

YOLO v5를 이용한 U-Net3+모델 기반 Semantic-Segmentation 성능 개선

조국한*, 송영준**

Performance Improvement of Semantic-Segmentation based on U-Net3+ Model using YOLO v5

Guk-Han Jo*, Young-Joon Song**

이 연구는 금오공과대학교 대학 연구과제비로 지원되었음(2022~2024)

요약

최근 영상처리 딥러닝 기술의 발달로 객체인식, 이미지 분할은 다양한 분야에 사용되고 있다. 하지만 객체 인식, 이미지 분할을 학습하기 위한 데이터 생성은 쉬운 일이 아니다. 따라서 본 논문에서는 추가적인 데이터 생성 없이 YOLO(You Only Look Once), U-Net3+기반의 모델에서 효율적인 학습이 가능하도록 하는 시스템을 제안 및 구현한다. 제안된 시스템은 YOLO v5의 v5s 모델을 사용하여 1차적으로 객체인식을 수행한다. 이후 YOLO를 거쳐 출력된 객체인식 데이터는 U-Net3+기반의 모델에 입력데이터로써 사용된다. 제안된 방법을 평가하기 위해 기존의 U-Net3+기반으로만 작성된 시스템과 비교하였다. 실험 결과 제안된 방법이 평균 0.1 이상의 향상된 mIoU 결과를 보였다. 이는 제안된 방법이 데이터 생성 없이 효과적으로 이미지 분할의 성능을 개선시킬 수 있음을 확인할 수 있었다.

Abstract

Recent advancements in deep learning technology for image processing have enabled the use of object detection and segmentation in various fields. However, data labeling and generation for learning object detection and segmentation are not easy tasks. Therefore, this paper proposes and implements a system that enables efficient training in a You Only Look Once(YOLO) and U-Net3+ based model without the need for additional data generation. The proposed system performs initial object detection using YOLO v5s. The object detection data output by YOLO v5s is then used as input for a U-Net3+ based model. To evaluate the proposed method, we compared it with a system based solely on the existing U-Net3+ model. Experimental results showed that the proposed method achieved an improvement of over 0.1 in mean Intersection over Union(mIoU) on average. This demonstrates that the proposed method can effectively improve segmentation performance without generating additional data.

Keywords

semantic-segmentation, YOLO, U-Net3+, object detection, mIoU

* 국립금오공과대학교 전자공학과 박사과정
- ORCID: <https://orcid.org/0000-0001-7077-3074>
** 국립금오공과대학교 전자공학부 교수(교신저자)
- ORCID: <https://orcid.org/0009-0003-6872-9404>

• Received: Aug. 12, 2024, Revised: Sep. 13, 2024, Accepted: Sep. 16, 2024
• Corresponding Author: Young-Joon Song
Dept. of EE Kumoh National Inst. of Tech., Korea
Tel.: 82+54-478-7486, Email: yjsong@kumoh.ac.kr

I. 서 론

휴대폰, 블랙박스, 카메라 등과 같이 영상 정보 수집 기기가 발달함에 따라 다양한 이미지 데이터들을 학습하는 CNN(Convolutional Neural Network)이 각광받고 있으며 CNN을 기반으로 하는 많은 논문들이 현재에도 지속적으로 제안되고 있다. 예를 들어 object detection을 위한 Regions with Convolutional Neural Networks features, FCN(Fully Convolutional Networks for Semantic Segmentation), YOLO(You only look once) 등이 있으며, segmentation에 특화된 모델 U-Net, DeepLab 등이 있다[1]-[4]. 이 제안된 모델들은 의료, 자율주행, 보안 카메라 등과 같은 다양한 산업 분야에 활용되고 있으며 이에 따라 다양한 오픈 데이터셋들이 제공되고 있다. 이렇게 다양한 CNN 기반 모델 중에서 본 논문에 기반이 되는 U-Net는 의료 분야에서 segmentation을 위해 주로 사용되며 YOLO는 객체 탐지에 사용된다. 예를 들어 U-Net을 이용하여 장기의 segmentation 검출과 객체를 인식하는 방법이 제안되었으며, YOLO를 사용한 객체인식에서는 CCTV 영상 내 군중 밀집도 분석 방법 및 자율주행을 위한 융복합 영상 식별 시스템 개발 등 지속적으로 다양한 기술들이 제안되고 있다[5]-[8]

본 논문에서는 제한된 데이터를 이용하여 도로환경에서 차량의 형태, 위치를 더 정밀하게 추정할 수 있는 방법을 연구하였다. 데이터는 도로환경 오픈 데이터인 BDD100K(Berkeley Deep Drive 100K)[9]를 사용하였다. 이 데이터를 YOLO를 통해 사전처리하고 이렇게 처리된 결과를 U-Net3+[10]의 입력으로 사용하여 U-Net3+ 단일 모델과 제안한 방법의 성능을 비교 분석한다.

YOLO v5[11]는 2020년 6월에 발표된 객체 인식 알고리즘이며 파이토치로 구현된 CSPNet(Cross Stage Partial Network) 기반 백본과 헤드로 구성되어 있다. 백본은 이미지의 특징맵 추출을 수행하며 헤드는 추출된 특징맵을 기반으로 클래스를 예측하거나 bounding box 예측을 수행한다. U-Net[2]은 2015년에 biomedical 분야에서 segmentation을 목적으로 제안된 모델이며 모델의 구조가 U자 형태로 되어 있다. U-Net은 CNN을 기반으로 제안된 모델이며

FCN의 확장 개념으로도 볼 수 있다. U-Net은 기본적으로 인코더에 해당하는 수축경로(Contracting path)와 디코더부분에 해당하는 확장경로(Expanding path)로 구성되어 있다. U-Net3+는 2020년에 U-Net, U-Net++를 발전시킨 모델로 U-Net++에서 발생하는 메모리 자원, 파라미터의 증가를 full scale skip connection을 통해 해결하는 모델이다. 이전에 제안한 YOLO를 통해 병렬로 U-Net을 연결하여 segmentation을 수행하는 방법[12]의 단점인 파라미터 수 감소와 추가적인 성능 개선을 위해 병렬구조를 제거하고 모델을 U-Net3+기반의 모델로 변경하여 실험을 진행하였다.

본 논문의 구성은 다음과 같다. 2장에서는 제안한 모델의 전체적인 구성을 설명하고 2.1에서는 YOLO v5를 통해 데이터를 생성하는 방법 2.2에서는 U-Net3+의 구조와 학습 방법에 대해 설명한다. 3장에서는 실험에 사용된 장비, 학습데이터에 대해 설명하고 기존 U-Net3+를 기반으로 구성된 모델을 통해 학습하는 방법과 제안한 방법의 결과를 비교한다. 4장에서는 결론에 대해서 기술하고 개선방향에 대하여 설명한다.

II. 관련 연구

Segmentation은 의료, 제조공정, 자율주행 분야에 주로 활용되고 있으며 크게 객체와 배경을 분리하여 위치와 형태를 예측하는 방법이다. 따라서 객체인식에서 사용되는 bounding box의 방법으로는 확인하기 어려운 객체의 형태를 파악할 수 있다는 장점이 있으며 픽셀단위로 클래스를 분류하기 때문에 더 세분화된 정보를 제공할 수 있다. 대표적인 segmentation 모델로는 U-Net, U-Net3+, DeepLab 등이 있다.

2.1 U-Net과 U-Net++

U-Net[2]은 semantic segmentation에 특화된 모델로 전체적인 구조는 U자의 형태로 구성되어 있다. U-Net은 크게 수축경로(Contracting path), bottle neck, 확장경로(Expanding path)로 나뉜다. 수축경로는 인코더의 역할을 담당하며 특징맵을 추출하거나 디코더 특징 값에 concat을 통한 위치 정보 전달의 역할

을 수행한다. Contracting step마다 double convolution과 maxpooling이 사용하여 의미정보를 추출하며 contracting step과 동일한 크기의 expanding step에 위치정보를 전달한다.

bottle neck은 수축경로에서 확장경로로 전환되는 중간지점이며 확장경로는 수축경로에서 추출한 특징맵을 높은 해상도로 전환하는 역할을 담당한다. 주로 upsampling을 적용하여 원래의 입력이미지에 해당하는 크기로 변환하며 이 과정에서 수축경로를 통해 위치정보를 결합하여 이미지를 확장한다.

U-Net++는 U-Net의 확장 개념이며 크게 2가지의 차이점이 존재한다. 첫 번째는 Re-designed skip pathways이며 중첩된 고밀도 합성곱 블록을 통해 연결된 인코더와 디코더로 구성되었다. 두 번째는 모든 분할 분기의 출력을 평균하여 최종결과로 사용한 부분이다. 그 결과 U-Net++는 U-Net에 비해 평균 3.4~3.9의 IoU의 향상되었다.

2.2 U-Net3+

U-Net3+[8]는 기존의 full scale connection을 통해 U-Net, U-Net++의 성능과 파라미터를 개선한 모델이다. 기존의 U-Net++에서는 레이어가 깊어짐에 따라 down sampling과 up sampling의 반복으로 정보가 희석될 수 있다는 단점이 존재한다. 이러한 단점을 U-Net3+는 full scale과 hybrid loss 방식으로 개선했다. 여기서 full scale은 low-level의 정보들을 다른 scale에 있는 high-level과 합치는 것을 의미한다.

2.3 DeepLab

DeepLab은 구글에서 제시한 segmentation 모델로 현재 v3+까지 제안되어 있다. v1에서는 atorus convolution을 처음 적용하여 세부적인 특징들을 추출하는 성능을 향상 시켰으며 v2에서는 atrous spatial pyramid pooling(ASPP)를 제안하여 멀티 스케일에서도 적용할 수 있게 하였다. v3에서는 기존의 ResNet 구조에 Atrous convolution를 활용하였으며 v3+에서는 depthwise separable convolution과 atorus convolution를 결합한 atrous separable convolution가 제안되었다. 이러한 segmentation 기술을 활용하여 segmentation의 성능을 높이기 위한 [13][14] 등 과 같은 많은 논문들이 제안되었다.

III. 시스템 개요

3.1 제안한 시스템 구성

그림 1은 YOLO v5와 U-Net3+를 이용한 도로환경 segmentation 시스템의 전체적인 구성을 나타낸 것이다. 첫 번째로 데이터를 수집하고 수집된 데이터들을 사전에 학습이 완료된 YOLO 모델을 사용하여 object detection 과정을 우선적으로 적용한다. YOLO를 통해 object detection이 완료된 이미지를 즉 bounding box가 적용된 이미지를 자동으로 U-Net3+의 입력 데이터로 사용하여 segmentation 학습을 수행한다. 이렇게 학습된 모델을 U-Net3+만 적용된 모델과 비교하여 성능을 테스트한다.

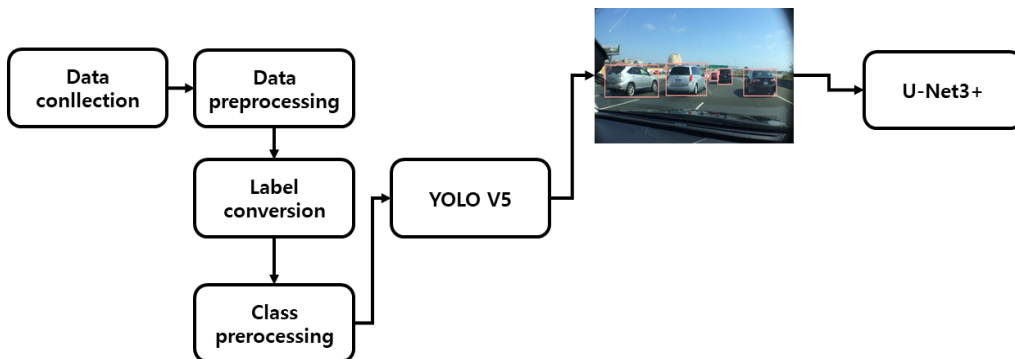


그림 1. YOLO와 U-Net3+를 활용한 Segmentation 시스템 구성
 Fig. 1. Segmentation system configuration using YOLO and U-Net3+

3.2 Segmentation 학습을 위한 데이터 생성

본 논문에서 semantic segmentation을 위한 학습 이미지 생성을 위해 [12][15] 논문에서 사용한 YOLO v5의 모델과 데이터를 거의 동일하게 구성하였다.

YOLO v5의 기본모델 v5s, v5m, v5l, v5x 중 v5s 모델을 사용하여 이미지 데이터를 생성하였다. 각 모델들은 depth_multiple과 width_multiple의 크기에 따라 성능의 차이가 발생하지만 모델이 깊어질수록 학습에 오랜 시간이 소요되기 때문에 가장 수행시간이 빠른 v5s를 선정하였다.

학습을 위해 사용된 로우 데이터는 도로환경 오픈 데이터셋인 BDD100K를 사용하였다. 총 8만개의 도로환경 데이터를 이미지로 제공하고 있으며 이중 7만개의 jpg 이미지 데이터를 학습데이터로 사용하고 1만개의 데이터를 검증 데이터로 사용하였다. 라벨과 물체의 종류에 관한 데이터는 json 파일로 제공되는데 이를 YOLO에 활용하기 위해서는 YOLO 라벨에 맞는 txt 파일로 변환하는 과정이 필요하다. 변환 과정 중 json 파일에서 누락된 부분이 존재하여 이를 삭제하였다. 따라서 총 학습에 사용된 데이터는 69,863개이다.

BDD100K의 json 파일에서는 총 10개의 클래스에 대한 클래스 좌표를 제공하는데 본 논문에서는 차량 및 보행자의 segmentation 정보만을 필요로 하기 때문에 이와 관련 없는 8개의 클래스는 제외하여 이미지 데이터를 생성하였다.

3.3 U-Net3+기반 모델 구성

그림 2는 U-Net3+를 기반으로 구성된 학습 모델의 구성을 나타낸다. 전체적인 구조는 U-Net3+와 동일하지만 학습에 소모되는 시간을 줄이기 위해 supervision by ground truth 과정을 삭제하였다.

①과 ④는 convolution layer와 batchnormalization으로 구성되어 있으며 ②는 x_{de}^1 와 크기를 맞춰주기 위해 maxpooling이 ①의 앞에 추가 되었다. ③은 ②와 비슷하지만 ②에서는 피쳐 크기를 줄이기 위해 maxpooling이 추가된 반면 ③은 upsampling이 추가 되었다.

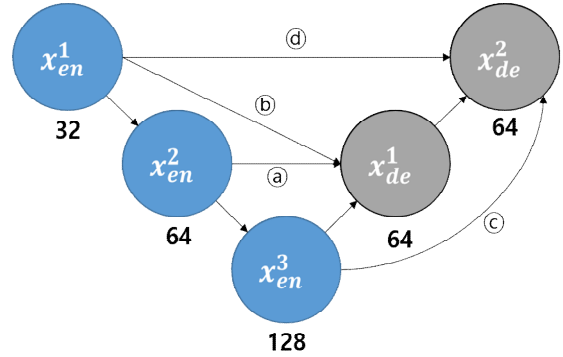


그림 2. U-Net3+기반 모델 구성
Fig. 2. Model configuration based on U-Net3+

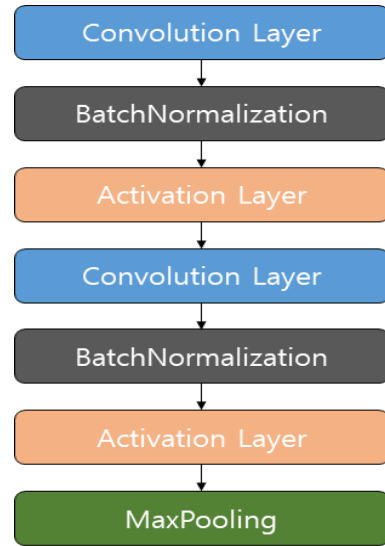


그림 3. x_{en} 레이어 구성
Fig. 3. x_{en} layer configuration

그림 3은 x_{en} 레이어 구성을 나타낸 것이며, 총 2개의 convolution layer와 1개의 maxpooling layer로 구성되어 있다. 각 레이어마다 배치 정규화를 적용함으로써 피쳐 데이터 값의 범위를 조정하도록 하였다. x_{en} 는 contracting path에 사용하였으며 필터의 개수를 제외한 다른 조건은 모두 동일하게 구성하여 총 3개의 x_{en} 를 사용하였다. 필터의 개수는 U-Net 특징에 따라 점차 늘어나는 형태로 사용하였다.

그림 4는 x_{de} 의 레이어 구성을 나타낸 것이며, 1개의 upsampling과 1개의 convolution layer로 구성되어 있다. U-Net3+기반으로 모델을 구성했기 때문에 full scale skip connection로 각 레이어들을 연결하였다. full scale connection을 구성할 때 x_{en}^2 는 x_{de}^1 와 크기가 동일하기 때문에 convolution layer만 적용하

여 연결하면 되지만 x_{en}^1 은 x_{de}^1 보다 높은 수준의 크기를 가지고 있어 maxpooling layer를 적용하여 크기를 줄인 다음 연결하였다.

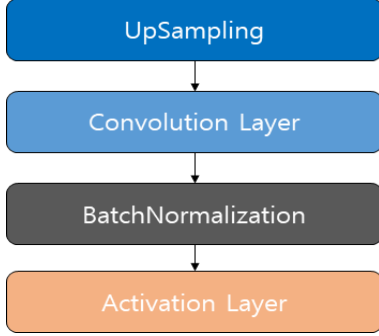


그림 4. x_{de} 레이어 구성
Fig. 4. x_{de} layer configuration

또한 x_{en}^3 의 정보를 x_{de}^2 와 연결하여 모델을 구성하였다. 이를 수식으로 나타내면 다음 식 (1)과 같다. 식 (1)에서 함수 $C(\cdot)$ 는 컨볼루션 연산을 나타내고, $H(\cdot)$ 은 컨볼루션과 일괄 정규화 및 ReLU 활성화를 통해 특징을 집계하는 역할을 한다. $D(\cdot)$ 와 $U(\cdot)$ 는 각각 업샘플링과 다운샘플링 동작을 나타내고, $[\cdot]$ 는 concatenation을 나타낸다[16]. U-Net3+에서는 기존 U-Net에서 사용하지 않는 새로운 손실함수가 사용되는데 이 손실함수를 hybrid loss function이라고 부르며 총 3가지의 손실함수들의 결합이라고 볼 수 있다. 각 손실함수는 focal loss, MS-SSIM loss, IoU loss로 구성되어 있다[10].

$$X_{De}^i = \begin{cases} X_{En}^i, & i = N \\ H\left(\left[\underbrace{C(D(X_{En}^k))_{k=1}^{i-1}}_{Scales: 1^{th} \sim i^{th}}, \underbrace{C(U(X_{De}^k))_{k=i+1}^N}_{Scales: (i+1)^{th} \sim N^{th}}\right]\right), & i = 1, \dots, N-1 \end{cases} \quad (1)$$

$$l_{seg} = l_{fl} + l_{ms-ssim} + l_{iou} \quad (2)$$

$$l_{fl} = -\sum_{i=1}^n a_i (1-p_i)^\gamma \log(p_i) \quad (3)$$

$$l_{ms-ssim} = 1 - \frac{(2\mu_p\mu_g + c_1)(2\sigma_{pg} + c_2)}{(\mu_p^2 + \mu_g^2 + c_1)(\sigma_p^2 + \sigma_g^2 + c_2)} \quad (4)$$

식 (2)~(4)에서 l_{seg} 는 hybrid loss이며, l_{fl} 은 Focal loss, $l_{ms-ssim}$ 은 MS-SSIM loss, l_{iou} 는 IoU loss를 나타낸 것이다[10]. l_{fl} 의 수식에서 a_i 는 분류 클래스에 더 큰 가중치를 부여하기 사용되며, p_i 는 각 클래스에 대한 모델의 추정확률, $(1-p_i)^\gamma$ 은 변조인자로 확률 값이 크면 낮은 가중치, 확률값이 작으면 높은 가중치를 부여한다. $l_{ms-ssim}$ 의 수식에서 p 는 분할 결과 마스크의 패치 영역, g 는 정답 마스크의 패치 영역을 의미한다. 또한 μ 는 평균값, σ 는 분산값을 나타낸다[17]. 이와 같이 물체 검출 분야에 주로 사용되는 IoU loss와 부류 불균형이 심한 경우 주로 사용되는 focal loss, 두 개의 이미지 비교에 사용되는 MS-SSIM loss를 합쳐서 사용함으로써 기존 U-Net보다 더 세밀하게 손실 값을 조정할 수 있게 되었다.

IV. 실험 결과

표 1은 실험에 사용된 CPU, GPU, Memory 성능을 나타낸 것이다. CPU는 i9-10920X를 사용했으며 GPU는 Titan RTX를 사용하여 segmentation 학습을 수행하였다. 학습 횟수는 100번 배치 사이즈는 100으로 설정하였다.

표 1. 실험 장비

Tabel 1. Equipment used in experiment

Device	CPU	GPU	Memory
Company	Intel	Nvidia	Samsung
Model name	i9-10920X	Titan RTX	PC4-21300

제안한 segmentation 예측 시스템의 성능 평가를 위해서 본 연구에서는 기존에 U-Net3+를 기반으로 작성된 동일한 모델 2개를 준비하고 하나는 YOLO를 거치지 않은 모델과 하나는 YOLO v5를 거쳐 U-Net3+에 학습한 결과의 mIoU 결과를 측정한다.

그림 5는 각 모델의 training loss의 결과를 나타낸 것이다. (a)는 기존 U-Net3+ 기반의 모델로만 학습한 것이고 (b)는 YOLO v5를 적용하여 U-Net3+를 학습한 손실 값이다. 학습 횟수와 배치 사이즈 모두 두 모델 동일하게 구성하였다. 그림을 살펴보면 (a)보다 (b)의 손실값이 더 낮은 것을 확인할 수 있다.

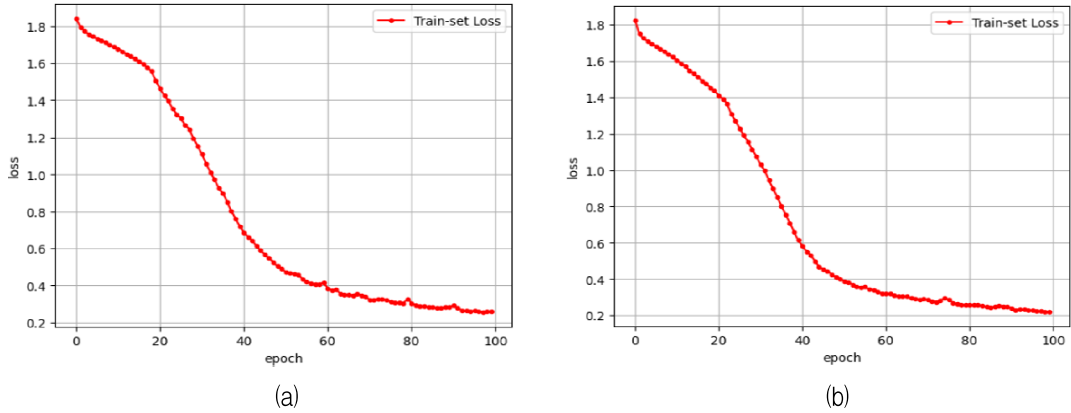


그림 5. U-Net3+기반 모델과 제안한 모델의 학습 손실값
 (a) U-Net3+ 모델 기반 시스템의 학습 손실값, (b) YOLO-U-Net3+ 모델 기반 시스템의 학습 손실값
 Fig. 5. Training loss of UNet3+ based model and the proposed model
 (a) U-Net3+ based system loss, (b) YOLO-U-Net3+ based system loss



그림 6. 이미지 분할 라벨 데이터
 Fig. 6. Segmentation ground truth

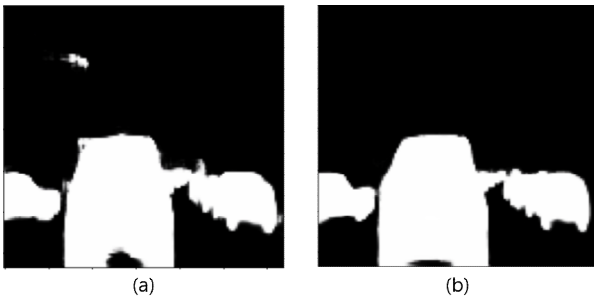


그림 7. U-Net3+기반 모델과 제안한 모델의 결과
 (a) U-Net3+ 의미론적 분할, (b) YOLO-U-Net3+ 의미론적 분할
 Fig. 7. Results of U-Net3+ based model and the proposed model (a) U-Net3+ segmentation, (b) YOLO-U-Net3+ segmentation

그림 6은 학습데이터의 라벨에 해당하는 ground truth의 이미지를 나타낸 것이다. segmentation 학습 시 학습 결과가 얼마나 ground truth와 유사한지에 따라 학습이 올바르게 되었는지 확인할 수 있다. 그림 7은 U-Net3+기반 모델과 제안한 모델의 segmentation 결과를 나타낸 것이다.

(a)는 YOLO 데이터를 U-Net3+기반 모델 학습에 사용하지 않은 결과이며, (b)는 YOLO 데이터를 사용한 결과이다. 결과를 살펴보면 육안으로 보았을 경우에도 (a)보다 (b)가 더 정확하게 예측한 것을 확인할 수 있다.

이러한 결과를 수치적으로 표현하기 위해 segmentation과 객체 인식의 성능 평가 지표로 많이 사용되는 mIoU를 이용하여 mIoU를 측정하였다. mIoU 식 (5)과 같다.

$$mIoU = \frac{1}{N_t} \sum_{i=1}^{N_t} \frac{|b_i \cap b_i^{gt}|}{|b_i \cup b_i^{gt}|} \quad (5)$$

N_t 는 테스트 데이터의 이미지 개수이며 b_i 는 i 번째 prediction instance segmentation이며 b_i^{gt} 는 i 번째 ground truth이다. 표 2는 U-Net3+기반 시스템과 제안한 시스템의 mIoU 결과를 측정한 것이다. threshold의 임계값은 동일하게 0.005로 설정하였다. 표 2의 측정 결과를 확인해보면 YOLO를 적용한 모델이 U-Net3+만 적용한 모델보다 더 좋은 mIoU를 가짐을 알 수 있다. 동일 조건에서 제안한 시스템의 신뢰성을 확인하기 위해 초기상태에서 여러 횟수의 학습을 시도하였는데 제안한 시스템은 YOLO에서 데이터를 생성하는 시간이 추가적으로 발생한다. 표 2의 Data creation time은 데이터 100개 당 평균 생성 시간을 나타낸 것이며 제안한 시스템이 단일 U-Net3+ 모델보다 생성시간이 6~7초 정도 더 소요되는 것을 확인하였다.

표 2. U-Net3+기반 모델과 제안한 모델의 성능 비교

Table 2. Performance comparison between U-Net3+ based model and the proposed model

Model	Performance	
	mIoU	Data creation time(sec)
U-Net3+ based system	0.538	11.3
YOLO-U-Net3+ based system	0.692	18.1

하지만 제안한 시스템이 기존의 U-Net3+기반으로 작성된 시스템보다 0.02~0.2 정도의 mIoU 향상을 보였다. 따라서 제안한 시스템이 YOLO를 통해 출력된 이미지가 segmentation 학습을 수행할 때 물체의 영역을 제한함으로써 더 좋은 성능의 segmentation 결과를 가지는 것을 확인할 수 있다.

V. 결론 및 향후 과제

일반적으로 모델의 정확도를 높이기 위해서는 모델의 레이어를 적층하거나 학습 데이터의 양을 늘리는 방법을 많이 사용하지만 데이터의 양을 늘리기 위해서는 라벨링 작업이 필요한데 이는 데이터 생성에 많은 시간이 소요된다. 본 논문에서는 같은 양의 데이터를 효과적으로 활용하도록 객체인식 결과를 segmentation에 활용하는 시스템을 구현하였다. 제안한 시스템은 YOLO v5s의 모델을 사용하여 BDD100K 데이터를 학습하였다. 이렇게 학습된 모델을 통해 bounding box가 포함된 객체인식 이미지를 U-Net3+의 입력으로 사용하여 semantic segmentation을 수행하였다. 그 결과 YOLO를 사용하지 않은 segmentation 시스템에서는 0.538, 제안한 시스템에서는 0.692의 mIoU를 얻어 제안한 시스템의 성능이 기존 U-Net3+기반의 모델만 사용한 시스템보다 향상되었음을 보였다.

하지만 구현과정에서 YOLO를 적용한 U-Net3+ 모델은 기존 U-Net3+만 사용한 모델에 비해 추가적인 객체 인식 과정이 추가된 점이 단점이다. 이로 인해 학습 과정에서 더 많은 시간이 소요될 수 있다. 이 부분은 U-Net3+과 YOLO 모델의 복잡도를 줄여 재구성하는 방법으로 개선해야 될 것으로 예상된다.

본 논문은 객체인식 모델 YOLO를 이용하여 U-Net3+의 semantic segmentation을 더 효과적으로

학습할 수 있다는 부분에서 연구의 의의를 찾을 수 있다. 그렇지만 더 깊은 모델에서의 성능과 supervision by ground truth 과정이 추가되었을 때 성능이 검증되지 않았다는 점에서 한계를 찾을 수 있다. 따라서 이러한 문제점을 개선하기 위해 더 깊은 모델, supervision by ground truth 과정이 추가되었을 때, 추가적으로 instance segmentation에서의 성능 평가 과정을 향후 과제로 한다.

References

- [1] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN", Proceedings of the IEEE international conference on computer vision, Venice, Italy, pp. 2961-2969, Oct. 2017.
- [2] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation", Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015, pp. 234-241, May 2015. https://doi.org/10.1007/978-3-319-24574-4_28.
- [3] J. Redmon, S. Divval, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 779-788, Jun. 2016.
- [4] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 4, pp. 834-848, Apr. 2018. <https://doi.org/10.1109/TPAMI.2017.2699184>.
- [5] S-M. Lee, S-Y Lee, and N. Kim, "Improved Pancreas Segmentation using Multiple Concatenated U-Net Model for Medical Image Systems", Journal of KIIT, Vol. 20, No. 5, pp. 81-87, May 2022. <http://dx.doi.org/10.14801/jkiit.2022.20.5.81>.
- [6] D. Kim and J. Huh, "Robust Object Tracking Framework for Occlusion Area using YOLOv5

and Dual Kalman Filter", Journal of KIIT, Vol. 21, No. 2, pp. 19-32, Feb. 2023. <http://dx.doi.org/10.14801/jkiit.2023.21.2.19>.

[7] S-Y. Cho and J-J. Kim, "Development of a Multi-disciplinary Video Identification System for Autonomous Driving", Journal of IIBC, Vol. 24, No. 1, pp. 65-74, Feb. 2024. <https://doi.org/JIIBC.2024.24.1.65>.

[8] S-Y. Hwang and J-J. Kim, "Development for Analysis Service of Crowd Density in CCTV Video using YOLOv4", Journal of IIBC, Vol. 24, No. 3, pp. 177-182, Jun. 2024. <https://doi.org/10.7236/JIIBC.2024.24.3.177>

[9] BDD100K, "BDD100k Label format", <https://doc.bdd100k.com/format.html> [accessed: Jun. 25, 2024]

[10] H. Huimin, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y-W. Chen, and J. Wu, "Unet 3+: A full-scale connected unet for medical image segmentation", 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1055-1059, May 2020.

[11] YOLOv5, "yolov5:v.70-YOLOv5 SOTA Realtime semantic segmentation(2022)", <https://github.com/ultralytics/yolov5/discussions/10258> [accessed: Jun. 26, 2024]

[12] G-H. Jo, K-M. Hyun, and Y-J. Song, "Parallel U-Net based Semantic Segmentation Method using Generated Data from YOLO V5", The Journal of the KICS, Vol. 48, No. 3, pp. 319-326, Mar. 2023. <http://dx.doi.org/10.7840/kics.2023.48.3.319>.

[13] H. Chen, Y. Jin, G. Jin, C. Zhu, and E. Chen, "Semisupervised semantic segmentation by improving prediction confidence", IEEE Transactions on Neural Networks and Learning Systems, Vol. 33, No. 9, pp. 4991-5003, Sep. 2022. <http://dx.doi.org/10.1109/TNNLS.2021.3066850>.

[14] Y. Zhu, Z. Zhang, C. Wu, Z. Zhang, T. He, H. Zhang, R. Manmatha, M. Li, and A. Smola, "Improving semantic segmentation via self-training", arXiv preprint arXiv:2004.14960, Apr. 2020.

<https://doi.org/10.48550/arXiv.2004.14960>.

[15] G-H. Jo and Y-J. Song, "Image Data Generation System for Classification of Traffic Accident Scenarios", Journal of KIIT, Vol. 20, No. 4, pp. 31-38, Apr. 2022. <http://dx.doi.org/10.14801/jkiit.2022.20.4.31>.

[16] T-H Kim and J-J Park, "Teeth Segmentation for Orthodontics based on Deep Learning", The Journal of the KIEE, Vol. 72, No. 3, pp. 440-446, Mar. 2023. <https://doi.org/10.5370/KIEE.2023.72.3.440>.

[17] M. J. Lee, Y-S. Oh, and H. Hong, "Enhanced Lung Cancer Segmentation with Deep Supervision and Hybrid Lesion Focal Loss in Chest CT Images", Journal of the Korea Computer Graphics Society, Vol. 30, No. 1, pp. 11-17, Mar. 2024. <https://doi.org/10.15701/kcgs.2024.30.1.11>.

저자소개

조 국 한 (Guk-Han Jo)



2016년 2월 : 국립금오공과대학교
컴퓨터공학과(공학사)
2019년 2월 : 국립금오공과대학교
전자공학과(공학석사)
2019년 3월 ~ 현재 : 국립금오공과
대학교 전자공학과 공학박사
과정

관심분야 : 머신러닝, 빅데이터, 채널코딩

송 영 준 (Young-Joon Song)



1987년 2월 : 한양대학교
전자통신공학과(공학사)
1994년 2월 : 한양대학교
전자통신공학과(공학석사)
1999년 2월 : 한양대학교
전자통신공학과(공학박사)
1994년 10월 ~ 2002년 2월 :

LG전자 책임연구원
2002년 3월 ~ 현재 : 국립금오공과대학교 전자공학부
교수 IEEE senior member
관심분야 : 머신러닝, 빅데이터, 채널코딩, 이동통신
시스템