



선박 탐지를 위한 딥러닝 알고리즘 조사

2021 SETsystem 인턴십 중간발표

김동화

2021.10.15



목차

- 과제 정의
- 객체 탐지
 - 정의 / 모델
- 이미지 분할
 - 정의 / 모델
- 모델 선택
 - 비교 / 결론

I. 과제 정의

RDM에서의 선박 탐지

1) 기존의 접근법

- 객체 탐지 알고리즘 중 Faster R-CNN를 이용한 선박 탐지 (2019)
- 객체 탐지와 이미지 분할에 사용되는 Detectron2 적용

2) 향후 개선 사항

- 최신 모델 적용을 통한 선박 탐지 성능 (정확도, 속도) 개선
- 실시간 탐지 가능성 확인

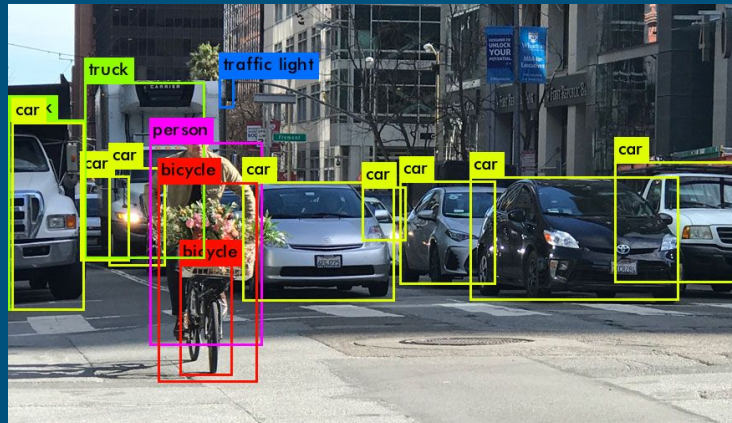
→ 컴퓨터 비전 영역의 객체 탐지(Object Detection)와 이미지 분할(Semantic Segmentation)!

I. 객체 탐지

객체 탐지

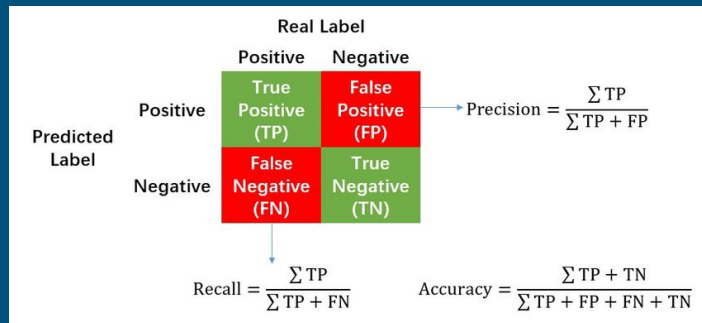
1) 목표

- 분류 : (주어진 분류 중) 물체가 어떤 분류에 속하는 가
- 위치 : 그 물체가 어디에 있는가
 - 박스형 경계로 표시



2) 성능 평가 지표

- mAP (mean Average Precision, 평균 정밀도)
 - Precision (정밀도) : 물체가 존재한다고 예측한 영역에 실제 물체가 존재할 확률
- FPS(Frame per Second, 초당 프레임 수)



객체 탐지 모델

Two Stage

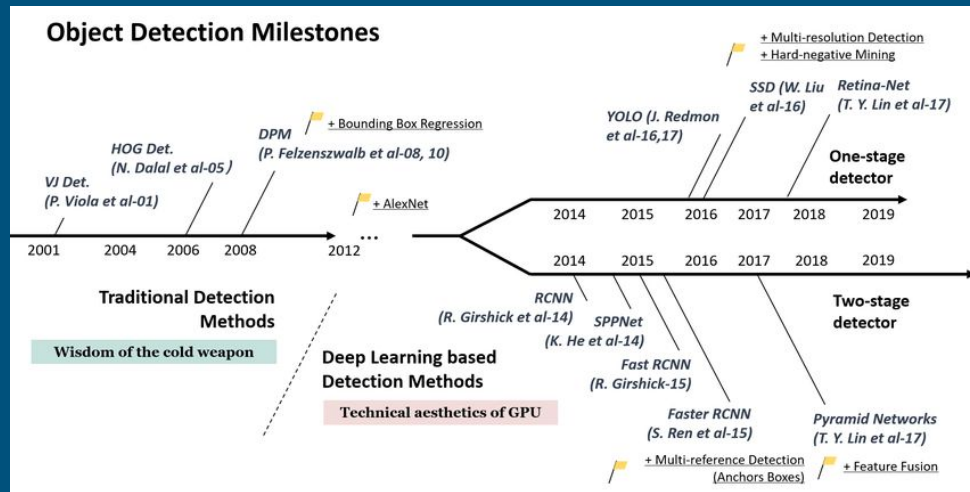
- R-CNN(14) → Fast R-CNN(15) → Faster R-CNN(15) → **Mask R-CNN** (17)
- FPN(Feature Pyramid Net, 17)

One Stage

- YOLO (You Only Live Once, 16) → YOLOv2(17) → YOLOv3(18) → **YOLOv4**(20)
- SSD (Single Shot Multibox Detector, 2016) → RefineDet(18)

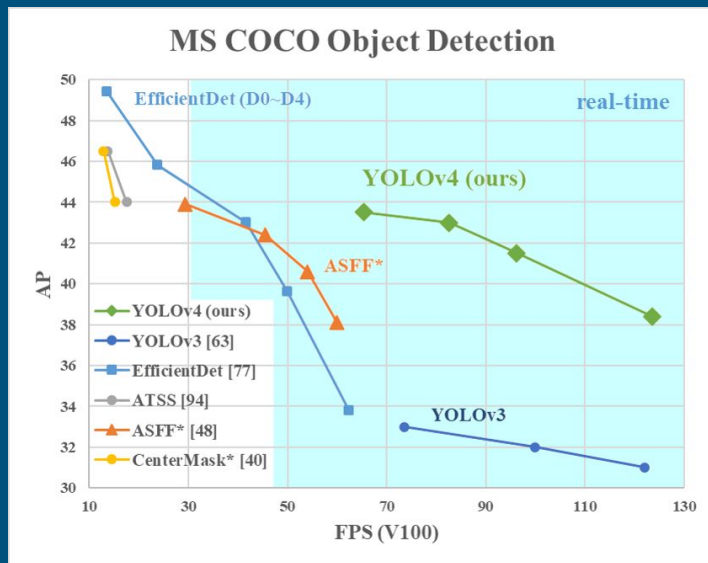
Multi Stage

- Cascade R-CNN(17) → Cascade Mask R-CNN(17)
- **HTC**(Hybrid Task Cascade, 19)
- RetinaNet (21)



YOLOv4 vs Mask R-CNN

- 1) YOLO 기반 모델들이 Mask R-CNN보다 higher-rank
 - 객체 탐지와 실시간 객체 탐지를 위한 데이터셋 모두에서
- 2) FPS(Frames per Second)가 30 이상으로 적절한 수준의 실시간 탐지가 가능할 것으로 예상
- 3) YOLO 모델이 Radar Map에 활용한 사례 O



II. 이미지 분할

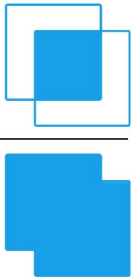
이미지 분할

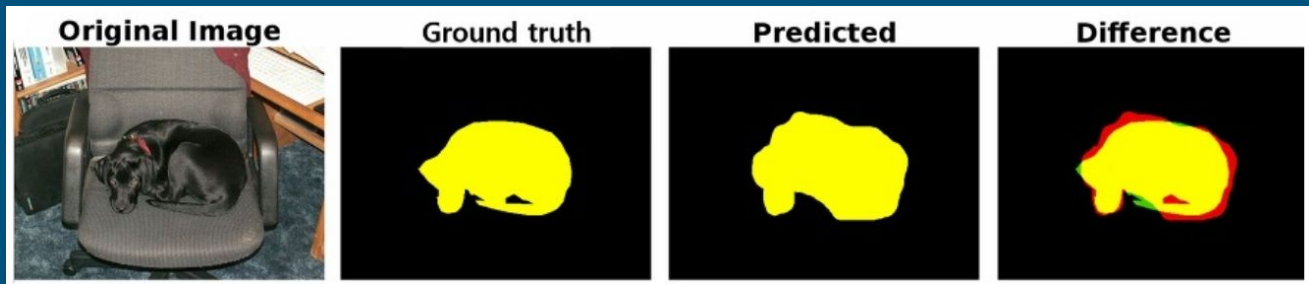
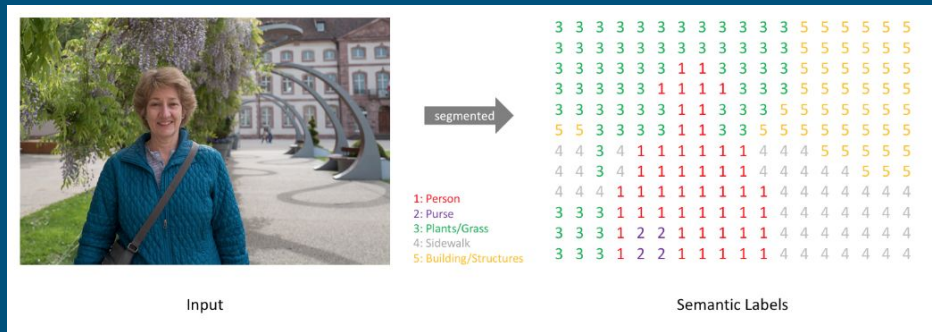
1) 목표

- 픽셀 단위의 이미지 분류

2) 성능 평가 지표

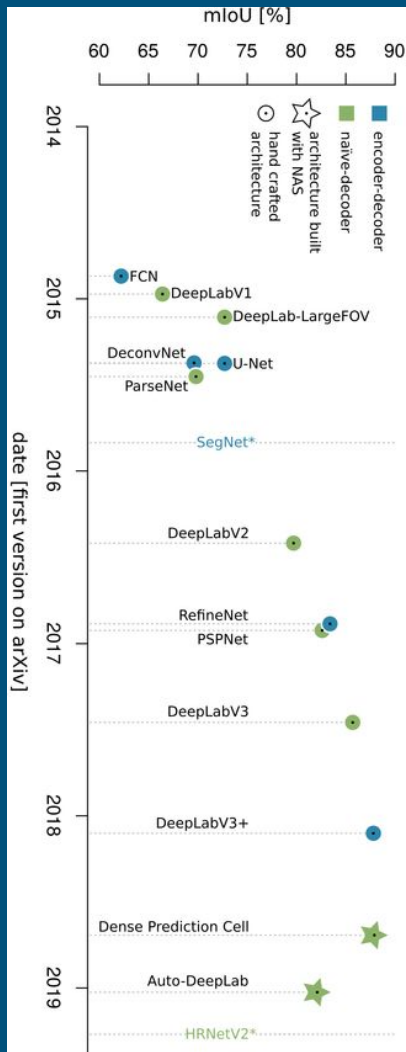
- mIoU (mean Intersection over Union)

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$




이미지 분할 모델

- Mask R-CNN
- FCN (Fully Connected Network)
- U-Net
- DeepLabv3+
- PSPNet
- HRNetV2



이미지 분할 모델

- Mask R-CNN
- FCN (Fully Connected Network)
- U-Net
- DeepLabv3+
- PSPNet
- HRNetV2

- Faster R-CNN 이후에 이미지 분할을 위한 모델 (객체 탐지에도 쓰임)
- 다양한 데이터셋에 적용 가능
- backbone에 따라서 성능 차이 O, 적용 과제 차이 O
- Detectron2과 동일한 성능보일 가능성
- RMD 적용한 사례 X

이미지 분할 모델

- Mask R-CNN
- FCN (Fully Connected Network)
- U-Net
- DeepLabv3+
- PSPNet
- HRNetV2

이미지 분할에서 두번째로 논문수가 많음

다른 모델에 비해서 성능은 낮지만 RDM에
사용된 사례가 있음 (feat. U-Net)

다만 backbone에 따라서 성능차이가
있어서 이를 실험적으로 확인하거나 가장
성능이 좋았던 모델을 사용해야함



(a): Detection at a far distance

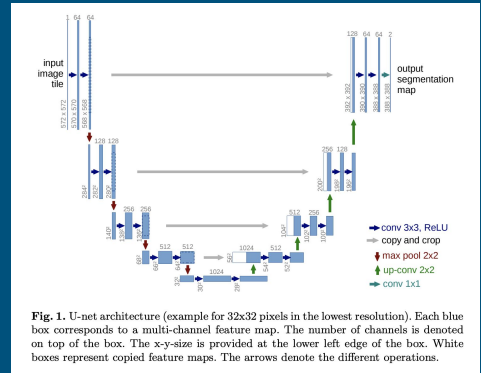
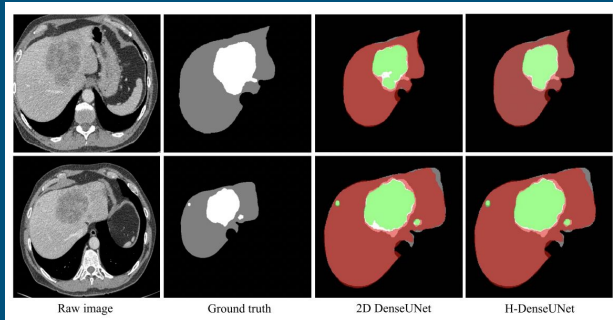


(b): Detection at a close distance

Fig. 6. Demonstration of two tested examples by applying the trained FCN on the validation dataset. The yellow box in the range-doppler spectrums indicates the cell positions for the detected car. The green circle in the images represents the estimated car orientations from the FCN.

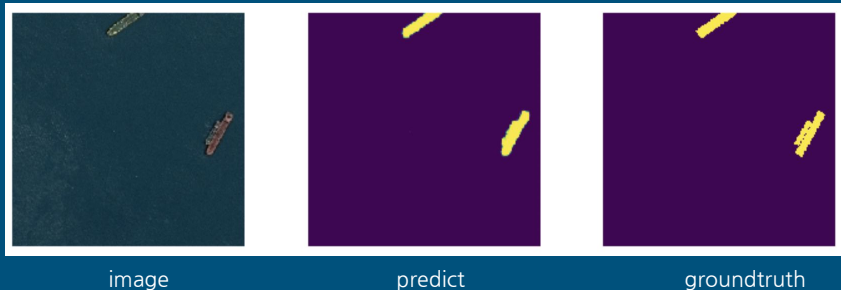
이미지 분할 모델

- Mask R-CNN
- FCN (Fully Connected Network)
- **U-Net**
- DeepLabv3+
- PSPNet
- HRNetV2



- Kaggle Airbus Ship Detection Challenge에서 우승한 알고리즘
- 모델 구조가 U자 형태여서 붙여진 이름, U-Net
- 이미지 분할 분야에서 가장 많은 논문 수
- 훈련할 이미지 데이터가 많지 않고, 이미지의 크기가 큰데(최소 512X512) 반해 탐지하고 싶은 물체가 작을 경우에 효과적
- biomedical image에 많이 쓰임

우리 데이터 사이즈가 512X256이라서... 탈락!



이미지 분할 모델

- Mask R-CNN
- FCN (Fully Connected N)
- U-Net
- DeepLabv3+
- PSPNet
- HRNetV2

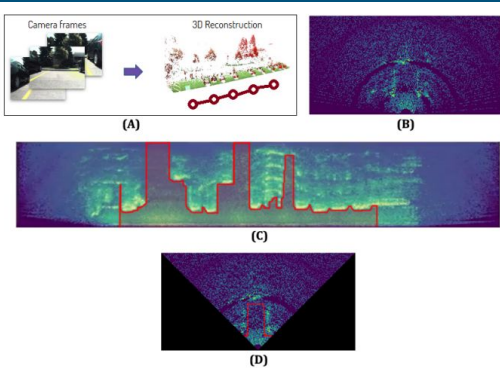


Fig. 2. Ground truth generation pipeline. (A) Camera frames are used to reconstruct the scene and extract odometry. (B) Cartesian DoA from a single frame. (C) Accumulated Cartesian DoA which is used to generate initial annotations. (D) Annotations are distributed to the corresponding frames, cropped for the radar's field of view, and are manually adjusted.

	cityscapes val		Pascal VOC val		
	ResNet50	ResNet101	ResNet50	ResNet101	
DeepLabV3	79.32	80.2	77.68	80.62	
DeepLabV3+	80.09	80.97	76.81	78.62	
FCN	78.8	80.65	72.9	76.24	
OCRNet	79.47	81.35	74.98	79.98	
PSPNet	78.55	79.76	77.29	80.76	
std	0.6023537	0.62580348	1.9855654	1.88060628	
mean	79.246	80.586	75.932	79.244	

	standardization				sum
DeepLabV3	0.1228514	-0.61680704	0.8803537	0.73167893	1.118077061
DeepLabV3+	1.4011700	0.61361115	0.4421914	-0.3318078	2.125164766
FCN	-0.7404287	0.10226852	-1.5270209	-1.5973572	-3.762538373
OCRNet	0.3718745	1.22083053	-0.4794604	0.39136315	1.504607797
PSPNet	-1.1554672	-1.31990316	0.6839361	0.80612301	-0.9853112504

- 최근 이미지 분할 분야에서 성능이 가장 좋은 세가지 모델
- *PADDLESEG: A HIGH-EFFICIENT DEVELOPMENT TOOLKIT FOR IMAGE SEGMENTATION (2021)*
- 위의 논문에서 이미지 분할을 수행하는 알고리즘을 모은 toolkit을 제공
- 각 알고리즘별 성능 지표를 비교한 결과, **Deeplabv3+**의 수행 결과가 가장 좋았다
- DeepLabv3+가 레이더 이미지에 적용된 사례 O

* 데이터셋은 이미지 분할에서 널리 사용되는 cityscapes val 과 PASCAL VOC val 을 사용하였다

III. 모델 선택

객체 탐지? 이미지 분할?

출력 결과물을 박스형 경계? 또는 픽셀단위 분할?

객체 탐지 = 박스형 경계	이미지 분할 = 픽셀 단위
(-) 디테일한 분류가 어렵다.	(+) 디테일한 분류가 가능하다 특히나, 브래그 영역에서 거리가 먼 부분의 시선 속도를 정확하게 측정할 수 있을 것으로 예상된다.
(+) 선박의 위치를 경계 박스의 중심으로 잡으면 되기 때문에 계산 방식이 명확하다.	(-) 비정형적인 형태로 객체가 탐지되면 중심점 계산이 어렵다
(-) 노이즈 영역을 잡아내기 어렵다.	(+) 노이즈를 새로운 카테고리 설정하여서 노이즈가 심한 픽셀을 탐지할 수 있다
(+) 실시간 탐지를 시도해볼 수 있다	(+) 이미지 분할 방식을 RDM에 새롭게 적용해볼 수 있다

결론

- 객체 탐지에서는 YOLOv4를 사용
- 이미지 분할에서는 RDM에 적용한 사례가 있는 FCN을 사용하면 좋겠지만 detectron 2보다 좋은 성능을 낼지는 미지수
- 이미지 분할에서는 레이더 이미지에 적용한 사례가 있고, 성능이 가장 좋은 DeepLabv3+를 사용

감사합니다



피드백

- 1) 객체 탐지보다 이미지 분할로
- 2) RDM상에서의 선박탐지는 일반적인 실시간 탐지에서 사용하는 스케일보다 더 길다. 따라서 일반적인 실시간 탐지에 사용하지 못한다고 하더라도 모델을 배제하지 말기. 웬만하면 가능하다.
- 3) 속도의 기준을 낮추되, 정확도를 높은 모델을 선택하기
- 4) 더 다양한 성능 지표가 있는지 알아보기
- 5) Detectron2, U-Net, DeepLabv3+ 구조와 특징, 장단점, 적용 사례 등을 표로 정리

*참고 : 모델 구현을 위한 세팅 작업 : 삽질하느라 시간이 오래 걸린다. 그런 과정도 겪어야하지만 하다가 모르겠으면 미리 해본 사람에게 물어보자.

참고문헌

[] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in Proc. Int. Conf. Medical Image Comput. Comput.-Assisted Intervention, 2015, pp. 234-241. [[Google Scholar](#)]

[] Li, X., Chen, H., Qi, X., Dou, Q., Fu, C.-W., Heng, P.A.: H-DenseUNet: hybrid densely connected UNet for liver and liver tumor segmentation from CT volumes. arXiv preprint [arXiv:1709.07330](#) (2017) [[Google Scholar](#)]

[] Kaggle Airbus Ship Detection Challenge “Unet34 (dice 0.87+)” [[Kaggle Notebook](#)]