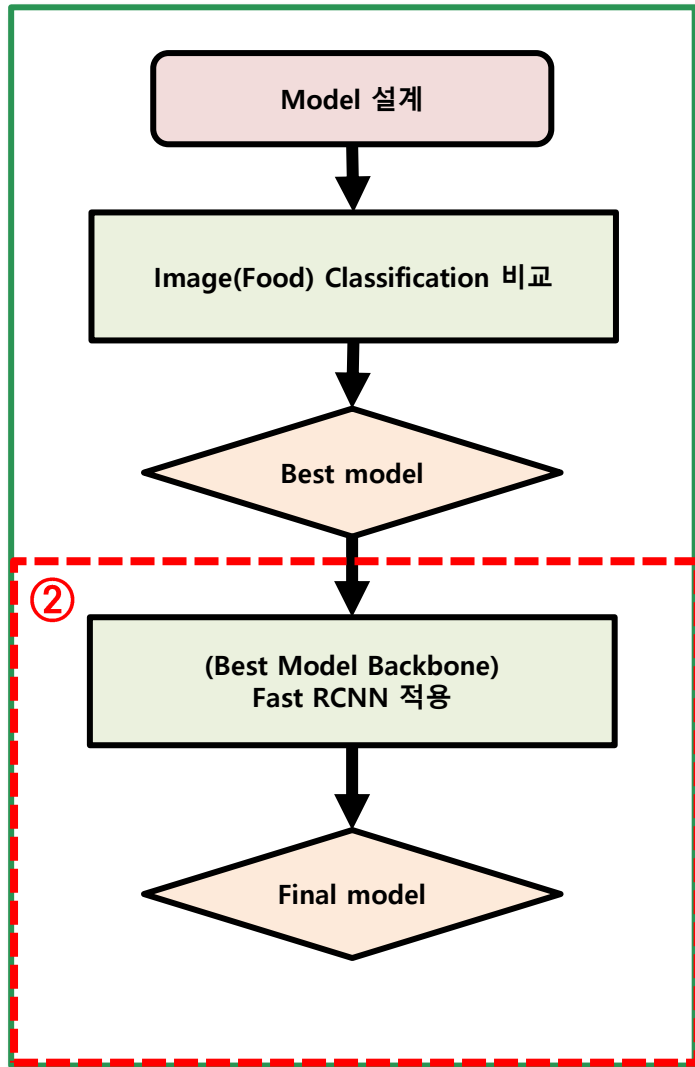
[ Augmentation 적용 방식에  
따른 Class별 정확도 ]

[ Optimizer 및 Augmentation 결과 ]

적용 하지 않음	부분 적용	전체 적용
Accuracy of 밥 : 100 %	Accuracy of 밥 : 100 %	Accuracy of 밥 : 100 %
Accuracy of 잡곡밥 : 66 %	Accuracy of 잡곡밥 : 33 %	Accuracy of 잡곡밥 : 100 %
Accuracy of 김치볶음밥 : 100 %	Accuracy of 김치볶음밥 : 100 %	Accuracy of 김치볶음밥 : 100 %
Accuracy of 전주비빔밥 : 100 %	Accuracy of 전주비빔밥 : 66 %	Accuracy of 전주비빔밥 : 100 %
Accuracy of 짜장면 : 85 %	Accuracy of 짜장면 : 0 %	Accuracy of 짜장면 : 100 %
Accuracy of 크림파스타 : 40 %	Accuracy of 크림파스타 : 50 %	Accuracy of 크림파스타 : 100 %
Accuracy of 시래기된장국 : 100 %	Accuracy of 시래기된장국 : 100 %	Accuracy of 시래기된장국 : 0 %
Accuracy of 떡갈비 : 20 %	Accuracy of 떡갈비 : 20 %	Accuracy of 떡갈비 : 75 %
Accuracy of 파전 : 0 %	Accuracy of 파전 : 66 %	Accuracy of 파전 : 100 %

## 6. 실험 및 결과

### “실험 시나리오(2) Detection”



[실험 시나리오]

#### [ 실험(2) 및 비교 모델 ]

학습한 Pre-Trained 모델이 Fast RCNN Backbone으로 유용한지 판단

→ Pytorch 제공한 Pre-trained Vgg16과 Inception v3를 Backbone으로 사용한 모델과 비교

→ 추가 : 학습한 Pre-trained 모델의 Fine tuning도 진행

평가지표: Bounding Box Regressor loss, Classification loss 활용

→ Image Classification 학습한 모델의 가중치(Parameter)를 Fast-RCNN의 Backbone 활용

## 6. 실험 및 결과

## “실험 결과(2) Detection”

학습시킨 Pre-trained Model 중 최적의 모델을 Detection Backbone 적용

비교 모델: Pre-trained Vgg16과 Inception v3

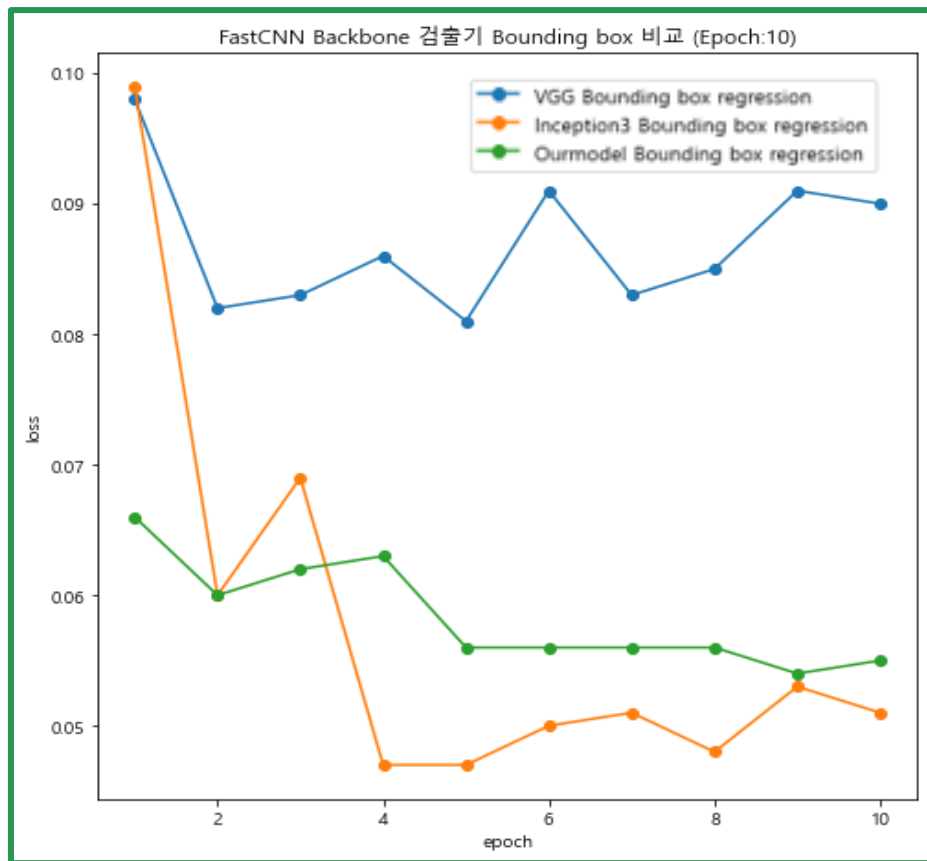
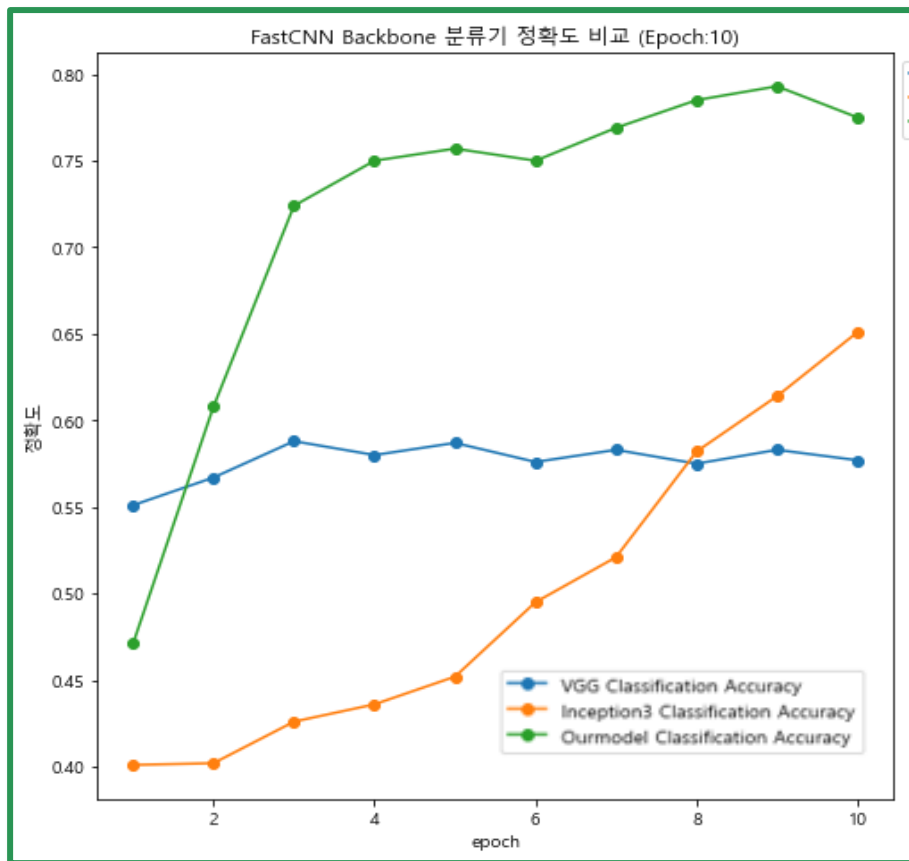
평가 척도 모델[Tuning]	Top (epoch)		Final (epoch)	
	Accuracy (Classification)	Loss (Bounding Box)	Accuracy (Classification)	Loss (Bounding Box)
제안 모델 Backbone (학습된 Pretrained Model)	0.793	0.054	0.775	0.054
Vgg-16 Backbone	0.588	0.083	0.577	0.09
Inception v3 Backbone	0.651	0.051	0.651	0.051



## 6. 실험 및 결과

### “실험 결과(2) Detection”

각 모델 별 정확도와 bbox regression loss 값 변화 Plot (왼쪽: 정확도, 오른쪽: Regression loss)



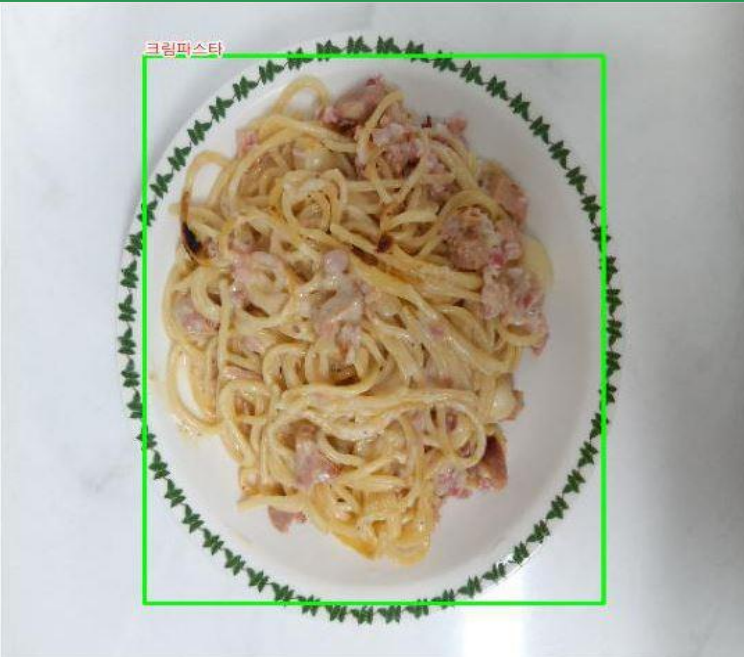
제안 모델이 **분류 부분**에서는 상대적으로 우수한 성능을 보였으나

검출 부분에서는 기존의 Inception v3 모델과 큰 성능 차이를 보여주지 못함

## 6. 실험 및 결과

## “실제 음식 이미지 적용”

### ▪ Single Label 데이터



index	영양성분	mg/g	%(kcal)
0	나트륨	950.0mg	0%
1	탄수화물	0.0g	0.0%
2	당류	0.0g	0%
3	지방	0.0g	0.0%
4	트랜스지방	0.0g	0%
5	포화지방	3.9g	0%
6	콜레스테롤	0.0mg	0%
7	단백질	14.0g	11.91%

[ 크림파스타: 470.0kcal ]

### ▪ Multi Label 데이터



index	영양성분	mg/g	%(kcal)
0	나트륨	703.12mg	0%
1	탄수화물	46.7g	71.85%
2	당류	7.9g	0%
3	지방	5.3g	18.35%
4	트랜스지방	0.0g	0%
5	포화지방	0.5g	0%
6	콜레스테롤	4.21mg	0%
7	단백질	6.3g	9.69%

[ 떡볶이: 260.0kcal ]

## 6. 실험 및 결과

## “한계 및 개선방향”

### [ 연구 한계점 ]

1. Computing Power 한계: (학습에 사용한) 음식의 Category 및 사진 수가 적음
  2. 음식 양 추정: 영양정보를 계산할 때 음식의 양과 상관없이 1인분 기준으로 진행
  3. 음식 분류 정확도는 잘 나오지만 bbox detector의 성능이 높지 않음
- Classification, Detector를 따로 학습시켜서 결합하여 더 높은 성능 기대

### [ 개선 및 제언 방향 ]

1. 음식 양 추정 모델 학습: 음식 양 추정 모델 개발, 음식 양의 맞춤형 영양정보 제공
2. Attention Module 추가: 여러 음식이 겹치는 경우 Box 중심부에 집중하여 성능 향상
3. 특정 객체를 어떤 음식과 함께 나오는지 보고 분류하는 Context aware 모델 활용
4. Object Detector의 Binary Classification화

## 참고자료

### ☑ 기존 식단 관리 상용화 및 문제점

<https://ddunang2.wixsite.com/portfolio/lotte-dutyfree>

[https://jhealthmedia.join.com/article/article\\_view.asp?pno=21928](https://jhealthmedia.join.com/article/article_view.asp?pno=21928)

<http://www.maeilmarketing.com/news/articleView.html?idxno=6642>

<http://www.datasom.co.kr/news/articleView.html?idxno=98904>

<http://www.dmorning.kr/news/articleView.html?idxno=28453>

### ☑ 한식 이미지 데이터 셋

<https://aihub.or.kr/aidata/27674>

<https://aihub.or.kr/aidata/13594>

<https://aihub.or.kr/aidata/30747>

### ☑ 선행 연구자료

<https://tech.kakaoenterprise.com/84>

<https://arxiv.org/pdf/1903.00858.pdf>

<https://arxiv.org/pdf/1909.05994.pdf>

<https://dl.acm.org/doi/abs/10.1145/3063592>

**감사합니다**

