

객체 사이의 관계를 고려한 객체검출 알고리즘

정지수^o, 박효진, 황지혜, 박규태, 곽노준
서울대학교 융합과학기술대학원
{soo3553,wolfrun,hjh881120,pgt4861,nojunk}@snu.ac.kr

요약

Deep Neural Network의 발전에 따라 객체 검출 알고리즘들의 성능은 매우 향상되었다. 그럼에도 불구하고 여전히 문제점을 가지고 있는데, 그중 하나가 객체 검출시 주변 상황을 고려하지 않는 것이다. 본 논문에서는 주변 객체의 정보를 이용하여 잘못된 정보를 교정해주는 연구를 진행하였다. 기존 객체 검출기를 사용해서 검출된 각 객체 스코어(confidence score)와 한 이미지 내에서 같이 나올 수 있는 객체들의 확률 값을 Conditional Random Field에 적용하여 잘못된 객체의 정보를 교정해 주었다.

1. 서론

최근 Deep Neural Network(DNN)이 높은 성능을 나타내면서 많은 분야에 적용되었다. 그중 하나가 객체 검출 알고리즘이고, 많은 연구자들의 노력으로 높은 정확도를 가지는 객체 검출 알고리즘들이 알려졌다[1-4]. 하지만 객체 검출 알고리즘에도 여전히 문제점을 가지고 있다. 그 중 하나로 객체 검출을 수행할 때, 주변의 객체 정보를 살펴보지 않고, 그 객체만으로 값을 결정한다는 것이다. 그림 1은 그 예를 나타낸 것이다.

그림 1은 You Only Look Once(YOLO)[1]라는 객체 검출 알고리즘을 수행한 결과이다. 이 객체 검출 알고리즘에서 박스를 생성할 때는, 가장 높은 값만을 선택하게 된다. 따라서 그림 1의 오른쪽 그림과 같이, 실제 두 마리의 소를 한 마리는 제대로 검출하고, 다른 한 마리는 양으로 검출하게 된 것이다. 본 논문에서 한 이미지 내에서 같이 나오는 객체들의 관계를 고려하여 이를 교정해주는 연구를 진행하였다. 한 이미지 내에서 같이 나오는 객체들(Co-Occurence)의 확률 값을 구하면, 소가 2마리 나올 확률이 소와 양이 나올 확률보다 높다. 즉, 각각의 객체의 확률과 Co-Occurence 확률을 Conditional Random Field(CRF)에 적용하여 값을 교정하였다.

2. 관련 연구

최근 객체 검출 알고리즘은 DNN을 이용하여 속도를 빠르게 하고 성능을 높였다. 그중 YOLO는 최초로 DNN을 이용해서 실시간 객체 검출을 가능하

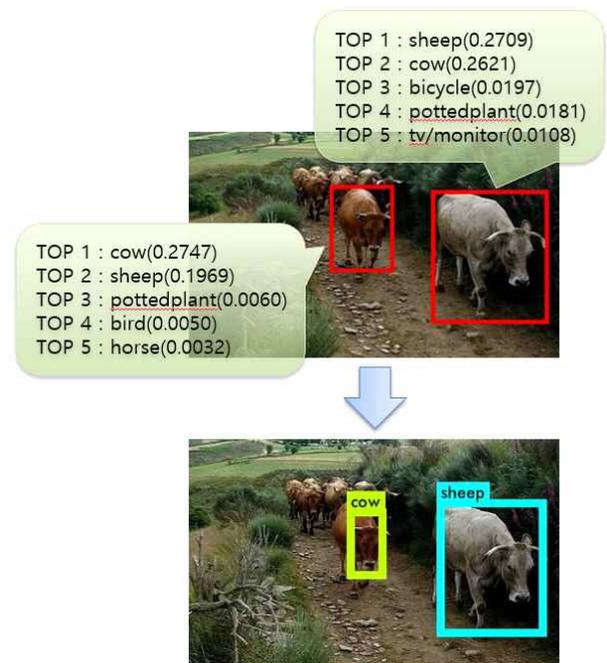


그림 1 기존 객체 검출의 문제점; YOLO 검출기에서 각 객체 스코어 0.2 이상 검출박스

게 한 알고리즘이다. 이후 R-FCN[2], Single Shot Multi-Box Detector(SSD)[3], Rainbow-SSD[4] 등이 연구되었고, 많은 성능 향상이 있었다.

CRF는 semantic segmentation에서 많이 사용되던 방식이다. semantic segmentation은 각 픽셀 별로 classification을 수행하는 것으로, 주변의 픽셀에 대한 영향을 객체 검출 알고리즘보다 더욱 많이 받는다. 따라서 [5]의 논문과 같이 CRF를 이용하여 주변 픽셀을 고려한 알고리즘들이 연구되었다.

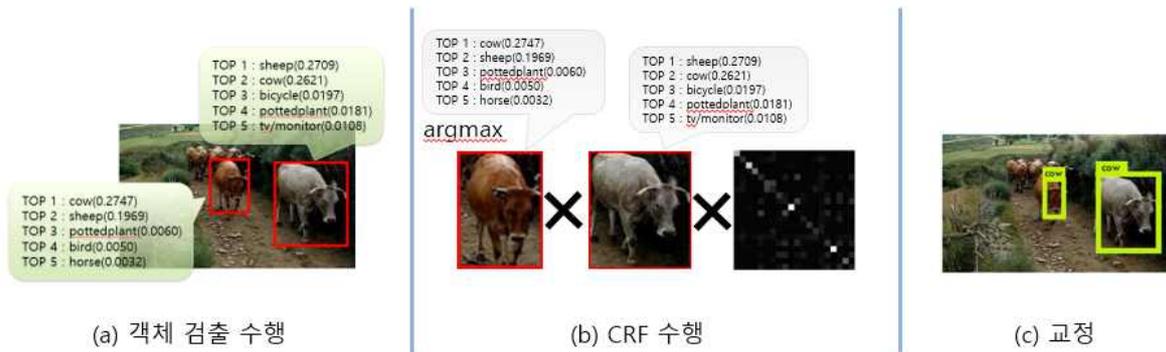


그림 2 전체적인 알고리즘

2. CRF를 이용한 객체 검출 교정 방법

본 논문의 연구는 CRF를 이용한 후처리 과정이고 객체 검출 알고리즘은 기존의 객체 검출 알고리즘을 사용하였다. 객체 검출 알고리즘은 YOLO 알고리즘을 사용하였다.

한 이미지 내에서 Co-Occurrence의 확률 값을 학습 데이터를 통하여 구하였다. 2.1은 데이터에 대한 설명으로 구성되어 있고, 2.2는 한 이미지 내에 Co-Occurrence의 확률 값을 구하는 방법으로 이루어져 있다. 2.3은 CRF를 이용해서, 각 객체들의 스코어와 같이 나오는 객체들의 확률 값들로 잘못 검출된 객체를 교정하는 방법으로 구성되어 있다.

2.1 데이터셋

데이터 셋은 PASCAL Visual Object Classes (VOC)[6]의 VOC2007 데이터 셋과 VOC2012 데이터 셋을 사용하였다. VOC 데이터 셋의 경우, 20가지 객체 종류가 있고, 사람, 운송수단(비행기, 자동차, 보트, 버스, 자동차, 오토바이, 기차), 동물(새, 고양이, 소, 개, 말, 양), 가전제품(물병, 의자, 식탁, 꽃병, 소파, TV/모니터)으로 이루어져 있다.

총 학습 데이터는 16,551장이 있고, 테스트 데이터는 4952장이 있다.

2.2 Co-Occurrence 확률 값

Co-Occurrence의 확률 값들을 VOC 학습 데이터 셋으로부터 배열의 형태로 만들었다. 그림 2의 (b) CRF에서 Co-Occurrence의 matrix를 만들기 위하여, 우선 각각의 이미지에서 동시에 나온 객체들의 수를 센다. 예를 들어 한 이미지에서 강아지가 세 마리, 소가 한 마리 나왔으면, 강아지에서는 강아지가 같이 나온 것으로 한번 카운트하고, 소가 같이 나온 것으로 카운트한다. 전체적인 데이터의 균형을

위하여 여러 마리가 나온 경우에도 한 번만 카운트하였다. 소에서는 강아지만 한번같이 나온 것으로 카운트한다. 이후, 행렬을 표준화(normalization) 해주었다.

$$n_{i,j} = \frac{n_{i,j}}{n_{i,i} * n_{j,j}} \tag{1}$$

Co-Occurrence matrix는 그림 2의 (b)에서 보는 것과 같이 나타나고, 대칭행렬이 된다.

표 1 Co-Occurrence 행렬 예제

	소	양
소	0.4379	0.0041
양	0.0041	0.0411

2.3 Conditional Random Field (CRF)

CRF를 수행하는 수식은 아래와 같다.

$$Y^* = \underset{y}{\operatorname{argmax}} \left[\prod_{i \in b} \varphi(y_i, X) + \left(\prod_{i \in b} \prod_{j \in N_i} \varphi_{ij}(y_i, y_j) \right)^\lambda \right] \tag{2}$$

X의 값은 데이터, y는 label, b는 박스, N은 주변 박스를 나타낸다. 앞의 식을 살펴보면, 데이터로부터 나온 객체의 확률을 나타내는 수식이다. 오른쪽은 Co-Occurrence의 값을 곱해주는 수식이다. 또한 Lambda를 추가하여 Co-Occurrence에 대한 비율을 정해줄 수 있다.

예를 들어 lambda를 0.05/2로 놓고 (b)를 수행해보면, 양쪽의 TOP1들의 결과 값은 0.2747*0.2709*(0.0041^0.05)는 약 0.057의 값을 가진다. 왼쪽의 TOP1, 오른쪽의 TOP2를 비교해보면, 0.2747*0.2621*(0.4379^0.05)는 약 0.070의 값을 갖는다. 왼쪽의 TOP2와 오른쪽의 TOP1의 값은 0.041의 값을 가진다. 즉 이 값 중에 가장 큰 값 0.070을 갖게 하는

객체의 종류로 바꾸어 준다.

추가로 같은 객체의 종류들은 자주 나오는 경우가 많다. 따라서 다른 객체임에도 불구하고, 같은 객체로 교정하는 경우를 막기 위해 Co-Occurrence 행렬의 diagonal 값에 D라는 한계 값을 주었다. 특정 값이 넘어가면, 고정된 D 값을 유지하였다.

3. 실험 결과 및 분석

본 논문에서는 기존 객체 검출에서 사용되는 mAP를 측정하지 않았다. mAP의 경우 모든 객체의 점수를 고려하여 사용되는 것이고, 실제 어플리케이션에 적용되는 평가 기법은 아니다. 또한, 본 연구는 새로운 객체 검출을 하는 것이 아니라 객체 검출 알고리즘에서 잘못된 부분을 교정해 주는 알고리즘이다. 따라서 본 연구에서는 잘못 검출된 박스를 얼마큼 교정하였는지에 대한 평가를 진행하였다.

3.1 실험 결과

객체 검출기 YOLO를 VOC에 테스트해 본 결과, 총 9,718개의 박스를 생성하였다. 이중, 8,910개는 제대로 검출을 하였고, 808개는 잘못 검출하였다.

표 2 오검출 박스의 수

	오검출한 박스의 수
YOLO	808
CRF($\lambda=0.01$)	808
CRF($\lambda=0.05$)	807
CRF($\lambda=0.1$)	809

표 3 λ 와 D를 적용하였을 때 오검출 박스 수

$\lambda \backslash D$	None	0.7	0.5	0.3
0.01	808	808	807	808
0.05	807	807	806	807
0.1	809	809	808	811

CRF만 적용하였을 경우, λ 가 0.05 일 때, 1개의 박스가 더 좋아졌다. 추가로 Co-Occurrence 행렬의 diagonal 값에 D라는 한계값을 주었을 때, 806개까지 2개가 줄어드는 것을 확인할 수 있었다.

3.2 분석

실험 결과 오검출 박스의 수에서는 많은 차이를 보이지 못하였다. 그 이유는 CRF를 통하여 잘못된 객체 정보를 교정해주기도 하지만, 제대로 검출된 객체의 정보도 바꾸는 문제점이 있다. 예를 들어

강아지와 나무의 이미지가 있고 이를 물고기와 나무로 검출한 경우, 본 연구에서는 나무의 점수가 높아서 물고기를 강아지로 교정하기를 원했다. 하지만, 물고기의 점수가 높은 경우, 나무를 해초로 바꾸어 원래 제대로 맞추었던 나무마저도 틀리게 해 버린 것이다. 즉, 제대로 교정한 것만큼, 제대로 맞춘 객체를 틀리게 만든 것이다.

4. 결론

본 논문에서는 객체 검출 알고리즘을 수행할 때, 단순히 하나의 객체 검출이 아니라 주변 객체를 고려한 객체 검출 알고리즘을 제안하였다. 이는 기존에 semantic segmentation에서 사용되던 CRF를 객체 검출 알고리즘에 맞게 변형 및 적용시켜보았다. 이를 객체 검출 알고리즘에 사용함으로써, 주변 객체 간의 상관관계를 고려하고 잘못된 값들을 교정하였다.

감사의 글

본 연구는 한국연구재단의 차세대정보컴퓨팅기술개발사업에 의해 진행되었음 (2017M3C4A7077582).

참고문헌

- [1] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788).
- [2] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks. In Advances in neural information processing systems (pp. 379-387).
- [3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
- [4] Jeong, J., Park, H., & Kwak, N. (2017). Enhancement of SSD by concatenating feature maps for object detection. The British Machine Vision Conference (BMVC2017), London, UK, Sep. 2017.
- [5] Lin, G., Shen, C., Van Den Hengel, A., & Reid, I. (2017). Exploring context with deep structured

models for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence.

- [6] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303 - 338, 2010.