

# 냉장고 속 재료 객체 인식을 통한 레시피 추천시스템

저자 1 박신혁<sup>○</sup> 저자 2 이정필 저자 3 이현호

저자 소속 아주대학교

저자1 qkr819@ajou.ac.kr, 저자2 ijpsm9@naver.com저자3hyunho0587@gmail.com

## An Recipe Recommend System with Finding Ingredients in Refrigerator Using Object detection

저자 1 Shin hyuk Park 저자 2 Jung Peel Lee 저자 3 Hyun Ho Lee<sup>○</sup>

저자 소속 Ajou Univ

### 요 약

본 논문에서는 convolution neural network을 이용하여 냉장고 속 식 재료 객체 인식을 통한 레시피 추천 서비스를 다루었다. 첫 번째로 객체 인식을 정확하게 하기 위해 어떠한 모델이 효과적 인지, 또 어떠한 학습 데이터 집합이 더 정확한 결과를 도출하는지를 연구하였다. 모델은 SSD(Single Shot Multibox Detector)와 faster R-CNN(Regions with Convolutional Neural Network Features)의 정확도 분석을 하였다. 데이터 집합에서는 각각의 데이터의 형태에 따른 분류로 학습을 시켜 더 효과적인 데이터 집합을 선택하였다. 두 번째로 인식된 재료들을 바탕으로 레시피를 추천 할 때 어떠한 추천방법이 적절한지에 대한 연구를 포함하였다. 종합하여 객체 인식의 정확도를 높이고, 적절한 레시피를 추천하는 방법을 다루었다.

### 1. 서 론

객체 인식(object detection)은 deep neural network를 이용하여 어떤 새로운 이미지에서 각 객체들을 찾아주는 기술이다.

기존 유사제품인 스마트 냉장고에서는 그 내부에 대한 전체 이미지를 한번에 인식하여 어떤 재료가 있는지에 대한 정보를 얻을 수 있는 방법은 아직 발표된 바가 없다. 이에 착안점을 두어 CNN(Convolution Neural Network)을 활용하여 물체를 인식하기 위한 모델 고안/인식한 물체를 바탕으로 레시피를 추천하는 서비스를 개발하였다.

서비스를 고안하기 위해 CNN 네트워크는 미리 학습된(Pre-trained) 모델을 사용하였으며, 객체 인식은 Tensorflow[1]를 사용하였다.

정확한 추천을 위해서는 정확한 인식이 바탕 되어야 한다. 따라서 먼저 객체 인식에서 어떤 요인들이 정확성에 영향을 미치는지 파악한 뒤, 인식하기 위한 모델에서 정확도를 높이기 위해 네트워크 별로 비교를 진행하고, 데이터 셋을 적합하게 바꾸었다.

정확성에 영향을 미치는 요인은 다양하다. 첫 번째로 어떠한 학습 모델의 영향이다. 본 논문에서는 SSD(Single Shot Multibox Detector)와 faster R-CNN(Regions with Convolutional Neural Network Features) 두 가지 모델의 성능을 비교하였다. 두 번째는 학습 데이터 집합의 영향이다. 학습 데이터는 실제 냉장고의 환경과 비슷한 데이터를 사용할수록 객체 인식에서 더 높은 정확성을 나타내었다.

또한 레시피 추천에서 어떠한 알고리즘을 사용하였을 때 사용자가 원하는 레시피에 가까운 결과를 출력하는지에 대한 방법을 연구하였다. 레시피 데이터는 각각의 재료에 대한 One-hot 벡터를 사용하여 생성하였으며, 각 재료에 대한 유무에 대한 판단으로 추천을 진행하였다. 이에 대한 방법으로 세 가지 알고리즘을 사용하였다. 첫 번째로 가지고 있는 재료가 레시피에 얼마나 포함되는지를 바탕으로 추천을 진행하였다. 두 번째로는 첫 번째 방식에서 포함된 재료의 수를 요리에 들어가는 재료의 개수로 나눠주어 높은 수치가 나오는 음식을 추천하였다. 세 번째는 첫 번째와 두 번째의 결과값을 곱하여 높은 값이 나오는 음식을 추천하는

방식이다. 이 세가지 추천방식 중 어떤 방법이 더 사용자의 기대에 가까운 레시피를 출력하는지를 연구하였다.

본 논문에서는 위 두 가지 관점에서 각각에 대해 서비스를 평가하고 데이터는 어떠한 데이터를 모아야 하는지를 확인하고자 한다.

## 2. 본 론

### 1. 모델의 선택

#### A. SSD(Single Shot Multibox Detector)

SSD는 네트워크 하나만 사용해서 오브젝트의 경계박스를 찾고 클래스를 인식한다.(Default box)이는 기초 네트워크(Base Network)위에 여러 Convolutional Layer를 쌓아서 output size가 점점 작아지는 구조이다. 각 레이어마다 사이즈가 달라지며, 그에 따른 그리드의 크기에 맞춰 경계박스를 찾아준다.따라서 각 사이즈 별로 다른 클래스를 확인 할 수 있다. 마지막으로 ssd는 여러 크기의 특징 맵(feature map)을 사용해 예측을 진행한다.[2] SSD의 구조는 Figure1 과 같다.

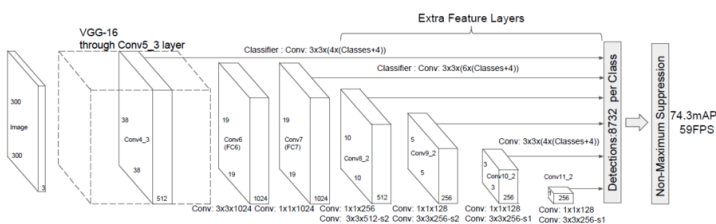


Figure1. structure of Ssd

#### B. Faster R-CNN(Regions with Convolutional Neural Network Features)

R-CNN은 이미지 입력을 받아서 Selective search로 2000개 정도의 region proposal을 추출한 다. 그 이후 각 region의 classification 결과와 bounding box regression을 계산한다. 이 때 SVM분류기를 사용한다.[3] 본 서비스에서는 R-CNN의 개선된 모델인 faster-RCNN을 사용한다. Faster R-CNN은 아래 그림과 같이 convolutional layer 에 RPN(Region Proposal Network)를 추가한 모델이다[4].

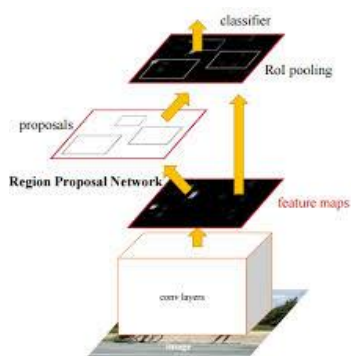


Figure2. Faster R-CNN

### 2. 데이터 선택

객체 인식의 정확성을 높이기 위해서는 실제로 들어 오는 input값과 비슷한 환경에서 많은 양의 데이터를 학습하여야 한다. 또한 학습시킨 데이터가 실제 사진과 얼마나 유사한지가 정확성에 가장 큰 영향을 미친다. 따라서 실제 학습할 때는 실제 냉장고의 사진으로 학습을 진행하였다. 또한 class의 개수가 많아질수록 실제 정확도는 더 낮아졌다.

따라서 정확성을 높이기 위해서는 첫째로, input과 유사한 환경의 데이터셋을 학습데이터로 사용하여야 한다. 또한 이 조건에 부합하는 데이터를 최대한 많이 모아서 사용하여야 한다.

### 3. 추천시스템

추천을 진행할 때 객체 인식이 어려운 재료에 대해서는 사용자의 input을 입력받아 추천을 진행하였다. 이에 알고리즘은 레시피 데이터를 재료별로 있는재료는 1 없는재료는 0으로 표현한 one-hot인코딩을 하여 실제 사용자의 냉장고와 비교를 진행하였다. 이후 각 레시피에 대한 점수를 매기는데 점수를 매기는 방법 중 세 가지 방법을 비교하였다. 첫째로 일치하는 재료의 개수만 점수에 포함하는 방법. 두번째로 일치하는재료의 레시피에 들어가는 전체재료개수로 나눠주는 방법. 세번째로 첫번째 방법의 점수와 두번째 방법의 점수를 곱해서 점수로 이용하는 방법이다.

### 4. 실험 결과

#### 1. 모델에 따른 object detection 실험 결과

##### A. SSD

SSD모델을 사용했을 때, 학습속도는 Faster R-CNN보다 빨랐다. 이는 실제 연산 양이 적기 때문이다. 하지만 Faster R-CNN보다 현저히 낮은 정확성을 보였다. 또한 loss graph는 0으로 잘 수렴하지 않았다. 이는 Figure4에서 확인이 가능하다.

$$L_{loc}(x, l, g) = \sum_{i \in Pos} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \quad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \quad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

Figure3. Loss function(SSD)

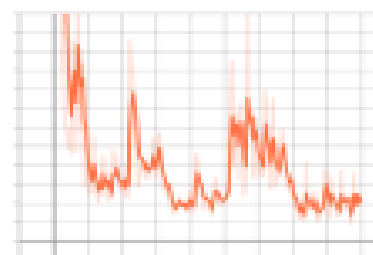


Figure4. Loss graph(SSD)

## B. Faster R-CNN

Faster R-CNN모델은 SSD보다 높은 정확성을 보였기에, 실제로 사용하기에 더 적합하였다. 최종 서비스를 구현할 때 Faster R-CNN을 사용하였다. Faster R-CNN의 loss function은 Figure5와 같으며 학습시킬때의 loss graph는 Figure6과 같다.

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

Figure5. Loss function(faster R-CNN)

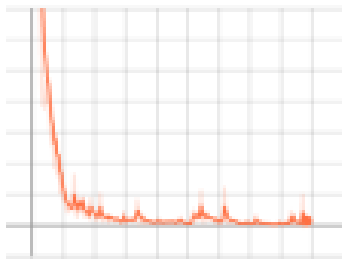


Figure6. Loss graph(faster R-CNN)

## 2. 데이터 집합에 따른 object detection실험 결과

## A. 하나의 object만 존재하는 데이터 집합

한 개 이미지에 한 개의 object만 존재하는 데이터 집합을 학습 데이터로 사용 하였을 때 실제 input 값에 대하여 10%이하의 매우 낮은 정확성을 보였다. 이 때 데이터 집합은 google image를 사용하였으며, 이 이미지들의 배경 화면은 실제 input 데이터의 환경과는 완전히 다른 환경이었다..

## B. 실제 input과 비슷한 데이터 집합

한 개 이미지에 대하여 여러 개의 object가 존재하는 데이터 셋을 사용하였을 때는 90%이상의 높은 정확성을 보였다. 이 때 직접 촬영한 데이터 집합을 사용하였다.

## 3. 추천시스템

추천시스템 알고리즘은 첫번째로 이미지에서 객체 인식된 재료들과 레시피 데이터를 일치하는 재료만 점수에 포함하는 방법을 사용하였다.

실제로 점수를 요리에 들어가는 재료의 개수로 나누는 방법은 재료가 적게 들어가는 요리를 무조건적으로 추천하는 문제가 생겼고, 이에 점수를 곱해서 사용하면 결과가 올바르게 도출되지 않았다.

따라서 요리재료 중 가지고 있는 재료의 개수만으로 추천시스템을 구현하였다.

## 4. 평가

## a. 이미지 인식에 대한 평가

구현은 가장 성능이 좋은 Faster R-CNN, 실제 input환경과 비슷한 데이터셋을 사용하여 실시하였다.

이미지 인식에서의 평가는 첫번째로 이미지가 잘 인식되는가? 두번째로 데이터를 학습시킬 때 loss값은

0으로 잘 수렴하는가? 두가지 측면을 평가의 기준으로 삼았다. Figure 7은 각각 이미지 인식이 잘 되었을 때와 아닐때의 모습을 보여준다.

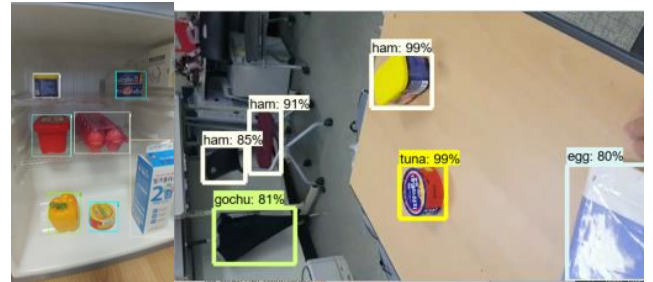


Figure7. Good&amp;Bad example (object detection)

## b. 레시피 추천에 대한 평가

추천시스템의 평가는레시피중 사용자 측면에서 만들수 레시피를 잘 추천해주는가?를 중점으로 평가하였다.

## 3.결 론

본 논문에서는 CNN(Convolution Neural Network)를 이용하여 object detection을 진행 후 냉장고 속에 있는 재료를 파악한 뒤 재료에 알맞은 레시피 추천 시스템에 대하여 다루었다.

객체 인식에서는 Faster R-CNN이 SSD보다 더 높은 정확성을 보였으며, 실제 input과 비슷한 학습 데이터 집합을 사용하였을 때 더 높은 정확성을 보였다.

추천시스템은 레시피에 들어가는 재료와 사용자가 가지고 있는 재료중 일치하는 개수가 가장 높은 요리를 추천해 줄 때 가장 적합한 요리를 추천해 주었다.

## 4.시사 문구

"본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학지원사업의 연구결과로 수행되었음"(2015-0-00908)

## Refrence

- [1] PAPAGEORGIOU, Constantine P.; OREN, Michael; POGGIO, Tomaso. A general framework for object detection. In: *Computer vision, 1998. sixth international conference on*. IEEE, 1998. p. 555-562.
- [2].ABADI, Martín, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.
- [3] LIU, Wei, et al. Ssd: Single shot multibox detector. In: *European conference on computer vision*. Springer, Cham, 2016. p. 21-37.
- [4]GIRSHICK, Ross, et al. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 2016, 38.1: 142-158.
- [5]Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." *Advances in neural information processing systems*. 2015.