

2021년도 1학기 IT융복합종합설계 최종보고서

영상처리 기반 냉장고 관리 모바일 어플리케이션



팀명: 영상철희
팀장: 한 상 현(201615434)
팀원: 김 영 한(201610878)

프로젝트 기간: 2021년 3월 2일 ~ 6월 2일




전북대학교 IT정보공학과

2021년도 1학기 IT융복합종합설계 최종보고서 제출문

소속	IT정보공학과		수강년도/학기	2021년도 1학기		
성명	학번	학년	이메일	비고		
한상현	201615434	4	luck2901@naver.com	팀장		
김영한	201610878	4	soosungp33@gmail.com	팀원		
				팀원		
프로젝트 제목	영상처리 기반 냉장고 관리 모바일 어플리케이션 refrigerator management mobile application					
프로젝트 수상내역	구분	세부구분	수여기관	수상내용		
	공모전	전국규모 공모전 수상(장/차관)				
		전국규모 공모전 수상(장려상)				
		전국규모 공모전 입선				
	논문	전국규모 논문 수상(최우수/우수)				
		전국규모 논문 수상(장려상)				
		전국규모 논문발표				
	교내	교내행사 수상				
		교내행사 참가				
	학과	학과행사 수상				
		학과행사 참가				
	기타					
	※ 수상내용 증빙자료 별첨					
	<div style="display: flex; justify-content: space-between; align-items: flex-end;"> <div style="text-align: center;"> <p>2021년 6월 2일</p> <p>팀장 한상현</p> <p>프로젝트 지도교수 곽영태</p> <p>전북대학교 IT정보공학과장 귀하</p> </div> <div style="text-align: right;">   </div> </div>					

2021년도 1학기 IT융복합종합설계 최종보고서

소속	IT정보공학과		수강년도/학기	2021년도 1학기	
성명	학번	학년	이메일	비고	
한상현	201615434	4	luck2901@naver.com	팀장	
김영한	201610878	4	soosungp33@gmail.com	팀원	
				팀원	
프로젝트 제목	영상처리 기반 냉장고 관리 모바일 어플리케이션 refrigerator management mobile application				
프로젝트 개요	<p>1인 가구의 증가와 코로나 19로 인해 집에서의 식사 빈도가 많아지고 있습니다. 이로 인해 집에 음식 재료를 많이 보관해야 할 일이 많아졌는데 냉장고에 있는 재료를 100% 사용하지 못하며 유통기한을 관리하지 못해 버리는 경우가 많습니다. 저희가 만든 어플리케이션은 이런 문제점들을 해결할 수 있는 어플리케이션으로 냉장고의 재료를 이미지 인식하여 내부의 재료를 정리 및 한눈에 살펴볼 수 있는 어플리케이션입니다.</p>				
<p>본 팀은 위와 같이 2021년도 1학기 IT융복합종합설계 최종보고서를 제출합니다.</p> <p style="text-align: center;">2021년 6월 2일</p> <p style="text-align: right;">팀장 한상현 </p> <p>전북대학교 IT정보공학과장 귀하</p>					

목 차

1. 서론

- 1.1 프로젝트 목적 및 필요성
- 1.2 프로젝트 목표 및 범위
- 1.3 조 편성 및 역할

2. 본론

- 2.1 문제 정의
- 2.2 문제 해결 아이디어
- 2.3 관련 기술의 현황

3. 프로젝트

- 3.1 시스템 개요
- 3.2 설계 조건
- 3.3 개발 환경
- 3.4 시스템 기능
- 3.5 사용자 인터페이스
- 3.6 실험 및 평가

4. 결론

- 4.1 프로젝트 결과물
- 4.2 기대 효과 및 활용 방안
- 4.3 프로젝트 수행후기 및 건의사항

참고 문헌

1. 서론

1.1 프로젝트 목적 및 필요성

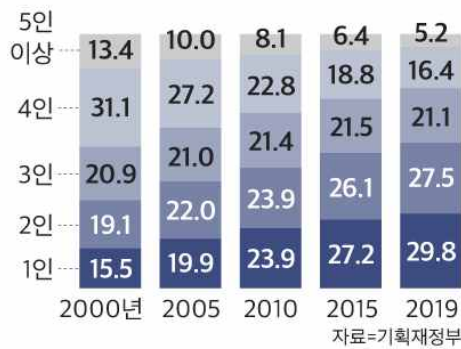
- 통계청에 따르면 1인 가구 비중 추이는 2019년 기준 30%이며 1인 가구의 수는 미래에도 꾸준히 증가할 것으로 예측되었으며 이에 따라 1인 가구 시장 또한 확장될 것으로 기대됩니다.

1인 가구 비중 추이와 가구원수 별 비중

1인 가구 비중 추이 장래가구추계



가구원수별 비중 장래가구추계



- 또한 최근 지속적인 인기를 끌고 있는 쿡방, 먹방과 함께 요리정보에 대한 수요도 급증하고 있는 추세입니다. 요리정보에 대한 수요는 가족 단위의 가구에서만 많은 것 같지만 1인 가구의 수요도 이에 못지않습니다.
- 따라서 위의 문제를 위한 서비스가 필요할 것으로 요망되는데 기존에 많은 요리 추천 어플리케이션이 존재하지만 이러한 어플리케이션의 추천 방식은 사용자의 냉장고 상황 즉, 재료를 고려하지 못한다는 점에서 한계가 있습니다.
- 이에 저희 프로젝트는 냉장고 내부의 재료를 이미지 인식하여 재료를 관리 용이, 재료에 따른 레시피를 추천하는 서비스를 제공합니다.

1.2 프로젝트 목표 및 범위

· 프로젝트 목표 : 사용자들이 냉장고 관리와 재료 선택을 수월하게 함으로써 음식물 쓰레기를 줄이고 효율적인 식습관을 가질 수 있다.

· 수행할 범위 : 가정 혹은 조직등 냉장고가 사용되는 모든 환경에서 사용 가능.

1.3 조 편성 및 역할

구성원	성명	역할	한 일
팀장	한상현	팀장	프로젝트 진행 일정을 조율
		설계, Front_End	프로젝트의 설계와 Front를 구현
		발표	프로젝트 발표를 담당
팀원	김영한	요구 분석	프로젝트의 요구 분석 작성
		Back_End	프로젝트의 Back을 구현
		보고서 작성	프로젝트의 보고서 작성

2. 본론

2.1 문제 정의

- 1인 가구 증가와 코로나 19로 인해 집에서 식사 빈도가 많아지고 있다.
- 냉장고에 무슨 재료가 있는지 한 눈에 파악하기 어려워서 재료를 100% 사용하지 못해 음식물 쓰레기 배출량이 늘어난다.
- 기존에 많은 요리 추천 어플리케이션은 사용자의 냉장고 상황 즉, 재료를 고려하지 못한다.

2.2 문제 해결 아이디어

- 냉장고에 무슨 재료가 있는지 한 눈에 파악하기 어려운 문제점을 해결하고자 저희 프로젝트는 냉장고 내부의 재료를 이미지 인식하여 재료를 관리 용이, 재료에 따른 레시피를 추천하는 서비스를 제공합니다.
- 기존의 냉장고 외부에서 사진을 찍은 뒤, 넣는 불편함을 보완하기 위해 냉장고 내부의 카메라를 통해 냉장고 내부의 재료를 한번에 인식.

2.3 관련 기술의 현황

가. 냉장고를 부탁해

- 구글플레이에 배포되어있는 모바일 어플리케이션.
- 냉장고 내부의 유통기한을 체크하는 기능.
- 한계 : 인공지능을 통한 재료 확인 기능이 아닌 사용자가 직접 상품을 등록/메모해야 함.

나. 밥타임

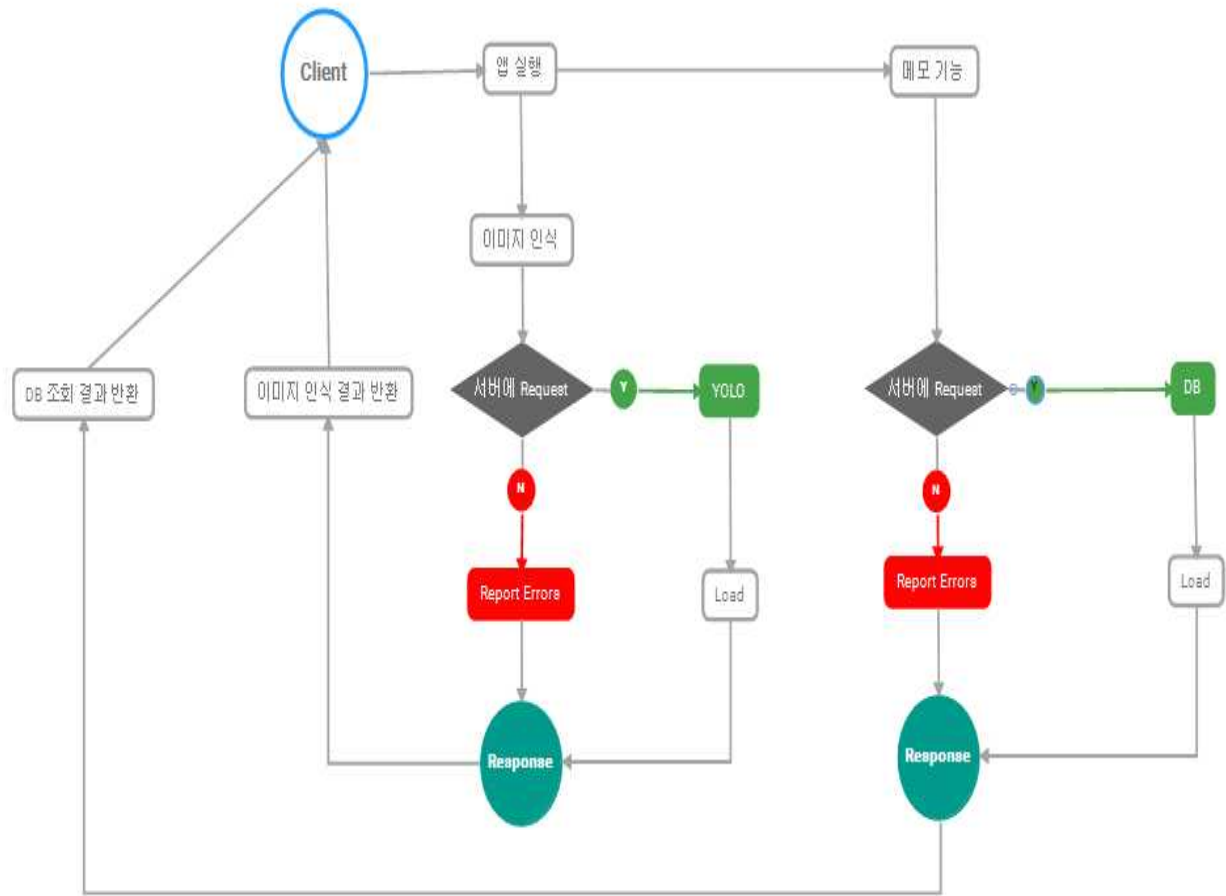
- 구글플레이에 배포되었다가 삭제된 모바일 어플리케이션.
- 냉장고 내부의 재료를 체크, 그에 따른 레시피 추천해주는 기능.
- 한계 : 위 어플리케이션과 마찬가지로 직접 상품을 등록/메모해야 함.

다. 삼성전자 2020년형 패밀리허브

- 삼성전자에서 세계 최대 IT, 가전전시회 CES 2020에 참가해 공개한 패밀리허브 냉장고.
- 패밀리허브 냉장고는 푸드 AI 기술을 적용하여 내부 식재료 자동 인식, 맞춤형 식단 및 레시피 제공, 식료품 온라인 주문 간편화 등을 실현.

3. 프로젝트

3.1 시스템 개요



3.2 설계 조건

1) 현실적 제한요소

제한요소	내용	고려여부
1. 산업표준	최신 트렌드 및 호환성 고려	■
2. 경제성(원가)	최적의 비용 추산 및 적용	■
3. 환경에 미치는 영향	설계 결과물이 환경오염을 유발할 가능성	■
4. 윤리성	정보 및 인터넷 윤리 고려	■
5. 안정성(안정성)	보안성 점검	□
6. 정치,사회에 미치는 영향	설계 결과물이 정치·사회적으로 미치는 영향	■

1. 산업 표준 : AOS, IOS 두 플랫폼에 쉽게 배포할 수 있는 React_native를 사용해 최신 트렌드 기술을 이용했다.
2. 경제성 : 웹 카메라 구매 비용과 서버 운영비만 발생한다.
3. 환경에 미치는 영향 : 음식물 쓰레기 배출량을 줄여서 오히려 환경에 도움을 준다.
4. 윤리성 : 인터넷 데이터를 사용하지 않고 실제 재료들을 구매해 찍은 데이터로 학습했다.
5. 안정성 : 회원가입 부분에서 일반적인 db에 저장하는 방식이라 보안성 문제가 발생할 수 있을 것 같다.
6. 정치, 사회에 미치는 영향 : 사회적 이슈인 코로나 19에 조금이나마 도움이 되고자 한다.

2) 컴퓨팅 문제수준 설명

구분	문제의 속성	컴퓨팅문제수준설명	부합여부 (●, △, X)
1	지식의 깊이	최신 정보와 관련 연구 결과를 활용하고 있다.	●
2	상충되는 요건의 범위	상충될 수 있는 기술적 또는 컴퓨팅적 이슈를 다루고 있다.	X
3	분석의 깊이	해답이 명확하지 않은 문제를 해결하기 위해 깊이 있는 사고와 분석과정을 다루고 있다.	●
4	생소한 주제	자주 접하지 않는 컴퓨팅 문제를 다루고 있다.	●
5	문제의 범위	전공분야의 일반적인 실무 영역을 벗어난 범위를 다루고 있다.	△
6	이해당사자의 요구 수준 및 범위	다양한 이해당사자들의 요구사항들을 고려하고 있다.	●
7	상호의존성	상호 의존적인 여러 세부문제들이 결합된 종합적인 문제로 구성되어 있다.	●
8	다양한 영향 고려	다양한 분야에 미치는 영향을 고려하고 있다.	X

1. 지식의 깊이

- 최신 정보와 관련 연구 결과를 활용하고 있습니다.
- React_Native, IntelliJ, Flask로의 최신 클라이언트/서버 구축 및 YOLO의 이미지 인식의 최신 연구 결과를 사용.

2. 상충되는 요건의 범위

- 상충되는 공학적 요건 및 현실적 제한 조건에 대해서는 해당되지 않는다.

3. 분석의 깊이

- 기존의 참고자료에는 안드로이드 스튜디오와 YOLO의 연결에 대한 자료만 존재했지만, 기존 계획대로 진행하기 위해 YOLO 모델을 FLASK로 연결시켜 해답이 명확하지 않은 문제임에도 해결.
- 이미지 학습 모델을 핸드폰 내부로 포함시키는 것이 아닌(많은 용량 차지) 독립적으로 두어 서버로 연결

4. 생소한 주제

- 해당 주제는 이미지 인식에 대한 접근으로는 생소하다고 볼 수는 없지만 기존의 존재하지

않는 애플리케이션을 만듦에 따라 이미지 인식에 따른 냉장고 정리 어플이라는 접근으로는 생소한 주제라고 생각.

5. 문제의 범위

- 전공 분야에서 어쩌면 가장 일반적일 수도 있는 클라이언트/서버 프로젝트를 함에 전공에 벗어나지 않는다.
- YOLO 또한 영상처리분야에서는 많이 사용되는 기술 스택으로 전공에 크게 벗어나지 않는다고 판단된다.

6. 이해 당사자의 요구 수준 및 범위

- “냉장고”라는 사물은 여러 사람들이 쓰는 만큼 여러 사람들에게서의 요구사항을 모아 요구사항 정의를 함.
- 보다 넓은 요구 수준과 범위를 만족시키도록 노력.
- 기존의 냉장고 외부에서 사진을 찍은 뒤, 넣는 불편함을 보완하기 위해 냉장고 내부의 카메라를 통해 냉장고 내부의 재료를 한번에 인식.

7. 상호의존성

- 냉장고 내부의 재료의 유통기한을 정리하기 어렵다는 점, 냉장고 내부에 어떤 재료가 있는지 정리하기 어렵다는 점, 해당 재료를 통해 만들 수 있는 레시피를 찾기가 어렵다는 세부분제들을 결합하여 종합적인 문제로 구성.

8. 다양한 영향 고려

- 냉장고 내부의 공간 크기에 따른 학습이 진행되지 않음에 다양한 영향을 고려하지 않는다고 판단.
- 추후 여러 높이에서 학습을 진행하여 다양한 영향에 대해서 고려할 계획.

3.3 개발 환경

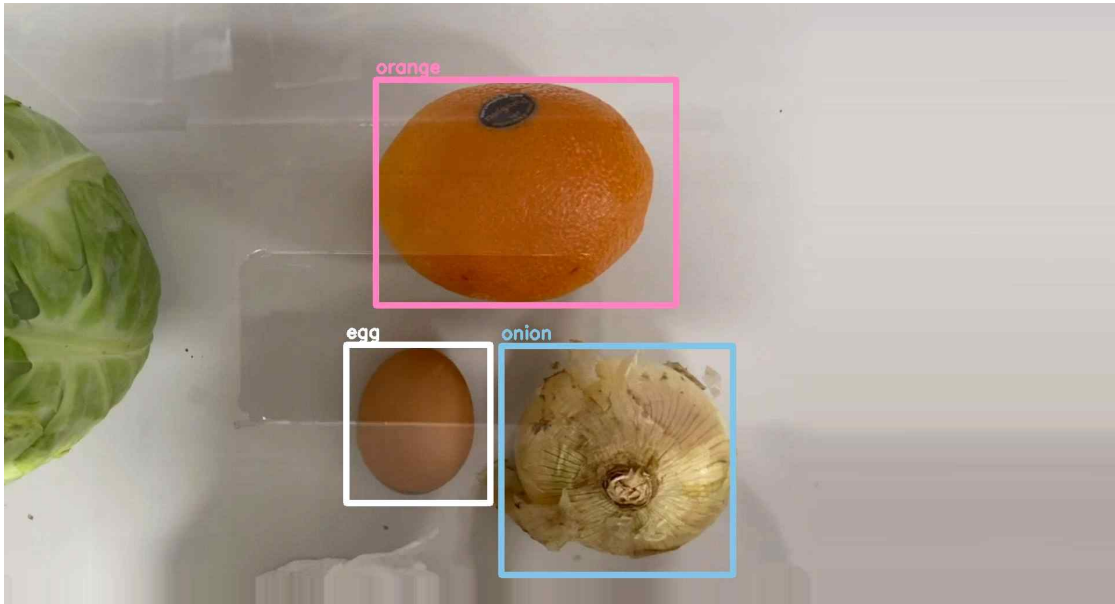
- H/W : 웹 카메라
- S/W
 - 개발 환경 : VS Code, IntelliJ, Pycharm
 - OS : Window – 이미지 학습, Mac – 개발

3.4 시스템 기능

1. 시스템은 로그인 및 로그아웃 기능을 사용한다. 로그인한 정보를 DB의 기본키로 사용하여 나중에 이미지 인식한 데이터를 기본키에 맞는 DB에 저장한다.

2. Object Detection 기능

2.1. YOLO 기술을 통해 냉장고 내부의 재료를 이미지 인식.



```
Anaconda Prompt (anaconda3) - python flask_rest.py
Load | Yep! | maxp 2x2p0.2 | | (? , 52, 52, 64)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 52, 52, 128)
Load | Yep! | maxp 2x2p0.2 | | (? , 26, 26, 128)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 26, 26, 256)
Load | Yep! | maxp 2x2p0.2 | | (? , 13, 13, 256)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 13, 13, 512)
Load | Yep! | maxp 2x2p0.1 | | (? , 13, 13, 512)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 13, 13, 1024)
Init | Yep! | conv 3x3p1_1 +bnorm leaky | | (? , 13, 13, 1024)
Init | Yep! | conv 1x1p0.1 linear | | (? , 13, 13, 50)

Running entirely on CPU
2021-06-01 21:17:26.085783: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2
2021-06-01 21:17:26.218760: I tensorflow/core/common_runtime/gpu/gpu_device.cc:984] Device interconnect StreamExecutor with strength 1 edge matrix:
2021-06-01 21:17:26.229719: I tensorflow/core/common_runtime/gpu/gpu_device.cc:990]
Loading from ./ckpt/my-tiny-yolo-123402
WARNING:tensorflow:From C:\Users\User\anaconda3\envs\GPW\lib\site-packages\tensorflow\python\training\saver.py:1266: checkpoint_exists (from tensorflow.python.training.checkpoint_management) is deprecated and will be removed in a future version.
Instructions for updating:
Use standard file APIs to check for files with this prefix.
Finished in 7.448784589767456s

[{'label': 'egg', 'confidence': 0.7865716, 'topleft': {'x': 589, 'y': 589}, 'bottomright': {'x': 837, 'y': 862}}, {'label': 'onion', 'confidence': 0.91486204, 'topleft': {'x': 856, 'y': 591}, 'bottomright': {'x': 1255, 'y': 986}}, {'label': 'orange', 'confidence': 0.90823495, 'topleft': {'x': 640, 'y': 132}, 'bottomright': {'x': 1157, 'y': 521}}]
210.117.181.74 - - [01/Jun/2021 21:17:32] "GET /detection HTTP/1.1" 200 -
```

이미지 인식 결과값 Json 반환

2.2. 인식된 재료는 사용자 값에 맞는 DB로 저장된다.

2.3. 학습할 자료는 총 5개로 직접 촬영 후 Image Augmentation을 통해 데이터 수를 늘려 학습했다.

```
Anaconda Prompt (anaconda3) - python flow --model ./cfg/my-tiny-yolo.cfg --labels ./labels.txt --trainer adam --dataset ./data/dataset/ --annota...
Use standard file APIs to check for files with this prefix.
Finished in 12.38413119316101s

Enter training ...

./cfg/my-tiny-yolo.cfg parsing ./data/annotations/
Parsing for ['egg', 'onion', 'orange', 'cabbage', 'cucumber']
[=====>]100% image378.xml
Statistics:
cabbage: 170
cucumber: 146
egg: 141
orange: 239
onion: 120
Dataset size: 364
Dataset of 364 instance(s)
Training statistics:
Learning rate : 0.0001
Batch size : 5
Epoch number : 100
Backup every : 50
2021-06-01 21:19:51.843834: W tensorflow/core/common_runtime/bfc_allocator.cc:211] Allocator (GPU_0_bfc) ran out of memo
ry trying to allocate 2.16GiB. The caller indicates that this is not a failure, but may mean that there could be perform
ance gains if more memory were available.
2021-06-01 21:19:51.897826: W tensorflow/core/common_runtime/bfc_allocator.cc:211] Allocator (GPU_0_bfc) ran out of memo
ry trying to allocate 2.30GiB. The caller indicates that this is not a failure, but may mean that there could be perform
ance gains if more memory were available.
2021-06-01 21:19:52.013185: W tensorflow/core/common_runtime/bfc_allocator.cc:211] Allocator (GPU_0_bfc) ran out of memo
ry trying to allocate 2.25GiB. The caller indicates that this is not a failure, but may mean that there could be perform
ance gains if more memory were available.
```

Learning Rate

```
Anaconda Prompt (anaconda3) - python flow --model ./cfg/my-tiny-yolo.cfg --labels ./labels.txt --trainer adam --dataset ./data/dataset/ --annota...
step 123412 - loss 1.9840996265411377 - moving ave loss 1.4767129916449802
Checkpoint at step 123412
step 123413 - loss 1.6742026805877686 - moving ave loss 1.496461960539259
step 123414 - loss 1.754169225692749 - moving ave loss 1.522326870546081
step 123415 - loss 2.0710368156433105 - moving ave loss 1.5771130999134784
step 123416 - loss 2.8265585899353027 - moving ave loss 1.7020576489156607
step 123417 - loss 1.7200384140014648 - moving ave loss 1.7038557254242412
step 123418 - loss 1.0268466472625732 - moving ave loss 1.6361548176080745
step 123419 - loss 1.7446813583374023 - moving ave loss 1.6470074716810075
step 123420 - loss 2.8045036792755127 - moving ave loss 1.762757092440458
step 123421 - loss 1.022149920463562 - moving ave loss 1.6886963752427686
step 123422 - loss 2.022873878479004 - moving ave loss 1.722114125566392
Checkpoint at step 123422
step 123423 - loss 1.798977255821228 - moving ave loss 1.7298004385918757
step 123424 - loss 2.0090267658233643 - moving ave loss 1.7577230713150247
step 123425 - loss 2.303776264190674 - moving ave loss 1.8123283906025895
step 123426 - loss 1.2557443380355835 - moving ave loss 1.756689985345889
step 123427 - loss 1.984492301940918 - moving ave loss 1.779452217005392
step 123428 - loss 0.5924832224845886 - moving ave loss 1.6607553175533116
step 123429 - loss 2.6174163818359375 - moving ave loss 1.7564214239815743
step 123430 - loss 2.1706490516662598 - moving ave loss 1.797844186750043
step 123431 - loss 1.5046789646148682 - moving ave loss 1.7685276645365255
step 123432 - loss 0.7569625377655029 - moving ave loss 1.6673711518594232
Checkpoint at step 123432
step 123433 - loss 1.487121343612671 - moving ave loss 1.6493461710347481
step 123434 - loss 1.6495281457901 - moving ave loss 1.6493643685102835
step 123435 - loss 1.4993950128555298 - moving ave loss 1.6343674323448081
step 123436 - loss 2.8094513416290283 - moving ave loss 1.7518758238132301
step 123437 - loss 1.641626623008728 - moving ave loss 1.74085110373278
```

이미지학습

2.4. 본 DB의 데이터는 추후에 자료 선택 기능에서 사용되어진다.

2.5. DB에서 데이터는 사용자의 어플리케이션 FrontEnd로 전해지며, 자료의 유무에 따라 자료의 UI를 어둡거나 밝게 보여진다.

2.6. 냉장고 내부의 카메라 -> 자료 인식 -> 학습된 모델로 이동 -> 자료의 인식 후 DB에 저장 -> 저장된 정보를 Front에서 보여줌.

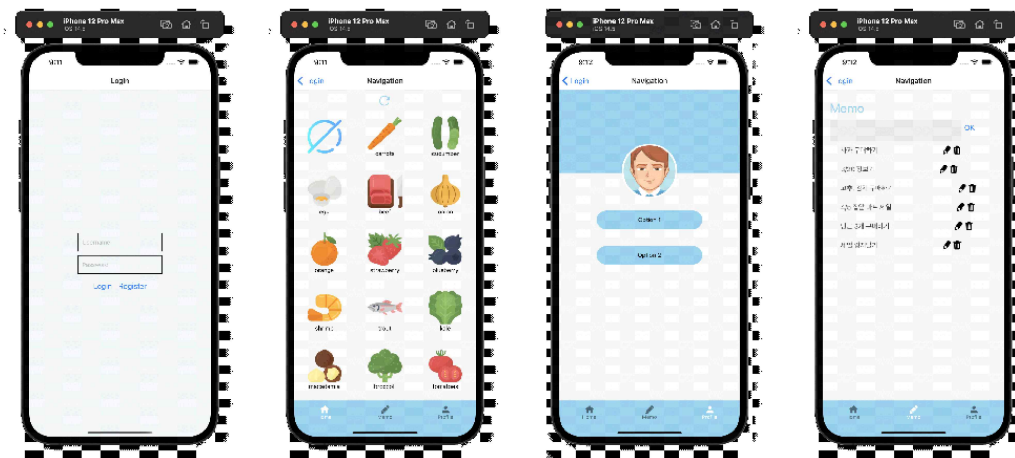
3. 자료 선택 기능

- 3.1. 검출된 재료 리스트에서 재료를 눌렀을 때 재료의 이름, 유통기한 등 관련 정보들을 나타낸다.
- 3.2. 검출된 재료 리스트에서 재료를 눌렀을 때 재료로 만들 수 있는 요리 정보를 나타낸다.
- 3.3. 만들기 쉽고 대중적인 요리를 우선으로 추천해준다.

4. 메모 기능

- 4.1. 레시피 정보, 장 볼거리 등 잠깐 정보들을 적어둬야 할 때 사용할 수 있는 기능이다.

3.5 사용자 인터페이스

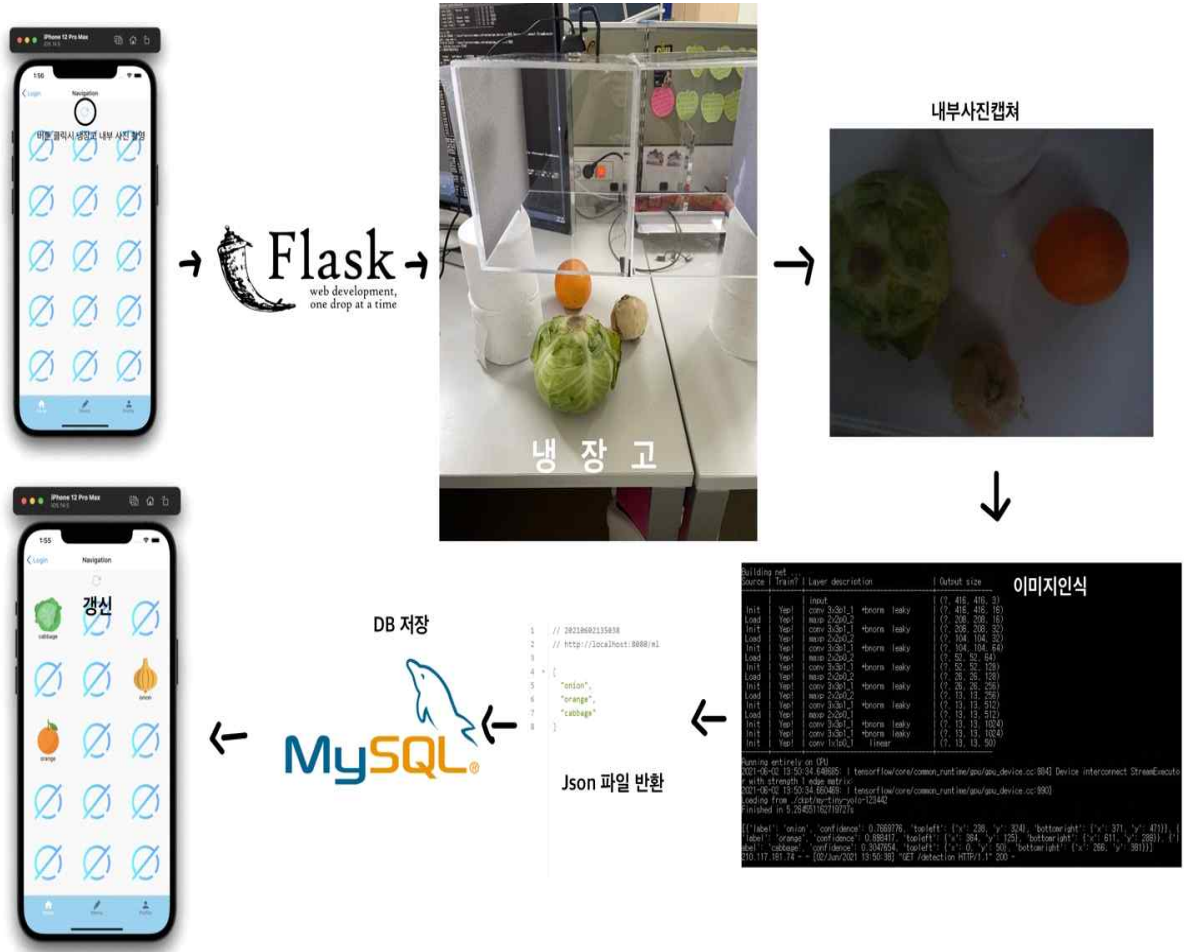


3.6 실험 및 평가

- 이미지 검출은 달걀, 양파, 오렌지, 양배추, 오이 총 5개를 기준으로 한다.
 - 학습한 데이터와 동일한 거리에 카메라를 설치하고 재료가 온전히 카메라에 나오게 테스트했을 때 결과는 매우 양호했다.
 - 학습한 데이터와 동일한 거리에 카메라를 설치하고 재료가 반쯤 가려진 상태로 카메라에 나오게 테스트했을 때 결과는 오이같은 길쭉한 재료는 검출이 잘 되는 편이었지만 그 외에 둥근 형태의 재료는 검출을 잘 하지 못했다.
 - 학습한 데이터보다 멀리 카메라를 설치하고 테스트 했을 때에는 오이같이 둥글지 않은 재료는 검출이 잘 되었지만 둥근 재료는 크기가 큰 재료를 크기가 작은 재료로 인식하는 경우가 있었다.
 - ex) 양배추를 오렌지로 인식
- 결과적으로 디테일한 부분까지 검출하기 위해서는 질 좋은 데이터와 많은 학습 시간이 더 필요할 것 같다.

4. 결론

4.1 프로젝트 결과물



4.2 기대 효과 및 활용 방안

• 기대 효과

- 코로나 19로 인한 외식 빈도의 하락세와 각종 먹방의 인기가 계속해서 상승하고 있습니다. 더불어 1인 가구의 수는 지속적으로 증가하는 추세입니다.
- 이 프로젝트는 재료 기반을 통해 실용적인 추천 서비스를 제공함으로써 기존 요리 추천 서비스와 차별점을 가집니다.
- 냉장고에 남아 있는 재료를 효율적으로 활용할 수 있는 방법을 제시해 음식물 쓰레기 배출량을 최소화할 수 있습니다.
- 이 프로젝트를 통해 증가하는 요리 정보의 수요를 충족시킬 수 있으며 요리가 미숙

한 1인 가구, 20~30대의 불편함을 해결해줄 수 있을 것입니다.

- 활용 방안

- 사용자의 기호를 고려하지 않은 프로젝트이므로 사용자의 식습관 패턴을 수집하고 분석해 다양한 마케팅부분에서 활용할 수 있습니다.
- 프로젝트에 사용된 이미지 인식 기술을 이용해 요리 추천뿐만 아니라 다양한 서비스를 개발할 수 있습니다.
- 가정 냉장고 관리 : 냉장고내 음식물 관리 및 요리 추천 서비스
- 대형 마트 냉장고 재고관리 및 유통기한 관리

4.3 프로젝트 수행 후기 및 건의사항

- 이번 프로젝트를 통해 이미지 검출하는 방법과 소스 코드는 많아도 실제로 학습시키고 원하는 결과가 나올 때까지 시간이 많이 걸리고 생각보다 정확한 결과가 나오기 힘들다는 것을 느꼈습니다. 가장 처음에는 크롤링을 통해 질보단 양의 데이터를 가지고 학습했는데 해당 방법으로는 테스트에서 검출된 경우가 극히 드물었습니다. 따라서 양보단 질의 방향으로 다시 계획을 수정하고 학습 후 테스트해보니 학습시킨 데이터에 한해서는 결과가 잘 나오는 걸 확인할 수 있었습니다. 여기서 그치지 않고 이미지 Argumentation을 통해 질 좋은 데이터를 늘려 학습했고 결과적으로 학습시킨 데이터가 아닌 즉석에서 찍은 이미지로도 결과가 잘 나오게 되었습니다. 따라서 질 좋은 데이터가 많이 필요하다는 것을 느꼈고 필요에 따라 잘 수집하는 것이 중요한 것 같습니다.

참고 문헌

<https://junyoung-jamong.github.io/deep/learning/2019/01/22/Darkflow%EB%A5%BC-%ED%99%9C%EC%9A%A9%ED%95%B4-YOLO%EB%AA%A8%EB%8D%B8-%EC%9D%B4%EB%AF%B8%EC%A7%80-%EB%94%94%ED%85%8D%EC%85%98-%EA%B5%AC%ED%98%84-in-windows.html>

<http://euzl.blogspot.com/2019/10/darkflow.html>

<http://blog.naver.com/PostView.nhn?blogId=mingu216&logNo=221302426428>