레시피 추천 시스템을 위한 식품 성분 감지

신유 후
huxinyu@stanford.edu

유헹 리우
lyhdk7@stanford.edu

유 싱
karenx@stanford.edu

초록

헬로프레시나 블루에이프런과 같은 밀키트 배달 서비스는 요즘 큰 인기를 끌고 있습니다. 이러한 밀키트는 다양한 종류로 제공되며 바쁜 라이프스타일을 가진 사람들의 큰 고민을 해결해줍니다. 여기서는 사용자가 이미 가지고 있는 식재료를 기반으로 레시피를 추천하는 아이디어에 대해 설명합니다. 추천 시스템은 먼저 이미지를 기반으로 음식 카테고리와 수량을 감지한 다음, 감지된 재료에 가장 적합한 레시피를 추천합니다. 이번 글에서는 첫 번째 단계인 식재료 감지를 딥러닝 기술로 어떻게 해결했는지 자세히 살펴보겠습니다. 기본 객체 감지 모델로 Mask-RCNN[1]을 사용했으며 양파, 가지, 오이의 세 가지 식품 카테고리를 감지할 때 90.12%의 $AP_{0.5}$ (평균 정밀도, IoU 임계값 0.5)를 달성했습니다. 데이터 세트를 확장하여 6개 클래스를 감지한 후 $AP_{0.5}$ 는 77.12%로 떨어졌습니다. 모델은 MSCOCO[13] 데이터 세트를 사용하여 사전 훈련된 가중치로 초기화되었습니다. 더 나은 결과를 얻기 위해 다양한 특징 추출 레이어를 동결하고, 데이터 증강을 적용하고, 다른 백본 컨볼루션 네트워크를 사용하고, 다른 정규화 방법을 사용한 결과를 분석했습니다. 그 결과, L2 정규화와 학습률 감쇠를 적용한 모델이 다른 모델보다 우수한 성능을 보였습니다.

1 소개

사람들은 종종 부엌에 많은 식료품과 식재료를 가지고 있지만, 기억나는 몇 가지 레시피로 요리를 계속 만들거나 유통기한이 다 될 때까지 잊어버리는 경우가 많습니다. 저희는 사람들의 삶을 더 편리하게 만드는 제품을 만들고자 하는 열정을 가지고 있으며, 사용자가 이미 가지고 있는 식재료를 기반으로 레시피를 추천하는 제품이 밀키트 배달 서비스와 비슷한 역할을 하지만 더 친근한 솔루션을 제공할 수 있을 것이라고 생각했습니다. 이 시스템은 사용자가 휴대폰으로 찍은 식재료 사진을 기반으로 레시피를 추천하는 모바일 앱이나 냉장고에 있는 식재료를 스캔하고 스캔 결과에 따라 매일 레시피를 추천하는 스마트 냉장고에 통합될 수 있습니다. 음식 카테고리와 양을 감지하는 것은 어려운 작업이므로 딥러닝 모델을 사용하면 도움이 됩니다. 네트워크의 입력은 이미지이고 출력은

입력 이미지에 있는 모든 식품의 카테고리와 수량입니다. 카테고리와 함께 수량도 감지해야 하기 때문에 분류 모델이 아닌 객체 감지 모델을 사용해야 합니다.

2 관련 작업

다음은 인터넷에서 찾은 몇 가지 컨볼루션 객체 감지 모델의 현재 상태를 보여주는 그림으로, Mask-RCNN[1]이 인스턴스 기능을 갖춘 다른 모델보다 성능이 뛰어납니다.

CS230: 딥 러닝, 2019년 가을, 스탠포드 대학교, 캘리포니아. (NIPS 2017에서 차용한 LateX 템플릿).

세분화 및 높은 정확도를 제공합니다. YOLO[2]는 여러 단계로 구성된 다른 모델에 비해 계산 속도가 훨씬 빠르기 때문에 특히 실시간 탐지 애플리케이션에 널리 채택되는 모델입니다. YOLO의 모델 성능은 Mask-RCNN[1]이 기반이 되는 Faster-RCNN[3]보다도 약간 떨어집니다.

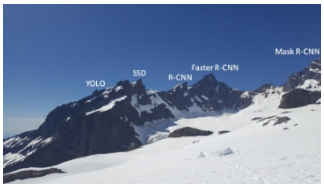

그림 1: 이미지 출처: http://deeplearning.csail.mit.edu/instance_ross.pdf

저희는 Faster-RCNN[3]과 YOLOv3[2]를 사용한 과일 감지에 관한 몇 가지 과거 연구를 발견했습니다.

Sa 등[4], Bargoti 등[5], Liu 등[6]은 Faster-RCNN을 사용한 과일 감지 시스템을 발표했습니다. 과수원의 이미지로 모델을 훈련하고 테스트하여 농업 작업을 지원했습니다. 이 논문의 결과는 특히 복잡한 배경을 가진 이미지에서 과일을 감지하는 데 높은 정확도를 보여주었습니다. 딥프루츠[4] 시스템은 사전 학습된 이미지넷 모델로 구축되었으며, 더 짧은 시간에 더 작은 데이터 세트를 사용하여 재학습할 수 있었기 때문에 새로운 과일을 더 빠르게 배포할 수 있었으며, 이는 우리 시스템에서 달성하고자 하는 목표와 유사합니다.

Tian 등의 Apple Detection[7]과 망고욜로[8]는 한 가지 과일 카테고리를 감지하기 위해 YOLOv3[2] 위에 구축된 모델을 제안했습니다. 이 모델들은 실시간 감지에 적용할 수 있는 빠른 감지 속도를 달성했습니다. [7]은 DenseNet 방식을 적용하여 YOLOv3의 특징 레이어를 개선했습니다.

겹치는 물체와 복잡한 배경과 같은 제약 조건은 과일 감지 모델에 큰 도전 과제입니다. 일반적으로 Faster-RCNN 모델이 더 정확하고 복잡한 이미지를 더 잘 처리하는 반면, YOLOv3 모델은 더 빠릅니다.

3 데이터 세트 및 기능

이 프로젝트에서는 객체 감지 위에 인스턴스 분할 마스크를 생성하는 Faster- RCNN[3]의 확장 버전인 Mask-RCNN[10]의 기존 구현을 사용했습니다. VIA[9]를 사용하여 이미지에 직접 주석을 달았습니다. 프로젝트를 합리적인 범위로 유지하기 위해 데이터 세트를 몇 가지 채소 및 과일 카테고리로 좁혔습니다. 먼저 가지, 오이, 양파의 세 가지 범주로 초기 결과를 모델에 학습시킨 후 소고기, 생선, 파인애플의 세 가지 범주를 추가하여 모델을 학습시켰습니다. 각 카테고리에 대해 약 60개의 훈련 세트와 약 15개의 검증 세트를 사용했습니다. 이미지는 Flickr[11] 및 [12]에서 다운로드했습니다. 이미지의 크기는 가로 세로 비율을 유지하면서 1024x1024로 조정했습니다. 다음은 데이터 세트의 예시입니다:

그림 2: 양파 이미지

그림 3: 세그멘테이션 마스크가 적용된 크기 조정된 이미지

4 방법

Mask-RCNN[1]은 영역 제안 네트워크를 사용하여 이미지에서 관련 오브젝트가 있을 수 있는 경계 상자를 제안한 다음 각 경계 상자에 대해 이진 분류를 실행하여 분류합니다. 병렬로

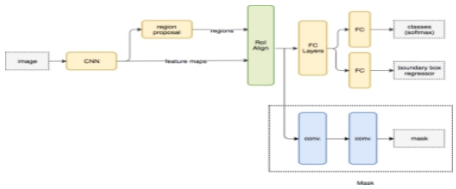

그림 4: https://medium.com/jonathan_hui/image-segmentation-with-mask-r-cnn-ebe6d793272

분류를 사용하면 모델은 다른 CNN 네트워크를 사용하여 각 경계 상자에 대해 의미론적 분할을 수행하여 마스크를 생성합니다.

이 프로젝트에서는 MS-COCO[13] 데이터 세트에서 사전 훈련된 가중치를 사용하는 모델로 시작합니다. 훈련 중에는 데이터 세트가 매우 작기 때문에 특징 추출 계층의 모든 가중치가 고정되고 분류기의 가중치만 훈련됩니다.

손실 함수에는 3개의 개별 손실 함수를 조합하여 사용합니다:

$$L = L_{cls} + L_{box} + L_{mask}$$

이 손실 함수는 [1]에서 확인되며, L_{cls} 및 L_{box} 모두 [3]에 정의되어 있습니다. 마스크 브랜치는 K 클래스에 대한 각 RoI에 대해 각 클래스에 대해 mxm 차원 마스크를 생성합니다. L_{mask} 은 평균 이진 교차 엔트로피 손실로 정의되며, RoI가 기준 실측 클래스 k와 연관된 경우 L_{mask} 은 k 번째 마스크[1]에만 정의됩니다. 따라서 L_{mask} 의 방정식은[14] 입니다:

$$L_{mask} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} [y_{ij} \log \hat{y}_{ij}^{k} + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^{k})]$$

여기서 y_{ij} 는 트루 마스크의 셀(i,j)에 대한 레이블입니다. y^k 는 기준값 클래스 k에 대한 학습된 마스크의 셀(i,j)에 대한 레이블입니다. 클래스별 오류를 계산하면 이 마스크 손실이 클래스 예측의 영향을 받지 않습니다.

각 분리된 손실 함수에 대해 L_{cls} 은 오브젝트 분류 성능을, L_{box} 은 네트워크가 오브젝트를 얼마나 잘 로컬라이즈하는지를 나타내며, 이번에 추가된 L_{mask} 은 네트워크가 오브젝트를 세그먼트하는 성능을 평가하는 데 도움이 됩니다.

5 실험/결과/토론

매터포트[10]가 만든 GitHub 리포지토리에서 마스크-RCNN을 구현하고 데이터 세트에 대한 전이 학습을 수행했습니다.

MS-COCO[13] 데이터 세트에 대해 사전 학습된 가중치 중 일부를 재사용하고 학습을 위해 세 가지 다른 접근 방식을 시도했습니다:

1. 네트워크의 RPN, 분류기 및 마스크 헤드만 학습합니다. ("head")
2. 훈련 리셋 5단계 및 접근법 1의 모든 레이어. (5+)
3. 훈련 리셋 4단계 및 접근 방식 2의 모든 레이어. (4+)

5.1 결과

3개 클래스의 데이터와 6개 클래스의 데이터를 모두 사용하여 모델을 여러 번 학습시킨 결과 다음과 같은 결과를 얻었습니다:

접근 방식	Epoch	훈련 손실	AP_{50}
3 클래스, Resnet101, 헤드	15	0.2180	79.56%
3 클래스, Resnet101, 5+	15	0.1800	81.57%
3 클래스, Resnet101, 4+	15	0.1589	77.10%
6 클래스, Resnet50, 4+	60	0.3491	76.14%
6 클래스, Resnet50, 5+	60	0.3681	77.12%
6 클래스, Resnet50, 헤드	60	0.5047	75.19%

손실 그래프는 아래와 같습니다:

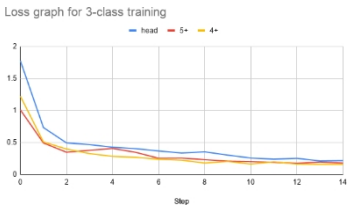

그림 5: 3등급 분류 시

손실그림 6: 6등급 분류 시 손실

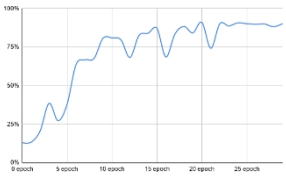

그림 7: 반복을 통한 AP_{50} (3클래스, Resnet101, 5+)

손실 그래프에서 모델의 복잡도가 높아질수록 손실이 감소하는 것을 볼 수 있습니다. 레이어 4 이상(접근 방식 3)을 훈련하면 실제로 가장 낮은 손실 값을 얻을 수 있습니다. 그러나 레이어 5 이상(접근 방식 2)을 훈련하면 가장 낮은 평가 AP_{50} 를 얻을 수 있는 것으로 보입니다. 이는 레이어 5와 헤드(접근 방식 2)를 훈련할 경우, 코코[13] 데이터 세트에서 훈련된 가중치의 대부분을 유지하면서 데이터에 맞게 모델을 사용자 정의할 수 있기 때문입니다. '머리'만 훈련하면 모델은 여전히 이전 데이터 세트를 대상으로 하며 일반화가 잘 되지 않습니다. 리셋 레이어 4 이상에서 학습하면 모델이 너무 복잡해져 과적합이 됩니다.

6개의 클래스를 훈련하려면 많은 연산 능력이 필요하기 때문에 이미지 인식을 위한 더 간단한 백본인 ResNet50을 사용했습니다. 3개 클래스를 훈련할 때와 마찬가지로 5계층 이상을 훈련할 때 가장 좋은 성능을 얻을 수 있습니다. 그러나 6개의 클래스를 훈련하면 검증 세트 성능이 낮아집니다. 데이터 세트를 살펴보고 몇 가지 가능한 이유를 요약해 보았습니다:

- 3 클래스 데이터 세트는 너무 쉽습니다. 이 데이터 세트의 AP는 COCO[13] 데이터 세트의 성능에 비해 너무 높습니다. 3 클래스 데이터 세트의 이미지는 배경이 단순합니다(예: 순수한 흰색 배경). 따라서 오브젝트 감지가 더 쉽습니다.
- 카테고리를 늘렸을 때, 전체 데이터 세트의 크기는 커졌지만 6개 클래스 데이터 세트의 이

미지에는 배경이 노이즈가 있는 이미지가 더 많이 포함되어 물체를 감지하기가 더 어려워졌습니다.

- ResNet101은 ResNet50 모델보다 더 나은 성능을 보였습니다.

저희는 AP IoU=0.5를 Mask R-CNN 논문[1]과 비교했습니다. AP 성능은 MS-COCO[13] 데이터 세트에 대한 논문의 성능보다 훨씬 높습니다. 이는 우리의 데이터 세트가 매우 구체적이고 과일에만 초점을 맞추고 있기 때문입니다. 또한 데이터 세트의 이미지 대부분은 배경이 단순하고 이미지에 한두 가지 종류의 물체가 포함되어 있습니다. 이는 학습 과정을 단순화합니다. 향후 데이터 세트를 확장하여 복잡한 이미지를 사용한다면 성능이 더 나빠질 것으로 예상됩니다.

결론적으로, 5단계 이상의 ResNet101 백본 모델로 훈련하는 것이 최적의 설정이었으며, 이를 최적화를 위한 기준으로 사용했습니다.

5.2 튜닝 성능

성능을 최적화하기 위해 다음과 같은 세 가지 방법을 시도했습니다:

- 감퇴 학습률
- L1 및 L2 정규화
- 데이터 증강 결과

는 다음과 같습니다:

접근 방식	Epoch	훈련 손실	AP_{50}
레이어 5+	15	0.1814	81.57%
레이어 5+, L1 정규화	15	0.1880	86.50%
레이어 5+, L2 정규화	15	0.1799	91.60%
레이어 5+, L2 정규화, 학습률 감쇠	15	0.1707	87.01%

학습률 감쇠가 학습 손실을 감소시키는 것을 볼 수 있습니다. 그러나 검증 세트에서 AP_{50} 에 비해 성능이 크게 향상되지는 않았습니다.

정규화를 위해 다양한 정규화 설정을 시도했습니다: L2, L1, 정규화 없음. L1과 L2 정규화는 비슷한 성능 향상을 가져왔으며, 네트워크가 과도하게 피팅되는 것을 방지하는 데는 L2 정규화가 가장 우수한 성능을 보였습니다.

또한 프로젝트에 데이터 증강을 시도했습니다. 6개의 클래스에 대해 증강을 적용하고 백본 이상의 5개 레이어를 *훈련한* AP_{50} 는 80.12%, 증강을 적용하지 않은 AP 는 82.90%였습니다. 검증 데이터가 너무 단순하고 일반적으로 훈련 데이터와 동일한 패턴을 가지고 있기 때문에 보강이 도움이 되지 않았습니다.

5.3 결론

훈련 과정에서 찾은 가장 좋은 모델은 ResNet101 백본과 L2 정규화를 사용한 모델이었습니다. 그 결과 AP_{50} 값이 91.60%였습니다. 이 모델을 테스트 이미지에 적용한 결과 가지와 양파가 성공적으로 세분화되어 검출되는 다음과 같은 결과를 얻었습니다:

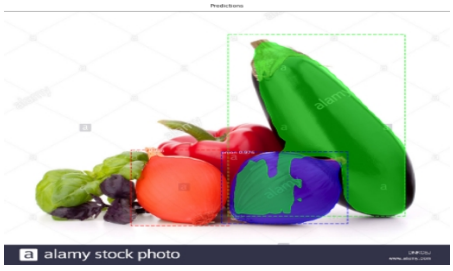

그림 8: 테스트 이미지에 대한 훈련 결과 [15]

6 결론 및 향후 작업

주방 내부의 식료품과 식재료를 감지하기 위해 Mask-RCNN 아키텍처를 사용했습니다. COCO[13] 데이터 세트에 대해 사전 학습된 가중치를 사용한 전이 학습, 하이퍼파라미터 튜닝 및 정규화를 적용하여 최상의 성능을 가진 모델을 찾았습니다. 오브젝트 감지 결과를 바탕으로 구글 검색을 통해 레시피 추천을 수행할 수 있었습니다.

앞으로는 각 카테고리에 대해 더 다양한 이미지를 포함하도록 데이터 세트를 확장하고 모델에 더 많은 클래스를 추가해야 합니다. 이 프로젝트를 진행하는 동안, 동일한 성능을 유지하려면 모델에 더 많은 클래스를 추가하는 것이 어려울 수 있다는 것을 확인했습니다. 데이터 세트를 확장하기 전에 성능 저하의 원인을 조사하고 이를 해결하기 위해 네트워크 아키텍처를 조정해야 합니다. 또한 이렇게 큰 모델을 훈련하는 데는 많은 시간과 리소스가 소모됩니다. 향후에는 모델 아키텍처를 단순화하여 시간과 리소스를 절약하는 방법에 대한 연구를 진행할 수 있습니다.

7 기여

모든 저자는 과거 작업 연구, 데이터 세트 준비, 모델 구축, 교육, 튜닝 및 보고서 작성에 동등하게 기여합니다.

참조

[1] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. 컴퓨터 비전에 관한 IEEE 국제 컨퍼런스 논문집 (2961-2969쪽).

[2] 레드몬, J., & 파르하디, A. (2018). Yolov3: 점진적 개선. arXiv 사전 인쇄물 arXiv:1804.02767.

[3] Ren, S., He, K., Girshick, R., & Sun, J. (2015). 더 빠른 R-CNN: 영역 제안 네트워크를 통한 실시간 물체 감지를 향하여. 신경 정보 처리 시스템의 발전 (pp. 91-99).

[4] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: 심층 신경망을 이용한 과일 감지 시스템. Sensors, 16(8), 1222. https://doi.org/10.3390/s16081222

[5] Bargoti, S., & Underwood, J. (2016). 과수원에서 깊은 과일 감지.

[6] Liu, X., Zhao, D., Jia, W., Ji, W., Ruan, C., & Sun, Y. (2019). 인스턴스 세분화에 기반한 온실에서 오이 열매 감지. IEEE Access, Access, IEEE, 139635.

[7] Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). 개선된 YOLO-V3 모델을 사용한 과수원의 다양한 성장 단계에서의 사과 감지. 농업의 컴퓨터와 전자, 157, 417-426.

[8] Koirala, A., Walsh, K. B., Wang, Z., & McCarthy, C. (2019). 실시간 과일 감지 및 과수원 과일 부하 추정을위한 딥 러닝 : '망고 욜로'의 벤치마킹 정밀 농업: 정밀 농업의 발전에 관한 국제 저널, 20(6), 1107.

[9] Dutta, A., Gupta, A., & Zisserman, A. (2016). VGG 이미지 어노테이터 (VIA). 에서 검색됨 http://www.robots.ox.ac.uk/~vgg/software/ via

[10] Abdulla, W. (2017). Keras 및 TensorFlow에서 객체 감지 및 인스턴스 세분화를 위한 Mask-RCNN. https://github.com/matterport/Mask_RCNN 에서 가져온 것

[11] 플리커. (nd). 에서 가져온 https://www.flickr.com

[12] Buyukkinaci, M. (2018). 이미지 현지화 데이터 세트. 에서 가져온 https://www.kaggle.com/mbkinaci/image-localization-dataset

[13] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). 마이크로 소프트 코코: 컨텍스트에서 일반적인 개체. 컴퓨터 비전에 관한 유럽 컨퍼런스에서 (pp. 740-755). Springer, Cham.

[14] Weng, L. (2019). 초보자를 위한 객체 감지 Part3: R-CNN 제품군. 검색: https://lilianweng.github.io/lil-log

[15] Limited, A. (nd). 1억 8천만 개의 스톡 이미지, 벡터, 동영상 및 360도 파노라마 이미지. 세계에서 가장 다양한 스톡 사진 컬렉션을 보유한 Alamy에서 더 많은 선택권을 제공합니다. 에서 검색됨 https://www.alamy. com/.