creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학박사 학위논문

확률적 모델링을 통한 불확실성 인식 심층 탐지 신경망

Uncertainty-Aware Deep Detection Neural Networks with Probabilistic Modeling

2021 년 8 월

서울대학교 대학원

전기 정보 공학부

최 지 웅

확률적 모델링을 통한 불확실성 인식 심층 탐지 신경망

Uncertainty-Aware Deep Detection Neural Networks with Probabilistic Modeling

지도 교수 이 혁 재

이 논문을 공학박사 학위논문으로 제출함
2021 년 8 월

서울대학교 대학원
전기 정보 공학부
최 지 웅

최지웅의 공학박사 학위논문을 인준함
2021 년 8 월

위 원 장 최 진 영
부위원장 이 혁 재
위 원 조 남 익
위 원 류 수 정
위 원 김 현

초 록

Object detection (객체 탐지)은 이미지 내 여러 물체들에 대해 각 물체의 카테고리 정보를 예측하는 classification과 각 물체의 크기 및 위치 정보를 예측하는 localization이 결합된 문제이다. 딥러닝과 GPU의 발전으로 object detection 알고리즘이 활발히 연구되고 있으며, 최근 object detection 알고리즘은 높은 정확도와 빠른 처리 속도로 자율주행과 무인 편의점 등 실시간 처리 속도를 요구하는 다양한 application에서 그 활용도가 높아지고 있다.

Object detection이 classification과 localization task가 결합된 문제임에도 불구하고, 기존 object detection 알고리즘은 classification 기반의 confidence score에 의존하여 물체의 카테고리와 위치 및 크기 정보를 예측한다. Localization 자체에 대한 confidence는 예측하지 못하기 때문에 localization에 의한 false positive에 대응하지 못한다는 문제가 있다. 특히, 자율주행과 같이 false positive로 인해 치명적인 사고가 유발되는 application에서는 이러한 mislocalization 문제를 해결하는 것이 필수적이다.

이뿐만 아니라, object detection 알고리즘은 높은 정확도를 달성하기 위해 annotation된 많은 양의 labeled 데이터를 학습 시 필요로 한다. 일반적으로 학습을 위한 이미지 데이터는 쉽게 수집이 가능하지만, 수집된 이미지에서 여러 물체에 대한 카테고리 정보와 물체의 위치 및 크기 정보를 얻는 것은 굉장히 많은 resource와 시간을 요구한다. 이러한 labeling cost 문제를 해결하기 위해 object detection을 위한 능동적 학습 (active learning)이 활발히 연구되어 왔다. 하지만, 기존의 object detection을 위한 능동적 학습 연구들은

uncertainty 예측을 위해 여러개의 모델을 사용하기 때문에 높은 연산량을 요구한다는 문제가 있으며, 능동적 학습 시 localization 정보는 사용하지 않고 classification 정보에만 의존하기 때문에 완전한 해결책이 되지 못한다.

Labeling cost를 줄이기 위한 또 다른 해결책으로 object detection을 위한 반지도 학습 (semi-supervised learning)이 활발히 연구되고 있다. 반지도 학습은 annotation 정보가 있는 labeled 데이터와 이미지로만 구성된 unlabeled 데이터를 함께 학습에서 사용하는 학습 방법이다. 일반적으로 unlabeled 데이터는 labeled 데이터에 비해 데이터 양이 훨씬 많은데, 기존 반지도 학습은 unlabeled 데이터의 유용성을 고려하지 않은 채 모든 데이터를 학습 시 사용한다. 그리하여 학습 시간을 대폭 증가시키는 문제가 있다.

본 논문은 앞서 언급한 3가지 문제에 대한 해결책을 제시한다. 첫째로, localization에 의한 false positive 문제를 해결하기 위해 localization uncertainty 모델링 방법과 이를 학습하기 위한 새로운 localization loss function을 제안한다. 제안 모델링으로 예측한 localization uncertainty를 검출 과정에서 활용하여, localization에 의한 false positive를 대폭 감소시킨다. 둘째로, 저비용 고성능의 object detection을 위한 새로운 능동적 학습 방법을 제안한다. 제안 방법은 능동적 학습 시 classification uncertainty 뿐만 아니라 localization uncertainty를 함께 고려하며, uncertainty 예측 시 single model과 single forward pass를 사용해 기존 대비 연산량을 대폭 줄이면서도 높은 정확도를 달성한다. 셋째로, 반지도 학습에서 uncertainty 기반 unlabeled 데이터 필터링 방법을 제안한다. 제안 방법으로 모델의 성능 향상에 도움이 되는 unlabeled 데이터만을 학습에 사용함으로써 기존 대비 학습 시간을 대폭 감소시키면서도 정확도를 향상시킨다.

앞서 언급한 문제들과 이를 해결하기 위한 제안 방법들을 본 논문의 각 장에서 자세히 소개한다.

주요어 : 불확실성, 객체 탐지, 능동적 학습, 반지도 학습, 딥러닝
학 번 : 2017-31914

목 차

제 1 장 서 론 1
1.1 연구의 배경 1
1.2 연구의 내용 3
1.3 논문 구성 5

제 2 장 연구의 배경지식 6
2.1 객체 탐지 (object detection) 6
2.1.1 Two-stage object detection 7
2.1.2 One-stage object detection 8
2.2 능동적 학습 (active learning) 9
2.3 반지도 학습 (semi-supervised learning) 11

제 3 장 Localization uncertainty를 사용한 정확하고 빠른 객체 탐지기 15
3.1 본 연구의 서론 15
3.2 본 연구의 배경 19
3.3 Gaussian YOLOv3 22
3.3.1 Gaussian 모델링 22
3.3.2 Loss function 재설계 25
3.3.3 Localization uncertainty의 활용 28
3.4 본 연구의 실험 결과 29
3.4.1 Localization uncertainty 평가 30
3.4.2 Gaussian YOLOv3의 성능 평가 32

3.4.3 False positive와 true positive의 시각적, 수치적 평가..38
3.5 본 연구의 결론 44

제 4 장 혼합 밀도 네트워크를 사용한 심층 객체 탐지 신경망을 위한 능동적 학습 46
4.1 본 연구의 서론 46
4.2 본 연구의 관련 연구 49
4.2.1 객체 탐지를 위한 심층 능동적 학습.....49
4.2.2 혼합 밀도 네트워크 50
4.3 객체 탐지를 위한 능동적 학습.....51
4.3.1 객체 탐지를 위한 mixture 모델링 53
4.3.2 능동적 학습을 위한 scoring function.....57
4.4 본 연구의 실험 결과.....59
4.4.1 객체 탐지에서 mixture 모델링의 효과 60
4.4.2 능동적 학습 평가.....65
4.4.3 확장성과 데이터 transferability 74
4.4.4 Parameter 민감도 76
4.4.5 Classification loss에 대한 논의 80
4.5 본 연구의 결론 82

제 5 장 반지도 학습을 위한 unlabeled 데이터 필터링 및 uncertainty 기반 반지도 능동적 학습 83
5.1 본 연구의 서론 83
5.2 본 연구의 관련 연구.....84
5.3 Unlabeled 데이터 필터링.....87

5.4 Uncertainty 기반 반지도 능동적 학습 90
5.5 본 연구의 실험 결과.....93
5.5.1 Unlabeled 데이터 필터링 93
5.5.2 Uncertainty 기반 반지도 능동적 학습.....96
5.6 본 연구의 결론 98

제 6 장 결 론.....99

참고문헌.....101

Abstract.....116

표 목차

[표 3.1] KITTI와 BDD training set에서 추출된 anchor box 결과 30
[표 3.2] KITTI validation set에서 기존 연구와의 성능 비교 34
[표 3.3] BDD test set에서 성능 비교 35
[표 3.4] MS-COCO 데이터에서 baseline과 제안 알고리즘의 성능 비교 36
[표 3.5] KITTI validation set에서 최신 detector인 YOLOv4와 제안 모델의 성능 비교 37
[표 3.6] BDD validation set에서 최신 detector인 YOLOv4와 제안 모델의 성능 비교 38
[표 3.7] False positive (FP)와 true positive (TP)를 통한 수치적 평가 44
[표 4.1] PASCAL VOC 2007 데이터에서 제안 모델과 여러 instance 및 baseline과의 정확도 비교 61
[표 4.2] MS-COCO 데이터에서 제안 모델과 여러 instance 및 baseline과의 정확도 비교 61
[표 4.3] PASCAL VOC 2007에서 여러 scoring aggregation function의 정확도 비교 67
[표 4.4] 각 task의 aleatoric 또는 epistemic uncertainty로 선택된 데이터의 overlapping ratio (%) 결과 68
[표 4.5] VOC07에서 능동적 학습의 정확도 및 계산 비용 비교 70

[표 4.6] VOC07+12에서 제안 방법과 기존 single model 기반 방법들의 정확도 비교 71
[표 4.7] VOC07+12에서 제안 방법과 기존 multiple models 기반 방법들의 정확도 비교 72
[표 4.8] 제안 방법과 multiple models 기반 방법의 계산 비용 비교 73
[표 4.9] MS-COCO에서 능동적 학습의 정확도 및 계산 비용 비교 74
[표 4.10] Faster-RCNN에서 제안 모델링 적용 전과 후의 정확도 및 계신 비용 비교 75
[표 4.11] 여러 네트워크 구조에서 제안 방법의 데이터 transferability 성능 비교 76
[표 4.12] VOC07 데이터에서 mixture model 개수에 따른 정확도 및 계산 비용 비교 77
[표 4.13] Input resolution에 따른 정확도 (%) 비교 78
[표 4.14] 능동적 학습에서 budget number에 따른 mAP 비교79
[표 4.15] VOC07에서 classification loss에 따른 능동적 학습의 정확도 비교 81
[표 4.16] MS-COCO에서 classification loss에 따른 능동적 학습의 정확도 비교 81
[표 5.1] 제안 방법 적용 전과 후의 반지도 학습 정확도 비교 95
[표 5.2] 다양한 필터링 비율에 따른 정확도 비교 96
[표 5.3] 제안 반지도 능동적 학습의 정확도 결과 97

그림 목차

[그림 2.1] Classification과 object detection의 차이 6
[그림 2.2] Two-stage object detection의 프로세스 7
[그림 2.3] One-stage object detection의 프로세스 8
[그림 2.4] 능동적 학습 10
[그림 2.5] 반지도 학습의 학습 데이터 예 11
[그림 2.6] Self-training 프로세스 12
[그림 2.7] Consistency 기반 반지도 학습의 loss 계산 예 14
[그림 3.1] YOLOv3의 네트워크 구조 21
[그림 3.2] YOLOv3의 prediction feature map 23
[그림 3.3] 제안 알고리즘의 prediction box의 구성요소 25
[그림 3.4] KITTI와 BDD validation set에서 IoU와 예측한 localization uncertainty의 관계 31
[그림 3.5] KITTI validation set에서 baseline인 YOLOv3의 검출 결과의 예 39
[그림 3.6] KITTI validation set에서 제안 알고리즘인 Gaussian YOLOv3의 검출 결과의 예 40
[그림 3.7] BDD test set에서 baseline인 YOLOv3의 검출 결과의 예 41
[그림 3.8] BDD test set에서 제안 알고리즘의 검출 결과의 예42
[그림 4.1] 제안 방법 overview 48
[그림 4.2] 제안 모델링을 적용한 object detection network 구조 52

[그림 4.3] 부정확한 예측에 대한 localization과 classification의 aleatoric uncertainty와 epistemic uncertainty의 예 65
[그림 5.1] 객체 탐지를 위한 consistency 기반 반지도 학습 [18] 85
[그림 5.2] [51]에서 사용한 mixup augmentation의 예..... 86
[그림 5.3] [51]에서 제안한 interpolation 기반 방법과 consistency 기반 방법 [18]의 결합 86
[그림 5.4] Labeled 데이터와 필터링된 unlabeled subset 데이터로 구성된 반지도 학습을 위한 최종 학습 데이터의 예..... 88
[그림 5.5] 제안 필터링 방법 프로세스..... 89
[그림 5.6] 제안 반지도 능동적 학습을 위한 loss function 및 uncertainty 예측 프로세스..... 92
[그림 5.7] 제안 방법 적용 전과 후의 학습 iteration과 정확도 비교..... 95

제 1 장 서 론

1.1 연구의 배경

최근 심층 탐지 신경망 (deep detection neural network)은 매우 높은 탐지 정확도 [1]로 자율주행 application [2], medical image 분석 [3], 그리고 다양한 computer vision을 활용한 social 서비스 [4] 등에 활발히 적용되어 사용되고 있다. 특히, 알고리즘의 발전뿐만 아니라 하드웨어 플랫폼의 발전은 심층 탐지 신경망의 활용 범위를 더욱 확대시키고 있으며, 향후 산업 전반적으로 보편화 될 것으로 기대된다.

심층 탐지 신경망은 크게 두 종류로 구별할 수 있다. 첫번째는 two-stage object detector로 region proposal stage와 classification+bounding box regression stage의 두 단계를 수행하는 방식 [5]이다. 이 방식은 일반적으로 높은 정확도를 보이지만 처리 속도가 느리다는 단점이 있다. 두번째로는 one-stage object detector로 region proposal stage 없이 classification과 bounding box regression을 한 번에 처리하는 방식 [6]이다. 이는 처리 속도가 빠르나 상대적으로 낮은 정확도를 보인다는 단점이 있다. 이러한 기존 두 종류의 심층 탐지 신경망의 공통적인 문제점 중 하나는 검출 대상의 localization 정보는 예측 가능하지만, 이 예측 결과를 얼마나 믿을 수 있는지는 알 수 없다는 것이다. 특히, 자율주행과 같이 localization에 대한 예측 신뢰도가 굉장히 중요한 application에서는 해당 localization 예측에 대한 검출 신뢰도, 즉, 불확실성을 추가로 예측하고 활용하는 것이 매우 중요 [2]하다. 이러한 이유로 심층 신경망에서 uncertainty를 예측하기

위한 연구들 [7], [8]이 많이 수행되어왔지만, 심층 탐지 신경망의 localization task에서 명확한 솔루션을 제시한 연구는 없었다.

이러한 문제 외에도, 심층 탐지 신경망을 학습하기 위한 학습 데이터 구성 시 높은 labeling cost를 요구한다는 문제가 있다. 일반적으로 지도 학습 (supervised learning) [9]에서 심층 탐지 신경망의 높은 정확도를 얻기 위해선, 많은 양의 labeled 데이터가 필요하다 [5], [10]. 단순히 이미지 데이터를 얻는 것은 쉽지만, 해당 이미지 내 물체의 종류 (즉, class category)와 물체의 localization 정보 (즉, bounding box coordinate)는 사람이 직접 labeling을 해야한다. 심층 탐지 신경망의 데이터는 이미지 내에 여러 물체가 있기 때문에 사람이 모든 데이터를 labeling하기에는 굉장히 많은 시간 및 인력 비용이 필요하다 [11]. 이러한 문제를 해결하기 위해 능동적 학습 (active learning) [12]이 활발히 연구되어 왔다. 능동적 학습은 현재 모델이 잘 모르는 데이터만을 labeling하여 학습 데이터로 사용하는 학습 기법이다. 해당 방법을 통해 적은 labeling cost로 심층 탐지 신경망의 높은 정확도 향상이 가능하다. 하지만, 이전의 심층 탐지 신경망을 위한 능동적 학습 연구들은 object detection이지만 classification 정보에만 의존해 왔다 [13]. 그리고 정확한 classification uncertainty 예측을 위해 multiple models 기반의 능동적 학습 방법을 사용 [14]했기 때문에 굉장히 큰 연산 비용을 요구하여 소비 전력 및 resource 측면에서 굉장히 비효율적이다. 이러한 단점을 보완하기 위해 single model 기반의 방법들 [15], [16] 및 localization 정보를 classification과 함께 활용한 연구들 [15], [17]이 제안되었지만, heuristic한 접근으로 multiple models 기반 방법 대비 정확도가 낮은 문제가 있다. 즉, 이전 연구들은 심층 탐지 신경망을 위한 능동적 학습에서 완전한 해결책을 제시하지 못했다.

능동적 학습 외에도 labeling cost를 줄이고 높은 정확도를 달성하기 위해 최근에는 심층 탐지 신경망을 위한 반지도 학습 (semi-supervised learning)이 연구 [18] 되고 있다. 반지도 학습은 annotation이 있는 labeled 데이터와 이미지만 있는 unlabeled 데이터를 함께 사용하는 학습 기법으로, unlabeled 데이터에 대해서는 labeling이 필요한 능동적 학습과는 다르게 추가 labeling을 요구하지 않는다. 그렇기 때문에 능동적 학습보다 labeling cost를 더 줄일 수 있지만, unlabeled 데이터에 대해서는 모델 예측 결과를 pseudo-label [19]로 사용하기 때문에 모델 예측이 잘못되거나 부정확한 경우에 대해서는 대응할 수 없다. 무엇보다도, 기존 반지도 학습은 unlabeled 데이터 자체의 유용성을 판단하지 못한 채 보유한 모든 unlabeled 데이터를 학습에서 그대로 사용하기 때문에 많은 학습 시간을 요구한다는 문제가 있다. 이는 결과적으로 소비 전력 및 resource 측면에서 비효율성을 증가시킨다.

1.2 연구의 내용

본 논문은 심층 탐지 신경망의 false positive (즉, mislocalization) 해결을 위한 localization uncertainty 모델링과 활용, 연산 효율과 정확도를 향상시킨 object detection을 위한 능동적 학습, 그리고 반지도 학습의 학습 데이터 효율성 향상을 위한 unlabeled 데이터 필터링을 제안한다.

첫째로, 기존 object detection의 false localization 문제를

해결하기 위해 본 논문은 bounding box coordinate의 confidence인 localization uncertainty를 모델링하는 방법을 제안한다. 제안 방법은 Gaussian distribution을 사용하며, bounding box coordinate과 localization uncertainty를 학습하기 위한 새로운 localization loss function을 제안한다. 그리고 이를 검출 과정에서 class score와 함께 활용함으로써 false localization 문제를 해결하고 정확도를 대폭 향상시킨다.

둘째로, 본 논문은 기존 능동적 학습 방법의 높은 연산량 문제를 해결하면서도, 높은 정확도를 유지하기 위해, mixture density network [20]를 사용하여 single model 기반의 single forward pass로 classification과 localization의 두 task에 대한 uncertainty 예측 방법을 제안한다. 그리고 두 task에서 aleatoric과 epistemic uncertainty 예측을 위해 새로운 classification loss와 localization loss를 제안한다. 능동적 학습에서 예측한 localization과 classification uncertainty를 모두 활용하여 기존 대비 연산량을 대폭 줄이고 정확도를 향상시킨다.

마지막으로, 기존 반지도 학습의 학습 효율성 문제를 해결하기 위해 본 논문은 능동적 학습에서 제안한 uncertainty 예측 방법을 반지도 학습에 활용하여, 전체 unlabeled 데이터에서 낮은 uncertainty를 보이는 데이터를 필터링하는 방법을 제안한다. 즉, 모델이 이미 잘 알고 있거나, 학습 데이터와 닮은 특징을 보이는 unlabeled 데이터를 전체 학습 데이터에서 제거함으로써, 기존 방법 대비 학습 시간을 대폭 감소시키고, 정확도를 향상시킨다.

1.3 논문 구성

본 논문의 구성은 다음과 같다. 2장에서는 본 논문을 이해하는데 필요한 배경지식을 설명한다. 3장에서는 심층 탐지 신경망에서 localization uncertainty를 모델링하고 활용하는 방법에 대해 제시한다. 4장에서는 확률적 모델링을 통한 고성능 저비용 능동적 학습 방법을 제시한다. 5장에서는 심층 탐지 신경망을 위한 반지도 학습에서 uncertainty 기반 unlabeled 데이터 필터링 방법과 제안 반지도 능동적 학습 방법을 제시한다. 마지막으로 6장에서 결론으로 논문을 마무리한다.

제 2 장 연구의 배경지식

2.1 객체 탐지 (object detection)

Object detection은 computer vision의 대표적인 문제 중 하나이다. Object detection은 그림 2.1과 같이 물체의 카테고리 정보를 예측하는 classification과 물체의 위치 및 크기를 예측하는 localization이 결합된 문제이다. 일반적으로, 하나의 이미지 내 여러 개의 물체를 예측해야 하므로 classification 보다 어려운 문제이며, 자율주행 [2] 등과 같은 application에서는 매우 필수적인 기술 [21] 이다. Object detection 문제를 풀고 그 활용 범위를 넓히기 위해 딥러닝 기반 object detection 알고리즘들이 활발히 연구되고 있다 [1], [22]. 최근까지 많은 네트워크 구조 및 방법론들이 제안되어 왔으며, 정확도뿐만 아니라 임베디드 환경에서의 알고리즘 처리 속도 또한 향상되고 있다 [23]. 이러한 딥러닝 기반 object detection 알고리즘은 크게 두 가지 방식으로 구별할 수 있다.

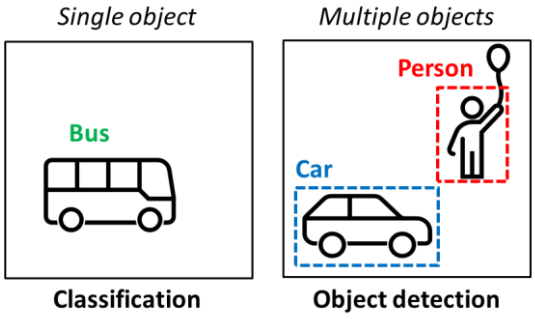


그림 2.1. Classification과 object detection의 차이.

2.1.1 Two-stage object detection

Two-stage object detection 방식은 그림 2.2와 같이 region proposal stage와 detection stage (즉, bounding box regression과 classification)의 두 가지 stage를 순차적으로 수행하는 방식이다. Region proposal generator란 이미지에서 모든 영역을 scanning하는 sliding window 방식 [24]과는 다르게, 물체가 있을 법한 영역을 빠르게 찾아내는 방식으로 기존의 sliding window 방식보다 효율성이 높다. 대표적인 알고리즘으로는 RCNN [24], Fast-RCNN [25], 그리고 Faster-RCNN [5]이 있다. 일반적으로 two-stage 기반의 object detection 방식은 정확도가 높으나 처리 속도가 느리다는 단점이 있다.

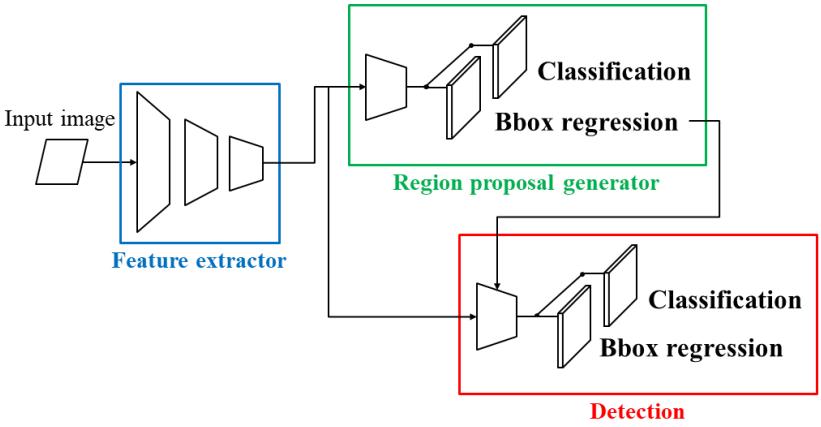


그림 2.2. Two-stage object detection의 프로세스.

2.1.2 One-stage object detection

One-stage object detection 방식은 그림 2.3과 같이 region proposal stage 없이 bounding box regression과 classification을 한 번에 수행하는 방식이다. 이 방식은 이미지 내 모든 위치에 물체가 있다고 가정하고, 해당 위치에서 objectness나 backgound class score를 통해 실제 물체 존재 여부를 판단한다. 대표적인 알고리즘으로는 YOLO [9]와 SSD [10]가 있으며, 일반적으로 two-stage 방식보다 정확도가 떨어지지만, 굉장히 빠른 처리 속도가 장점이다. 그렇기 때문에 real-time application에서 주로 많이 사용된다 [2].

본 연구에서는 처리 속도가 빠르고 효율성이 뛰어난 one-stage object detection을 기반으로 알고리즘 모델링을 설명한다. 하지만, 제안 방식은 one-stage 방식뿐만 아니라, two-stage 방식에도 적용이 가능하다. 제안 방식의 scalability는 본 논문의 실험 결과 장에서 보인다.

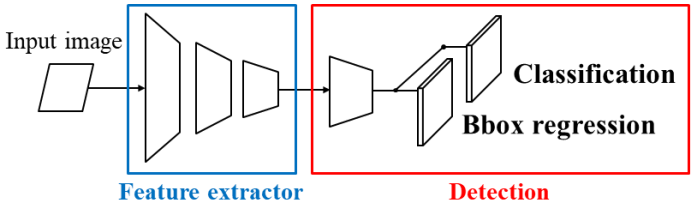


그림 2.3. One-stage object detection의 프로세스.

2.2 능동적 학습 (active learning)

능동적 학습은 그림 2.4와 같이 딥러닝 모델이 labeling할 데이터 선택에 관여하여, labeling cost를 줄이고 정확도를 향상시키는 학습 기법이다. 능동적 학습의 프로세스는 초기 labeled 데이터로 학습된 모델을 기반으로 모델의 성능 향상에 도움이 되는 informative한 unlabeled 데이터를 선택한다. 그 후 annotator (예를 들어, 사람)이 선택된 unlabeled 데이터를 labeling하여 정답 annotation을 만든다. 그 후 이 데이터를 기존 학습 데이터에 추가하고, 수집된 labeled 데이터를 사용해 모델을 재학습한다. Unlabeled 데이터 중에서 모델이 잘 모르는 데이터를 선택해야 하기 때문에 “informative” 정도를 나타내는 정량적 지표를 모델링하는 것이 능동적 학습에서 가장 중요하다. 이러한 정량적 지표를 모델링하는 접근법으로는 능동적 학습 survey [12]에 잘 나와 있듯이, uncertainty 기반 방법 [26], [27], [28], [29], [30], SVM 기반 방법 [31], [32], query-by-committee 기반 방법 [33], [34], 그리고 distribution 기반 방법 [35], [36], [37], [38], [39], [40] 등이 있다.

능동적 학습을 위한 정량적 지표를 계산할 때, uncertainty 기반 방법들은 entropy [41] 혹은 mutual information [42] 등 예측 class의 posterior probability을 사용하거나, 예측 class와 두 번째 예측 class의 margin [28], [43] 등을 사용하였다. SVM 기반 방법은 distance [31], [32], [44]를 uncertainty처럼 사용하였고, Query-by-committee 방법은 여러 모델들을 독립적으로 구성하여 각각 예측의 편차를 uncertainty로 정의하였다 [33], [34]. Distribution 기반 방법 [16]은 데이터 distribution에서 가장 대표되는 데이터들을 선택하는 방법으로 다른 방법들과는 다르게 모델의 예측 결과에 의존하지 않는다.

기존의 능동적 학습 연구들은 classification task에 집중되어 왔다 [45], [46], [47]. Object detection에서 능동적 학습을 다룬 연구들은 많지 않으며, 있다고 하더라도 기존 classification 기반 방법들을 object detection으로 간단히 확장한 방식이 대부분이다 [14], [48]. 본 논문에서는 object detection을 위한 novel한 능동적 학습 방법을 제안하며, 본 논문의 제안 방법은 대표적인 4가지 능동적 학습 방법 중 uncertainty 방식을 기반으로 한다.

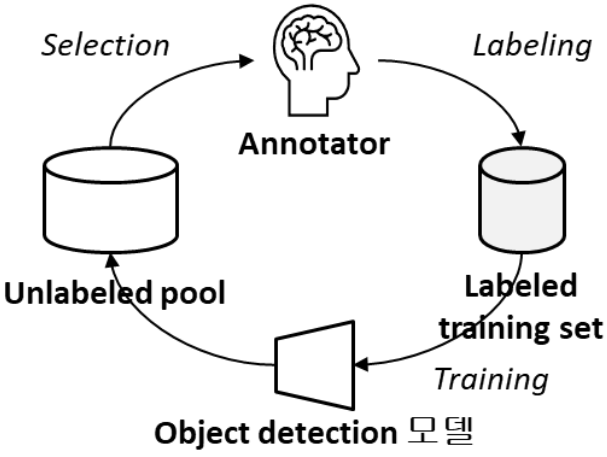


그림 2.4. 능동적 학습 loop.

2.3 반지도 학습 (semi-supervised learning)

반지도 학습은 그림 2.5와 같이 정답 label이 제공되는 labeled 데이터와 이미지 데이터로만 구성된 unlabeled 데이터를 함께 사용하여 모델을 학습하는 기법으로, labeling cost 감소와 정확도 향상이 목표이다. Real-world problem에서는 labeled 데이터보다 unlabeled 데이터의 양이 훨씬 많으며, 많은 양의 unlabeled 데이터를 labeling하는 것은 굉장히 많은 resource와 시간을 요구한다. 이러한 문제를 해결하기 위해 다양한 task에서 반지도 학습이 활발히 연구되고 있다 [18], [49], [50].

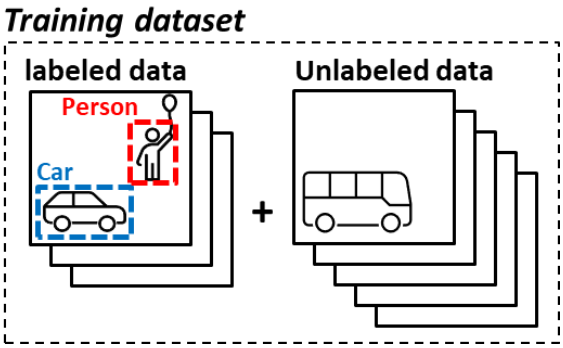


그림 2.5. 반지도 학습의 학습 데이터 예.

반지도 학습은 크게 두 가지 방법으로 구별된다. 첫째는 self-training [49], [50]으로 그림 2.6과 같이 labeled 데이터로 학습한 모델을 기반으로 unlabeled 데이터를 예측하고, 이 예측된 결과를 unlabeled 데이터의 정답 label (즉, pseudo-label)로 사용하는 것이다. 이 방식은 신뢰할 수 있는 예측 결과만을 pseudo-label로 얻기 위해 threshold를 사용하며, confidence가 낮은 예측 결과는 필터링하고 confidence가 높은 예측 결과만을 학습 label로 사용한다. 이 방식은 간단함에도 불구하고 좋은 성능을 보이지만, confidence 기반 필터링 과정이 필요하므로 전체 학습 시간을 증기시키는 문제가 있다. 또한 label 선택을 위해 미리 정한 threshold에 의존한다는 문제가 있다. 즉, threshold가 너무 작으면 매우 많은 sample이 label되어 성능을 감소시키며, 너무 높은 threshold는 pseudo-label의 수를 줄여 큰 성능 향상 효과를 주지 못한다. 그렇기 때문에 self-training은 hyper-parameter로 인해 성능이 불안정하다는 문제가 있다.

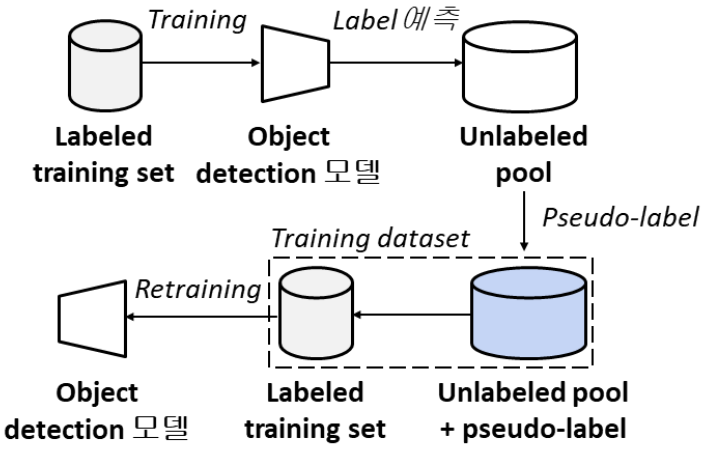


그림 2.6. Self-training 프로세스.

두번째 방법은 consistency regularization으로 그림 2.7과 같이 원본 이미지와 변형된 이미지 (flip된 이미지)를 모델의 입력으로 넣은 후 각 출력의 예측 차이가 최소화되도록 모델을 학습시키는 것이다 [18]. 이 방법은 출력 차이를 최소화하는 loss (즉, consistency loss)를 사용하기 때문에 self-training처럼 추가적인 pseudo-label을 구하는 과정을 필요로 하지 않는다. 그림 2.7(a)와 같이 labeled 데이터가 입력된 경우 ground-truth가 있으므로, 기존 supervised learning loss와 consistency loss를 함께 고려하여 모델을 학습하며, 그림 2.7(b)와 같이 unlabeled 데이터의 경우는 ground-truth가 없으므로 consistency loss로만 모델을 학습한다. 이 방법은 threshold 등 hyper-parameter에 크게 의존하지 않기 때문에 최근 object detection의 반지도 학습 연구는 이 방식을 기반으로 활발히 연구되고 있다 [18] [51].

본 논문의 제안 방법은 consistency regularization [18]을 기반으로 한다.

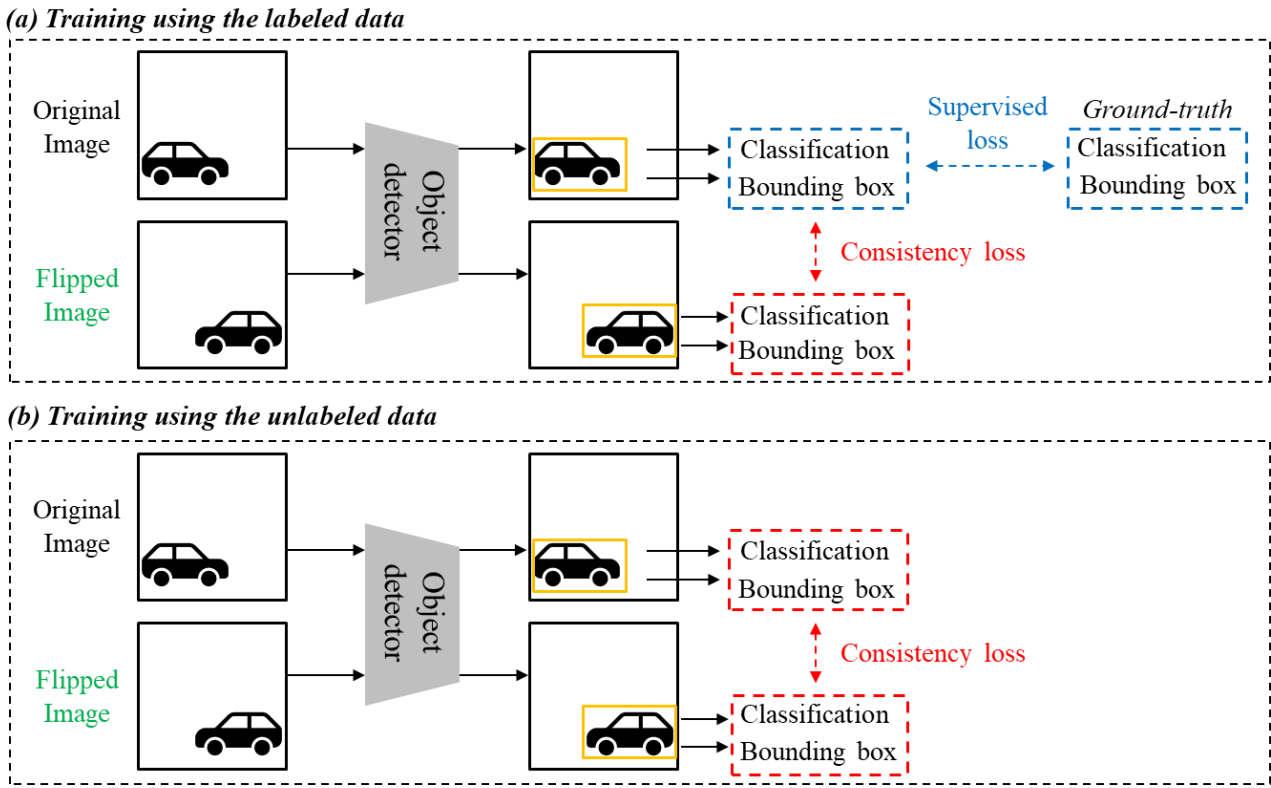


그림 2.7. Consistency 기반 반지도 학습의 loss 계산 예.

제 3 장 Localization uncertainty를 사용한 정확하고 빠른 객체 탐지기

3.1 본 연구의 서론

최근 딥러닝은 기존 machine learning 기반 방식보다 뛰어난 성능을 보이며 컴퓨터 비전 [52], 자율주행 [53], 그리고 소셜 네트워크 서비스 [4] 등 다방면의 분야에 활발히 적용되고 있다. 특히, 딥러닝과 더불어 영상 센서와 GPU의 발달은 인공지능 기반의 자율주행 자동차 연구를 가속시켰다. 자율주행 자동차는 운전자의 개입 없이 스스로 안전 주행을 하는 자동차로, 안전을 보장하는 올바른 제어 결정을 내리기 위해 차량, 보행자, 표지판, 신호등 등을 실시간으로 정확히 탐지 [54]해야 한다. 이러한 객체 탐지 (object detection)를 위해 자율주행 자동차는 카메라 센서, 라이다 (Lidar), 레이더 (Radar) 등 다양한 센서들을 융합하여 사용하는 것이 일반적 [55]이다. 이러한 다양한 센서들 중에서 카메라 센서의 경우, texture와 color feature들을 통해 물체의 종류를 정확하게 파악할 수 있을 뿐만 아니라 여러 센서들 중 가장 cost-effective [56]하다. 최근 딥러닝 기반의 카메라 센서를 활용한 객체 탐지 기술은 사람보다 더 높은 검출 정확도를 보여 자율주행 자동차의 인식 처리에서 중요도가 높아지고 있으며, 이로 인해 이미지 기반의 객체 탐지는 자율주행 시스템에서 대체할 수 없는 필수적인 방식 [21]이 되었다.

자율주행 자동차를 위한 object detection 알고리즘은 다음과 같은 두 가지 조건을 만족해야 한다. 첫째, 도로객체에 대한 높은 인식

정확도가 필요하다. 둘째, 검출 결과에 대한 자동차 제어 단의 빠른 대응과 latency 감소를 위해 실시간의 처리 속도가 필수적이다. 자율주행 자동차에서 필수적으로 사용되는 딥러닝 기반 object detection 알고리즘은 2장에서 언급하였듯이 두 종류로 분류할 수 있다: Two-stage detector [25], [5], [57]와 one-stage detector [9], [10]. 최근에는 각 방법의 장점을 취하고, 단점을 보완하기 위한 object detection 알고리즘들이 활발히 연구되어왔다 [21], [58], [59], [60], [61]. Two-stage detector인 MS-CNN [58]은 detection을 network의 다양한 중간 layer에서 수행하여 처리 속도를 향상시켰고, two-stage detector인 SINet [21]은 scale-insensitive network를 통해 빠른 검출을 가능하게 하였다. One-stage detector인 CFENet [59]은 SSD를 기반으로 comprehensive feature enhancement 모듈을 사용해 검출 정확도를 향상시켰고, one-stage detector인 RefineDet [60]은 anchor refinement module과 object detection module을 이용하여 검출 정확도를 향상시켰다. 또 다른 one-stage detector인 RFBNet [61]은 receptive field block을 적용하여 정확도를 향상시켰다. 하지만, 기존 연구들 [21], [59], [59], [60]은 높은 정확도를 위한 512×512 이상의 input resolution에서 자율주행 application의 필수조건인 30 fps (frames per second) 이상의 실시간 처리 속도를 만족하지 못한다. 실시간 처리가 가능하다 할지라도 [61], 낮은 정확도로 인해 자율주행 application에 사용되기에는 한계가 있다. 즉, 이전 연구들은 정확도와 처리 속도 간의 trade-off 측면에서 완전하지 못하며, 결과적으로 자율주행 자동차에 적용되어 사용되기에는 한계가 있다.

이러한 기존의 딥러닝 기반 object detection 알고리즘들의 또 다른 문제점 중 하나는 검출 대상의 bounding box coordinates (즉,

localization) 정보는 알 수 있지만 이 결과가 얼마나 불확실한지는 알지 못한다는 점이다. 즉, 결과의 불확실성에 대한 정보가 없이 deterministic한 결과만을 출력하므로 mislocalization (즉, false positive) 문제에 대응하지 못한다. 자율주행에서 false positive는 해당 물체의 ground-truth 대비 잘못된 bounding box 예측 또는 부정확한 bounding box 예측을 의미하며, true positive는 ground-truth 대비 정확한 bounding box 예측을 의미한다. 자율주행 자동차에서 mislocalization에 의한 false positive는 예상치 못한 제동과 같은 과도한 반응을 일으켜 주행 안정성과 효율성을 크게 떨어뜨리고, 정확한 객체 탐지 결정에 혼란을 주어 치명적인 사고 [62], [63]를 유발할 수 있기 때문에 매우 위험하다. 즉, false positive를 줄이면, 자율주행 사고를 줄일 수 있기 때문에 검출된 bounding box의 uncertainty를 파악하고 이를 활용해 object detection에서 false positive를 줄이는 것이 굉장히 중요하다. 더욱이, 최근 자율주행 자동차 관련 사고로 인해 자율주행 application에서 uncertainty의 예측과 활용의 필요성은 더욱 증가하고 있다. 이러한 이유로 딥러닝에서 uncertainty를 예측하기 위한 연구들이 활발히 수행되어 왔다. Kendall *et al.* [7]은 Bayesian neural network를 사용하여 딥러닝에서 uncertainty 예측을 위한 모델링 방법을 제안하였고, Feng *et al.* [64]은 Kendall *et al.* [7]의 방식을 Lidar 센서의 3D vehicle detection에 적용하여 uncertainty를 예측하는 방법을 제안하였다. 하지만, Kendall *et al.* [7]과 Feng *et al.* [64]은 uncertainty를 예측만 하였고 이를 실제 application에서 활용하지 못했다. Choi *et al.* [8]는 Gaussian mixture model을 이용해 uncertainty를 실시간으로 예측하여 자율주행 application에 활용하였으나, object detection이 아닌 steering angle에 적용하였으며, steering angle 처리를 위해 복잡한 분포를 모델링하기 때문에 연산

복잡도가 증가하는 문제가 있다. He *et al.* [65]는 object detection에서 불확실성을 예측하고 활용하는 방법을 제안했다. 하지만, two-stage object detection을 기반으로 하여 실시간 처리가 어렵고 bounding box의 overlap 문제가 발생하기 때문에 자율주행 application에 적용되기에는 한계가 있다.

기존 연구들의 이러한 문제들을 보완하기 위하여 본 장에서는 YOLOv3 [6]를 기반으로 자율주행에 적합한 새로운 object detection 알고리즘을 제안한다. YOLOv3는 한번의 inference로 여러 object 검출이 가능해 처리 속도가 굉장히 빠르며, multi-stage detections 방식을 적용해 처리 속도 대비 정확도가 우수하다. 이러한 장점으로 YOLOv3는 자율주행 application에 사용하기 적합하지만, 일반적으로 two-stage 방식보다는 상대적으로 낮은 정확도를 보이기 때문에 실시간의 처리 속도를 유지하면서 정확도를 개선하는 것이 필수적이다. 본 장에서는 YOLOv3를 기반으로 기존에 deterministic한 결과만을 출력하던 bounding box coordinates 정보를 평균과 분산의 Gaussian parameter로 모델링하고, 이를 고려해 loss function을 재설계함으로써 검출 정확도를 향상시키는 방법을 제안한다. 그리고 이러한 Gaussian 모델링을 통해 bounding box regression task를 위한 localization uncertainty 예측 방법을 제안한다. 또한, 예측된 localization uncertainty를 YOLOv3의 검출 과정에 활용하여 bounding box coordinates의 localization uncertainty와 objectness score, 그리고 class들에 대한 confidence scores를 검출 과정에서 모두 고려하여 false positive를 대폭 줄이고 검출 정확도를 향상시키는 방법을 제안한다. 즉, 본 장은 최초로 YOLOv3에 localization uncertainty를 모델링하고 이를 활용한 연구로 기존의 자율주행을 위한 카메라 기반 영상 처리에서는 불가능했던 mislocalization에 대한 대응이 가능하다.

제안 방식은 YOLOv3의 detection layer (즉, output layer)의 bounding box head에만 모델링이 적용되기 때문에 연산 증가량이 매우 적다. 즉, 제안 알고리즘은 정확도를 대폭 향상시키면서도 512×512의 입력 resolution에서 42 fps 이상의 처리 속도를 보여 실시간 처리를 지원한다. Baseline 알고리즘인 YOLOv3와 비교 시, 제안 알고리즘인 Gaussian YOLOv3는 KITTI [66]와 Berkeley Deep Drive (BDD) [67] 데이터에서 각각 3.09 mAP (mean average precision), 3.5 mAP의 높은 정확도 향상을 보인다. 게다가, 제안 알고리즘은 KITTI와 BDD 데이터 각각에 대해 false positive를 41.40%, 40.62% 감소시키며, true positive를 7.26%, 4.3% 증가시킨다. 결과적으로, 제안 알고리즘은 정확도와 처리 속도의 trade-off 측면에서 우수하며, 정확도를 대폭 향상시키고 mislocalization 문제를 해결함과 동시에 실시간 탐지를 지원하기 때문에 자율주행 자동차에 가장 적합하다.

3.2 본 연구의 배경

YOLO [9]는 two-stage detector들의 region proposal 방식 대신에 이미지를 grid 단위로 나누어 처리하며, detection layer (즉, output layer)의 feature map을 bounding box coordinates, objectness score 그리고 class (confidence) scores가 출력되도록 설계하여 한번의 inference로 여러 object 검출이 가능하게 하였다. 그렇기 때문에 기존 방식들에 비해 검출 처리 속도가 매우 빠르다. 하지만 grid 단위의 처리로 인해 localization error가 크고 검출 정확도가 낮아

자율주행 application에서 사용되기에는 적합하지 않다. 이러한 문제를 해결하기 위해 YOLOv2 [68]가 제안되었다. YOLOv2는 모든 convolution layer에 batch normalization을 추가하고 anchor box, multi-scale training, 그리고 fine-grained feature 등을 적용하여 기존 YOLO [9]보다 검출 정확도를 개선하였다. 하지만 여전히 작은 물체와 dense한 물체들에 대해 검출 정확도가 낮다는 문제가 있다. 그러므로 표지판, 신호등과 같은 작은 물체와 밀집된 차량 등에 대해 높은 검출 정확도가 필요한 자율주행 application에 사용되기에는 한계가 있다.

이러한 YOLOv2의 단점을 보완하기 위해 YOLOv3 [6]가 제안되었다. YOLOv3는 그림 3.1과 같이 convolution layer로만 구성되며, deep network에서 vanishing gradient 문제 처리에 도움이 되는 residual skip connection과 작은 물체 검출을 위해 fine grained feature를 보존하는 up-sampling 및 concatenation 기법을 적용하였다. 가장 두드러진 특징은 feature pyramid network [69]와 비슷한 방법으로 3개의 다른 scale에서 detection을 수행하는 것이다. 이를 통해 YOLOv3는 여러 크기의 물체를 검출할 수 있다. 좀 더 자세히 설명하면, YOLOv3의 입력으로 R, G, B의 3 channel 이미지가 입력되면, 그림 3.1의 네트워크를 거쳐 3개의 detection layer에서 물체 검출을 위한 정보 (즉, bounding box coordinates, objectness score, class scores)를 출력한다. 그리고 이들 결과를 종합하여 non-maximum suppression으로 후 처리함으로써 최종 검출 결과를 얻는다. YOLOv3는 1×1와 3×3의 작은 크기의 convolution filter로만 구성된 fully convolution network로 기존 YOLO [9], [68]처럼 처리 속도가 빠르며, 여러 크기의 물체에 대해서 고른 정확도를 보인다. 이러한 관점에서 YOLOv3는 자율주행 application에 사용하기 적합하여, 자율주행

연구에서 활용 [70]되고 있다. 하지만, region proposal stage를 사용하는 two-stage object detector보다는 상대적으로 낮은 정확도를 보인다. 이러한 문제를 보완하기 위해, 기존의 two-stage detector들보다 연산량이 작은 YOLOv3에 localization uncertainty를 예측하고 알고리즘의 성능을 향상시킬 수 있는 기법을 적용함으로써 자율주행 application에 더 효율적인 알고리즘을 설계할 수 있다. Bounding box에 대한 Gaussian 모델링과 loss function 재설계를 통한 loss attenuation으로 학습 시 noisy한 데이터의 영향을 줄여 정확도를 향상시킨다. Inference 시 localization uncertainty를 예측하고, 예측한 localization uncertainty를 검출 과정에 활용함으로써 false positive를 대폭 줄인다. 다음 장에서 이에 대해 자세히 다룬다.

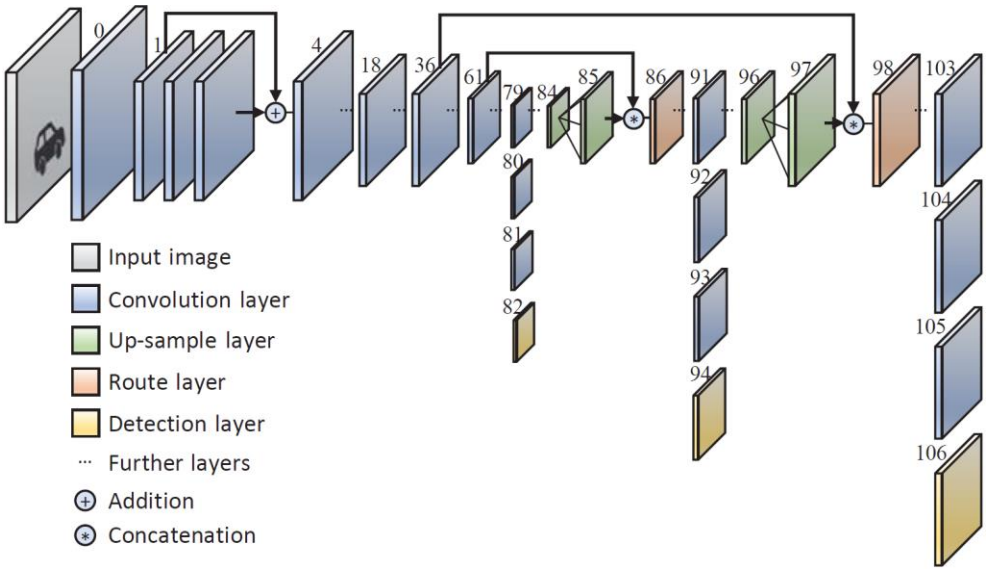


그림 3.1. YOLOv3의 네트워크 구조.

3.3 Gaussian YOLOv3

3.3.1 Gaussian 모델링

YOLOv3 [6]의 prediction feature map은 그림 3.2와 같이 grid 당 3개의 prediction box를 가지며, 각 prediction box는 bounding box coordinates인 t_x, t_y, t_w, t_h와 objectness score, 그리고 class scores로 구성된다. YOLOv3는 물체 존재 여부를 의미하는 objectness와 해당 물체가 어떤 카테고리인지 의미하는 class score를 0과 1사이 값으로 출력하여, 이 두 값의 곱을 기준으로 이미지 내의 물체들을 검출한다. 이때, bounding box coordinates은 objectness와 class들과 다르게 score가 아닌 deterministic한 좌표 값으로 출력되기 때문에 검출 bounding box의 신뢰도는 알 수 없다. 즉, 검출 bounding box가 얼마나 불확실한지 알 수 없다. 반면에, 본 장에서 제안하는 bounding box의 localization uncertainty는 bounding box에 대한 score 역할을 하기 때문에 검출 bounding box가 얼마나 불확실한지를 나타내는 지표로 활용 가능하다.

YOLOv3에서 bounding box regression은 bounding box의 center 정보인 t_x, t_y coordinate과 box 크기 정보인 t_w (width), t_h (height) coordinate의 추출이 목적이다. 해당 정보들은 물체에 대한 정답이 하나로 정해져 있기 때문에 본 장에서는 복잡한 분포를 사용하지 않고 t_x, t_y, t_w, t_h 각각에 대한 single Gaussian distribution으로 localization에 대한 uncertainty를 모델링한다. 출력이 Gaussian parameter로 구성된 주어진 test input x에 대한 출력 y의 single Gaussian model의 식은 아래와 같다:

$$p(y|x)=N(y;\mu(x),\Sigma(x)), \tag{3.1}$$

$\mu(x)$, $\Sigma(x)$은 각각 mean function 과 variance function 이다.

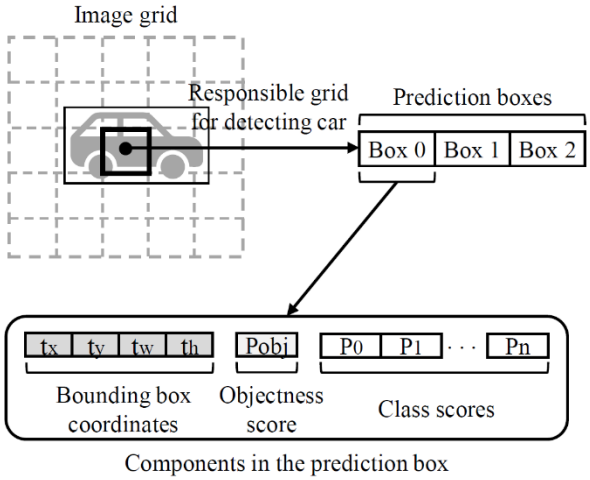


그림 3.2. YOLOv3의 prediction feature map.

Bounding box의 uncertainty 예측을 위해 그림 3.3 과 같이 기존 YOLOv3의 prediction feature map의 bounding box coordinate 각각을 Gaussian parameter인 평균(μ)과 분산(Σ)으로 모델링 한다. 즉, bounding box의 output은 $\hat{\mu}_{tx}$, $\hat{\Sigma}_{tx}$, $\hat{\mu}_{ty}$, $\hat{\Sigma}_{ty}$, $\hat{\mu}_{tw}$, $\hat{\Sigma}_{tw}$, $\hat{\mu}_{th}$, $\hat{\Sigma}_{th}$가 된다. YOLOv3의 detection 처리를 위해 t_x, t_y, t_w, t_h에 대한 Gaussian parameter는 다음과 같이 전처리된다:

$$\mu_{tx}=\rho(\hat{\mu}_{tx}),\ \mu_{ty}=\rho(\hat{\mu}_{ty}),\ \mu_{tw}=\hat{\mu}_{tw},\ \mu_{th}=\hat{\mu}_{th} \tag{3.2}$$

$$\Sigma_{tx}=\rho(\hat{\Sigma}_{tx}),\ \Sigma_{ty}=\rho(\hat{\Sigma}_{ty}),\ \Sigma_{tw}=\rho(\hat{\Sigma}_{tw}),\ \Sigma_{th}=\rho(\hat{\Sigma}_{th}) \tag{3.3}$$

$$\rho(x)=\frac{1}{1+exp(-x)} \tag{3.4}$$

Inference 시 detection layer (즉, output layer)의 각 coordinate 의 평균이 검출 bounding box 의 좌표가 되며, 각각의 분산이 각 좌표의 uncertainty 를 나타낸다. 식 (3.2)에서 t_x, t_y 의 좌표 값을 의미하는 μ_{tx}, μ_{ty}는 각각 grid 내부의 x, y 좌표를 표현해야 하므로 0 과 1 사이의 값을 갖도록 식 (3.4)의 sigmoid function 으로 처리하며, 식 (3.3)은 각 coordinate 의 분산으로, 이 또한 0 과 1 사이의 값을 갖도록 식 (3.4)의 sigmoid function 으로 처리한다. YOLOv3 에서 width 와 height 는 bounding box prior 와 exponential function [6]을 통해 처리되기 때문에 식 (3.2)에서 bounding box 의 t_w, t_h 정보인 μ_{tw}, μ_{th}는 사전 처리없이 출력 값을 그대로 사용한다.

YOLOv3 에서 bounding box 의 uncertainty 예측을 위한 single Gaussian 모델링은 그림 3.1 의 detection layer 의 bounding box head 에만 적용하기 때문에 이로 인한 알고리즘의 전체 연산량은 거의 증가하지 않는다. 512×512 의 input resolution, 10 개 class 기준으로 기존 YOLOv3 는 135.24×10^9 FLOPs (FLoating point Operations Per Second)의 연산량을 요구하지만, single Gaussian 모델링 후에는

135.28×10^9 의 연산량을 요구한다. 즉, 기존에 비해 연산량이 0.03% 밖에 증가하지 않기 때문에 처리 속도 penalty 는 거의 없다.

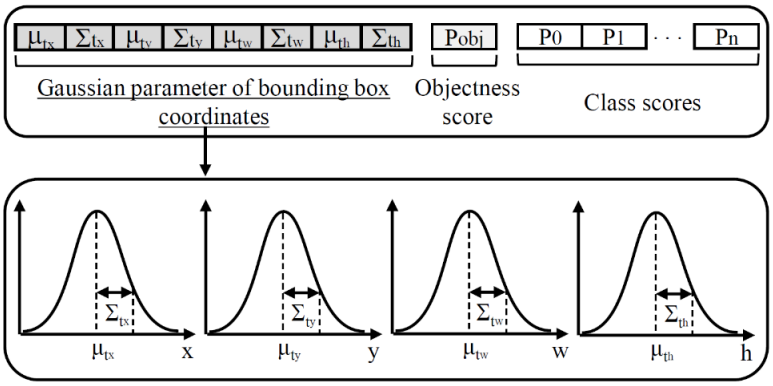


그림 3.3. 제안 알고리즘의 prediction box의 구성요소.

3.3.2 Loss function 재설계

기존 YOLOv3 [6]는 bounding box coordinate에 대해서는 sum of squared error loss를 사용하며, objectness와 class에 대해서는 binary cross-entropy loss를 사용한다. Gaussian 모델링으로 YOLOv3의 bounding box coordinate이 Gaussian parameter로 출력되기 때문에 이를 고려하여 기존 bounding box coordinate의 loss function을 negative log likelihood (NLL) loss로 재설계한다. Objectness와 class에 대해서는 기존 loss function을 그대로 사용한다. Bounding box

coordinate에 대한 제안 loss function은 다음과 같다:

$$L_x = -\sum_{i=1}^{W}\sum_{j=1}^{H}\sum_{k=1}^{K}\gamma_{ijk}\, log(N(x_{ijk}^{G}|\mu_{tx}(x_{ijk}),\Sigma_{tx}(x_{ijk}))+\varepsilon), \qquad (3.5)$$

식 (3.5)의 L_x은 t_x coordinate의 NLL loss이며, 다른 coordinate에 대한 loss들 (즉, L_y, L_w, L_h)은 L_x에서 x parameter를 제외하고는 모두 동일하다. 식에서 W와 H는 *width*와 *height* 각각의 grid 수를 의미하며, K는 anchor 수를 의미한다. $\mu_{tx}(x_{ijk})$는 제안 알고리즘의 detection layer의 t_x coordinate의 출력으로 (i, j)번째 grid center와 k번째 anchor에서 bounding box coordinate을 의미한다. $\Sigma_{tx}(x_{ijk})$은 detection layer의 출력으로 t_x coordinate의 uncertainty를 의미한다. 학습을 위해 bounding box의 ground-truth는 다음과 같이 계산된다:

$$x_{ijk}^{G}=x^{G}\times W-i, \quad y_{ijk}^{G}=y^{G}\times H-j \qquad (3.6)$$

$$w_{ijk}^{G}=log(\frac{w^{G}\times IW}{A_k^w}), \quad h_{ijk}^{G}=log(\frac{h^{G}\times IH}{A_k^h}) \qquad (3.7)$$

x^G, y^G, w^G, 그리고 h^G는 이미지 내 ground-truth box의 ratio를 의미하며, IW와 IH는 resize된 이미지의 *width*와 *height*를 의미한다. A_k^w와 A_k^h는 k 번째 anchor box prior의 *width*와 *height*를 의미한다. YOLOv3에서 bounding box의 centroid는 grid 단위로 계산되며, bounding box의 size는 anchor box를 기반으로 계산된다. 따라서 학습을 위해 식 (3.5)의 hyper-parameter는 다음과 같이 계산된다:

$$\gamma_{ijk} = \frac{\omega_{scale} \times \delta_{ijk}^{obj}}{2} \tag{3.8}$$

$$\omega_{scale} = 2 - w^G \times h^G. \tag{3.9}$$

식 (3.8)의 w_{scale}은 식 (3.9)와 같이 사전에 정의된 anchor box prior 의 width 와 height 를 이용하여 계산되는 값으로 학습 시 물체 크기에 따라 가중치를 다르게 주는 역할을 한다. δ_{ijk}^{obj}은 사전에 정의된 anchor 들 중 현재 detection layer 에서 가장 적합한 anchor 에 대해서만 loss 에 포함시키는 역할을 하는 parameter 로 (i, j)에 있는 k 번째 anchor 와 ground-truth 의 intersection over union (IoU)이 가장 큰 경우에 1 로 할당되며, 해당 위치에서 알맞은 ground-truth 가 없는 경우 0 으로 할당된다. Logarithm function 의 numerical stability 를 위해 식 (3.5)의 ε는 10^{-9}로 할당한다.

기존 YOLOv3 는 deterministic 한 bounding box coordinate 만을 출력하고, sum of squared error 를 loss 로 사용하였기 때문에 학습 시 noisy 한 데이터에 대응하지 못한다. 하지만, 본 장의 bounding box 에 대한 Gaussian 모델링은 bounding box 좌표에 대한 uncertainty 를 출력할 수 있고, bounding box 에 대해 새롭게 재설계한 NLL loss 는 training 시 inconsistent 한 데이터에 대해 observation noise parameter 인 sigma (즉, uncertainty)로 loss 에 penalty [7]를 줄 수 있다. 즉, training 시 noisy 한 데이터의 경우 NLL loss function 에서 높은 uncertainty 로 인해 loss 값이 줄고, 반대의 경우 loss 값이 커지기 때문에 학습 데이터 중 consistent 한 데이터에 집중해서 학습할

수 있다. 이러한 loss attenuation 으로 기존 대비 정확도 향상이 가능하다.

3.3.3 Localization uncertainty의 활용

제안 알고리즘인 Gaussian YOLOv3는 이미지 전체에 대해 하나의 uncertainty를 출력하는 것이 아니라, 이미지 내 검출 대상마다 bounding box의 uncertainty를 모두 예측한다. 즉, 각각의 검출 결과에 대해 uncertainty를 적용하는 것이 가능하다. 기존 YOLOv3는 물체 검출 시 objectness score와 class scores만을 고려하며, bounding box 좌표에 대한 score 정보는 알 수 없기 때문에 검출 시 이를 고려하지 못한다. 그러나 Gaussian YOLOv3는 bounding box score인 localization uncertainty를 출력할 수 있기 때문에 검출 시 objectness score, class scores와 함께 이를 고려할 수 있다. 제안 알고리즘은 물체 검출 시 uncertainty가 높은 bounding box의 경우 detection 결과에서 필터링 되도록 기존 YOLOv3의 검출 기준에 bounding box의 localization uncertainty를 적용한다. 이를 통해 objectness score, class score 그리고 bounding box의 신뢰도가 모두 높은 검출 대상이 최종 검출되므로, baseline 대비 mislocalization에 의한 false positive를 대폭 줄여 검출 정확도를 향상시킨다. Localization uncertainty를 고려한 제안 검출 기준식은 다음과 같다:

$$Cr. = \sigma(Object) \times \sigma(Class_i) \times (1\text{-}Uncertainty_{aver}). \qquad (3.10)$$

식 (3.10)의 Cr.는 Gaussian YOLOv3의 검출 기준이며, $\sigma(Object)$는 objectness score, $\sigma(Class_i)$는 i번째 class의 score 값을 의미한다. σ은 sigmoid function을 의미한다. $Uncertainty_{aver}$는 localization uncertainty로 검출 bounding box의 t_x, t_y, t_w, t_h uncertainty의 평균을 의미한다. Localization uncertainty는 sigmoid로 전처리 했기 때문에 object score, class score와 같이 0과 1사이의 값을 가지며, 이 값이 높을수록 검출한 bounding box의 신뢰도가 낮다는 것을 의미한다.

3.4 본 연구의 실험 결과

제안 알고리즘의 우수성을 보이기 위해 자율주행 연구에서 보편적으로 많이 쓰이는 KITTI [66]와 BDD [67] 데이터로 실험을 진행한다. KITTI 데이터의 class는 car, pedestrian 그리고 cyclist로 총 3개이며, 7,481개 이미지의 training set과 7,518개 이미지의 test set으로 구성된다. Test set에 대한 ground-truth가 제공되지 않기 때문에 실험에서 training set을 랜덤하게 절반으로 나눠 [54] training set과 validation set으로 사용한다. BDD 데이터의 class는 bike, bus, car, motor, person, rider, traffic light, traffic sign, train 그리고 truck으로 총 10개이며, training set, validation set 그리고 test set이 7:1:2의 비율로 구성되어 있다. 실험에서는 validation과 test set에 대하여 성능을 평가한다. 정량적 성능 평가 시 각 데이터의 공식 평가 IoU threshold를 사용한다. KITTI 데이터의 경우 car는 0.7, pedestrian과 cyclist는 0.5의 IoU threshold [66]를 사용하며, BDD는

모든 class에 대해 0.75의 IoU threshold [67]를 사용한다. YOLOv3와 Gaussian YOLOv3 학습 시 batch size는 64로, learning rate는 0.0001로 한다. Anchor 크기는 표 3.1과 같이 KITTI와 BDD training set 각각에 대해 k-means clustering으로 추출한 값을 사용한다. 실험에서 GPU는 NVIDIA TITAN Xp를 사용하였으며, CUDA 8.0와 CuDNN v7 환경에서 실험하였다.

표 3.1. KITTI와 BDD training set에서 추출된 anchor box 결과.

	Anchor 0	Anchor 1	Anchor 2
KITTI training set			
First detection layer	(49,240)	(82,170)	(118,206)
Second detection layer	(45,76)	(27,172)	(67,116)
Third detection layer	(13,30)	(23,53)	(17,102)
BDD training set			
First detection layer	(73,175)	(141,178)	(144,291)
Second detection layer	(32,97)	(57,64)	(92,109)
Third detection layer	(7,10)	(14,24)	(27,43)

3.4.1 Localization uncertainty 평가

그림 3.4은 KITTI와 BDD validation set에서 제안 알고리즘인 Gaussian YOLOv3가 예측한 검출 bounding box의 localization uncertainty와 IoU의 관계를 나타낸다. 각 데이터에서 가장 dominant한 class인 car에 대한 결과이며, 대표적인 경향을 보이기 위해 IoU를 0.1

범위로 나눠 범위 별로 IoU의 평균값과 uncertainty의 평균값을 계산해 대표값으로 사용하였다. 그림 3.4를 통해 두 dataset에서 모두 localization uncertainty가 작아질수록 IoU 값이 커지며, localization uncertainty가 커질수록 IoU 값이 작아지는 경향을 확인할 수 있다. 이 결과는 IoU가 클수록 예측 bounding box의 coordinate이 ground-truth의 coordinate에 가까워지는 것을 의미하므로, 이러한 경향성을 토대로 제안 알고리즘의 localization uncertainty는 예측 bounding box의 신뢰도를 효과적으로 나타낼 수 있음을 알 수 있다. 이를 통해, 검출 과정에서 제안 방식이 예측한 localization uncertainty를 활용하여 mislocalization에 대응함으로써 false positive 감소와 정확도 향상이 가능하다.

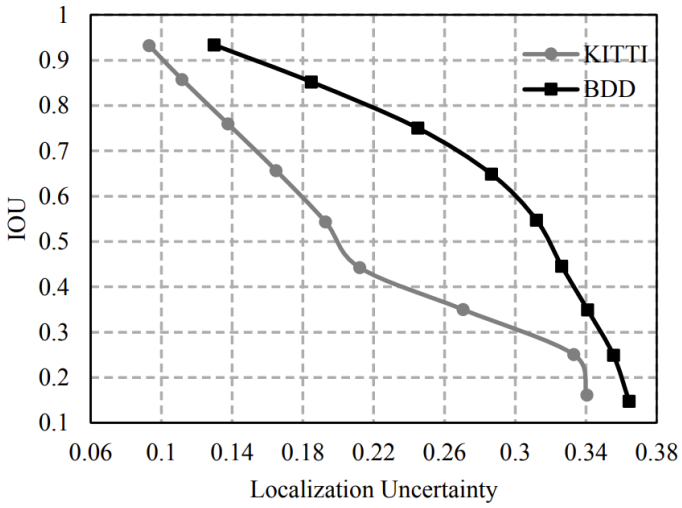


그림 3.4. KITTI와 BDD validation set에서 IoU와 예측한 localization uncertainty의 관계.

3.4.2 Gaussian YOLOv3의 성능 평가

제안 알고리즘의 우수성을 보이기 위해 기존 연구들 [6], [10], [21], [58], [59]. [60], [61]과 정확도 및 처리 속도를 비교한다. KITTI 데이터에 대한 실험에서는 random하게 split한 training set과 validataion set을 기존 연구들 [6], [10], [21], [58], [60], [61]의 공식 publish된 코드를 사용해 학습 및 평가하였다. CFENet [59]은 공식 코드가 공개되지 않아 KITTI object detection leader board의 결과를 사용하였다. BDD 데이터의 경우 CFENet [59]에 SSD [10], CFENet [59] 그리고 RefineDet [60]의 BDD test set에 대한 결과가 명시되어 있기 때문에 이를 본 연구에서 사용하였고, 그 외의 기존 연구들 [6], [21], [58], [61]은 공식 publish 코드를 사용하여 학습 및 평가하였다. 공정한 성능 비교를 위해 제안 방식의 input resolution을 CFENet [59]과 같이 맞췄으며, two-stage detector는 공식 publish 코드의 default size를 사용한다. 정량적 성능 비교 시 각 데이터의 공식 평가 코드를 사용하며, 각 데이터의 IoU threshold는 앞에 명시된 값들을 사용한다. 정확도 비교를 위해 이전 object detection 연구들에서 일반적으로 사용하는 mAP를 평가 metric으로 사용한다.

표 3.2는 KITTI validation set에서 제안 알고리즘과 기존 연구들의 성능을 나타낸다. 제안 알고리즘인 Gaussian YOLOv3는 baseline인 YOLOv3 대비 mAP를 3.09 percent points (pp) 향상시킨다. 처리 속도는 43.13 fps로 YOLOv3와 큰 차이 없이 여전히 실시간 처리가 가능하다. Gaussian YOLOv3는 YOLOv3를 제외한 이전 연구들 중에서 가장 빠른 RFBNet [61]보다 3.93 fps 빠르다. 그러면서도 정확도는 Gaussian YOLOv3가 RFBNet보다 10.17 pp 높다. 정확도와 처리

속도는 trade-off 관계에 있으므로 공정한 비교를 위해 제안 알고리즘의 input resolution을 기존 연구들 중 가장 높은 정확도를 보이는 SINet [21]의 fps와 비슷하게 맞춘 후 평가한다. 즉, input resolution을 704×704로 바꾸었을 때, Gaussian YOLOv3의 정확도는 표 3.2의 마지막 row에서 볼 수 있듯이 86.79 mAP로 SINet 보다 우수하며, 처리 속도 또한 24.91 fps로 SINet보다 빠르다. 즉, Gaussian YOLOv3는 이전 연구들보다 정확도와 처리 속도의 trade-off 측면에서 가장 우수하다.

표 3.2. KITTI validation set에서 기존 연구와의 성능 비교. E, M, 그리고 H는 각각 easy, moderate, 그리고 hard를 의미한다.

Detection algorithm	Average precision (%)									mAP (%)	FPS	Input size
	Car			Pedestrian			Cyclist					
	E	M	H	E	M	H	E	M	H			
MS-CNN	92.54	90.49	79.23	87.46	81.34	72.49	90.13	87.59	81.11	84.71	8.13	1920×576
SINet	99.11	90.59	79.77	88.09	79.22	70.30	94.41	86.61	80.68	85.42	23.98	1920×576
SSD	88.37	87.84	79.15	50.33	48.87	44.97	48.00	52.51	51.52	61.29	28.93	512×512
RefineDet	98.96	90.44	88.82	84.40	77.44	73.52	86.33	80.22	79.15	84.36	27.81	512×512
CFENet	90.33	90.22	84.85	-	-	-	-	-	-	-	0.25	-
RFBNet	87.41	88.35	83.41	65.85	61.30	57.71	74.46	72.73	69.75	73.44	39.20	512×512
YOLOv3	85.68	76.89	75.89	83.51	78.37	75.16	88.94	80.64	79.62	80.52	43.57	512×512
Gaussian YOLOv3	90.61	90.20	81.19	87.84	79.57	72.30	89.31	81.30	80.20	83.61	43.13	512×512
Gaussian YOLOv3	98.74	90.48	89.47	87.85	79.96	76.81	90.08	86.59	81.09	86.79	24.91	704×704

표 3.3은 BDD test set에서 제안 방법과 이전 연구들의 정확도와 처리 속도를 나타낸다. 제안 알고리즘인 Gaussian YOLOv3는 baseline인 YOLOv3 대비 mAP를 3.5 pp 향상시킨다. 처리 속도는 42.5 fps로 YOLOv3와 거의 비슷하게 실시간 처리를 지원한다. Gaussian YOLOv3는 YOLOv3를 제외한 이전 연구들 중에서 가장 빠른 RFBNet [61]보다 3.5 fps 빠르다. 그러면서도 정확도는 Gaussian YOLOv3가 RFBNet보다 3.9 pp 높다. 게다가, 이전 연구들 중 가장 높은 정확도를 보이는 CFENet [59]과 비교했을 때, 제안 알고리즘은 표 3.3의 마지막 row에서 볼 수 있듯이 736×736 input resolution에서 mAP가 1.7 pp 높으며 처리 속도는 1.5 fps 빠르다. 즉, Gaussian YOLOv3는 CFENet 포함 이전 연구들보다 정확도와 처리 속도의 trade-off 측면에서 가장 우수하다.

표 3.3. BDD test set에서 성능 비교.

Detection algorithm	mAP (%)	FPS	Input size
MS-CNN	5.7	6.0	1920×576
SINet	9.0	18.2	1920×576
SSD	14.1	23.1	512×512
RefineDet	17.4	22.3	512×512
CFENet	19.1	21.0	512×512
RFBNet	14.5	39.0	512×512
YOLOv3	14.9	42.9	512×512
Gaussian YOLOv3	18.4	42.5	512×512
Gaussian YOLOv3	20.8	22.5	736×736

Gaussian YOLOv3는 자율주행 데이터뿐만 아니라 general한 object detection 데이터에서도 높은 정확도 향상 효과를 보인다. 표 3.4는 MS-COCO [71] 데이터에서 제안 알고리즘과 baseline 알고리즘의 정확도와 처리 속도 결과를 나타낸다. MS-COCO 데이터는 총 80개의 class로 구성되며, 성능 평가 시 공식 evaluation code를 사용한다. Input resolution과 학습 및 평가에 사용한 데이터 버전은 YOLOv3 [6]의 MS-COCO 실험과 동일하다. 표를 통해 알 수 있듯이, 제안 알고리즘의 AP (average precision)는 36.1%로 baseline인 YOLOv3보다 3.1 pp 높으며, 처리 속도는 baseline과 큰 차이가 없다. 즉, 이 결과를 통해 제안 알고리즘은 자율주행 데이터뿐만 아니라 일반적인 object detection 데이터에서도 baseline보다 높은 정확도를 보인다.

표 3.4. MS-COCO 데이터에서 baseline과 제안 알고리즘의 성능 비교.

Detection algorithm	AP (%)	FPS	Input size
YOLOv3	33.0	24.91	608×608
Gaussian YOLOv3	36.1	24.82	608×608

표 3.5와 표 3.6은 KITTI와 BDD 데이터에서 최신 object detector인 YOLOv4 [1]를 baseline으로 한 정확도와 처리 속도 결과를 나타낸다. 추가로, 최신 후처리 방법인 DIOU-NMS [72]를 baseline에 적용한 알고리즘과도 성능을 비교한다. 실험을 위해 YOLOv4의 공식 코드를 사용했으며, 실험 hyper-parameter는 3.4 장 도입부에 언급한 실험 세팅을 따른다. 표 3.5의 KITTI 데이터에 대한 결과를 통해, 제안 방법이 적용된 Gaussian YOLOv4는 baseline 대비 정확도를 1.32 pp 향상시키며, baseline과 비슷한 처리 속도를 보이는 것을 알 수 있다. Baseline에 DIOU-NMS를 적용한 알고리즘보다도 정확도와 처리 속도가 우수하다. 표 3.6의 BDD 데이터에 대한 결과에서도 제안 방법을 적용한 알고리즘은 baseline인 YOLOv4 대비 정확도가 1.49 pp 향상되며, baseline과 비슷한 처리 속도를 보인다. DIOU-NMS를 적용한 baseline과 비교하더라도 제안 방법을 적용한 알고리즘은 여전히 정확도와 처리 속도가 우수하다. 이러한 결과들을 기반으로, 제안 방법을 적용한 최신 object detection 알고리즘은 baseline의 처리 속도를 비슷하게 유지하면서도 정확도를 대폭 향상시킨다.

표 3.5. KITTI validation set에서 최신 detector인 YOLOv4와 제안 모델의 성능 비교.

Detection algorithm	mAP (%)	FPS	Input size
YOLOv4	84.62	53.75	512×512
YOLOv4+DIOU-NMS	84.99	53.53	512×512
Gaussian YOLOv4	85.94	53.67	512×512

표 3.6. BDD validation set에서 최신 detector인 YOLOv4와 제안 모델의 성능 비교.

Detection algorithm	mAP (%)	FPS	Input size
YOLOv4	21.50	51.08	512×512
YOLOv4+DIOU-NMS	21.65	50.84	512×512
Gaussian YOLOv4	22.99	50.89	512×512

3.4.3 False positive과 true positive의 시각적, 수치적 평가

그림 3.5와 그림 3.6은 각각 KITTI validation set에서 baseline과 Gaussian YOLOv3의 visual example 결과를 보여준다. Detection threshold로는 YOLOv3의 default threshold인 0.5를 사용한다. 그림 3.5의 첫번째 행과 그림 3.6의 첫번째 행은 제안 알고리즘이 YOLOv3가 찾지 못한 물체를 찾을 수 있어 true positive를 향상시킴을 보여준다. 이는 제안한 Gaussian 모델링과 loss function 재설계를 통해 학습 과정에서 loss attenuation 효과로 bounding box의 학습 정확도가 향상되기 때문이다. 그림 3.5의 두번째 행과 그림 3.6의 두번째 행을 통해 제안 알고리즘이 baseline의 잘못된 검출 결과를 보완할 수 있음을 알 수 있다. 게다가, 그림 3.5의 세번째 행과 그림 3.6의 세번째 행을 통해 제안 알고리즘은 baseline의 부정확한 bounding box 검출 결과 대비 정확한 bounding box 예측을 할 수 있음을 보여준다.

그림 3.7와 그림 3.8은 각각 BDD test set에서 baseline과 Gaussian YOLOv3의 visual example 결과를 보여준다. KITTI와 마찬가지로, 그림 3.7의 첫번째 행과 그림 3.8의 첫번째 행은 제안 알고리즘이 YOLOv3가 찾지 못한 물체를 찾을 수 있어 true positive를

향상시킴을 보여준다. 그림 3.7의 두번째 행과 그림 3.8의 두번째 행을 통해 제안 알고리즘이 baseline의 잘못된 검출 결과를 보완할 수 있음을 알 수 있다. 게다가, 그림 3.7의 세번째 행과 그림 3.8의 세번째 행을 통해 제안 알고리즘은 baseline의 부정확한 bounding box 검출 결과 대비 정확한 bounding box 검출을 할 수 있음을 보여준다.

그림 3.5. KITTI validation set에서 baseline인 YOLOv3의 검출 결과의 예.

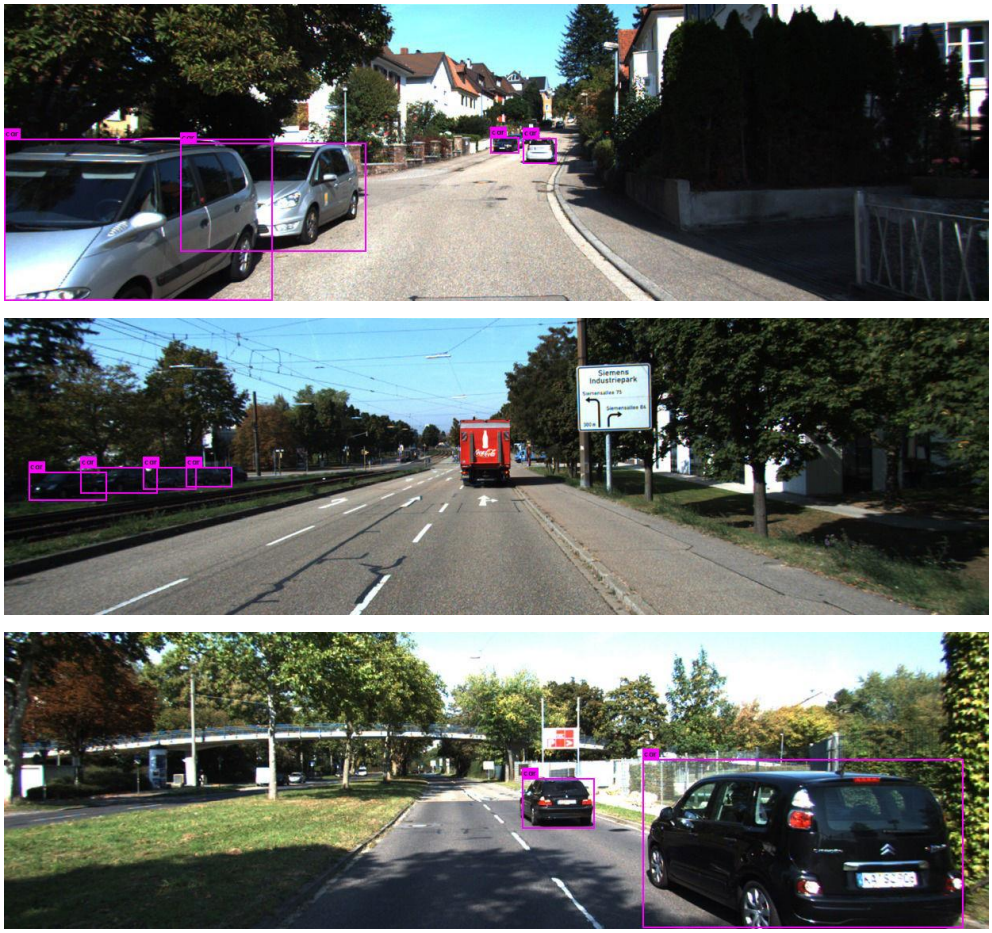


그림 3.6. KITTI validation set에서 제안 알고리즘인 Gaussian YOLOv3의 검출 결과의 예.

그림 3.7. BDD test set에서 baseline인 YOLOv3의 검출 결과의 예.

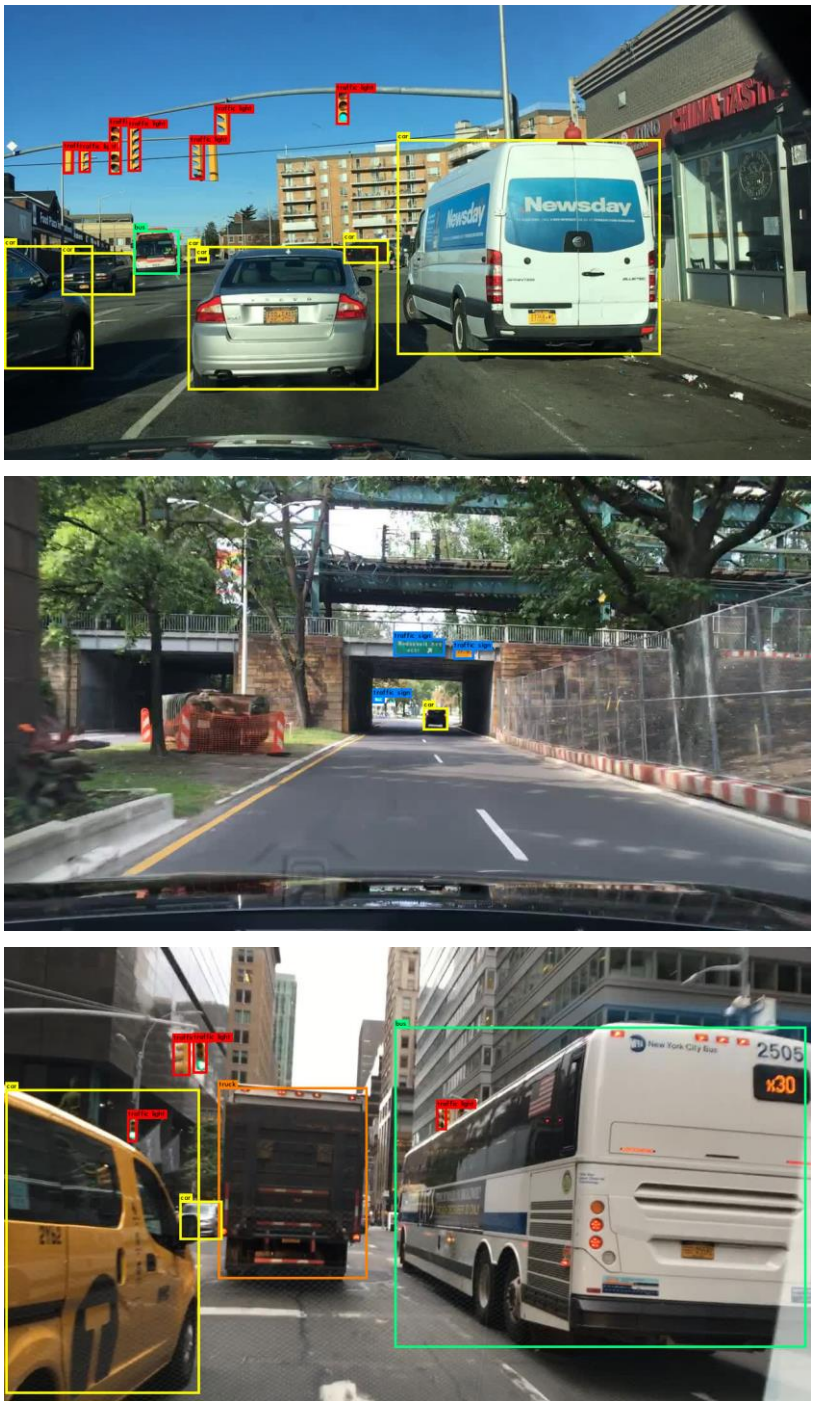


그림 3.8. BDD test set에서 제안 알고리즘의 검출 결과의 예.

False positive와 true positive의 수치적 평가를 위해 표 3.7은 baseline과 Gaussian YOLOv3의 false positive와 true positive 수를 나타낸다. Detection threshold는 이전에 언급한 YOLOv3의 default threshold 값을 사용한다. False potive와 true positive를 계산하기 위해 KITTI와 BDD 모두 ground-truth가 제공되는 validation set을 사용한다. 정확한 측정을 위해 BDD 공식 평가 코드를 사용하여 false positive와 true positive를 계산한다. 왜냐하면 KITTI 공식 평가 코드의 경우 특정 크기 이하인 bounding box를 false positive로 간주하지 않기 때문이다. KITTI와 BDD validation set에서 Gaussian YOLOv3는 YOLOv3 대비 false positive를 각각 41.40%, 40.62% 감소시킨다. 게다가 true positive는 KITTI와 BDD validation set 각각에 대해 7.26%, 4.3% 증가시킨다. 자율주행에서 false positive 감소는 예상치 못한 급제동을 방지하여 mislocalization로 인한 사고를 줄이며, true positive 증가는 object detection 에러로 인한 치명적인 사고를 방지한다. 이러한 결과를 바탕으로 제안 알고리즘인 Gaussian YOLOv3는 baseline 대비 false positive를 대폭 감소시키고, true positive를 증가시켜, 결과적으로 자율주행 자동차의 주행 안정성과 효율성을 향상시키고 치명적인 사고를 방지할 수 있다.

표 3.7. False positive (FP)와 true positive (TP)를 통한 수치적 평가. (GT는 ground-truth를 의미한다.)

	YOLOv3	Gaussian YOLOv3	Variation rate (%)
KITTI validation set			
# of FP	1,681	985	-41.40
# of TP	13,575	14,560	+7.26
# of GT	17,607	17,607	0
BDD validation set			
# of FP	86,380	51,296	-40.62
# of TP	57,261	59,724	+4.30
# of GT	185,578	185,578	0

3.5 본 연구의 결론

자율주행 자동차의 안전성과 실시간 제어를 위해 검출 알고리즘의 정확도와 처리 속도는 매우 중요하다. 그리하여 다양한 검출 알고리즘 연구들이 수행되어 왔지만, 정확도와 처리 속도의 trade-off 측면에서 완전하지 못하였다. 이러한 이유로, 본 장에서는 정확도와 처리 속도의 trade-off 측면에서 가장 우수한, 자율주행을 위한 object detection 알고리즘을 제안하였다. 제안 알고리즘은 기존 YOLOv3의 bounding box 출력을 Gaussian parameter로 모델링하고 loss function을 재설계하여, 학습 시 noisy한 데이터의 영향을 최소화하여 정확도를 향상시킨다. 그리고 localization uncertainty를 활용하여 false positive를 크게 낮추고, true positive를 향상시켜 정확도를 대폭

증가시키면서도 실시간 처리 속도를 유지한다. Baseline 대비 제안 알고리즘인 Gaussian YOLOv3은 KITTI와 BDD 데이터에서 mAP를 각각 3.09 및 3.5 pp 향상시킨다. 또한 제안 알고리즘은 유사한 처리 속도를 가진 이전 연구보다 정확도가 높기 때문에 정확도와 처리 속도의 trade-off 측면에서 가장 우수하다. 이 외에도, 제안 방법은 다양한 object detector에 적용 가능하여 확장성이 우수하다. 그 결과 제안 방법은 자율주행을 위한 카메라 기반 객체 탐지 시스템을 획기적으로 개선할 수 있으며, 이에 따라 자율주행 application에서 크게 기여할 것으로 기대된다.

제 4 장 혼합 밀도 네트워크를 사용한 심층 객체 탐지 신경망을 위한 능동적 학습

4.1 본 연구의 서론

심층 객체 탐지 네트워크의 성능은 labeled 데이터의 양에 의존적 [5], [10], [73]이다. 즉, 학습에 사용할 labeled 데이터가 많을수록 객체 탐지 네트워크의 정확도는 향상된다. 학습 데이터를 취득하는 것은 쉬우나, 취득한 모든 데이터에 annotation을 하는 것은 많은 시간과 비용을 요구한다. 이러한 문제를 해결하기 위해, 여러 데이터들 중 가장 informative한 데이터를 선택하고 labeling하여, annotation cost를 줄이면서 정확도를 향상시키는 능동적 학습 (active learning) [12]이 활발히 연구되어 왔다. 일반적으로, 능동적 학습은 네트워크의 predictive uncertainty 기반 scoring function을 통해 데이터의 informativeness를 판단하여 수행 [15], [45], [74] 된다.

네트워크의 predictive uncertainty는 aleatoric uncertainty와 epistemic uncertainty로 분류 [75], [76] 할 수 있다. Aleatoric uncertainty는 데이터 노이즈와 같이 inherent 노이즈와 관련된 uncertainty로 occlusion 또는 visual feature의 부족 [7], [64] 등에 따라 변하는 uncertainty이다. Epistemic uncertainty는 모델의 지식 결여로 발생되며, 학습 데이터의 양에 반비례하는 특징 [77]이 있다. 능동적 학습에서 이 두 종류의 uncertainty를 모델링하고 구별하는 것은 매우 중요하다. 왜냐하면, aleatoric uncertainty는 의심스러운 예측을 판단하고, epistemic uncertainty는 학습 데이터와 닮지 않은

데이터를 판단하므로, 두 uncertainty를 통해 현재 딥러닝 모델의 한계 [77], [78]를 알 수 있기 때문이다. 일반적으로 이러한 두 종류의 uncertainty를 예측하기 위해, ensemble [45] 또는 Monte Carlo (MC) dropout [46]과 같은 multiple models 기반 방식들을 사용해 왔다. 이러한 방법들은 좋은 정확도를 보였지만, 몇 가지 문제점 [14], [79]이 있다. Multiple models 기반 방식은 매우 큰 computing cost를 요구하기 때문에 능동적 학습 시 많은 시간과 resource를 필요로 하며, 특히 ensemble의 경우 네트워크의 parameter 수가 대폭 증가 [45] 하는 문제가 있다. 게다가, 기존 객체 탐지 네트워크를 위한 능동적 학습 방식은 localization uncertainty는 고려하지 않고, 오직 classification uncertainty에만 의존하는 한계가 있다. 그러므로 능동적 학습을 위한 완전한 해결책이 되지 못한다.

이 장에서는 심층 객체 탐지를 위한 novel한 능동적 학습 방법을 제안한다. 제안 방법은 single model 기반 single forward pass를 사용하여 multiple models 기반 방법 대비 연산량을 대폭 감소시킨다. 그럼에도 불구하고, 제안 방법은 높은 정확도를 유지한다. 이를 위해서, 제안 방법은 localization과 classfication task의 aleatoric과 epistemic uncertainty를 능동적 학습에서 모두 활용한다. 그림 4.1에서 볼 수 있듯이, 제안 방법은 localization과 classification task에서 두 가지 uncertainty를 모두 예측하기 위해 네트워크의 각 출력 head를 Gaussian mixture model을 학습하는 mixture density network [20]로 모델링 한다. 제안 네트워크를 학습하기 위해, inconsistent 데이터에 대해 regularizer 역할을 하는 새로운 loss function을 제안하며, 이를 통해 더욱 robust한 모델을 만든다. 제안 scoring function은 이미지 내 모든 물체의 localization과 classification 기반 uncertainty들을 모두 활용하여 informativeness score를 계산하다. 능동적 학습에서 모든

task의 uncertainty들을 활용하는 것이 정확도 향상에 중요한 요소라는 것을 실험에서 보이며, 제안 방법의 우수성을 일반적인 object detection 데이터 셋인 PASCAL VOC [80]와 MS-COCO [71]에서 보인다. 제안 능동적 학습 방법은 single-model 기반의 기존 연구들보다 정확도가 우수하며, 기존의 multiple models 기반 방식들과는 유사한 정확도를 보이면서 계산 비용을 대폭 감소시킨다.

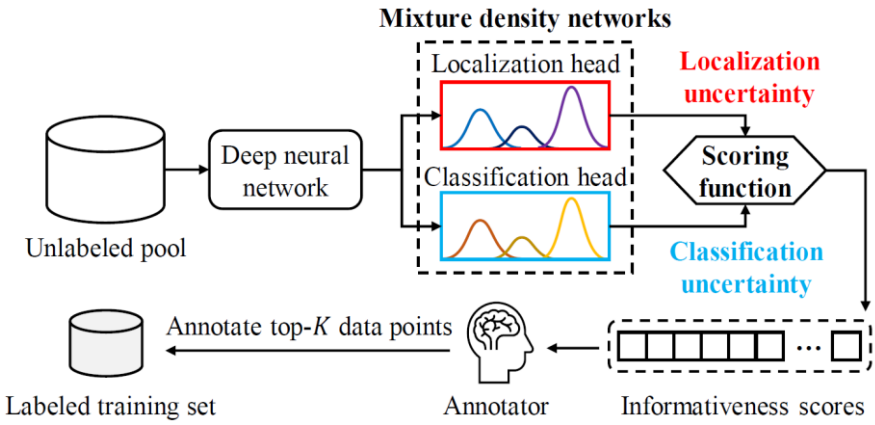


그림 4.1. 제안 방법 overview.

4.2 본 연구의 관련 연구

4.2.1 객체 탐지를 위한 심층 능동적 학습

객체 탐지를 위한 능동적 학습은 최근에 연구가 활발히 되어 왔다. Haussmann *et al.* [14]는 ensemble [45]을 사용해 entropy [41] 혹은 mutual information [42]을 계산하여 데이터를 선택하는 방법을 제안하였다. Feng *et al.* [79]은 MC-dropout [47], [81]을 사용해 uncertainty를 예측하여 능동적 학습을 위한 데이터 선택 방법을 제안하였다. Aghdam *et al.* [48]은 pixel score를 계산해 informative한 데이터를 선택하는 방법을 제안하였으며, Roy *et al.* [13]는 informative 데이터 선택을 위해 query by committee 기반 방법을 제안하였다. Sener *et al.* [16]은 학습 데이터 중에서 대표 데이터를 선택하기 위해 feature space를 사용하는 방법을 제안하였으며, 해당 방법이 object detection에서도 우수한 성능을 보이는 것이 실험적으로 증명 [15] 되었다. Kao *et al.* [17]는 두 가지 score를 정의하여 능동적 학습에 사용하였다. 첫번째는 region proposal과 최종 예측 사이의 overlapping ratio를 기반으로 계산되는 localization tightness score이다. 두번째는 입력 noise에 따른 예측 localization의 변화량을 나타내는 localization stability score이다. 여러 데이터들 중 해당 점수가 가장 높은 데이터를 선택하여 labeling 한다. Yoo *et al.* [15]는 heuristic하지만 새로운 접근 방법을 제안하였다. 해당 방법은 single model 기반의 능동적 학습 방법들 중 가장 우수한 성능을 보인다. 이 방법은 입력 데이터의 target loss를 예측하기 위한 prediction module을 제안한다. 능동적 학습 과정에서 가장 높은 예측 loss를 보이는 데이터를 선택하여

labeling한다.

앞서 언급한 대부분의 방법들 [14], [17], [79]은 이미지의 informativeness score 계산을 위해 multiple models 혹은 multiple forward passes를 사용하기 때문에 매우 높은 계산 비용이 발생한다. 게다가 이전의 모든 연구들은 객체 탐지를 위한 능동적 학습일지라도 localization uncertainty를 예측하기 위해 heuristic한 방법 [15], [17]을 사용하거나 아예 localization uncertainty를 예측하지 못한다는 문제가 있다 [13], [14], [16], [48], [79]. 즉, 이전 연구들은 흥미로운 접근법을 제안했지만, 정확도와 계산 비용의 trade-off 측면에서 완전하지 못하다. 이와 반대로, 제안 방법은 높은 정확도를 얻기 위해 localization과 classfication 기반 uncertainty를 능동적 학습에서 모두 사용한다. 또한, single model 기반 single forward pass로 이 uncertainty들을 예측하여 능동적 학습을 위한 계산 비용을 대폭 감소시킨다.

4.2.2 혼합 밀도 네트워크

Mixture density network는 최근 다양한 딥러닝 task에 적용되어 왔다. Choi *et al.* [8]는 steering angle의 regression task에 mixture density network를 사용했으며, He *et al.* [82]와 Varamesh *et al.* [83]는 multi-model regression task에 mixture density network를 적용하였다. Yoo *et al.* [84]는 density estimation에 이를 적용했으며, Choi *et al.* [85]는 corrupted 데이터와 supervised learning에 이를 사용하였다. 하지만, 이전 연구들 [8], [82], [83]은 object detection의

필수 task인 classifiation task를 고려하지 못했다. 게다가, 앞서 언급한 모든 연구들은 bounding box regression과 classification 기반의 aleatoric uncertainty와 epistemic uncertainty를 고려하지 못했으며, object detection을 위한 능동적 학습 문제를 해결하지 못했다. 본 장의 제안 방법은 object detection의 localization과 classification task에서 aleatoric과 epistemic uncertainty를 모두 예측하고, 이들을 능동적 학습에서 활용한다.

4.3 객체 탐지를 위한 능동적 학습

제안 방법의 핵심 아이디어는 신경망 네트워크의 출력을 새롭게 정의하는 것이다. 그림 4.2와 같이 기존에 단일 값을 출력하는 네트워크 head 대신에 확률 distribution을 출력하도록 head를 모델링한다. 이를 위해, Gaussian mixture model의 구성요소가 출력되도록 하는 mixture density network를 사용하여 새로운 object detection 모델을 제안한다. Gaussian mixture model의 k번째 요소는 mean (μ^k), variance (Σ^k), 그리고 mixture weight (π^k)로 구성된다. 주어진 parameter로 aleatoric uncertianty (u_{al})와 epistemic uncertainty (u_{ep})를 다음과 같이 계산할 수 있다 [8]:

$$u_{al} = \sum_{k=1}^{K} \pi^k \, \Sigma^k, \tag{4.1}$$

$$u_{ep} = \sum_{k=1}^{K} \pi^k \left\| \mu^k - \sum_{i=1}^{K} \pi^i \mu^i \right\|^2, \tag{4.2}$$

식에서 K는 Gaussian mixture model의 component 개수를 의미한다.

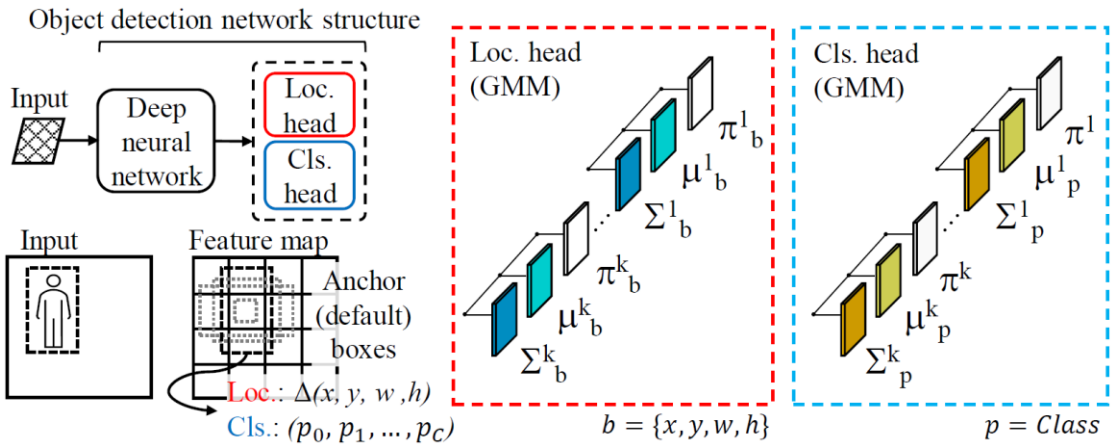


그림 4.2. 제안 모델링을 적용한 object detection network 구조.

4.3.1 객체 탐지를 위한 mixture 모델링

본 장에서는 먼저 localization task에 대한 제안 모델링 방법을 소개하고, 그 후 classfication task에 대해 소개한다. 향후 실험에서 볼 수 있듯이, 제안 방법은 single-stage 뿐만 아니라 two-stage object detector에도 적용 가능하다.

본 설명에서 bounding box (b)는 center (x, y)와 width (w), 그리고 height (h) coordinate으로 정의한다. 제안 mixture model은 deterministic 값을 예측하는 것 대신에 bounding box의 각 coordinate을 위한 3 개의 parameter를 예측한다:

- Mean ($\hat{\mu}_x$, $\hat{\mu}_y$, $\hat{\mu}_w$, $\hat{\mu}_h$)
- Variance ($\hat{\Sigma}_x$, $\hat{\Sigma}_y$, $\hat{\Sigma}_w$, $\hat{\Sigma}_h$)
- Mixture weight ($\hat{\pi}_x$, $\hat{\pi}_y$, $\hat{\pi}_w$, $\hat{\pi}_h$).

$\left\{\hat{\pi}_b^k, \hat{\mu}_b^k, \hat{\Sigma}_b^k\right\}_{k=1}^{K}$, $b \in \{x, y, w, h\}$ 을 제안 네트워크의 bounding box 출력이라고 하면, 각 coordinate에 대한 K 개의 Gaussian mixture model의 parameter를 다음과 같이 처리한다:

$$\pi_b^k = \frac{e^{\hat{\pi}_b^k}}{\sum_{j=1}^{K} e^{\hat{\pi}_b^j}}, \tag{4.3}$$

$$\mu_b^k = \hat{\mu}_b^k, \tag{4.4}$$

$$\Sigma_b^k = \sigma(\hat{\Sigma}_b^k), \qquad (4.5)$$

식에서 π는 각 component의 mixture weight, μ는 각 bounding box coordinate의 예측 값, 그리고 Σ는 각 coordinate의 variance로 aleatoric uncertainty를 의미한다. π를 probability space에 유지하기 위해 softmax function으로 처리하고 variance가 0 이상 1이하의 값을 가지도록 sigmoid function으로 처리 [8] 한다.

일반적으로 bounding box regression loss로는 smooth L1 loss [25]가 주로 사용되었다. 이 loss function은 예측 bounding box와 ground-truth box의 좌표 값만을 고려한다. 그렇기 때문에 loss에서 bounding box의 모호성 (즉, aleatoric uncertainty)을 고려할 수 없다는 문제가 있다. Localization task에서 제안 mixture density network를 학습하기 위해, negative log-likelihood loss 기반 localization loss를 제안한다. 제안 loss는 아래 식과 같이 Gaussian mixture model의 parameter가 positive match된 anchor (default) box (즉, d_x, d_y, d_w, d_h)를 기준으로 center (x, y), width (w), 그리고 height (h)의 offset coordinate을 regression 하도록 학습한다:

$$L_{loc}(\lambda, l, g) = -\sum_{i \in Pos}^{N} \sum_b \lambda_G^{ij} log(\sum_{k=1}^{K} \pi_b^{ik} N(\hat{g}_b^j | \mu_b^{ik}, \Sigma_b^{ik}) + \varepsilon), \qquad (4.6)$$

$$\lambda_G^{ij} = \begin{cases} 1, & if\ IoU > 0.5. \\ 0, & otherwise. \end{cases}, \qquad (4.7)$$

$$\hat{g}_x^j = \frac{(g_x^j - d_x^i)}{d_w^i}, \qquad (4.8)$$

$$\hat{g}_y^j = \frac{(g_y^j - d_y^i)}{d_h^i}, \tag{4.9}$$

$$\hat{g}_w^j = log(\frac{g_w^j}{d_w^i}), \tag{4.10}$$

$$\hat{g}_h^j = log(\frac{g_h^j}{d_h^i}), \tag{4.11}$$

식에서 l은 예측 bounding box의 Gaussian parameters, N은 positive match 개수 (즉, ground-truth box와 match된 anchor box 개수), K는 mixture distribution의 수, λ_G^{ij}는 i번째 anchor box와 카테고리 G의 j번째 ground-truth box의 matching 여부를 알려주는 indicator, 그리고 $\hat{g}_b^j$는 j번째 ground-truth box를 의미한다. 실험에서 logarithm 함수의 수치 안정성을 위해 ε은 10^{-9}로 설정하였다.

그림 4.2와 같이 classification task의 모델링 또한 localization과 비슷하게 모든 class 출력을 Gaussian mixture model의 parameter가 출력되도록 모델링한다. 제안 모델은 각 class의 평균 ($\hat{\mu}_p^k$)과 분산 ($\hat{\Sigma}_p^k$), 그리고 Gaussian mixture model의 각 component의 mixture weight ($\hat{\pi}^k$)을 예측한다. Classification task에서 Gaussian mixture model의 각 parameter는 식 (4.3), (4.4), (4.5)에 따라 처리된다. k번째 mixture의 class probability distribution은 다음 식과 같이 Gaussian noise, variance (Σ_p^k) 그리고 mean (μ_p^k)으로 계산된다 [7]:

$$\hat{c}_p^k = \mu_p^k + \sqrt{\Sigma_p^k}\gamma, \quad \gamma \sim N(0,1) \tag{4.12}$$

식에서 γ 는 auxiliary noise variable이며, 그 크기는 Σ_p^k 와 μ_p^k 과 동일하다.

Classification에서 mixture density network를 학습하기 위해, 본 장에서 anchor box와 ground-truth box의 IoU를 고려하고, hard negative mining을 적용한 classification loss function을 제안한다. 좀 더 자세히 말하면, 아래 식과 같이 positive sample과 negative sample 각각에 대한 loss인 L_{cl}^{Pos} 과 L_{cl}^{Neg} 을 따로 계산한 후 결합하여 최종 classification loss function을 설정한다:

$$L_{cl}^{Pos}(\lambda, c) = -\sum_{i \in Pos}^{N} \lambda_G^{ij} \sum_{k=1}^{K} \pi^{ik} (\hat{c}_G^j - log \sum_{p=0}^{C} e^{\hat{c}_p^{ik}}), \quad (4.13)$$

$$L_{cl}^{Neg}(c) = -\sum_{i \in Neg}^{M \times N} \sum_{k=1}^{K} \pi^{ik} (\hat{c}_0^i - log \sum_{p=0}^{C} e^{\hat{c}_p^{ik}}), \quad (4.14)$$

식에서 N은 positive match 수, K는 mixture distribution의 수, C는 class 수, M은 hard negative mining ratio를 의미한다. $\hat{c}_G^j$ 는 i번째 anchor (default) box와 matching된 j번째 ground-truth box의 ground-truth class (G category), $\hat{c}_0^i$ 는 matching 되지 않은 i번째 anchor box의 background class (0 category)를 의미한다. $\hat{c}_p^{ik}$ 는 식 4.12로 계산된 값이며, λ_G^{ij} 는 식 4.6에서 사용된 것과 동일하다. Negative로 matching된 모든 sample을 쓰는 것 대신에, 제안한 mixture classification loss를 사용하여 모든 negative sample들을 정렬한 후 가장 높은 loss를 갖는 상위 sample부터 $M \times N$까지만 선택하여 학습 시 사용한다. 실험에서 M 은 3 [10]으로 설정하였다.

Mixture density network를 사용한 제안 object detector의 최종 학습 loss는 다음과 같다:

$$L = \begin{cases} \frac{1}{N}\left(L_{loc}(\lambda, l, g)/\eta + L_{cl}^{Pos}(\lambda, c) + L_{cl}^{Neg}(c)\right), & if\ N > 0. \\ 0, & otherwise. \end{cases} \quad (4.15)$$

식에서 N는 positive match 수를 의미하며, 실험에서 η은 2 [2]로 설정한다.

Inference에서 bounding box coordinate (R_b)과 각 class의 confidence score (P_i)는 아래 식과 같이 mixture model의 coomponent들의 합과 곱을 통해 계산한다:

$$Localization: R_b = \sum_{k=1}^{K} \pi_b^k \mu_b^k, \quad (4.16)$$

$$Classification: P_i = \sum_{k=1}^{K} \pi^k \frac{e^{\mu_i^k}}{\sum_{j=0}^{C} e^{\mu_j^k}}. \quad (4.17)$$

4.3.2 능동적 학습을 위한 scoring function

능동적 학습에서 scoring function은 이미지의 informativeness score를 계산한다. 제안 scoring function은 이미지 내 모든 물체의 bounding box와 class에 대한 aleatoric uncertainty와 epistemic uncertainty를 고려한다. 즉, 예측한 모든 uncertainty를 사용하여 입력 이미지의 최종 informativeness score를 계산한다.

좀 더 자세한 설명을 위해, U = $\{u^{ij}\}$를 이미지 그룹에서 aleatoric uncertainty 또는 epistemic uncertainty의 집합으로 설정한다. 여기서

u^{ij} 는 i번째 이미지의 j번째 object에 대한 uncertainty를 의미한다. Localization에서 u^{ij} 는 4개의 bounding box coordinate에 대한 uncertainty 최댓값을 의미한다. Bounding box의 coordinate이 unbound 되어 있고, 이미지에 따라 uncertainty 값의 범위가 다를 수 있기 때문에 각각 uncertainty들을 아래 식과 같이 z-score normalization으로 처리한다.

$$\tilde{u}^{ij} = \frac{(u^{ij} - \mu_U)}{\sigma_U}, \tag{4.18}$$

그런 다음 아래 식과 같이 여러 물체들 중 각 uncertainty의 최댓값만을 수집한다.

$$u^{i} = max_j \tilde{u}^{ij}. \tag{4.19}$$

앞서 언급한 과정을 통해 아래 식과 같이 각 이미지 당 총 4개의 서로 다른 normailzed uncertainty를 얻는다: classification에 대한 epistemic uncertainty와 aleatoric uncertainty, localization에 대한 epistemic uncertainty와 aleatoric uncertainty.

$$\mathbf{u} = \{u^{i}_{ep_c}, u^{i}_{al_c}, u^{i}_{ep_b}, u^{i}_{al_b}\}. \tag{4.20}$$

마지막 과정은 각 이미지에 대한 총 4가지 normalized uncertainty들을 single score로 집계하는 것이다. 다른 능동적 학습 연구 [13], [14]와 같이 최댓값 또는 합을 사용하는 scoring function을 통해, 예측한 uncertainty들을 집계하는 다양한 조합을 실험에서 탐구한다.

4.4 본 연구의 실험 결과

본 장에서 제안 방법의 우수성을 실험을 통해 보인다. 먼저, 심층 객체 탐지 네트워크에 제안 mixture 모델링을 적용했을 때의 효과를 보인 후, 능동적 학습에서 제안 scoring function과 이전 연구들과의 성능을 비교한다.

실험에서 대표적인 object detection 데이터 셋인 PASCAL VOC [80]와 MS-COCO [71] 데이터를 사용한다. PASCAL VOC는 총 20개의 class 카테고리로 구성된 데이터이며, 학습에서는 VOC07 (VOC2007) trainval 또는 VOC07+12 trainval (VOC2007과 VOC2012의 결합)을 사용한다. 평가 시 VOC07 test를 사용한다. MS-COCO는 총 80개의 class 카테고리로 구성된 데이터이며, 학습 시 MS-COCO train2014를 사용하고 val2017로 평가한다.

능동적 학습을 위한 baseline 알고리즘으로는 이전 연구들 [13], [15]에서 널리 쓰이던 VGG-16 [87] backbone의 Single Shot MultiBox Detector (SSD) [10]를 사용한다. 실험에서 총 iteration은 120,000, batch 크기는 32이며, stochastic gradient descent를 사용한다. 최대 learning rate는 0.001이며, 1,000 iteration까지 learning rate를 warm-up한다. 80,000 iteration과 100,000 iteration에서 learning rate를 각각 10으로 나누어 학습한다. 실험에서 Gaussian mixture 개수는 4개이며, 3번의 독립 시행을 통한 mAP의 평균값과 표준편차로 성능을 평가한다.

4.4.1 객체 탐지에서 mixture 모델링의 효과

본 장에서 object detection에 대한 제안 probabilistic 모델링의 효과를 PASCAL VOC와 MS-COCO 데이터를 통해 보인다. MS-COCO 데이터에서는 train2014의 전체 데이터를 사용하지 않고, random sampling하여 5,000개의 데이터를 사용하여 학습한다. 제안 모델인 $Ours_{gmm}$을 SSD [10]와 여러 네트워크 구성들 (즉, localization head 또는 classification head에서 single 또는 multiple Gaussian 모델링이 적용된 모델)과 비교한다.

표 4.1과 4.2는 각각 PASCAL VOC07과 MS-COCO에서 제안 모델과 baseline 그리고 여러 configuration에 대한 정확도 결과를 나타낸다 (표에서 Loc는 localization을, Cl은 classification을 의미). 표에서 볼 수 있듯이, probabilistic 모델링을 적용한 모든 모델이 baseline인 SSD보다 향상된 정확도를 보인다. 특히, mixture network를 적용한 모델의 정확도가 baseline보다 우수하다. 이는 aleatoric uncertainty로 loss attenuation 효과를 주는 제안 loss function 때문이다. 이를 통해, noisy 데이터에 robust한 모델을 얻을 수 있다. Normal metric (IoU>0.5)과 strict metric (IoU>0.75)을 모두 고려했을 때, 제안 모델인 $Ours_{gmm}$은 PASCAL VOC07와 MS-COCO에서 모든 instance들 및 baseline보다 정확도가 우수하다.

표 4.1. PASCAL VOC 2007 데이터에서 제안 모델과 여러 instance 및 baseline과의 정확도 비교.

Method	Head	IoU > 0.5	IoU > 0.75
SSD	–	69.29±0.51	43.36±1.24
SGM	Loc	70.20±0.27	45.39±0.23
MDN	Loc	70.09±0.22	46.01±0.27
SGM	Cl	69.95±0.41	44.25±0.26
MDN	Cl	70.47±0.17	44.47±0.06
$Ours_{gmm}$	Loc + Cl	**70.19**±0.36	**46.11**±0.38

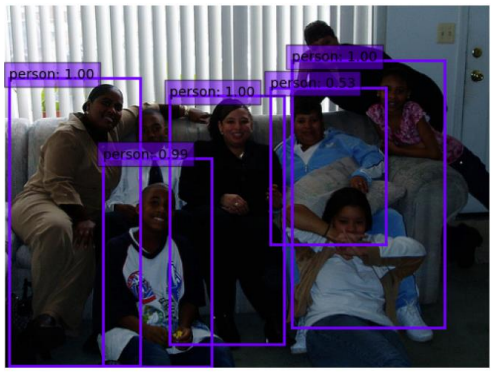
표 4.2. MS-COCO 데이터에서 제안 모델과 여러 instance 및 baseline과의 정확도 비교.

Method	Head	IoU > 0.5	IoU > 0.75
SSD	–	25.63±0.40	11.93±0.60
SGM	Loc	27.20±0.08	12.70±0.16
MDN	Loc	27.67±0.12	13.53±0.05
SGM	Cl	27.23±0.12	12.50±0.08
MDN	Cl	27.33±0.09	12.67±0.09
$Ours_{gmm}$	Loc + Cl	**27.70**±0.08	**13.57**±0.19

그림 4.3는 object detector의 부정확 혹은 잘못 예측한 결과에 대한 시각적 예시와 그에 따른 4가지 uncertainty들의 score를 보여준다. 그림에서 볼 수 있듯이, 각 uncertainty 값은 특정 검출 에러에 대해 유의미한 정보를 제공한다 (그림에서 잘못된 예측과 관련된 uncertainty 값을 굵게 표시). 그림에서 u_{al_b}와 u_{ep_b}는 각각 bounding box coordinate에 대한 aleatoric uncertainty와 epistemic

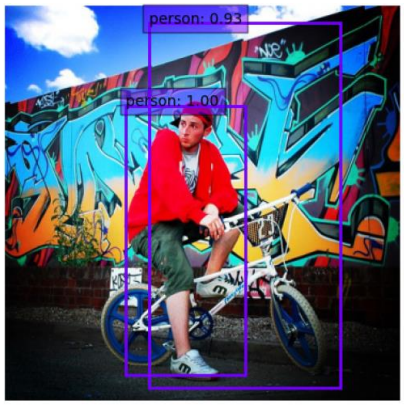
uncertainty를 나타내며, u_{al_c} 와 u_{ep_c} 는 각각 classification에 대한 aleatoric uncertainty와 epistemic uncertainty를 나타낸다. 그림을 통해 알 수 있듯이, localization uncertainty들은 bounding box의 예측 정확도와 관련이 있으며, classification uncertainty들은 class의 카테고리 예측과 연관이 있다. 흥미롭게도, 이 예시들에서 예측이 잘못 됐음에도 불구하고 각 uncertainty 값들은 correlation이 작다. 즉, 각 uncertainty 값들이 부정확한 결과들을 독립적으로 예측할 수 있음을 나타낸다. 이러한 결과들로부터 제안 모델링을 적용한 object detector는 single model 기반 single forward pass로 uncertainty를 예측하며, object detection 네트워크의 성능을 향상시킨다는 것을 알 수 있다. 다음 실험에서 이러한 uncertainty 값들을 조합해 능동적 학습에 사용함으로써, 데이터 sampling 성능을 향상시킬 수 있음을 보인다.

u_{al_b}: **3.60** u_{al_c}: 0.96
u_{ep_b}: 1.06 u_{ep_c}: −0.19



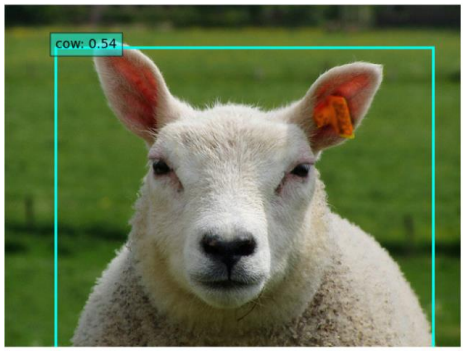
u_{al_b}: **3.45** u_{al_c}: -0.57
u_{ep_b}: 0.88 u_{ep_c}: 0.05

u_{al_b}: 1.49 u_{al_c}: 0.86
u_{ep_b}: **10.39** u_{ep_c}: -0.32

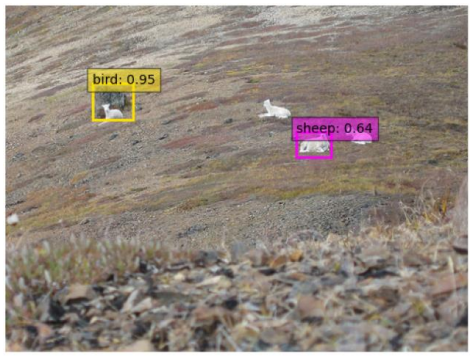


u_{al_b}: 1.77 u_{al_c}: -0.75
u_{ep_b}: **8.10** u_{ep_c}: 0.06

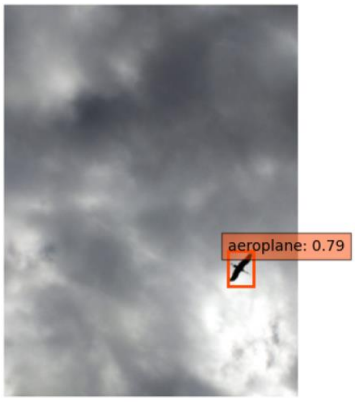
u_{al_b}: -0.82 u_{al_c}: **11.71**
u_{ep_b}: -0.37 u_{ep_c}: 1.31



u_{al_b}: -1.09 u_{al_c}: **8.80**
u_{ep_b}: -0.38 u_{ep_c}: 1.35



u_{al_b}: 0.74 u_{al_c}: 1.06
u_{ep_b}: 0.80 u_{ep_c}: **7.14**



$u_{al_b}: -0.19$ $u_{al_c}: 1.58$
$u_{ep_b}: -0.35$ $u_{ep_c}: \mathbf{10.06}$

그림 4.3. 부정확한 예측에 대한 localization과 classification의 aleatoric uncertainty와 epistemic uncertainty의 예.

4.4.2 능동적 학습 평가

본 장에서 제안 능동적 학습 방법의 우수성을 PASCAL VOC와 MS-COCO 데이터에서 성능 평가하여 보인다. 능동적 학습에서 initial set을 VOC07 데이터에서 2,000개, VOC07+12 데이터에서 1,000개 [15], 그리고 MS-COCO 데이터에서는 5,000개 [17] 데이터를 random하게 선택하여 학습한다. Initial set으로 학습된 모델로 unlabeled 데이터의 객체들을 예측하고, 예측 객체들을 non-maximum suppression으로 처리한 후, 필터링된 검출 결과의 uncertainty들을 집계한다. 제안한 scoring function은 이미지의 최종 informativeness

score를 계산하기 위해 합 또는 최댓값을 사용하여 예측한 uncertainty들을 집계한다. 그 후 모든 unlabeled 이미지들의 score를 기반으로 정렬 후 가장 높은 score를 갖는 데이터부터 총 1,000개의 데이터를 선택 [15]한다. 그리고 선택된 데이터를 학습 데이터 셋에 추가하여 모델을 다시 학습한다. 이러한 능동적 학습 과정을 계속 수행하며, 학습 시 ImageNet pretrained weight를 사용해 모델을 초기화한다.

첫번째 실험에서는 localization과 classification head 각각의 aleatoric unceratinty와 epistemic uncertainty를 여러 조합으로 집계하고, 이들의 능동적 학습 성능을 비교한다. 좀 더 자세히 말하면, random sampling 외 총 7가지 경우의 scoring function의 성능을 비교한다. 첫번째는 classificaton 또는 localizatoin task 각각의 aleatoric uncertainty 또는 epistemic uncertainty만을 사용한 것이다. 두번째는 localization 또는 classification head에서 aleatoric과 epistemic uncertainty의 합이며, 세번째는 localization과 classification head에서 aleatoric uncertainty 또는 epistemic uncertainty의 합이다. 네번째는 localization과 classification head에서 aleatoric uncertainty와 epistemic uncertainty의 합이다. 다섯번째는 localization 또는 classification의 aleatoric uncertainty와 epistemic uncertainty의 최댓값이며, 여섯번째는 localization과 classification의 aleatoric uncertainty 또는 epistemic uncertainty의 최댓값이다. 마지막은 4개의 모든 uncertainty들의 최댓값이다. 본 실험의 결과는 표 4.3에 나타나며, 사용 데이터는 PASCAL VOC07이고, 4.4장에서 언급한 실험 세팅을 따른다. 표에서 볼 수 있듯이, localization과 classification task에서 aleatoric uncertainty와 epistemic uncertainty의 최댓값을 사용한 scoring function이 가장 우수한 성능을 보인다. 다시 말하면,

능동적 학습에서 모든 task들의 모든 uncertainty들의 최댓값을 사용한 경우가 다른 scoring function을 사용한 경우보다 우수한 데이터 sampling 성능을 보인다. 이러한 결과를 기반으로 다음 실험에서는 모든 uncertainty들의 최댓값을 scoring function으로 사용하여 기존 능동적 학습 연구들과 성능을 비교한다.

표 4.3. PASCAL VOC 2007에서 여러 scoring aggregation function의 정확도 비교.

Aggregation function	mAP in % (# images)		
	1st (2k)	2nd (3k)	3rd (4k)
random sampling	62.43±0.10	66.36±0.13	68.47±0.09
u_{al_b}	62.43±0.10	67.06±0.18	68.84±0.18
u_{ep_b}	62.43±0.10	66.75±0.26	69.01±0.17
u_{al_c}	62.43±0.10	67.09±0.09	68.75±0.08
u_{ep_c}	62.43±0.10	66.51±0.12	68.95±0.13
$\sum_{j\in\{u_{al_b},u_{ep_b}\}} u_j$	62.43±0.10	67.01±0.10	68.58±0.29
$\sum_{j\in\{u_{al_c},u_{ep_c}\}} u_j$	62.43±0.10	67.07±0.27	69.03±0.20
$\sum_{j\in\{u_{al_b},u_{al_c}\}} u_j$	62.43±0.10	66.96±0.08	68.92±0.23
$\sum_{j\in\{u_{ep_b},u_{ep_c}\}} u_j$	62.43±0.10	66.49±0.14	68.62±0.24
$\sum_{j\in\{u_{al_b},u_{ep_b},u_{al_c},u_{ep_c}\}} u_j$	62.43±0.10	67.04±0.28	69.09±0.30
$\max_{j\in\{u_{al_b},u_{ep_b}\}} u_j$	62.43±0.10	66.82±0.21	68.95±0.22
$\max_{j\in\{u_{al_c},u_{ep_c}\}} u_j$	62.43±0.10	66.87±0.14	68.99±0.31
$\max_{j\in\{u_{al_b},u_{al_c}\}} u_j$	62.43±0.10	67.18±0.10	69.06±0.25
$\max_{j\in\{u_{ep_b},u_{ep_c}\}} u_j$	62.43±0.10	66.72±0.10	68.99±0.21
$\max_{j\in\{u_{al_b},u_{ep_b},u_{al_c},u_{ep_c}\}} u_j$	62.43±0.10	**67.32**±0.12	**69.43**±0.11

표 4.4은 localization 또는 classification task에서 aleatoric uncertainty 또는 epistemic uncertainty를 사용해 선택된 데이터의 overlapping 비율 결과이다. Localization task에서 aleatoric uncertainty와 epistemic uncertainty로 선택된 데이터의 overlapping 비율은 48% 이다. Classification task에서는 aleatoric과 epistemic uncertainty의 overlapping 비율이 33% 이다. 더욱이, localization과 classification을 함께 사용할 경우 overlapping 비율이 거의 14%로 대폭 감소한다. 즉, 이 결과를 통해 localization과 classification task의 두 가지 uncertainty를 함께 고려하면, 각각의 정보들은 서로 다른 이미지를 선택할 수 있기 때문에 선택의 다양성이 증가하며, 이는 능동적 학습에서 이미지 sampling 성능을 향상시킨다.

표 4.4. 각 task의 aleatoric 또는 epistemic uncertainty로 선택된 데이터의 overlapping ratio (%) 결과.

	Localization		Classification	
	Aleatoric	Epistemic	Aleatoric	Epistemic
	u_{al_b}	u_{ep_b}	u_{al_c}	u_{ep_c}
u_{al_b}	100	48	6	11
u_{ep_b}	48	100	7	14
u_{al_c}	6	7	100	33
u_{ep_c}	11	14	33	100

표 4.5는 VOC07에서 제안 방법과 이전 연구들의 능동적 학습 정확도 및 계산 비용 비교 결과이다. 계산 비용을 비교하기 위해, 각 방법의 모델 parameter 수와 forward time을 사용한다. 일반적으로 빠른 forward (backward) step과 작은 모델 크기는 학습 비용과 능동적 학습의 데이터 sampling 시간을 감소시킨다 [88], [89].

능동적 학습 성능만을 공정하게 비교하기 위해, 기존 sampling 방법들을 제안 Gaussian mixture model architecture에 적용하여 모든 수치를 reproduce한다. 즉, 기존 방법들은 baseline으로 $Ours_{gmm}$과 같은 모델을 사용한다. Ensemble의 경우 비교를 위해 3개의 독립된 모델 [45]을 구성하였다. MC-dropout은 SDD 기반 모델의 extra feature layer에서 5번째 convolutional layer에 p=0.1의 dropout layer를 추가하였다. Image의 score 계산을 위해 MC-dropout에서 25 forward pass [45]를 사용하였다. Ensemble, MC-dropout과 entropy 기반 방법의 경우, 기존 연구들이 했던 것처럼 classification head의 평균 entropy를 image의 최종 score로 사용하였다 [14]. Core-set [16]은 VGG-16 backbone에서 fully connected layer-7의 feature를 사용 [15]하였다. State-of-the-Art (SOTA) 알고리즘인 LLAL [15]과 비교를 위해, [15]에서 제안한 *learning loss prediction module*을 제안 Gaussian mixture model 기반 object detector에 구현하였다. 마지막으로, 제안 모델의 random sampling을 baseline으로 사용하였다. 실험에서 기존 연구들과 제안 방법은 앞서 언급한 실험 환경에 따라 모두 동일한 hyperparameter를 사용하였다. 표 4.5에서 볼 수 있듯이, 제안 접근 방법인 $Ours_{gmm}$은 모든 single-model 기반 능동적 학습 방법들 [10], [13], [15], [16]보다 우수한 성능을 보인다. Multi-model 기반 능동적 방법들 [14], [79]과 비교했을 때, 제안 방법은 더 높은 정확도를 보이며, 필요 계산 비용이 매우 적다. 즉, 제안

방법은 정확도와 계산 비용의 trade-off 측면에서 가장 우수하다.

표 4.5. VOC07에서 능동적 학습의 정확도 및 계산 비용 비교.

	mAP in % (# images)			Number of	Forward
	1st (2k)	2nd (3k)	3rd (4k)	para. ($\times 10^6$)	time (sec)
Random	62.43±0.10	66.36±0.13	68.47±0.09	52.35	0.031
Entropy	62.43±0.10	66.85±0.12	68.70±0.18	52.35	0.031
Core-set	62.43±0.10	66.57±0.20	68.57±0.26	52.35	0.031
LLAL	62.47±0.16	67.02±0.11	68.90±0.15	52.71	0.036
MC-dropout	62.43±0.19	67.10±0.07	69.39±0.09	52.35	0.689
Ensemble	62.43±0.10	67.11±0.26	69.26±0.14	157.05	0.093
$Ours_{gmm}$	62.43±0.10	67.32±0.12	69.43±0.11	52.35	0.031

지금부터는 VOC07+12 데이터에서 기존 연구들과 제안 방법의 능동적 학습 성능을 비교한다. VOC07+12 실험에서는 기존 연구들 중 가장 높은 정확도를 보이는 LLAL [15]에서 이미 제공된 여러 실험 결과들을 참고한다. LLAL [15]에서는 core-set [16], entropy [13], random [10], 그리고 LLAL [15]의 VOC07+12에서의 능동적 학습 정확도를 제공한다. 능동적 학습 성능만을 비교하기 위해, LLAL의 baseline인 SSD를 기반으로 성능을 비교하며, LLAL [15]의 실험 조건 및 환경을 따른다. 이를 위해, 제안 모델인 Gaussian mixture model 기반 네트워크에서 제안 scoring function으로 sampling된 데이터를 가지고 SSD를 학습하여 정확도를 비교한다. Initial training set에 따른 성능 효과를 검증하기 위해, 서로 다른 random seed로 총 5번 독립 시행 (즉, 학습 및 평가)을 하였다. 그 결과 평균 mAP는 0.5246이고 표준 편차는 0.003으로 매우 작은 편차를 보였기 때문에 실험에서

initial subset에 따른 영향이 크지 않음을 알 수 있다. 표 4.6은 VOC07+12에서 single model 기반 능동적 학습 방법과 제안 방법을 비교한 결과이다. 결과를 통해 알 수 있듯이, 제안 방법은 모든 single-model 기반 방식들보다 성능이 우수하다. 특히, 마지막 능동적 학습 iteration에서 제안 방법은 0.7598 mAP의 성능으로 가장 높은 정확도를 보였던 LLAL의 0.7338 mAP보다 2.6 percent points가 높다. 즉, single model 기반 능동적 학습 방식들 중 제안 방법이 가장 우수하다.

표 4.6. VOC07+12에서 제안 방법과 기존 single model 기반 방법들의 정확도 비교.

# of labeled images	Random	Entropy	Core-set	LLAL	$Ours_{gmm}$
1k	0.5262±0.0062	0.5262±0.0062	0.5262±0.0062	0.5238±0.0028	0.5254±0.0017
2k	0.6082±0.0019	0.6123±0.0081	0.6236±0.0052	0.6095±0.0042	0.6130±0.0040
3k	0.6423±0.0022	0.6357±0.0091	0.6590±0.0043	0.6491±0.0047	0.6556±0.0051
4k	0.6633±0.0018	0.6694±0.0021	0.6763±0.0021	0.6690±0.0028	0.6843±0.0043
5k	0.6751±0.0017	0.6870±0.0015	0.6888±0.0048	0.6905±0.0045	0.7077±0.0019
6k	0.6860±0.0050	0.6982±0.0011	0.6944±0.0032	0.7035±0.0055	0.7252±0.0027
7k	0.6927±0.0016	0.7018±0.0027	0.7016±0.0013	0.7149±0.0066	0.7352±0.0025
8k	0.7010±0.0017	0.7112±0.0012	0.7083±0.0012	0.7213±0.0060	0.7453±0.0025
9k	0.7044±0.0047	0.7166±0.0031	0.7115±0.0016	0.7273±0.0030	0.7509±0.0014
10k	0.7117±0.0016	0.7222±0.0024	0.7171±0.0025	0.7338±0.0028	0.7598±0.0021

이제는 VOC07+12에서 제안 방법과 기존 multiple models 기반 방식인 ensemble [14]과 MC-dropout [79]을 비교한다. Ensemble과 MC-dropout은 표 4.5에서 언급한 구현 상세 내용을 SSD에 적용하여 성능을 비교한다. 공정한 비교를 위해, LLAL [15]의 실험 조건 및 환경을 따른다. 표 4.7과 표 4.8은 제안 방법과 multiple models 기반 방법들의 정확도 및 계산 비용 비교 결과이다. 표 4.7에서 볼 수 있듯이,

제안 방법은 ensemble 및 MC-dropout과 비슷한 정확도를 보인다. 그러나, 제안 방법은 single model 기반에서 single forward pass로 능동적 학습을 수행하므로, ensemble 및 MC-dropout 방법들보다 훨씬 효율적이다. 표 4.7에서 MC-dropout의 fwd는 forward를 의미하며, 25 fwd와 50 fwd의 성능 차이는 거의 없다. 하지만, 50 fwd의 경우 25 fwd보다 훨씬 많은 계산 비용을 요구한다. Parameter와 관련해서는, MC-dropout의 경우 dropout layer가 추가 parameter를 요구하지 않기 때문에 baseline인 SSD와 parameter 수가 동일하다. 하지만, uncertainty 예측을 위해 여러 번의 forward pass가 필요하다는 문제가 있다. 제안 방법은 localization과 classification head에서 aleatoric uncertainty와 epistemic uncertainty 예측을 위해 출력 layer에 추가 parameter가 필요하다. 그렇기 때문에 제안 방법의 parameter 수는 baseline인 SSD보다 크다. Ensemble 기반 방법의 경우 parameter 수는 ensemble에서 사용한 SSD 수에 비례한다 [90]. 표 4.8에서 볼 수 있듯이, 제안 방법은 MC-dropout과 ensemble기반 방법보다 계산 비용이 훨씬 적다. 즉, 제안 방법은 정확도와 계산 비용의 trade-off 측면에서 가장 우수하다.

표 4.7. VOC07+12에서 제안 방법과 기존 multiple models 기반 방법들의 정확도 비교.

# of images	MC-dropout (50 fwd)	MC-dropout (25 fwd)	Ensemble	$Ours_{gmm}$
1k	0.5235±0.0004	0.5235±0.0004	0.5254±0.0017	0.5254±0.0017
2k	0.6059±0.0026	0.6059±0.0028	0.6020±0.0093	0.6130±0.0040
3k	0.6660±0.0023	0.6690±0.0030	0.6570±0.0099	0.6556±0.0051
4k	0.6890±0.0018	0.6840±0.0019	0.6920±0.0034	0.6843±0.0043
5k	0.7060±0.0045	0.7080±0.0041	0.7150±0.0018	0.7077±0.0019
6k	0.7200±0.0012	0.7190±0.0050	0.7290±0.0027	0.7252±0.0027
7k	0.7367±0.0015	0.7381±0.0003	0.7429±0.0004	0.7352±0.0025
8k	0.7468±0.0027	0.7475±0.0056	0.7491±0.0041	0.7453±0.0025
9k	0.7549±0.0013	0.7558±0.0023	0.7589±0.0025	0.7509±0.0014
10k	0.7567±0.0048	0.7601±0.0018	0.7590±0.0032	0.7598±0.0021

표 4.8. 제안 방법과 multiple models 기반 방법의 계산 비용 비교.
(표에서 M는 million을 의미하며, sec는 seconds를 의미한다.)

	SSD	Ensemble	MC-dropout	$Ours_{gmm}$
# of parameters (M)	26.29	78.87	26.29	52.35
Forward time (sec)	0.023	0.069	0.412	0.031

표 4.9는 MS-COCO 데이터에서 제안 능동적 학습 방법과 기존 밥법들의 성능 비교 결과이다. 능동적 학습 성능만을 비교하기 위해, 기존 방법들을 제안 Gaussian mixture model 기반 네트워크에 적용하여 성능을 reproduce한다. 즉, $Ours_{gmm}$과 동일한 네트워크에 제안 sampling 방법 및 기존 방법들을 적용하여 비교한다. 실험 세팅은 표 4.5에서 사용한 세팅을 따른다. 결과를 통해 알 수 있듯이, 제안 방식은 기존 single model 기반 능동적 학습 방식들 [10], [13], [15], [16]보다 성능이 우수하다. 특히, VOC07+12에서 기존 연구들 중 가장 높은 성능을 보였던 LLAL [15]은 MS-COCO에서 random sampling과 비슷한 성능을 보였다. 이는 LLAL이 데이터의 diversity나 density를 고려하지 못했고, LLAL의 loss prediction module이 복잡한 task에서 정확도가 낮은 경향을 보였기 때문이다. 하지만, 제안 방법은 MS-COCO에서도 우수한 성능을 보인다. Multiple models 기반 방식들 [14], [79]과 비교했을 때, 제안 방법은 기존 multiple models 기반 방식들보다 매우 적은 연산량으로 높은 정확도를 달성한다. 이러한 결과를 통해 제안 방법은 많은 class로 구성된 diversity가 높은 데이터에서도 효과가 있으며, 기존 연구들보다 우수하다.

표 4.9. MS-COCO에서 능동적 학습의 정확도 및 계산 비용 비교.

	mAP in % (# images)			Number of	Forward
	1st (5k)	2nd (6k)	3rd (7k)	para. ($\times 10^6$)	time (sec)
Random	27.70±0.08	28.70±0.13	29.83±0.04	116.51	0.152
Entropy	27.70±0.08	28.93±0.11	29.89±0.09	116.51	0.152
Core-set	27.70±0.08	28.99±0.01	29.93±0.06	116.51	0.152
LLAL	27.71±0.03	28.71±0.06	29.53±0.15	116.87	0.194
MC-dropout	27.70±0.10	29.20±0.09	30.30±0.08	116.51	3.718
Ensemble	27.70±0.08	29.03±0.07	30.02±0.06	349.53	0.456
$Ours_{gmm}$	27.70±0.08	29.28±0.05	30.51±0.12	116.51	0.152

4.4.3 확장성과 데이터 transferability

본 장의 제안 모델링 방법은 single-stage object detector 뿐만 아니라 two-stage object detector에도 적용 가능하다. 제안 모델링 방법의 확장성을 보이기 위해 대표적인 two-stage object detector인 Faster-RCNN [5]에 제안 모델링 방법을 적용하여 성능을 비교한다. 실험에서는 Faster-RCNN에 FPN [69]이 적용된 모델을 사용한다. 실험 데이터로는 PASCAL VOC07 데이터를 사용한다. 표 4.10은 Faster-RCNN에 제안 모델링 방법 적용 후의 정확도 및 계산 비용 결과를 보여준다. 결과를 통해 알 수 있듯이, 제안 모델링이 적용된 모델은 baseline인 Faster-RCNN보다 정확도가 0.66 mAP 향상된다. 특히, 제안 모델링 방법은 Faster-RCNN의 region proposal stage가 아닌 detection 네트워크의 출력 layer에 적용되기 때문에 연산 비용 및 latency가 거의 증가하지 않는다.

표 4.10. Faster-RCNN에서 제안 모델링 적용 전과 후의 정확도 및 계산 비용 비교.

		Faster-RCNN	$Ours_{gmm}$
mAP (%)	IoU>0.5	75.31±0.22	75.90±0.09
	IoU>0.75	48.70±0.11	49.36±0.07
# of parameters (M)		41.17	42.23
Forward time (sec)		0.059	0.062

Object detection 알고리즘은 계속해서 업데이트 되며, 새로운 detector가 나올 때 마다 처음부터 모든 데이터를 능동적 학습하는 것은 시간 및 연산 비효율적이다. 이전 모델에서 능동적 학습으로 sampling된 데이터가 새로운 모델에 효과가 있는지를 확인하기 위해 데이터 transferability 실험을 한다. Baseline인 SSD를 기반으로 Resnet-34, Resnet-50 [52] 등 다른 backbone을 사용한 모델과 아예 다른 네트워크 구조를 갖진 Faster-RCNN [5]에서 실험을 진행한다. 본 실험에서는 표 4.5의 마지막 능동적 학습 iteration에서 sampling된 데이터를 사용한다. 제안 방법의 우수성을 보이기 위해 비교 대상으로는 ranom sampling된 데이터를 사용한다. 표 4.11은 random하게 sampling된 데이터와 제안 방법 (localization과 classification의 모든 uncertainty의 최댓값을 scoring function으로 사용)으로 sampling된 데이터의 transferability 결과를 나타낸다. 결과를 통해 알 수 있듯이, 제안 방법으로 sampling된 데이터는 random하게 sampling된 데이터보다 정확도가 1.47 mAP 우수하다. 그리고 다양한 네트워크 구조에서 제안 sampling 방법으로 추출된 데이터가 random하게 추출된 데이터보다 우수한 정확도를 보인다. 이러한 결과들을 바탕으로 제안 방법은 다른 object detection

네트워크로 확장 가능하며, 제안 능동적 학습 방법으로 선택된 데이터는 다른 architecture에 적용 가능하다.

표 4.11. 여러 네트워크 구조에서 제안 방법의 데이터 transferability 성능 비교.

Model	Backbone	mAP in %	
		Random selection	$Ours_{gmm}$ selection
SSD	VGG-16	67.77±0.12	68.71±0.18
	Resnet-34	65.53±0.17	67.00±0.14
	Resnet-50	64.28±0.39	65.73±0.32
Faster-RCNN	Resnet-50-FPN	72.93±0.41	73.60±0.18

4.4.4 Parameter 민감도

본 장의 첫번째 실험에서 mixture 개수 (K)에 따른 성능을 비교한다. 표 4.12는 제안 Gaussian mixture model 기반 네트워크에서 mixture 개수가 1, 2, 4, 그리고 8일 때의 정확도 및 계산 비용 결과를 나타낸다. 정확도의 경우 IoU>0.5인 normal metric과 IoU>0.75인 strict metric에서 3번의 독립 시행에 대한 평균 정확도와 표준 편차를 제시한다. 계산 비용의 경우 모델 parameter 수와 forward time을 사용해 비교한다. 표에서 M는 million을 의미하며, sec는 seconds를 의미한다. 실험 결과에서 K가 1과 8인 경우 정확도가 하락함을 확인할 수 있다. K가 1인 경우 single Gaussian 모델이기 때문에 복잡한 분포를 모델링하지 못하며, epistemic uncertainty를 예측할 수 없다. K 개수가

많아지는 경우 fluctuation 때문에 실제 구현 모델을 학습시키기가 어렵다. 이러한 이유로 정확도 하락이 발생한다. Parameter 수와 forward time은 K에 따라 많은 차이를 보인다. 본 연구에서는 normal metric과 strict metric의 정확도와 계산 비용의 trade-off를 고려하여 가장 좋은 성능을 보인 K=4를 사용해 모든 실험을 진행하였다.

표 4.12. VOC07 데이터에서 mixture model 개수에 따른 정확도 및 계산 비용 비교.

K # of mixture	mAP (%) IoU>0.5	IoU>0.75	# of parameters (M)	Forward time (sec)
1	69.89±0.23	45.18±0.24	29.6	0.021
2	70.29±0.29	45.98±0.38	37.6	0.025
4	70.19±0.36	46.11±0.38	52.3	0.031
8	70.01±0.29	45.69±0.28	81.8	0.051

이 외에도, 제안 모델링 방법이 input resolution에 강경하다는 것을 보이기 위해 제안 모델과 baseline의 input resolution에 따른 성능 결과를 표 4.13에 제시한다. Input resolution으로는 512×512와 이전 결과에서 사용한 300×300을 사용하여 비교한다. 해당 실험은 표 4.1에서 한 실험과 input resolution을 제외하고는 동일하다. Baseline으로는 SSD [10]를 사용하며, 표 4.13에서 볼 수 있듯이 input resolution이 증가해도 제안 방법은 baseline 대비 mAP를 대폭 향상시킨다 (높은 input resolution에서 strict metric (IoU>0.75) 기준 SSD보다 정확도가 2.49 percent points 향상). 즉, 제안 방법은 예측

bounding box와 ground-truth box의 높은 intersection이 필요한 시나리오에서 두드러진 성능 향상을 보인다.

표 4.13. Input resolution에 따른 정확도 (%) 비교.

Method	SSD 512 (512×512)		SSD 300 (300×300)	
	IoU>0.5	IoU>0.75	IoU>0.5	IoU>0.75
SSD	73.22±0.35	45.74±0.70	69.29±0.51	43.36±1.24
$Ours_{gmm}$	73.50±0.12	48.23±0.53	70.19±0.36	46.11±0.38

제안 능동적 학습 방법의 budget number에 따른 성능 비교 결과를 표 4.14에 제시한다. 4.4.2 장에서 언급한 VOC07+12 데이터에서의 능동적 학습 결과는 이전 연구 [15]와의 비교를 위해 1k (1,000개)의 budget number를 사용하였다. 본 실험에서는 추가로 3k와 9k budget number일 때의 능동적 학습 결과를 비교한다. 표에서 결과 값은 이전과 동일하게 3번의 독립 시행에 따른 평균 정확도와 표준편차이다. 표 4.14에서 볼 수 있듯이, budget number가 작을수록 능동적 학습의 정확도 향상폭이 크다. 능동적 학습의 마지막 iteration인 10k에서 정확도 비교 시, 9k budget인 경우 0.7493 mAP 성능을 보이지만, 3k에서는 0.7550 mAP 성능을 보인다. 반면에 1k인 경우 0.7598 mAP로 가장 높은 정확도를 보인다. 일반적으로 budget number가 작을수록 전체 능동적 학습 시간이 길어지기 때문에 사용자의 application 및 데이터 양에 따라 알맞은 budget number 설정이 필요하다.

표 4.14. 능동적 학습에서 budget number에 따른 mAP 비교.

# of images	Budget number		
	9k	3k	1k
1k	0.5254±0.0017	0.5254±0.0017	0.5254±0.0017
2k	-	-	0.6130±0.0040
3k	-	-	0.6556±0.0051
4k	-	0.6797±0.0011	0.6843±0.0043
5k	-	-	0.7077±0.0019
6k	-	-	0.7252±0.0020
7k	-	0.7335±0.0049	0.7352±0.0025
8k	-	-	0.7453±0.0025
9k	-	-	0.7509±0.0014
10k	0.7493±0.0043	0.7550±0.0012	0.7598±0.0021

능동적 학습을 통한 정확도의 saturation point 확인을 위해, 표 4.14의 1k budget number에 대해 능동적 학습을 계속 수행한다. VOC07+12의 모든 데이터를 사용해서 학습한 경우 (즉, 총 16,551개의 데이터), 77.20% mAP의 정확도를 보인다. 하지만, 제안 능동적 학습을 사용한 경우 14k (14,000개)의 데이터만으로 77.28% mAP를 보여, 적은 데이터를 사용했음에도 더 높은 정확도를 달성한다. Random sampling으로 14k 데이터를 sampling한 후 학습한 경우에는 76.32% mAP를 보인다. 즉, 제안 능동적 학습 방법으로 정확도의 saturation point에 빨리 도달할 수 있으며, random sampling보다 성능이 우수하다.

4.4.5 Classification loss에 대한 논의

4.3.1 장에서 제안한 classification loss 외에 다음과 같은 classification loss로 제안 mixture 모델을 학습할 수 있다:

$$L_{cl}^{Pos}(y,c) = -\sum_{i \in Pos}^{N} y_G^i log \sum_{k=1}^{K} \pi^{ik} Softmax(\hat{c}_p^{ik}), \qquad (4.21)$$

$$L_{cl}^{Neg}(y,c) = -\sum_{i \in Neg}^{M \times N} y_0^i log \sum_{k=1}^{K} \pi^{ik} Softmax(\hat{c}_p^{ik}), \qquad (4.22)$$

식에서 y_G와 y_0는 각각 ground-truth class와 background class에서 1을 갖는 one-hot vector이다. 나머지 parameter는 식 4.13, 식 4.14와 동일하다. 본 장의 제안 classification loss인 식 4.13과 4.14의 경우 (*type 1 loss*로 정의), 학습 시 여러 개의 분포 중 하나의 분포에 가중치가 집중되는 문제가 있었다. 식 4.21과 4.22의 경우 (*type 2 loss*로 정의), 이러한 문제가 완화되는 경향을 보인다. 표 4.15와 표 4.16은 각각 PASCAL VOC07와 MS-COCO에서 두 가지 classification loss의 능동적 학습 결과를 나타낸다. 표를 통해 알 수 있듯이, PASCAL VOC에서 두 loss function의 능동적 학습 성능은 큰 차이가 없지만, MS-COCO에서는 두 loss function의 성능 차이가 크다. Type 2 loss가 type 1 loss 대비 가중치 쏠림 문제를 완화시키지만, class가 많은 데이터에서는 성능 향상 효과가 미비하다. 본 장에서는 최종 성능이 뛰어난 type 1 loss로 모든 실험을 진행하였지만, 가중치 쏠림 문제를 해결하면서도 전체적인 능동적 학습 성능을 향상시킬 수 있는 classification loss 설계에 대한 연구가 향후 필요하다.

표 4.15. VOC07에서 classification loss에 따른 능동적 학습의 정확도 비교.

Cls. loss	mAP in % (# images)		
	1st (2k)	2nd (3k)	3rd (4k)
Type 1	62.43±0.10	67.32±0.12	69.43±0.11
Type 2	62.64±0.21	67.20±0.22	69.40±0.14

표 4.16. MS-COCO에서 classification loss에 따른 능동적 학습의 정확도 비교.

Cls. loss	mAP in % (# images)		
	1st (5k)	2nd (6k)	3rd (7k)
Type 1	27.70±0.08	29.28±0.05	30.51±0.12
Type 2	27.38±0.16	28.69±0.22	29.55±0.14

4.5 본 연구의 결론

본 장에서 object detection을 위한 새로운 심층 능동적 학습 방법을 제안하였다. 제안 방법은 mixture density network를 사용하여, single model기반 single forward pass로 localization과 classification task에서 aleatoric uncertainty와 epistemic uncertainty를 예측하는 방법을 제안하였다. 그리고 예측한 모든 uncertainty들을 능동적 학습을 위한 scoring function에서 고려하였다. 제안한 mixture modeling과 scoring function은 정확도와 계산 비용 측면에서 뛰어난 성능 향상을 보였다. 본 장에서는 일반적인 object detection 데이터셋인 PASCAL VOC와 MS-COCO를 사용하여 다양한 범위의 실험들을 통해 제안 방법의 우수성을 보였다. 게다가, 제안 방법이 다양한 backbone 및 네트워크 architecture에 적용 가능함을 실험을 통해 보였다.

제 5 장 반지도 학습을 위한 unlabeled 데이터 필터링 및 uncertainty 기반 반지도 능동적 학습

5.1 본 연구의 서론

일반적으로 심층 객체 탐지 신경망 학습을 위한 데이터 형성 시, 높은 정확도 달성을 위해 이미지 내 여러 물체에 대한 카테고리 정보와 bounding box coordinates을 labeling 한다. 하지만 labeling 작업은 많은 resource와 시간을 요구하기 때문에, 이러한 labeling cost를 줄이기 위해 능동적 학습 (active learning), 반지도 학습 (semi-supervised learning) [49], [50], weakly-supervised learning [91], [92], [93], [94], weakly-semi-supervised learning [95], [96], 그리고 반지도 능동적 학습 (semi-supervised active learning) [97] 등 다양한 학습 기법들이 연구되어 왔다.

다양한 학습 기법들 중 반지도 학습은 unlabeled 데이터와 labeled 데이터를 함께 사용하는 학습 방법이다. 이 방법은 unlabeled 데이터에 추가적인 labeling을 요구하지 않기 때문에, 여러 학습 방법들 중 labeling cost를 가장 많이 줄일 수 있으며, 최근에는 object detection을 위한 반지도 학습 연구들이 활발히 수행되고 있다 [18], [51]. 하지만, 기존 연구들은 반지도 학습의 정확도 향상에만 초점을 맞춰왔으며, 학습에서 unlabeled 데이터의 유용성은 고려하지 못했다. 일반적으로 unlabeled 데이터의 양은 labeled 데이터보다 훨씬 많으며, 모델이 모든 labeled 데이터와 unlabeled 데이터를 학습하기 위해서는 많은 학습 시간이 필요하다는 문제가 있다.

본 장에서는 object detection의 localization과 classification 기반 informativeness score를 통해 unlabeled 데이터 중 모델이 이미 잘 알고 있는 데이터를 필터링하는 방법을 제안한다. 즉, 모델의 정확도 향상에 기여도가 낮은 unlabeled 데이터를 제거함으로써, 전체 학습 데이터를 줄여 모델의 학습 시간을 대폭 감소시킨다. 제안 방법은 기존 반지도 학습 방법 대비 50% 적은 unlabeled 데이터를 사용하여 더 적은 학습 iteration으로 정확도를 향상시킨다. 이 외에도, 본 장은 4장의 제안 능동적 학습 방법과 본 장의 제안 반지도 학습을 결합한, uncertainty 기반 반지도 능동적 학습 모델을 제시한다. 두 가지 학습 방법이 결합된 제안 반지도 능동적 학습 방법은 단일 학습 방법들 대비 정확도를 대폭 향상시킨다.

5.2 본 연구의 관련 연구

Jeong *et al.* [18]은 object detection에서 consistency 기반으로 반지도 학습을 시도한 최초의 논문이다. 이 방법은 그림 5.1과 같이 원본 이미지와 flip된 이미지를 객체 탐지 네트워크의 입력으로 넣고, 두 예측 결과의 차이를 최소화하는 방향으로 모델을 학습한다. 이를 위해, classification과 localization task의 consistency loss를 각각 제안하였다. 반지도 학습에서 labeled 데이터가 입력이 되면, supervised loss와 consistency loss를 함께 사용해 학습을 하며, unlabeled 데이터가 입력되면, consistency loss만으로 학습한다.

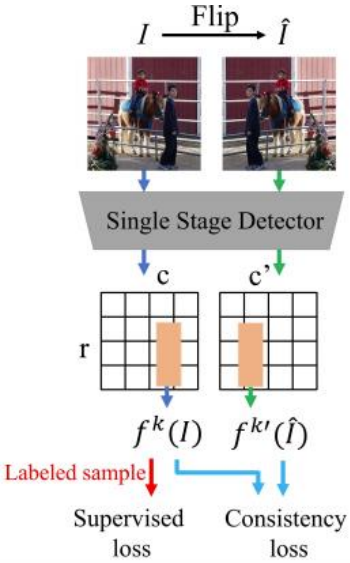


그림 5.1. 객체 탐지를 위한 consistency 기반 반지도 학습 [18].

현재 가장 좋은 성능을 보이는 interpolation 기반 반지도 학습 방법 [51]은 그림 5.2와 같이 mixup 기반의 데이터 augmentation을 활용해 새로운 unlabeled 데이터 학습 loss인 interpolation-based loss function을 제안하였다. 그리고 이를 기존 consistency 기반 반지도 학습 [18]의 consistency loss와 함께 사용하여, 가장 높은 정확도를 얻었다. 그림 5.3와 같이 원본 입력 이미지 (A)와 flip된 원본 이미지 ($\hat{A}$)로 mixup된 이미지인 M을 얻어야하는데, 이를 위해 flip된 이미지를 shuffle하여 B를 만들고, 이를 A와 함께 사용하여 M을 만든다. 원본 데이터인 A와 filp된 데이터 $\hat{A}$는 [18]에서 제안한 consistency loss를 사용해 학습되며, mixup된 M과 A 또는 B는 interpolation 기반 loss를 사용해 학습된다.

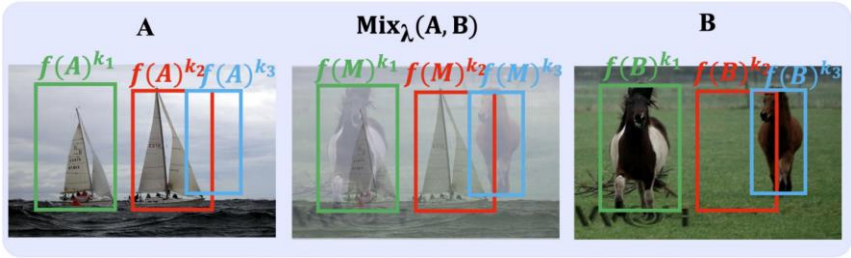


그림 5.2. [51]에서 사용한 mixup augmentation의 예.

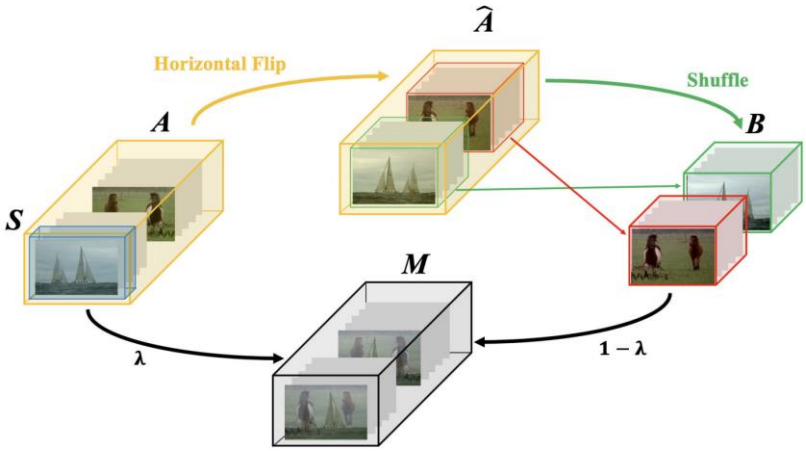


그림 5.3. [51]에서 제안한 interpolation 기반 방법과 consistency 기반 방법 [18]의 결합.

이 방법 외에도, self-training과 consistency regularization을 결합한 방법 [98], 원본 예측과 noisy 예측의 consistency를 반지도 학습에 활용한 연구 [99]가 제안되었다. 하지만, 기존 대부분의 반지도 학습 연구들은 정확도 향상에만 중점을 두었고, 학습 데이터의 효율성 측면은 고려하지 못했다. 본 연구는 unlabeled 데이터의 유용성을 고려해 기존 반지도 학습의 정확도를 유지하면서 학습 시간을 대폭 감소시키는 방법을 제안한다. 본 장의 제안 방법은 consistency 기반 반지도 학습 [18]을 baseline으로 한다.

5.3 Unlabeled 데이터 필터링

4장에서 제안한 predictive uncertainty 기반 informativeness score는 모델이 현재 입력 데이터에 대해 모르는 정도를 정량적으로 나타내는 지표이다. 즉, informativeness score가 낮은 unlabeled 데이터는 모델이 이미 잘 알고 있는 데이터로 판단할 수 있고, 반대의 경우 모델이 잘 모르는 데이터라고 판단할 수 있다. 그렇기 때문에 낮은 informativeness score를 갖는 unlabeled 데이터는 전체 학습 데이터에서 필터링을 하여, 그림 5.4와 같이 기존보다 unlabeled 학습 데이터를 줄여 반지도 학습 시간을 감소시키는 것이 가능하다. 이 때, labeled 데이터의 양은 기존과 동일하다.

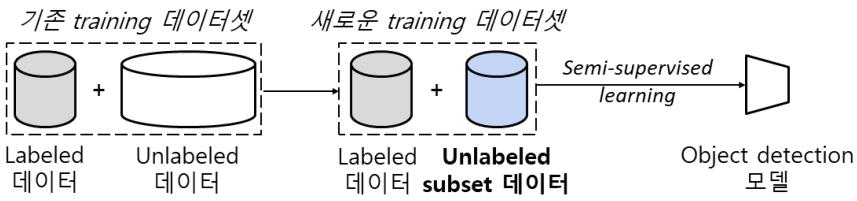


그림 5.4. Labeled 데이터와 필터링된 unlabeled subset 데이터로 구성된 반지도 학습을 위한 최종 학습 데이터의 예.

제안 방법의 데이터 필터링 과정은 그림 5.5과 같다. Labeled 데이터로 학습된 모델로 전체 unlabeled 데이터들의 informativeness score를 계산한다. Informativeness score은 localization과 classification head의 aleatoric uncertainty와 epistemic uncertainty를 모두 고려한 scoring function으로 계산된다. 이 과정은 4장에서 제안한 능동적 학습을 위한 scoring function과 동일하다. 그 후, informativeness score 기반으로 가장 높은 score를 갖은 unlabeled 데이터가 상위 index에 위치하도록 정렬한다. Informativenss score가 높은 unlabeled 데이터는 현재 모델이 잘 모르는, 신뢰하지 못하는 데이터이며, 반대의 경우는 현재 모델이 잘 알고 있는 데이터로 판단할 수 있다. 최종적으로 현재 모델이 잘 알고 있는 데이터를 학습 데이터에서 제거하여, 모델이 어려워하는 데이터로 구성된 unlabeled subset 데이터를 추출한다. 이렇게 얻어진 unlabeled subset 데이터와 labeled 데이터로 최종 학습 데이터를 구성하고, 기존 반지도 학습 [18]을 적용한다. 제안 방법은 학습 프로세스를 바꾸는 것이 아닌, 새로운 학습 subset 데이터를 구성하는 것이므로 기존의 여러 반지도 학습 방법들에 그대로 적용할 수 있다.

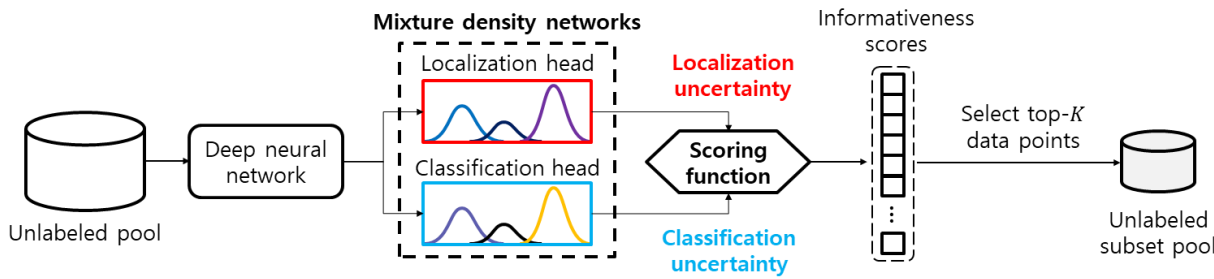


그림 5.5. 제안 필터링 방법 프로세스.

5.4 Uncertainty 기반 반지도 능동적 학습

이 장에서는 4장의 제안 능동적 학습 방법과 5장의 제안 반지도 학습 방법을 결합한 uncertainty 기반 반지도 능동적 학습 방법을 제시한다. 두 가지 학습 방법을 함께 사용하기 위해선, object detection 네트워크의 loss function 재설계가 필요하다. 이를 위해, 그림 5.6과 같이 mixture density network를 사용한 object detection 모델에 unlabeled 데이터 학습을 위한 consistency loss를 모델링 한다. 이 때, 원본 이미지에 대한 각 task의 예측 aleatoric uncertainty를 기존 consistency loss에 적용한다:

$$L_{cons} = \frac{\alpha}{4}\left(\frac{l_{con-loc_x}}{{u_{al_x}}^2} + \frac{l_{con-loc_y}}{{u_{al_y}}^2} + \frac{l_{con-loc_w}}{{u_{al_w}}^2} + \frac{l_{con-loc_h}}{{u_{al_h}}^2}\right) + \beta\frac{l_{con-cls}}{u_{al_c}} \qquad (5.1)$$

식에서 $l_{con-loc}$ 와 $l_{con-cls}$ 는 [18]에 언급된 localization과 classification의 consistency loss이며, u_{al} 은 식 4.20과 같이 각 task의 aleatoric uncertainty를 의미한다. 실험에서 α는 7, β는 1.5를 사용한다. 기존 반지도 학습 방법과 동일하게 labeled 데이터의 경우 supervised learning loss와 consistency loss를 함께 사용하여 학습하고, unlabeled 데이터의 경우 consistency loss를 사용해 학습한다.

첫번째 단계로, labeled 데이터로 학습된 모델로 그림 5.6의 파란색 부분 (즉, 능동적 학습을 위한 uncertainty 예측 및 informativeness score 계산)과 같이 모든 unlabeled 데이터에 대해 informativeness score를 계산한다 (그림 5.6에서 사용한 notation은 4장과 동일). 계산한 informativeness score를 기준으로 정렬한 후, 낮은

informativeness score를 갖는 unlabeled 데이터를 사전에 정의된 비율만큼 학습 데이터에서 필터링한다. 그리고 기존 labeled 데이터와 필터링된 unlabeled 데이터를 사용해 반지도 학습으로 제안 모델을 재학습한다. 재학습 완료 후, 이전과 동일하게 제안 scoring function으로 informativeness score를 계산하고, 계산한 informativeness score를 기준으로 가장 높은 값을 갖는 데이터부터 특정 budget만큼을 선택하여 labeling 한다. 이렇게 labeling된 데이터를 기존 labeled 학습 데이터에 추가하고, 낮은 informativeness score를 갖는 unlabeled 데이터는 사전에 정의된 비율만큼 학습 데이터에서 필터링한다. 그런 다음, 새로운 labeled 데이터와 필터링된 unlabeled 데이터를 사용해 반지도 학습으로 제안 모델을 재학습한다. 이러한 과정을 반복 수행하며, 이를 통해 반지도 능동적 학습이 가능하다.

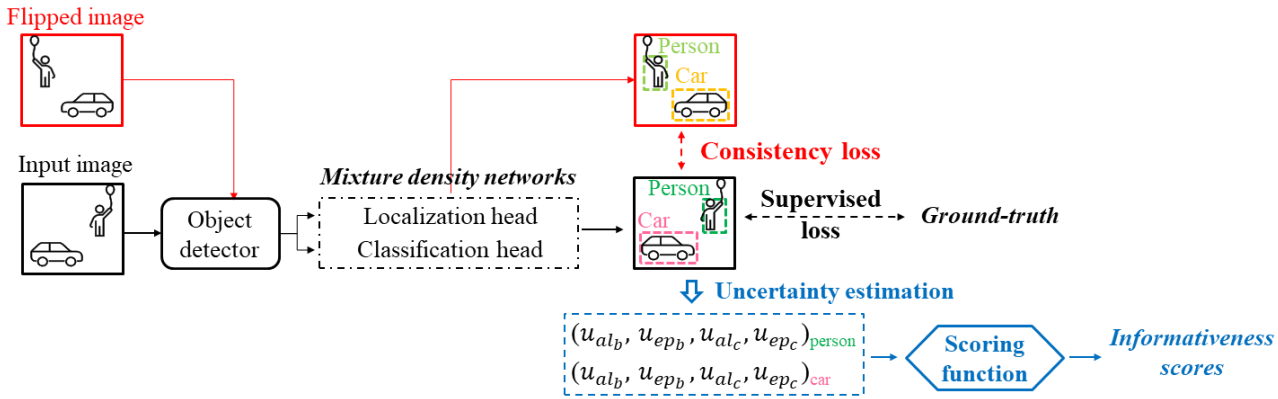


그림 5.6. 제안 반지도 능동적 학습을 위한 loss function 및 uncertainty 예측 프로세스.

5.5 본 연구의 실험 결과

본 장의 실험에서는 제안 모델의 mixture distribution의 개수를 2로 설정하여 실험한다.

5.5.1 Unlabeled 데이터 필터링

제안 방법의 우수성을 보이기 위해 대표적인 object detection 데이터 셋인 PASCAL VOC [80]를 사용해 평가한다. PASCAL VOC 데이터의 구성은 4.4장과 같다. 반지도 학습 평가를 위해 labeled 데이터와 unlabeled 데이터 구별이 필요하다. 이를 위해, 기존 연구 [18]와 같이 labeled 데이터는 총 5,011개로 구성된 PASCAL VOC 2007 trainval을 사용하며, unlabeled dataset은 총 11,540개로 구성된 PASCAL VOC 2012 trainval를 사용한다. 성능 평가 시 총 4,952개로 구성된 PASCAL VOC 2007 test를 사용한다. 실험을 위한 baseline으로는 SSD [10] 기반의 consistency-regularization [18]을 사용하며, localization과 classification의 uncertainty 예측을 위해 baseline 모델 [10]에 4장에서 제안한 mixture 모델링을 적용한다.

표 5.1은 제안 필터링 방법 적용 전과 후의 정확도 (mAP) 비교 결과이다. 표의 정확도 수치는 3번의 독립 시행에 따른 평균 값과 표준 편차이다. 실험에서 제안 방법의 unlabeled 데이터 필터링 비율을 50%로 설정한다. Ablation study로 50% 필터링 시 random sampling한 경우와 모델이 잘 모르는 데이터를 50% 필터링한 경우를 추가로 제시한다. 표에서 *Random*은 random sampling을 의미하며, *Low UC*는 모델이 이미 잘 알고 있는 (low uncertainty) unlabeled 데이터를

학습 시 사용했다는 의미이다. 제안 방법으로 50% 데이터 필터링 (*High UC*) 시 (즉, 모델이 잘 모르는 데이터를 반지도 학습에 사용) 정확도는 72.39% mAP로 전체 데이터를 다 사용한 경우 (72.27% mAP)보다 데이터를 적게 사용했음에도 불구하고 0.12 percent points 정확도가 높다. 반대로 낮은 uncertainty를 갖는 unlabeled 데이터를 학습에 사용한 경우 (*Low UC*) 71.79% mAP로 정확도가 하락한다. Random sampling으로 50% 데이터 필터링 시 71.83% mAP로 *Low UC*보다 정확도가 높다. 이러한 결과를 통해, 제안한 informativeness score는 모델이 잘 모르는 데이터와 잘 알고 있는 데이터를 판단할 수 있으며, 이를 기반으로한 unlabeled 데이터 필터링은 기존보다 적은 학습 데이터로 정확도를 향상시킨다.

그림 5.7은 제안 unlabeled 데이터 필터링 적용 전과 후의 학습 iteration 비교 결과이다. 공정한 비교를 위해, 전체 데이터를 사용한 경우와 제안 방법으로 unlabeled 데이터를 필터링한 경우 모두 230 epoch으로 학습 [10] 하였다. 그림에서 각 수치는 3번의 독립 시행에 대한 평균 값이다. 필터링 없이 전체 unlabeled 데이터를 사용한 경우, 학습 iteration이 총 120,000번 필요하며 정확도는 72.27% mAP를 보였다. 반면에, 제안 방법으로 필터링을 한 경우 학습 iteration이 80,000번 필요하며, 정확도는 기존보다 높은 72.39% mAP를 보였다. Ablation study로 전체 학습 데이터를 80,000 iteration으로 학습한 경우, 정확도가 71.79% mAP로 감소한다. NVIDIA TITAN Xp GPU 기반 환경에서 10,000 iteration 학습 시 대략 2.7 시간이 소요된다. 즉, 제안 필터링은 기존 대비 학습 iteration을 40,000번 감소시키므로 학습 시간을 대략 10.8 시간 줄인다. 학습 시간이 짧아질수록 resource 및 소비 전력이 감소하므로, 제안 방법은 딥러닝 모델의 학습 효율성을 크게 향상시킨다.

표 5.1. 제안 방법 적용 전과 후의 반지도 학습 정확도 비교.

	Labeled data	Unlabeled data	mAP (%)
Baseline	VOC 2007	-	70.19±0.36
	VOC 2007	VOC 2012	72.27±0.02
Subset (*Random*)	VOC 2007	VOC 2012 (50%)	71.83±0.03
Subset (*Low UC*)	VOC 2007	VOC 2012 (50%)	71.79±0.13
Subset (*High UC*)	VOC 2007	VOC 2012 (50%)	72.39±0.04

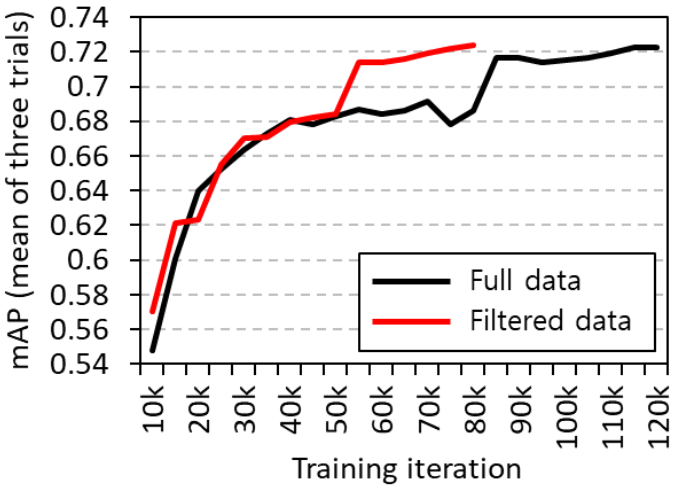


그림 5.7. 제안 방법 적용 전과 후의 학습 iteration과 정확도 비교.

표 5.2는 다양한 필터링 비율에 따른 정확도와 학습 iteartion 결과를 나타낸다. 25%만 필터링한 경우 50% 필터링한 경우보다 정확도는 거의 유사하지만, 데이터 양이 많기 때문에 학습 iteration이 증가한다. 75%로 필터링한 경우 학습 iteration은 대폭 감소하지만, 정확도가 많이 하락한다. 사용자의 application에 따라 비율을 조절할 필요가 있으며, 본 장에서는 가장 높은 정확도를 보인 50%를 필터링 비율로 사용하여 기존 연구와 비교하였다.

표 5.2. 다양한 필터링 비율에 따른 정확도 비교.

Filtering ratio	Labeled data	Unlabeled data	mAP (%)	Iteration
25%	VOC 2007	VOC 2012 (25%)	72.34±0.08	99,000
50%	VOC 2007	VOC 2012 (50%)	72.39±0.04	80,000
75%	VOC 2007	VOC 2012 (75%)	71.69±0.13	58,000

5.5.2 Uncertainty 기반 반지도 능동적 학습

표 5.3은 제안 반지도 학습과 제안 능동적 학습을 동시에 적용한 실험 결과이다. 실험에서 능동적 학습을 위한 unlabeled 데이터로는 PASCAL VOC 2007를 사용하고, 반지도 학습을 위한 unlabeled 데이터로는 PASCAL VOC 2012 데이터를 사용한다. 성능은 PASCAL VOC 2007 *test*로 평가한다. 초기 학습을 위한 1,000개 labeled 데이터는 PASCAL VOC 2007에서 random sampling 한다. 표에서 *Proposed*는 *baseline* [10]에 4장에서 제안한 mixture 모델링을 적용한

object detection 모델이다. *Random*은 random sampling을 의미하며 $Ours_{AL}$은 4장의 능동적 학습 결과이다. $Ours_{SSAL}$은 제안 반지도 학습과 제안 능동적 학습을 결합한 결과이다. 첫번째 iteration에서 baseline (기존)과 proposed (제안) 모델의 random sampling 결과를 통해, 제안 모델의 정확도가 기존 모델보다 우수함을 확인할 수 있다. 제안 모델에서 제안 능동적 학습 ($Ours_{AL}$)과 random sampling과의 비교를 통해, 제안 능동적 학습의 성능이 random sampling보다 우수함을 알 수 있다. 여기에 반지도 학습까지 적용하게 되면 ($Ours_{SSAL}$), 마지막 iteration (3rd iteration)에서 제안 능동적 학습 방법 대비 정확도가 1.81 mAP 향상되며, 제안 방법을 적용하지 않은 baseline 대비 정확도가 3.47 mAP로 대폭 향상된다.

표 5.3. 제안 반지도 능동적 학습의 정확도 결과.

Model	Method	mAP in % (# labeled images)		
		1st (2k)	2nd (3k)	3rd (4k)
Baseline	*Random*	61.21±0.12	65.49±0.12	67.77±0.10
Proposed	*Random*	62.43±0.10	66.36±0.13	68.47±0.09
	$Ours_{AL}$	62.43±0.10	67.32±0.12	69.43±0.11
	$Ours_{SSAL}$	64.61±0.43	69.66±0.08	71.24±0.02

5.6 본 연구의 결론

본 장에서 object detection의 반지도 학습을 위한 unlabeled 데이터 필터링 방법을 제안하였다. 제안 방법은 localization과 classification uncertainty를 모두 고려한 informativeness score를 사용하여, 모델이 잘 알고 있는 unlabeled 데이터를 필터링한다. 이를 통해, 전체 학습 데이터 수를 크게 줄여 학습 시간을 대폭 감소시켰다. 이뿐만 아니라, 모든 제안 방법을 결합한 제안 반지도 능동적 학습은 능동적 학습 및 반지도 학습을 따로 수행한 것 대비 월등히 높은 정확도 향상을 보였다. 본 장의 제안 방법은 일반적인 object detection 데이터셋인 PASCAL VOC에서 다양한 실험들을 통해 그 우수성을 보였으며, computing과 labeling cost 감소가 필요한 application에 사용되기 적합하다.

제 6 장 결 론

본 논문은 첫째로, 기존 object detection 알고리즘의 false localization 문제를 해결하기 위해, Gaussian distribution 기반 localization uncertainty 모델링 방법과 localization uncertainty를 예측하고 이를 활용하는 방법을 제안하였다. 기존 알고리즘들은 bounding box coordinate에 대한 confidence score를 예측하지 못했기 때문에 잘못된 bounding box 예측에 대응하지 못하는 문제가 있었다. 하지만, 제안 방법은 localization uncertainty를 통해 mislocalization 문제를 해결하였고, baseline 대비 비슷한 처리 속도를 유지하며 정확도를 대폭 향상시켰다.

본 논문은 둘째로, 기존 능동적 학습의 높은 연산량 문제와 localization information을 활용하지 못하는 문제를 해결하기 위해, mixture density network를 사용해 single model에서 single forward pass로 object detection을 위한 uncertainty 예측 방법을 제안하였다. 이를 통해, 기존 multiple-models 방법 대비 연산량을 대폭 감소시켰다. 그리고 능동적 학습에서 localization과 classification task의 aleatoric uncertainty와 epistemic uncertainty를 모두 활용하는 방법을 제안하여, 기존 single model 방법 대비 정확도를 대폭 향상시켰고, multiple-models 방법과 비슷한 정확도를 달성하였다. 즉, 제안 능동적 학습 방법은 정확도와 연산량의 trade-off 측면에서 기존 연구들보다 우수한 성능을 보였다.

본 논문은 셋째로, 기존 반지도 학습의 unlabeled 데이터 학습 비효율성 문제를 해결하기 위해, localization과 classification uncertainty를 모두 고려한 unlabeled subset 데이터 추출 방법을

제안하였다. 제안 방법은 모델이 이미 잘 알고 있는 unlabeled 데이터를 학습 데이터에서 필터링하여, 전체 학습 데이터 수를 크게 감소시켜 기존 대비 학습 시간을 대폭 줄이고 정확도를 향상시켰다.

본 논문은 마지막으로, 제안 능동적 학습 방법과 제안 반지도 학습 방법을 결합한, object detection을 위한 uncertainty 기반 반지도 능동적 학습 모델을 제안하였다. 제안한 반지도 능동적 학습 방법은 기존 및 제안 능동적/반지도 학습 방법들 대비 가장 높은 정확도를 보였다.

본 논문의 제안 방법들은 다양한 object detection 네트워크에 적용 가능하며, object detection을 필요로 하는 다양한 application에서 그 활용도가 높아질 것으로 기대된다.

참고 문헌

[1] A. Bochkovskiy, C-Y. Wang, and H-Y. Mark Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” *arXiv preprint* arXiv:2004.10934, 2020.

[2] J. Choi, D. Chun, H. Kim, and HJ Lee, "Gaussian YOLOv3: An Accurate and Fast Object Detector Using Localization Uncertainty for Autonomous Driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[3] D. Wang, Y. Zhang, K. Zhang, and L. Wang, “FocalMix: Semi-Supervised Learning for 3D Medical Image Detection,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[4] F. Liu, B. Liu, C. Sun, M. Liu, and X. Wang, “Deep learning approaches for link prediction in social network services,” in *Proceedings of the International Conference on Neural Information Processing (ICONIP)*, 2013.

[5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[6] J. Redmon and A. Farhadi, Yolov3: An incremental improvement,” *arXiv preprint* arXiv:1804.02767, 2018.

[7] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[8] S. Choi, K. Lee, S. Lim, and S. Oh, "Uncertainty-aware learning from demonstration using mixture density networks with sampling-free variance modeling," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proccedings of the European Conference on Computer Vision (ECCV)*, 2016.

[11] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, 34(4):743-761, 2012.

[12] B. Settles, "Active Learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2012.

[13] S. Roy, A. Unmesh, and V. P. Namboodiri, "Deep active learning for object detection," in *Proceedings of the British Machine Vision Conference (BMVC),* 2018.

[14] E. Haussmann, M. Fenzi, K. Chitta, J. Ivanecky, H. Xu, D. Roy, A. Mittel, N. Koumchatzky, C. Farabet, and J. M. Alvarez, "Scalable active learning for object detection," in *Proceedings of the IEEE Intelligent Vehicles (IV) Symposium*, 2020.

[15] D. Yoo and I. Kweon, "Learning loss for active learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] O. Sener and S. Savarese, "Active learning for convolutional neural networks: A core-set approach," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[17] C-C. Kao, T-Y. Lee, P. Sen, and M-Y. Liu, "Localization-aware active learning for object detection," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2018.

[18] J. Jeong, S. Lee, J. Kim, and N. Kwak, "Consistency-based Semi-supervised Learning for Object Detection," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[19] O. Chapelle, B. Schölkopf, and A. Zien, "Semi-supervised learning," *IEEE Transactions on Neural Networks*, 20(3):542-542, 2009.

[20] Christopher M Bishop, "Mixture density networks," *Technical Report*. Aston University, Birmingham 1994.

[21] X. Hu, X. Xu, Y. Xiao, H. Chen, S. He, J. Qin, and P-A. Heng,

"Sinet: A scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, 20(3):1010-1019, 2019.

[22] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[23] J. Choi, D. Chun, H-J. Lee, and H. Kim, "Uncertainty-based Object Detector for Autonomous Driving Embedded Platforms," in *Proceedings of the IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2020.

[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[25] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[26] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 1994.

[27] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the International Conference on Research and Development in Information Retrieval*, 1994.

[28] D. Roth and K. Small, "Margin-based active learning for structured output spaces," in *Proceedings of the European Conference on Machine Learning (ECML)*, 2006.

[29] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP)*, 2008.

[30] W. Luo, A. G. Schwing, and R. Urtasun, "Latent structured active learning," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2013.

[31] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of Machine Learning Research*, 2:45-66, 2001.

[32] S. Vijayanarasimhan and K. Grauman, "Large-scale live active learning: Training object detectors with crawled data and crowds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[33] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the Conference on Computational Learning Theory (COLT)*, 1992.

[34] J. E. Iglesias, E. Konukoglu, A. Montillo, Z. Tu, and A. Criminisi, "Combining generative and discriminative models for semantic segmentation of CT scans via active learning," in *Proceedings of the Information Processing in Medical Imaging*, 2011.

[35] H. T. Nguyen and A. Smeulders, "Active learning using preclustering," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2004.

[36] Y. Yang, Z. Ma, F. Nie, X. Chang, and A. G. Hauptmann, "Multi-class active learning by uncertainty sampling with diversity maximization," *International Journal of Computer Vision (IJCV)*, 113(2):113-127, 2015.

[37] E. Elhamifar, G. Sapiro, A. Yang, and S. Shankar Sasrty, "A convex optimization framework for active learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2013.

[38] Y. Guo, "Active instance sampling via matrix partition," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2010.

[39] M. Hasan and A. K. Roy-Chowdhury, "Context aware active learning of activity recognition models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[40] O. Mac Aodha, N. Campbell, J. Kautz, and G. J. Brostow, "Hierarchical subquery evaluation for active learning on a graph," in *Proceedings of the IEEE/CVF International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[41] C. E. Shannon, "A mathematical theory of communication," *Mobile Computing and Communications Review*, 5(1):3-55, 2001.

[42] K. Chitta, J. M. Alvarez, and A. Lesnikowski, "Large-Scale Visual Active Learning with Deep Probabilistic Ensembles, *arXiv*, 2018.

[43] A. JOSHI, "Multi-class active learning for image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[44] X. Li and Y. Guo, "Multi-level adaptive active learning for scene classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[45] W. H. Beluch, T. Genewein, A. N¨urnberger, and J. M. K¨ohler, "The power of ensembles for active learning in image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[46] Y. Gal, R. Islam, and Z. Ghahramani, "Deep bayesian active learning with image data," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2017.

[47] A. Kirsch, J. V. Amersfoort, and Y. Gal, "Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[48] H. H. Aghdam, A. Gonzalez-Garcia, A. M. L´opez, and J. Weijer, "Active learning for deep detection neural networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[49] K. Wang, X. Yan, D. Zhang, L. Zhang, and L. Lin, "Towards human-machine cooperation: Self-supervised sample mining for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[50] N-V Nguyen, C. Rigaud, and J-C Burie, "Semi-supervised object detection with unlabeled data," in *Proceedings of the International conference on computer vision theory and applications*, 2019.

[51] J. Jeong, V. Verma, M. Hyun, J. Kannala, N. Kwak, "Interpolation-based semi-supervised learning for object detection," *arXiv preprint* arXiv:2006.02158, 2020.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[53] X. Dai, "Hybridnet: A fast vehicle detection system for autonomous driving," *Signal Processing: Image Communication*, 70:79-88, 2019.

[54] B. Wu, F. Iandola, P. H Jin, and K. Keutzer, "Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[55] C. Zhang, Y. Liu, D. Zhao, and Y. Su, "Roadview: A traffic scene simulator for autonomous vehicle simulation testing," in

Proceedings of the International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014.

[56] J. Wei, J. M Snider, J. Kim, J. M Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2013.

[57] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[58] Z. Cai, Q. Fan, R. S Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[59] Q. Zhao, Y. Wang, T. Sheng, and Z. Tang, "Comprehensive feature enhancement module for single-shot object detector," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2018.

[60] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z Li, "Single-shot refinement neural network for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[61] S. Liu, and D. Huang, "Receptive field block net for accurate and fast object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[62] A. Marshall, "False positive: Self-driving cars and the agony of knowing what matters," *WIRED Transportation*, 2018.

[63] Y-W Seo, N. Ratliff, and C. Urmson, "Self-supervised aerial images analysis for extracting parking lot structure," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.

[64] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in *Proceedings of the International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[65] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding Box Regression with Uncertainty for Accurate Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[66] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,2012.

[67] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[68] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger,"

in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[69] T-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[70] A. Corovic, V. Ilic, S. Duric, M. Marijan, and B. Pavkovic, "The real-time detection of traffic participants using yolo algorithm," in *Proceedings of the Telecommunications Forum (TELFOR)*, 2018.

[71] T-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2014.

[72] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[73] J. Choi, I. Elezi, H-J. Lee, C. Farabet, and J.M. Alvarez, "Active Learning for Deep Object Detection via Probabilistic Modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021., to be published.

[74] K. Chitta, J.M. Alvarez, E. Haussmann, and C. Farabet, "Less is more: An exploration of data redundancy with active dataset subsampling," *arXiv preprint* arXiv:1811.03542, 2019.

[75] A. Harakeh, M. Smart, and S. L Waslander, "Bayesod: A bayesian approach for uncertainty estimation in deep object detectors," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[76] S. C. Hora, "Aleatory and epistemic uncertainty in probability elicitation with an example from hazardous waste management," *Reliability Engineering and System Safety*, 54:217-223, 1996.

[77] N. Tagasovska and D. Lopez-Paz, "Single-model uncertainties for deep learning," in *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[78] E. Hullermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *arXiv preprint* arXiv:1910.09457, 2019.

[79] D. Feng, X. Wei, L. Rosenbaum, A. Maki, and K. Dietmayer, "Deep active learning for efficient training of a lidar 3d object detector," in *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[80] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal in Computer Vision (IJCV)*, 88(2):303-338, 2010.

[81] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[82] Y. He and J. Wang, "Deep multivariate mixture of gaussians for object detection under occlusion," *arXiv preprint* arXiv:1911.10614, 2019.

[83] A. Varamesh and T. Tuytelaars, "Mixture dense regression for object detection and human pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[84] J. Yoo, G. Seo, and N. Kwak, "Mixturemodel-based bounding box density estimation for object detection," *arXiv preprint* arXiv:1911.12721, 2019.

[85] S. Choi, S. Hong, K. Lee, and S. Lim, "Task agnostic robust learning on corrupt outputs by correlation-guided mixture density networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[86] Y. Kwon, J-H Won, BJ Kim, and MC Paik, "Uncertainty quantification using bayesian neural networks in classification: Application to biomedical image segmentation," *Computational Statistics & Data Analysis*, 142:106816, 2020.

[87] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[88] T. Highlander and A. Rodriguez, "Very efficient training of convolutional neural networks using fast fourier transform and

overlap-and-add," *arXiv preprint* arXiv:1601.06815, 2016.

[89] F. N Iandola, S. Han, M. W Moskewicz, K. Ashraf, W. J Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size," *arXiv preprint* arXiv:1602.07360, 2016.

[90] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Proceedings of the Advances in neural information processing systems (NeurIPS)*, 2017.

[91] Y. Zhu, Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, "Soft proposal networks for weakly supervised object localization," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[92] M. Shi, H. Caesar, and V. Ferrari, "Weakly supervised object localization using things and stuff transfer," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[93] Z. Jie, Y. Wei, X. Jin, J. Feng, and W. Liu, "Deep self-taught learning for weakly supervised object localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[94] J. Wang, J. Yao, Y. Zhang, and R. Zhang, "Collaborative learning for weakly supervised object detection," *arXiv preprint* arXiv:1802.03531, 2018.

[95] Y. Tang, J. Wang, B. Gao, E. Dellandréa, R. Gaizauskas, and L. Chen,

"Large scale semi-supervised object detection using visual and semantic knowledge transfer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[96] Z. Yan, J. Liang, W. Pan, J. Li, and C. Zhang, "Weakly-and semi-supervised object detection with expectation-maximization algorithm," *arXiv preprint* arXiv:1702.08740, 2017.

[97] M. Gao, Z. Zhang, G. Yu, S. O. Arik, "Consistency-based Semi-supervised Active Learning: Towards Minimizing Labeling Cost," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[98] K. Sohn, Z. Zhang, C-L. Li, H. Zhang, C-Y. Lee, and T. Pfister, "A Simple Semi-Supervised Learning Framework for Object Detection," *arXiv preprint* arXiv:2005.04757, 2020.

[99] P. Tang, C. Ramaiah, Y. Wang, R. Xu, and C. Xiong, "Proposal learning for semi-supervised object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2021.

Abstract

Uncertainty-Aware Deep Detection Neural Networks with Probabilistic Modeling

Jiwoong Choi
Electrical and Computer Engineering
The Graduate School
Seoul National University

Object detection combines the localization and classification tasks to classify and localize one or more objects in an image or video data. The development of GPU along with deep learning algorithm has accelerated research of deep learning-based object detection. Recently, deep learning-based object detection achieves a better level of accuracy than humans and has become an essential method in various applications such as autonomous driving systems or unmanned stores.

Although object detection combines both the localization and classification tasks, conventional object detection algorithms rely on classification-based confidence score to estimate object category and location information. Conventional method cannot estimate a confidence score of the localization task and therefore it cannot cope

with false positive caused by mislocalization. Especially, this problem leads to severe accidents in autonomous driving application, thereby, it is essential to resolve this problem. In addition, conventional object detection algorithms require a large amount of labeled data to obtain high accuracy. While crawling a large amount of image data is a trivial task, labeling these data is an expensive and time-consuming activity. An image typically contains multiple objects, and each object requires a category and a bounding box information. To solve this problem, active learning has been studied actively. However, most of active learning studies for object detection are based on multiple models or are straightforward extensions of active learning for classification, hence using only the classification-based information. Thus, these previous methods require high computing costs and fail to utilize localization-based information for active learning, therefore these methods cannot be a complete solution.

As a solution to reduce labeling cost, semi-supervised learning for object detection is also being actively studied. Semi-supervised learning is a method of using labeled data with annotation and unlabeled data consisting only of images in training. In general, the amount of unlabeled data has a much larger than that of labeled data. The conventional semi-supervised learning increases the training time by using all these data during training without grasping the usefulness of unlabeled data, thereby reducing efficiency in training.

This dissertation proposes solutions to the three problems mentioned above. First, this dissertation proposes a method of localization uncertainty modeling and novel localization loss function

to solve the false positive problem caused by mislocalization. By utilizing the predicted localization uncertainty in the inference step, the proposed method significantly reduces the false positive and improves the accuracy. Second, this dissertation proposes a high accuracy and low-cost active learning method for object detection. The proposed method estimates both classification and localization uncertainties with a single model and a single forward pass, and leverages them in active learning, thereby achieving outstanding performance gains in terms of accuracy and computational cost. Third, this dissertation proposes a method of an uncertainty-based unlabeled data filtering for semi-supervised learning. The proposed method significantly reduces the training time and improves the accuracy by selecting only the unlabeled data that helps to improve the performance of the model.

The problems and proposed methods are dealt with in detail in each chapter of this dissertation.

Keywords : Uncertainty, object detection, active learning, semi-supervised learning, deep learning
Student Number : 2017-31914