

YOLO V5의 생성 데이터를 이용한 병렬 U-Net 기반 의미론적 분할 방법

조국한*, 현광민*, 송영준^o

Parallel U-Net Based Semantic Segmentation Method Using Generated Data from YOLO V5

Guk-Han Jo*, Kwang-Min Hyun*, Young-Joon Song^o

요 약

본 연구에서는 객체 인식 모델인 YOLO(You Only Look Once) V5로 바운딩 박스 이미지를 생성하고 이 이미지를 병렬 U-Net에 학습 데이터로 사용하여 의미론적 분할을 수행하는 시스템을 제안 및 구현한다. YOLO V5 학습에는 yolov5s 모델을 사용하고 차량 및 보행자를 바운딩 박스의 형태로 검출하여 이미지를 생성한다. 이렇게 생성된 이미지 데이터는 병렬 U-Net의 학습 데이터로 사용한다. 제안한 병렬 U-Net은 원본 이미지와 YOLO V5에서 출력한 이미지를 병렬 형태로 입력받아 학습하여 의미론적 분할을 수행한다. 제안한 시스템의 검출 성능을 평가하기 위해 기존의 U-Net 및 U-Net3+의 의미론적 분할 결과와 비교하였다. YOLO V5의 학습 데이터로는 BDD100K(Berkeley Deep Drive 100K)를 사용하였으며 실험 결과, 병렬 U-Net이 기존의 단일 U-Net보다 0.1~0.2, U-Net3+보다 0.03~0.15 향상된 mIoU(Mean Intersection over Union)를 얻을 수 있었다.

키워드 : YOLO V5, 병렬 U-Net, 의미론적 분할, BDD100K, mIoU

Key Words : YOLO V5, Parallel U-Net, semantic segmentation, BDD100K, mIoU

ABSTRACT

In this study, we propose and implement an semantic segmentation method using a pair of U-Net in parallel where its training data are images generated from the YOLO(You Only Look Once) V5 object detection technique using yolov5s model. Image for vehicles and pedestrians from the YOLO model are in the form of bounding box and they are used for training data of a parallel U-Net. The proposed U-Net receives training images for learning in such a way that one U-Net receives original images and the other parallelly connected U-Net receives output images from YOLO V5, and then performs semantic segmentation. We compared the detection performance of the proposed method with the conventional semantic segmentation methods using a U-Net or a U-Net3+. BDD100K(Berkeley Deep Drive 100K) data set was used as training data for YOLO V5, and we found that mIoU(Mean Intersection over Union) of the proposed method is about 0.1~0.2 higher than a conventional U-Net and 0.03~0.15 higher than U-Net3+.

※ 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2021R1A6A3A13043168)

• First Author : Dept. of Electronic Engineering, Kumoh National Institute of Technology, mrgks12@kumoh.ac.kr, 학생회원

o Corresponding Author : Dept. of Electronic Engineering, Kumoh National Institute of Technology, yjsong@kumoh.ac.kr, 종신회원

* Gangneung-Wonju National University, Dept. of Information and Telecommunication Eng., kamihyun@gwnu.ac.kr, 종신회원

논문번호 : 202212-302-B-RU, Received December 19, 2022; Revised January 12, 2023; Accepted January 26, 2023

I. 서론

최근 머신러닝의 발달로 인해 도로환경에서 활용될 수 있는 다양한 기술들이 개발되고 있다. 그 중에서도 객체 인식, 의미론적 분할 기술은 자율주행, 사고 예측에 필수적으로 사용되고 있어 기업 및 개인 개발자들이 지속적으로 연구를 수행하고 있는 기술이다. 따라서 R-CNN, FCN(Fully Convolutional Network), DoTA(Object Detection in Aerial Images), U-Net과 One-stage 방식의 YOLO 등과 같은 다양한 알고리즘 및 모델들이 제안되었으며 이에 맞게 다양한 오픈 데이터셋이 제공되고 있다¹⁻⁵. 차량 인식 대표적인 오픈 데이터셋은 BDD100K⁶과 Tusimple, Waymo 등이 있으며 라이다, 객체 인식, 의미론적 분할을 위한 정보들을 제공한다⁷.

하지만 의미론적 분할 학습을 위해 사용되는 데이터들은 객체 인식에 비해 적게 제공된다. 예를 들어 본 논문에서 사용된 BDD100K는 객체 인식 데이터가 총 100,000개 제공되는 반면 의미론적 분할 데이터는 8,000개가 제공된다. 따라서 추가적인 데이터 생성이 필요할 경우 새로운 이미지를 수집하고 라벨링 하는 작업이 필요하여 많은 시간이 소요될 수 있다.

따라서 본 논문에서는 동일한 데이터를 활용하여 더 정확한 의미론적 분할을 수행하기 위해 YOLO V5⁸와 병렬 U-Net을 이용한 방법을 제안하고 기존의 U-Net 및 U-Net3+⁹와 성능을 비교한다. YOLO V5는 2020년 6월에 발표된 객체 인식 알고리즘으로써 이전 YOLO 버전 및 기존의 대부분 객체 인식 알고리즘과 동일하게 백본(backbone)과 헤드(head)로 구성되어 있다. 하지만 이전 YOLO 버전에서는 C기반의 Darknet을 이용하였지만 YOLO V5에서는 파이토치로 구현된 CSPNet 기반 백본을 가지고 있다. 객체 인식에서 백본은 입력 데이터나 이미지의 특징맵 추출을 수행하며 헤드에서는 추출된 특징맵을 바탕

로 클래스 예측과 바운딩 박스 작업을 수행한다⁸. 본 논문에서는 Glenn jocher가 공개한 YOLO v5 오픈 소스를 이용하여 BDD100K 학습을 수행하였다¹⁰. U-Net은 FCN의 확장 개념으로 적은 데이터양으로 준수한 성능을 보이는 모델이다. 기본적으로 수축경로(contracting path)와 확장 경로(expanding path)로 구성되어 있으며 엔드 투 엔드 구조로 속도가 빠르다는 장점이 있다. 일반적으로 차선 검출 및 분할(segmentation) 작업에 주로 사용되고 있다^{4,7}. U-Net3+는 2020년에 발표되었으며 풀-스케일 연결(full scale connection)을 통해 U-Net++의 성능을 보완하고 네트워크 파라미터를 줄이기 위해 제안된 모델이다. 본 논문에서 제안한 시스템은 먼저 객체 인식을 수행하고 출력된 검출 결과와 원본데이터를 U-Net 모델에 병렬로 입력한다. 병렬 U-Net을 통해 각각 학습된 결과는 출력 레이어 직전 레이어에서 결합 되어 출력된다.

본 논문의 구성은 다음과 같다. 2장은 전체적인 시스템 구성에 대해 설명하고 2.1에서는 객체 인식을 위해 수행한 YOLO V5 모델 설명과 학습방법, 2.2에서는 의미론적 분할을 위해 구성한 U-Net의 구조와 YOLO V5에서 생성한 검출 데이터를 U-Net 학습으로 사용하는 방법에 대하여 설명한다. 3장에서는 실험에 사용된 장비와 학습데이터, YOLO V5 모델의 검출 결과에 대하여 설명하고 이 결과 데이터를 입력으로 하는 병렬 U-Net의 결과와 U-Net, U-Net3+의 결과를 비교한다. 마지막으로 4장에서는 결론에 대해 기술하고 개선방향 및 향후 개발 내용에 대하여 설명한다.

II. 시스템 개요

그림 1은 본 논문에서 제안하는 의미론적 분할 방법의 구성도이다. 먼저 수집한 이미지 및 라벨 데이터들을 YOLO V5 모델에 입력하여 학습을 수행한다.

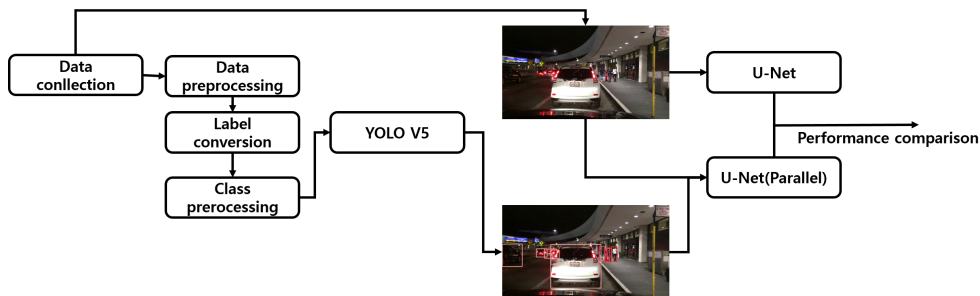


그림 1. 전체적인 시스템 구성
Fig. 1. Overall system configuration

학습 후 U-Net의 입력으로 사용될 이미지들을 YOLO 모델에 입력하여 차량 및 보행자의 검출이 바운딩 박스를 통해 이루어진 이미지를 얻는다. 이후 YOLO를 이용하여 출력된 결과이미지와 YOLO 입력으로 사용한 원본 이미지를 U-Net의 학습 데이터로 사용한다. U-Net 입력을 병렬로 구성하여 원본 데이터와 YOLO 출력 이미지를 각각 학습하고 각각의 학습 결과를 마지막에 취합하여 의미론적 분할 결과를 얻는다. 이후 기존 원본 데이터만 사용한 U-Net과 U-Net3+의 결과와 비교하여 성능을 테스트한다.

2.1 학습 이미지 생성을 위한 YOLO V5

YOLO V5에서는 총 4개의 모델이 존재하는데 각 모델은 depth_multiple과 width_multiple의 크기에 따라 v5s, v5m, v5l, v5x로 구성되어 있다.

depth_multiple과 width_multiple의 값이 크면 모델의 깊이가 깊어져 mAP(mean Average Precision) 성능이 좋아지지만 수행시간이 증가한다⁸⁾. 4가지 모델 중 v5s 모델이 depth_multiple과 width_multiple 값이 가장 작은 모델이며 v5x가 가장 큰 모델이다. 본 논문에서는 학습에 시간을 최소화 하기 위해 v5s 모델을 사용하였다.

본 논문에서 YOLO V5 학습에 사용한 BDD100K 데이터셋은 학습(training)에 사용할 jpg 형태의 차량의 이미지 70,000개와 검증(validation)에 사용할 10,000개의 이미지, 학습과 검증 데이터의 물체의 좌표 및 종류에 관한 데이터를 json 파일로 제공한다. 하지만 제공된 데이터들을 YOLO의 학습 데이터로 사용하기 위해서는 그림 2와 같이 BDD100K json 파일을 YOLO label txt로 변환하는 작업이 필요하다. BDD100K json 파일을 YOLO 라벨 데이터로 변환하기 위해 1차적으로 COCO 형태의 json 파일로 변환¹¹⁾한 후에 Convert2Yolo 코드를 이용하여 YOLO label txt로 변환하였다¹¹⁻¹²⁾. 변환과정에서 학습에 사용될 70,000개의 데이터 중에 BDD100K json 데이터 자체에 누락된 부분이 존재하여 누락된 이미지를 제외하였다. 모델 학습에 사용된 총 데이터는 학습 데이터 69,863개 검증 데이터 10,000이며 이를 크기로 변환하여 사용하였다. 또한 BDD100K에서 제공하는 json

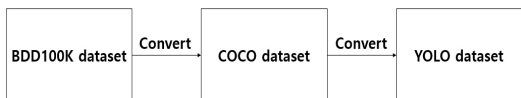


그림 2. BDD100k 데이터셋 변환
Fig. 2. Transformation of the BDD100k dataset.

파일에는 총 10개의 클래스에 대한 좌표를 제공하는데 본 논문에서는 차량 및 보행자 인식이 필요로 하여 이와 관련이 없는 클래스는 삭제하고 버스, 트럭 클래스는 자동차 클래스로 변환하여 총 2개의 클래스로 구성된 라벨 데이터를 생성하였다.

학습 횟수는 총 100회, 배치 사이즈는 32, 초기 가중치(weight)는 YOLO V5에서 제공하는 가중치를 사용하지 않고 학습을 수행하였다. 학습 후 BDD100K 7000개의 이미지 데이터를 YOLO V5 모델에 입력한다. 그리고 차량 및 보행자 정보가 바운딩 박스 형태로 검출된 이미지를 출력하고 이를 U-Net의 입력으로 사용한다.

2.2 의미론적 분할을 위한 U-Net 모델

본 논문에서는 의미론적 분할을 위해 U-Net을 사용하였다. U-Net은 일반적으로 수축 경로와 확장 경로로 구성되어 있는데 수축 경로는 인코딩 경로(encoding path)이라고도 부르며 입력 이미지의 의미 정보(context) 포착을 목표로 한다. 확장 경로는 디코딩 경로(decoding path)라고도 부르며 업샘플링(up-sampling) 과정에서 데이터 결합(concatenate)을 이용하여 풀링 이전의 특징맵을 복사하고 이를 컨볼루션을 통해 소실된 픽셀 값을 보정한다⁴⁾.

그림 3은 본 논문에서 구성한 U-Net 모델의 학습과정을 플로우차트를 통해 나타낸 것이다. U-Net 모델에 이미지 데이터 및 라벨 데이터가 입력되면 수축 경로에서 더블 컨볼루션과 배치정규화(batch normalization), 풀링을 통해 이미지의 의미정보를 추출한다. 모델 깊이에 따라 수행 횟수가 다르며 본 논문에서는 모델의 깊이를 2로 설정하였다. 이후 수축 경로가 완료되면 확장 경로에서는 수축 경로와 반대의 연산을 수행함으로써 특징맵을 확장한다.

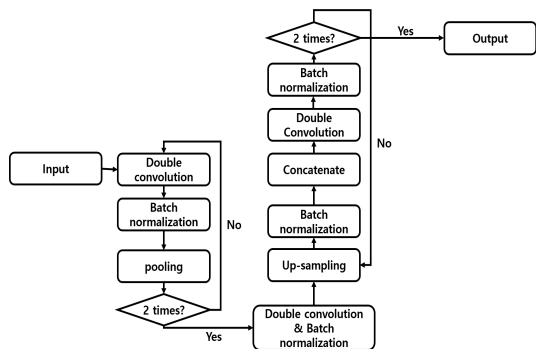


그림 3. U-Net 플로우차트
Fig. 3. Flowchart of U-Net

그림 3의 U-Net 모델에서 사용된 활성화 함수는 ReLU이며 출력 레이어에서만 시그모이드(sigmoid) 사용하였다. 그림 4의 U-Net3+ 모델은 그림 3의 U-Net과 동일한 구조로 이루어져 있지만 풀-스케일 연결, 스킵 연결(skip connection)의 형태를 가진다. 따라서 U-Net3+의 각 디코더들은 현재 스케일과 같거나 작은 특징맵을 받고 이를 디코더에서 나온 스케일의 특징맵과 통합한다. 여기서 E1은 2회 컨볼루션과 ReLU를 수행하며 E2, E3는 E1에서 맥스풀링이 추가된 레이어이다. D1, D2는 확장 경로로 각 스케일의 특징값들을 합쳐주고 컨볼루션과 ReLU를 수행하는 레이어이다. U-Net3+는 기존의 U-Net의 손실함수 이진 교차 엔트로피(binary cross entropy)가 아닌 focal, MS-SSIM,(multi-scale structural similarity index), IoU loss를 합하여 사용한다. 그림 5는 본 논문에서 구성한 병렬 U-Net 모델의 구성이며 모델의 깊이는 U-Net, U-Net3+와 비교하기 위해 동일하게 구성하였다. 그림 5를 살펴보면 해당 모델은 2개의 이미지 데이터셋이 입력되어 각각 학습을 수행하는 것을 확인할 수 있다. Input 1에는 원본 이미지를 입력으로 넣고 input 2에서 입력한 동일한 이미지를 YOLO V5를 통해 바운딩 박스가 포함된 이미지로 출력한다. 이 이미지를 input 2의 입력으로 넣는다. 따라서 라벨 데이터는 동일하다. Input 1과 Input 2는 확장

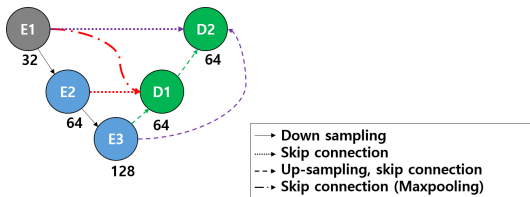


그림 4. U-Net3+ 모델의 구성
Fig. 4. Construction of U-Net3+

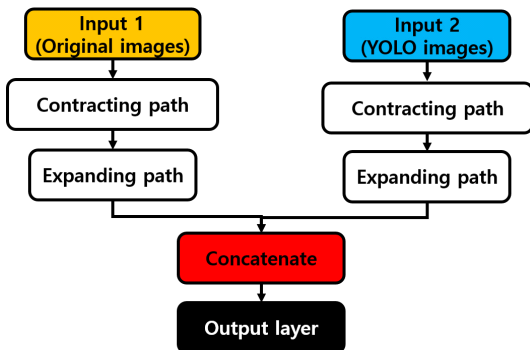


그림 5. 제안된 병렬 U-Net 모델의 구성
Fig. 5. Construction of proposed parallel U-Net model

경로를 모두 지날 때 까지 각각 학습을 수행하다가 출력 레이어를 지나기 전에 결합된다.

III. 실험

표 1은 실험에 사용된 장비의 종류와 제조회사이며 그림 6은 YOLO V5에 학습된 BDD100K 이미지 데이터와 좌표 형식의 라벨 데이터를 이미지에 나타낸 것이다.

표 1. 실험에 사용된 장비
Table 1. Equipments used in the experiment

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



그림 6. YOLO V5 학습 데이터
Fig. 6. Learning data for YOLO V5

3.1 YOLO V5 학습 결과

그림 7은 YOLO V5 학습 결과의 Precision-Recall curve 그래프를 나타낸 것이다. Precision은 정밀도이며 Recall은 재현도이다. 그래프에서 파란색 그래프의 PR curve의 면적이 mAP@0.5가 되며 이를 통해 객체의 정확도를 평가하였다. mAP@0.5는 IoU(Intersection over Union)가 0.5이상을 true positive로 하여 모든 AP의 값을평균한 것이다. IoU 수식은 (1)과 같으며 mAP 수식은 (2)와 같다.

$$IoU = \frac{|b \cap b^{gt}|}{|b \cup b^{gt}|} \quad (1)$$

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i \quad (2)$$

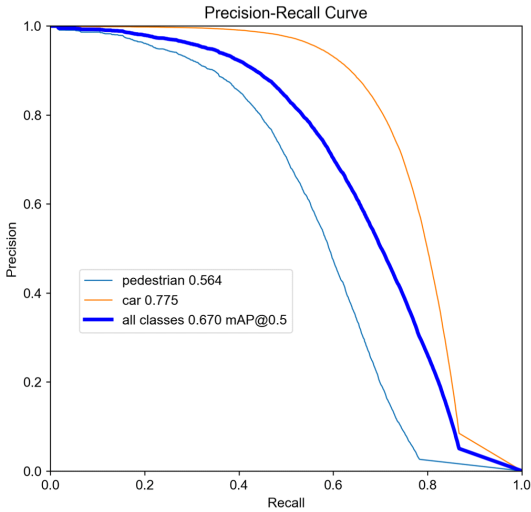


그림 7. PR curve 그래프 결과
Fig. 7. Results of PR curve graph

수식(1)에서 b 는 예측 바운딩 박스며 b^{gt} 는 라벨 데이터(ground truth)이다. 수식 (2)에서 N 은 총 클래스의 수이며 AP_i 는 i 번째 클래스의 AP 값이다 [13-14]. 본 논문에서는 학습된 결과로 0.670의 mAP를 얻었다.

그림 8은 학습된 모델을 바탕으로 테스트 이미지를 입력하여 바운딩 박스를 출력한 결과이다. 그림 8의 결과를 살펴보면 차량 및 보행자가 존재하는 곳을 올바르게 검출하는 것을 확인할 수 있다. 본 논문에서는 그림 8과 같은 데이터를 병렬 U-Net의 추가적인 입력으로 사용한다.

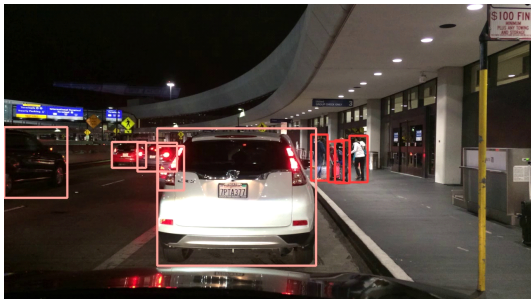


그림 8. YOLO V5 학습 결과
Fig. 8. Training result of YOLO V5

3.2 U-Net 학습 결과

본 논문에서는 보행자와 차량을 한 클래스로 묶고 배경과 물체를 분할하였다. 이후 의미론적 분할 성능 평가를 위해 BDD100K 데이터 6500개일 때 mIoU와

3250개일 때 mIoU를 계산하고 각 모델의 성능을 비교하였다.

그림 9는 모델의 입력으로 사용한 데이터이며 (a)는 원본 이미지 데이터, (b)는 YOLO V5 출력 이미지 (c)는 라벨 데이터이다. (a), (c) 데이터는 BDD100k에서 제공되는 의미론적 분할을 위한 학습 데이터이며 일반 U-Net에는 이 2가지 데이터만 사용된다.

그림 10의 (a)는 라벨 데이터, (b)는 U-Net에 YOLO에서 출력한 바운딩 박스 이미지를 입력하여 학습한 의미론적 분할 결과이다. 원본 이미지 데이터 없이 YOLO V5를 통해 출력된 이미지를 단일로 U-Net 모델에 학습하게 되면 테스트 이미지를 입력했을 경우에 그림 10의 (b)와 같이 의미론적 분할이 올바르게 출력되지 않았다. 이는 학습 데이터와 테스트 데이터 형태가 동일하지 않아 발생하는 문제이며 이를 해결하기 위해 병렬 U-Net을 구성하여 1개 입력에는 원본 이미지 데이터를 또 다른 1개의 입력에는 YOLO V5의 출력 이미지를 입력하여 학습을 수행하였다. 그 결과 테스트 데이터에서 의미론적 분할을 올바르게 수행하였다.

그림 11의 (a)는 train 데이터 6500개를 사용하여 출력한 병렬 U-Net의 의미론적 분할이며 (b)와 (c)는 각각 U-Net과 U-Net3+의 분할 결과이다.

표 2는 U-Net, U-Net3+, 병렬 U-Net 학습 후

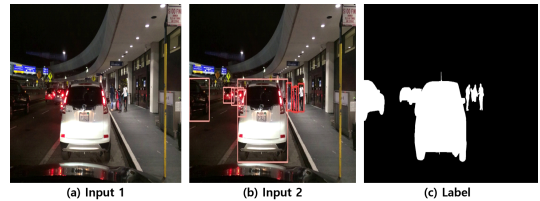


그림 9. 병렬 U-Net 학습 데이터
Fig. 9. Learning data for parallel U-Net

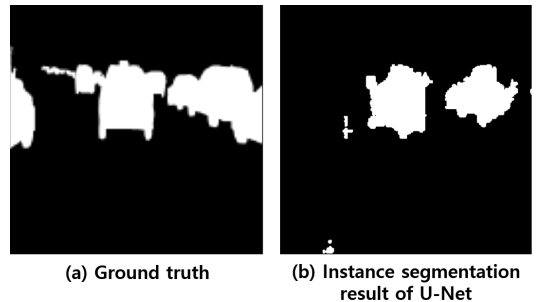


그림 10. YOLO V5 결과로 학습한 의미론적 분할
Fig. 10. Semantic segmentation learned from YOLO V5 results

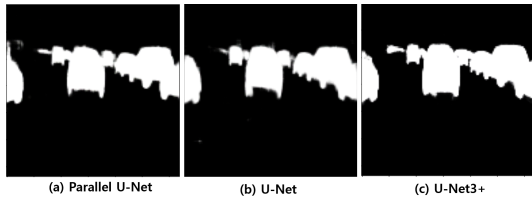


그림 11. (a) 병렬 U-Net (b) U-Net (c) U-Net3+ 결과
Fig. 11. Results of (a) parallel U-Net (b) U-Net (c)U-Net3+

표 2. U-Net, U-Net3+, 병렬 U-Net 성능 결과
Table 2. Results of U-Net, U-Net3+ and parallel U-Net

Model	Data set size	Performance (mIoU)
U-Net	3250	0.463
	6500	0.512
U-Net3+	3250	0.548
	6500	0.587
Parallel U-Net	3250	0.595
	6500	0.708

테스트 데이터의 mIoU 결과를 정리한 것이다. mIoU는 입력된 이미지의 평균적인 IoU로 수식은 (3)과 같다.

$$mIoU = \frac{1}{N_t} \sum_{i=1}^{N_t} \frac{|b_i \cap b_i^{gt}|}{|b_i \cup b_i^{gt}|} \quad (3)$$

N_t 는 테스트 데이터의 이미지 개수이며 i 는 I 번째 예측 의미론적 분할이며 i 번째 라벨 데이터이다. 표 2의 결과를 살펴보면 U-Net 및 U-Net3+ 모델에서 6500개의 학습 데이터를 사용했을 경우에 각각 0.512, 0.587의 mIoU를 얻을 수 있었고 3250개에서는 각각 0.463, 0.548의 mIoU를 얻을 수 있었다. 병렬 U-Net에서는 입력 당 6500개의 이미지를 사용했을 경우 0.708의 결과를 얻을 수 있었고 3250개에서는 0.595의 결과를 얻을 수 있었다.

이를 통해 YOLO 데이터를 입력으로 사용해 병렬 U-Net을 수행한 결과 mIoU가 기존 U-Net, U-Net3+에 비해 높은 것을 확인 할 수 있었다. 또한 데이터 수 3250개로 적을 경우 원본데이터만 활용한 U-Net, U-Net3+보다 원본데이터에 YOLO를 적용한 바운딩 박스가 포함된 데이터도 함께 사용한 병렬 U-Net이 mIoU가 더 높았다.

이때 YOLO 데이터가 실제 학습에 도움이 되는지

를 알아보기 위해 YOLO 데이터 없이 단순히 U-Net을 병렬로 연결하여 사용하였다. 그 결과 3250개의 데이터에서 mIoU의 성능은 0.49였다. 이를 통해 U-Net은 기존 원본데이터를 사용한 의미론적 분할 수행과정에서 인식 하지 못한 물체를 바운딩 박스가 포함된 이미지 데이터 학습을 통해 보완하는 것으로 생각된다. 따라서 병렬 U-Net의 입력으로 사용된 원본이미지와 YOLO 이미지는 서로 같은 라벨 데이터를 사용하기 때문에 같은 이미지를 사용하는 것처럼 보이나 그림 10과 같이 YOLO 이미지를 단일로 사용했을 경우에는 분할이 올바르게 되지 않았고 YOLO 데이터 없이 단순히 U-Net을 병렬로 구성하여 학습을 진행한 것보다 YOLO 데이터를 사용하여 병렬로 학습한 모델의 성능이 더 좋은 것을 알 수 있다.

하지만 병렬 U-Net을 사용했을 경우 입력이 2개 존재하기 때문에 U-Net에 비해 총 파라미터의 개수도 약 2배 U-Net3+에 비해 1.5배 정도로 증가한다.

IV. 결 론

본 논문에서는 의미론적 분할의 성능 향상을 위해 YOLO V5를 적용하여 객체 인식이 적용된 이미지를 생성하고 이를 병렬 U-Net의 학습 데이터로 사용함으로써 기존에 U-Net, U-Net3+보다 더 좋은 mIoU를 가지는 시스템을 구현하였다. 학습 이미지는 BDD100K에서 제공하는 도로환경 이미지 세트와 객체 인식, 의미론적 분할의 라벨 데이터를 이용하였다. YOLO V5를 이용한 데이터 생성을 위해 총 79863개의 BDD100K 데이터를 학습 및 검증 데이터로 사용하였으며 학습 결과 0.670의 mAP를 얻었다. 또한 바운딩 박스가 출력된 이미지 7000개와 원본 이미지 7000개를 병렬 U-Net에 학습 및 테스트 데이터로 사용하여 기존에 U-Net보다 0.1~0.2, U-Net3+보다 0.03~0.15 향상된 mIoU를 얻을 수 있었다. 병렬 U-Net은 입력이 2개 존재하기 때문에 파라미터가 1.5~2배가량 증가하는 단점과 더 깊은 모델에서의 성능이 검증되지 않았다는 한계점이 있지만 본 연구에서 수행한 깊이의 스케일에서는 의미론적 분할을 위한 추가적인 라벨링 작업 없이 YOLO 데이터만으로 성능을 개선할 수 있다는 장점이 있다.

향후에는 본 연구에서 제안한 병렬 U-Net 모델의 깊이를 늘리거나 모델의 구조를 변환하여 학습하고 이때에도 U-Net 및 U-Net3+ 보다 높은 mIoU를 가지는지를 연구할 예정이다. 이 연구 결과를 바탕으로 최종적으로 IGLAD(Initiative for the GLocal

harmonization of Accident Data)을 이용한 사고유형 분류 시스템을 구현하고자 한다.

References

- [1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 2961-2969, Venice, Italy, Oct. 2017. (<https://doi.org/10.48550/arXiv.1703.06870>)
- [2] G.-S. Xia, et al., "DOTA: A large-scale dataset for object detection in aerial images," in *Proc. IEEE Conf. CVPR*, pp. 3974-3893, Salt Lake City, UT, USA, Nov. 2018. (<https://doi.org/10.1109/CVPR.2018.00418>)
- [3] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. CVPR*, pp. 3431-3440, UC Berkeley, USA, Nov. 2015. (<https://doi.org/10.1109/CVPR.2015.7298965>)
- [4] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Int. Conf. Med. Image Comput. and Computer-Assisted Intervention*, pp. 234-241, Munich, Germany, May 2015. (<https://doi.org/10.48550/arXiv.1505.04597>)
- [5] J. Redmon, S. Divval, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. CVPR*, pp. 779-788, Las Vegas, USA, Jun. 2016. (<https://doi.org/10.1109/CVPR.2016.91>)
- [6] BDD100K, "BDD100k Label format," Dec. 6, 2022, <https://doc.bdd100k.com/format.html>
- [7] G.-H. Jo and Y.-J. Song, "Image data generation system for classification of traffic accident scenarios," *J. KIIT*, vol. 20, no. 4, pp. 31-38, Apr. 2022. (<http://doi.org/10.14801/jkiit.2022.20.4.31>)
- [8] G. Jocher, "yolov5:v.7.0 - YOLOv5 SOTA Realtime semantic segmentation(2022)," Dec. 2022. (<https://doi.org/10.5281/zenodo.7347926>)
- [9] H. Huang, et al., "Unet 3+: A full-scale connected unet for medical image segmentation," *ICASSP*, Barcelona, Spain, Apr. 2020. (<https://10.1109/ICASSP40776.2020.9053405>)
- [10] G. Jocher, Nioolek, "yolov5 v7.0 by ultralytics," Nov. 2022. (<https://github.com/ultralytics/yolov5>)
- [11] Y. P. Loh and C. S. Chan, "Getting to know low-light images with the exclusively dark dataset," *Comput. Vision and Image Understanding*, vol. 178, pp. 30-42, Jan. 2019. (<https://doi.org/10.1016/j.cviu.2018.10.010>)
- [12] J. Gao, et al., "Detecting and learning city intersection traffic contexts for autonomous vehicles," *J. Smart Cities and Soc. Preprint*, vol. 1, no. 3, pp. 213-239, Sep. 2022. (<https://doi.org/10.3233/SCS-220010>)
- [13] X. Song and W. Gu, "Multi-objective real-time vehicle detection method based on yolov5," *Int. Symp. Artificial Intell. and its Appl. on Media*, pp. 142-145, Xi'an, China, May 2021. (<https://doi.org/10.1109/ISAIAM53259.2021.00037>)
- [14] I. Ahmad, et al., "Deep learning based detector YOLOv5 for identifying insect pests," *Appl. Sci.*, vol. 12, no. 19, 10167, 2022. (<https://doi.org/10.3390/app121910167>)

조 국 한 (Guk-Han Jo)



2016년 2월 : 금오공과대학교 컴
퓨터공학과 공학사
2019년 2월 : 금오공과대학교 전
자공학과 공학석사
2019년 3월~현재 : 금오공과대
학교 전자공학과 공학박사 과
정 재학

<관심분야> 머신러닝, 채널코딩
[ORCID:0000-0002-4384-2852]

송 영 준 (Young-Joon Song)



1987년 2월 : 한양대학교 전자
통신공학과 공학사
1994년 2월 : 한양대학교 전자
통신공학과 공학석사
1999년 2월 : 한양대학교 전자
통신공학과 공학박사
1994년 10월~2002년 2월 : LG
전자 책임연구원

2002년 3월~현재 : 금오공과대학교 전자공학부 교수
<관심분야> 머신러닝, 채널코딩, 이동통신 시스템
[ORCID:0000-0002-9736-602X]

현 광 민 (Kwang-Min Hyun)



1989년 2월 : 한양대학교 전자
통신공학과 학사
1995년 2월 : 한양대학교 전자
통신공학과 석사
2004년 8월 : 한양대학교 전자
통신전공공학 박사
1989년~1994년 : (주) 대우통신

1989년~1994년 : (주) 대우통신
1995년~1996년 : (주) 고등기술연구원
1997년~2001년 : (주) 케이엠더블유
2004년~2005년 : (주) 인텔 코리아
2005년~현재 : 강릉원주대학교 정보통신공학과 교수
<관심분야> 디지털통신, 이동통신, 통신신호처리
[ORCID:0000-0002-2036-6558]