Journal of the KIMST, Vol. 25, No. 6, pp. 606-616, 2022
ISSN 1598-9127(Print) · 2636-0640(Online)
DOI https://doi.org/10.9766/KIMST.2022.25.6.606

Research Paper 정보·통신 부문

비정형 야지환경 주행상황에서의 실시간 의미론적 영상 분할 알고리즘 성능 향상에 관한 연구

김대영*,1) · 안승욱2) · 서승우1)

1) 서울대학교 전기정보공학부
2) 한화디펜스㈜ 국방로봇사업부 로봇개발그룹

A Study of Real-time Semantic Segmentation Performance Improvement in Unstructured Outdoor Environment

Daeyoung Kim*,1) · Seunguk Ahn2) · Seung-Woo Seo1)

1) *Department of Electrical and Computer Engineering, Seoul National University, Korea*
2) *Department of Defense Robotics and Autonomous Systems Development, Hanwha Defense Co., Ltd., Korea*

(Received 8 April 2022 / Revised 31 October 2022 / Accepted 18 November 2022)

Abstract

Semantic segmentation in autonomous driving for unstructured environments is challenging due to the presence of uneven terrains, unstructured class boundaries, irregular features and strong textures. Current off-road datasets exhibit difficulties like class imbalance and understanding of varying environmental topography. To overcome these issues, we propose a deep learning framework for semantic segmentation that involves a pooled class semantic segmentation with five classes. The evaluation of the framework is carried out on two off-road driving datasets, RUGD and TAS500. The results show that our proposed method achieves high accuracy and real-time performance.

Key Words : Deep Learning(딥러닝), Semantic Segmentation(의미론적 영상 분할), Autonomous Driving(자율주행)

1. 서 론

의미론적 영상 분할은 자율주행 분야의 실시간 이동 가능 경로를 탐색하는 데 중요한 기술이며 현재 실시간 처리속도와 함께 정확도 향상을 위한 많은 연구가 진행되었다. 하지만 이러한 연구 방향은 대부분 Cityscapes benchmark[1]와 같이 도심 주행 환경에 대해서 발전해왔으며 야지 환경에서의 이동 가능 경로 탐지에 대해선 많은 연구가 진행되지 않았다. 야지 환경은 도심 환경과 달리 풀과 나무와 같이 불분명한 경계를 갖는 텍스처로 이루어져 있으며 모호한 비정형 특징점들을 갖는다. 야지 환경의 지면 또한 도심 환경의 포장로와 달리 흙길, 자갈길 등이 포함되고 매우 불규칙적이다. 야지 환경에서의 의미론적 영상 분할 알고리즘의 또 다른 한계점은 Cityscapes, KITTI semantic

* Corresponding author, E-mail: ddykim@snu.ac.kr

segmentation benchmark[1,2]와 같은 대규모 공개 데이터 세트의 부재이다. 의미론적 영상 분할 특성상 레이블링 작업이 매우 힘들기에 양질의 데이터 세트를 손쉽게 구할 수 없고 현재 공개된 데이터 세트는 주로 도심 환경에서의 주행 데이터이다.

본 논문에서는 먼저 현재 공개된 모든 야지 환경 주행 데이터를 조사하여 환경 특징부터 도심 도로환경과의 차이점, 활용 가능성을 확인하고 기존 의미론적 영상 분할 알고리즘의 야지 환경에서의 한계점을 분석하였다. 다음으로 기존 도심 환경에서 실시간성과 정확도가 검증된 의미론적 영상 분할 알고리즘인 BiSeNetV2[8] 기반으로 야지 환경에서의 이동 경로 탐지에 있어 최적의 프레임워크를 제안한다. 네트워크 학습을 위해 현재 공개된 야지 환경 데이터 세트를 이동 경로 탐지 업무에 맞도록 클래스를 재정의하여 통합하고 이에 따른 클래스 불균형 문제를 해결한다.

2. 주행 가능 영역 탐지 알고리즘

본 논문에서는 BiSeNetV2 기반의 의미상 영상분할을 적용하여 주행 가능 영역 탐지 알고리즘을 제안한다. 적용 환경은 기존 도심 환경과 다른 야지 환경을 대상으로 한다. 풀(grass)과 같은 주행 가능한 식물과 덤불 및 나무(bush and tree)와 같은 주행 불가능한 식물을 구별하고 아스팔트, 흙길 그리고 자갈길(asphalt, dirt, gravel)과 같은 다양한 종류의 지면에 대해 주행 가능한 영역을 식별하는 것을 목표로 한다.

2.1 의미상 영상분할 알고리즘

의미상 영상분할(semantic segmentation)은 입력받은 이미지에 대해 의미상으로 비슷한 영역으로 분류하는 것을 말하며 출력 이미지는 각 픽셀 위치에 자신이 해당하는 클래스의 값을 갖는 픽셀들로 만들어진다. 현재 많은 주목을 받는 CNN(Convolutional Neural Network) 기반의 방식[12,13]에서는 입력받은 이미지의 특징들을 추출하는 인코더 부분과 추출된 특징들로부터 다시 원하는 결과로 복원하는 디코더 부분으로 구성된다. 디코더 과정에서 원래 이미지 크기로 복원된 결과는 원본 이미지에 대응하는 클래스를 픽셀값으로 갖는다. 이러한 인코더-디코더 구조는 기존 방식보다 훨씬 좋은 성능을 보였지만 이미지를 압축했다가 복원하는 과정에서 세부 정보(low-level detail)가 손실되는 한계를 갖고 있다. 여러 논문[14,15]에서 이를 보완하여 정확도를 개선하였지만 많은 연산량을 요구하는 구조로 실시간의 처리속도를 보장하지 않는다. 하지만 최근 세부 정보(low-level detail) 손실을 최소화하며 실시간 처리속도를 보장하는 경량화된 CNN 기반 네트워크 구조의 Bilateral Segmentation Network(BiSeNet), BiSeNetV2가 제안되었다[7,8]. 의미상 영상분할 알고리즘은 저수준 세부 정보(low-level detail)와 고수준 의미 정보(high-level semantics)를 모두 고려해야 한다. 기존에는 하나의 네트워크로 두 가지 정보를 모두 처리하려 했으나 BiSeNetV2는 실시간 처리속도를 유지하기 위해 두 개의 경량화된 네트워크를 병렬적으로 배치하여 각각 하나씩 담당하게 하였다. Detail branch에서는 저수준 세부 정보(low-level detail)를 얻고 높은 해상도의 특징들을 추출하기 위해 각 필터의 채널 수는 많지만 적은 수의 필터를 배치하여 얇은 층(shallow layers)으로 구성한다. Semantic branch는 고수준 의미 정보(high-level semantics)를 정보를 얻기 위해 깊은 층(deep layers)으로 구성하지만 필터 채널 수는 줄이고 depth-wise CNN[11]으로 이루어진 경량화된 구조로 global average pooling등을 통해 빠르게 수준을 내려(down-sample) 수용 영역(receptive field)을 확장한다. 이와 같이 의미론적 경로(Semantic branch)와 세부 정보 경로(Detail branch)에서 추출된 특징들은 집계 레이어(Aggregation layer)에서 융합된다. 기본적인 특징 융합 방법으로는 특징들의 원소별 합(element-wise summation)을 하거나 혹은 이어 붙여 연결하는 방식(concatenation)이 있다. 하지만 의미론적 경로에서 추출된 특징은 높은 수준 의미 정보(high-level semantics)이고 세부 정보 경로의 추출된 특징은 낮은 수준 세부 정보(low-level detail)이기에 단순 합(summation) 또는 연결(concatenation)의 구조는 한계가 존재한다. 따라서, 두 가지 경로의 다양성을 모두 살리기 위해 멀티 스케일(multi-scale) 방식으로 한번은 낮은 수준으로 의미론적 경로의 특징들의 수준을 내려 융합하고 한번은 높은 수준으로 세부 정보 경로의 특징들의 수준을 올려 원소별 합(element-wise summation)과 곱(element-wise product) 연산을 통해 융합한다. 이러한 방식은 낮은 수준 및 높은 수준의 정보들을 모두 활용할 수 있는 방식이다. Detail branch와 Semantic branch의 자세한 구조는 Table 1, 2와 같다. 여기서 opr은 처리 모듈을 의미하고 k는 각 처리 모듈에서의 커널 크기이며 c, e, s, r은 각각 채널 수, 확장 계수, stride, 처리 반복

횟수를 의미한다. 먼저 Detail branch를 보면 입력 이미지에 대해 2차원 합성곱필터(convolutional layer)인 Conv2d를 쌓아 총 3단계에 걸쳐 저수준 세부 정보를 얻는다. 여기서 각 필터의 채널 수를 64, 128으로 구성해 풍부한 정보를 얻으나 총 3단계만으로 구성하여 메모리 사용량을 줄이고 연산 속도를 빠르게 하였다.

Semantic branch에서는 처리 모듈을 단순 Conv2d 필터를 쌓는 것이 아닌 Stem, GE(Gatherand-Expansion), CE(Context Embedding) 블록 구조로 구성하였다. Stem 블록은 입력 이미지를 3단계에 맞는 크기로 빠르게 줄이기 위해 다운 샘플링을 하기 위한 구조로 고수준 정보를 얻기 위한 연결 고리 역할을 한다. GE 모듈은 이미지에 대한 고수준 특징들을 추출하기 위한 경량화된 구조로 3×3 Conv2d 필터들을 통해 채널을 확장계수 e만큼 늘렸다가 다시 원상 복귀하는 구조로 확장하고 모으는 단계에서 많은 양의 연산 없이 특징 정보들이 잘 추출된다. 마지막으로 CE는 추출된 특징으로부터 전역 정보를 획기적으로 추출하는 구조이다. 이 모듈에는 GPooling(global average pooling) 연산이 포함되어 있는데 이미지 전체에 대한 문맥상의 정보를 효율적으로 추출할 수 있게 한다. 이 정보는 최종 단계에서 GE를 통해 추출한 정보와 결합한다.

이 외에 모듈에는 마지막으로 일종의 결과 출력을 위한 Head 역할을 하는 Booster 모듈이 있다. 네트워크의 마지막 단계 이후 최종 출력을 위해선 의미상 영상분할 결과를 입력 이미지와 동일한 크기로 업 샘플링 해야 하고 출력값의 채널과 분류해야 할 클래스 수에 맞춰 각 픽셀에 대한 채널에 최종 클래스에 대한 네트워크 결과(score)를 갖도록 해야 한다. 이를 위해 Booster 모듈에서는 먼저 출력의 채널을 조정하는 Conv2d 필터들과 함께 업 샘플링 연산이 이루어진다. 해당 모듈은 또한 의미상 영상분할 정확도를 올리기 위해 Booster 학습 전략에 사용될 수 있다. Booster 모듈은 최종 단계뿐만 아니라 Semantic branch의 중간 단계에도 활용히 가능하다. 따라서 네트워크 중간 결과를 네트워크 최종 결과 형태로 변환을 할 수 있다. 학습 상황에서는 상대적으로 연산 복잡도에 대한 부담이 적기 때문에 이 모듈을 여러개 추가하여 변형된 중간 단계의 결과물들을 또한 최종 결과에 보조적으로 활용하여 더욱더 학습이 잘되게 활용할 수 있다.

BiSeNetV2는 실시간 의미론적 영상분할을 위해 경량화에 초점을 맞춰 주로 설계되었지만 대형 모델로도 일반화할 수 있다. 주로 두 가지 측면에서 네트워크 확장이 가능하며 첫째는 필터 채널 수 증가를 통해 더 넓은 모델을 만들 수 있고 두 번째로 필터 수 자체를 늘리고 쌓는 방식으로 더 깊은 모델을 만들 수 있다. 따라서 필터 조정으로 더 넓고 깊은 모델을 설계하여 정확도를 올릴 수 있고 반대로 경량화된 구조로 설계할 수도 있다. 또한, Semantic branch는 기존 연구된 모든 경량 CNN 모델[9,10]로 대체가 가능하다. 본 논문에서는 도심 환경에 맞춰진 BiSeNetV2을 비정형 클래스와 고르지 않은 지형 그리고 불규칙한 특징점들을 갖는 야지 환경에서의 적용을 위해 Table 1, 2와 같은 네트워크 구조를 활용하였다.

Table 1. Detail branch

stage	*opr*	*k*	*c*	*s*	*r*	Output Size
Input						512×1024
S1	Conv2d Conv2d	3 3	64 64	2 1	1 1	256×512 256×512
S2	Conv2d Conv2d	3 3	64 64	2 1	1 2	128×256 128×256
S3	Conv2d Conv2d	3 3	128 128	2 1	1 2	64×128 64×128
S4						32×64 32×64
S5						16×32 16×32 16×32

Table 2. Semantic branch

stage	*opr*	*k*	*c*	*e*	*s*	*r*	Output Size
Input							512×1024
S1	Stem	3	8	-	4	1	256×512 256×512
S2							128×256 128×256
S3	GE GE	3 3	16 16	6 6	2 1	1 2	64×128 64×128
S4	GE GE	3 3	64 64	6 6	2 1	1 1	32×64 32×64
S5	GE GE CE	3 3 3	128 128 128	6 6 -	2 1 1	1 3 1	16×32 16×32 16×32

2.2 학습 데이터 세트 및 클래스 적응적 손실함수

현재 의미론적 영상 분할의 연구에는 다양한 도심 주행 환경에서 취득한 Cityscapes[1]와 KITTI semantic segmentation benchmark[2]와 같은 데이터 세트가 주로 활용된다. 하지만 이와 달리 야지 환경은 주행 가능한 식물과 주행 불가능한 식물을 구분해야하고 아스팔트 길 뿐만 아니라 흙길, 자갈길 등 다양한 지형이 존재하는 비정형 환경이다. 본 연구진이 확인한 현재까지 공개된 야지환경 데이터 세트는 다음 Table 3과 같다.

Table 3. Off-road environment dataset

	데이터 수량 (학습용)	해상도 ($H \times W$)	클래스 수
Freiburg Forest[3]	366	487×880	6
RELLIS-3D[4]	6,235	1200×1920	20
RUGD[5]	7,422	550×688	24
TAS500[6]	540	620×2026	21

Freiburg Forest[3]는 산림 환경에서의 카메라 이미지를 제공한다. 카메라 해상도가 낮고 가을 혹은 겨울의 계절에만 취득하여 식물과 지형의 다양성이 부족하고 동일한 구간을 반복하며 데이터 수량 또한 부족하다.

RELLIS-3D[6]는 다른 데이터와 다르게 카메라 이미지뿐만 아니라 라이다 데이터를 함께 제공한다. 환경은 낮은 초록색 풀과 나뭇잎 없이 가지만 있는 나무들이 주로 존재하며 넓은 초원과 같은 환경이다. 같은 장소에서 취득했기에 유사한 장면의 반복이 많다.

RUGD[5]는 RELLIS-3D와는 달리 다양한 환경이 존재하고 특히 깊은 숲 속과 같은 환경이 존재한다. 냇가(creek), 공원(park), 숲길(trail) 그리고 마을(village) 총 4가지 환경에 대해서 각각 평균 3분 길이의 연속된 이미지를 취득하였다. 흙 길, 자갈길, 아스팔트 등 지형의 다양성이 존재하나 장애물이 많지 않고 낮은 위치에서 취득하여 시야가 좁고 이미지 해상도가 낮다.

TAS500[6]은 지형과 식물 모두에 대해 정밀한 레이블을 제공한다. 주행 가능한 표면을 4개의 하위 클래스로 식물을 7개의 하위 클래스로 세분화하였다. 취득 환경은 봄, 여름, 가을 계절에 대해서 맑은 날과 흐린 날을 포함하여 총 5개의 연속된 영상을 취득하였다. 총 약 50분 분량으로 여러 장소와 다양한 환경의 데이터가 존재하나 레이블링 된 데이터는 약 500장으로 수가 적다.

본 논문에서는 최근까지 공개된 야지 환경에서의 자율 주행 데이터 세트 중 적당하다고 판단한 RUGD, TAS500에 대하여 학습을 진행하고 성능 확인을 진행하였다. 여기서 Table 4와 같이 이동 가능 공간 탐지에 맞게 몇몇 유사한 클래스는 통일시켜 학습을 진행하였다. 야지환경에서는 클래스 간의 불균형이 매우 크기에 학습이 적절하게 이뤄지기 위해서 학습 클래스를 지면, 주행 가능 식물, 주행 불가능 식물, 하늘, 기타 장애물로 총 5가지로 재정의하여 불균형 문제를 해결하였다.

기존 도심 환경 의미상 영상분할 알고리즘에서 사용되는 손실함수는 Softmax cross-entropy이다. 본 논문에서는 Fig. 1과 같이 야지 환경에 대한 데이터 학습 시 클래스 불균형에 따른 문제를 해결하기 위해 다음과 같은 클래스 적응적 손실함수를 제안한다.

$$Loss = \sum_{p} \left[-\log \left(\frac{e^{s_p}}{\sum_{j}^{C} e^{s_j}} \right) \circ (W_c) \right] \quad (1)$$

여기서 p는 입력 이미지에 대한 픽셀들을 의미하며 C는 모든 클래스에 대한 집합이다. S_p는 참값(Ground truth)을 의미하고 S_j는 j번째 클래스에 대한 네트워크 결과값(score) 그리고 W_c는 각 클래스에 대한 가중치 값이다. 클래스 가중치 값은 두 가지 방식으로 정할 수 있다. W_c^d는 입력 영상에서 각 해당 클래스에 대한 비율이고 W_c^f는 전체 데이터세트 분포에서의 해당 클래스 비율이다. 본 논문에서는 두 가지 방식에 대해 각각 실험을 진행하고 Table 5에서 결과를 비교하였고 W_c^f의 경우 약간의 성능 우세함을 보여 해당 손실 함수로 실험을 진행하였다.

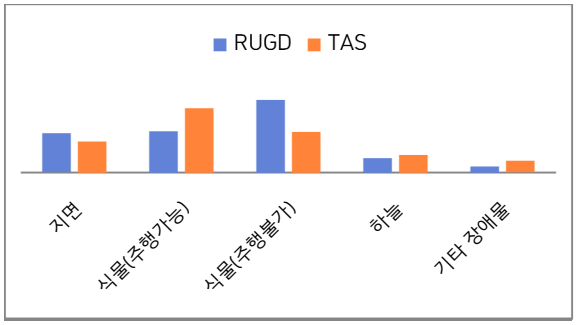

Fig. 1. Class distribution for RUGD and TAS500

Table 4. Pooled class distribution for RUGD[RU] and TAS500[TAS]

클래스 재정의 (TAS500, RUGD)					
지면 (ground)	식물 (vegetation)		하늘 (sky)	기타 장애물 (obstacle)	
	주행 가능 (drivable)	주행 불가능 (non-drivable)			
dirt[RU] soil[TAS] sand[TAS,RU] gravel[TAS, RU] asphalt[TAS,RU] concrete[RU] mulch[RU] rockbed[RU]	grass[RU] low grass[TAS] high grass[TAS]	bush[TAS,RU] tree[RU] trunk[TAS] crown[TAS] forest[TAS]	sky[TAS,RU]	person[TAS,RU] vehicle[RU] bicycle[RU] car[TAS] bus[TAS] pole[RU,TAS] sign[TAS] rock[RU] table[RU]	building[TAS,RU] fence[TAS,RU] wall[TAS] container[RU] log[RU] sign[RU] rock[RU] bridge[RU] water[RU]

3. 실험 결과

본 논문에서는 RUGD 데이터 세트와 TAS500 데이터 세트에 대해서 실험을 진행하였다. RUGD 데이터 세트에서는 4765장의 이미지를 학습(training)으로 733장의 이미지를 검증(validation) 그리고 1924장의 이미지를 테스트(testing)으로 활용하였다. TAS500 데이터 세트에서는 440장의 이미지를 학습(training)으로 100장의 이미지를 검증(validation)으로 활용하였다. 입력 영상 크기($w \times h$)의 경우 RUGD 데이터 세트는 688×550이며 TAS500 데이터 세트는 2026×620이다. 네트워크 학습에는 사전 학습 없이 kaiming normal 방식으로 초기화된 네트워크에서 학습을 시작하였으며 확률적 경사 하강법 알고리즘으로 최적화를 진행하였다. 학습 데이터 증강을 위해 본 논문에서는 수평 플립(horizontally flip)과 함께 0.75, 1, 1.25, 1.5 중 랜덤하게 하나의 값을 선택하여 해당 배율로 영상 크기를 변경한다. 입력 영상 크기보다 커지는 경우는 랜덤하게 crop을 진행하고 작은 경우는 padding을 진행하여 네트워크 입력 크기에 맞게 조절한다. 정확도 평가는 각 클래스에 대해 Intersection-over-Union(IoU)를 채택하였다.

$$IoU = \frac{TP}{TP + FP + FN} \tag{2}$$

여기서 TP, FP 그리고 FN는 각각 true positive, false positive 그리고 false negative를 의미한다.

정확도 평가와 함께 본 논문에서는 네트워크의 추론 시간 측정을 위해 5000번 반복하여 수행한 결과의 평균을 구하여 평균 추론 소요시간과 fps(Frame Per Second)를 구하였다. 본 논문의 실험은 Tensorflow 기반으로 추론 시간 측정은 NVIDIA GeForce GTX 2080Ti with the CUDA 9.0, CUDNN 7.0 and TensorRT v5.1.5에서 이뤄졌다.

본 논문에서는 도심 주행 상황에서 최적화되고 경량화된 알고리즘인 BiSeNetV2 네트워크를 야지 환경이라는 특성에 맞게 구조를 변경하여 실험을 진행하였다. Fig. 1과 같이 야지 환경의 데이터 세트는 도심 환경과 확연한 차이가 있다. 도심 환경에서는 차량, 사람과 같은 장애물에 해당하는 객체들이 다수 존재하지만 야지 환경에서는 지면과 식물과 같은 유사한 배경이 반복된다. 따라서 야지 환경에서의 나눠진 장애물들을 하나의 클래스로 묶었으며 식물에 대해서는 주행 가능한지 주행 불가능한지 세부 구분이 가능하도록 클래스를 나누고 결과를 확인하였다.

Table 5는 클래스 재정의와 손실 함수 변경에 따른 성능 결과를 나타낸다. CE는 기존 손실 함수를 나타내며 W_c^f와 W_c^d는 변형된 손실함수이다. 클래스 재정의에 따른 결과를 보면 RUGD는 25개, TAS500은 총

Fig. 2. Samples images from cityscapes, TAS500, and RUGD

Table 5. Performance evaluation on validation set

Method	class	mIOU (%)	
		RUGD	TAS500
CE (baseline)	25/23	36.95	59.68
	6	73.92	85.14
W_c^d	6	74.25	86.62
W_c^f		75.61	86.77

Table 6. Ablations on validation set

Method	mIOU (%)		fps
	RUGD	TAS500	
baseline	73.92	85.14	130
ν = 1/4, σ = 2, W_c^f	75.61	86.77	130
ν = 1/4, σ = 4, W_c^f	75.57	87.11	123
ν = 1/8, σ = 2, W_c^f	75.34	85.91	140
ν = 1/8, σ = 4, W_c^f	74.67	86.73	135

23개의 클래스에 대해 분류해야한다. Table 4와 같이 table, bridge, container등 장애물에 대해 불필요한 구분으로 전체적인 평균 성능은 낮은 결과를 보여준다. 손실 함수 실험 결과를 보면 기존 방식에 비해 변형된 방식들은 개선된 성능을 보였다. 도심환경에서는 차선 혹은 표지판, 차량처럼 일부 구간에만 등장하는 클래스들이 많이 존재하지만 야지환경에서는 같은 배경이 주로 반복되는 환경이기에 전체 데이터에서의 클래스 분포와 학습 배치(batch) 크기만큼 샘플링 된 데이터에서의 분포가 크게 다르지 않다. 따라서, W_c^f와 W_c^d 간의 성능 차이는 비슷하다. 변경된 클래스 방식과 손실 함수 W_c^f에 따라 본 논문에서는 Semantic branch 구조의 채널 용량 비율(channel capacity ratio) ν와 최종 네트워크 결과 출력을 위한 Head 모듈의 확장 비율(expansion ratio) σ를 변경하였으며 연산량과 성능 분석을 통해 최적의 구조를 확인하였다. 채널 용량 비율이란 Semantic branch 네트워크에서 첫 번째, 두 번째 단계에서의 Detail branch 네트워크 대비 채널 용량의 비율을 말한다. Semantic branch에서는 고수준 의미의 정보를 얻는 역할을 하기에 채널이 많다면 더 풍부한 의미 정보를 뽑을 수 있지만 연산량은 증가될 것이다. 따라서 야지 환경의 특성상 Semantic branch의 채널 용량 확장이 성능 개선에 대한 영향이 적다면 이를 줄여 실시간성을 더 확보할 수 있다.

Head 모듈의 확장 비율 σ는 네트워크의 마지막 단계 이후 최종 출력을 위한 Conv2d 필터들의 출력 채널 비율로 네트워크 표현 능력에 영향을 준다. 따라서 확장 비율을 조절하면 야지 환경 특성에 맞는 효율적인 구성이 가능하다. Table 6은 네트워크 구조 변경에 따른 성능 결과이며 Table 6, 7, 8은 클래스별로 성능을 분석한 결과이다. 여기서 RUGD 데이터세트의 실험은 두 가지로 구분되는데 'park-8'는 숲 보다는 일반 아스팔트 도로와 함께 공원 내 풀밭을 함께 포함하는 세트이고 'trail-5'는 더 깊은 숲 속을 주행하는 세트이다. Table 6에서 ν = 1/4은 Semantic branch의 네트워크 채널 용량이 Detail branch에 비해 0.25배로 경량화된 것을 의미한다. ν = 1/8는 0.125배로 더욱더 경량

Table 7. Performance evaluation for RUGD park-8

Class	mIOU (%)			
	ν = 1/4, σ = 2	ν = 1/4, σ = 4	ν = 1/8, σ = 2	ν = 1/8, σ = 4
ground	81.37	81.28	81.26	81.84
vegetation (drivable)	91.57	91.39	91.38	91.62
vegetation (non-drivable)	89.89	90.23	90.07	90.09
sky	72.38	71.73	71.34	71.12
obstacle	45.49	47.05	45.99	45.49
total	76.14	76.34	76.01	76.03

Table 8. Performance evaluation for RUGD trail-5

Class	mIOU (%)			
	ν = 1/4, σ = 2	ν = 1/4, σ = 4	ν = 1/8, σ = 2	ν = 1/8, σ = 4
ground	90.74	90.42	90.59	90.94
vegetation (drivable)	78.22	78.63	76.00	74.93
vegetation (non-drivable)	91.97	92.23	91.49	90.72
sky	63.07	61.49	62.58	58.42
obstacle	39.56	38.72	40.78	37.44
total	72.71	72.3	72.29	70.49

Table 9. Performance evaluation for TAS500

Class	mIOU (%)			
	ν = 1/4, σ = 2	ν = 1/4, σ = 4	ν = 1/8, σ = 2	ν = 1/8, σ = 4
ground	89.76	89.55	88.78	89.78
vegetation (drivable)	89.75	90.44	89.36	90.07
vegetation (non-drivable)	86.63	88.12	86.09	86.94
sky	95.60	95.60	95.44	95.32
obstacle	72.12	71.83	69.90	71.54
total	86.77	87.11	85.91	86.73

화된 것으로 이에 따라 fps가 더 증가한 것을 알 수 있다. σ = 2는 Head 모듈의 채널을 2배로 확장한 것이며 σ = 4는 4배로 확장한 것이다. 따라서 연산량 증가로 fps가 감소한다. RUGD 데이터 세트에서의 결과를 보면 연산량이 가장 많은 ν = 1/4, σ = 4 보다 ν = 1/4, σ = 2의 성능이 더 좋은 것을 알 수 있다.

Table 6, 8의 TAS500 결과를 보면 Semantic branch의 Head 모듈의 확장이 성능 개선에 영향을 주지 못하는 것으로 생각할 수 있으나 Table 6, 7을 보면 'trail-5'의 하늘에 대한 성능 저하가 원인으로 이는 'trail-5' 데이터 세트의 하늘 클래스에 대한 부정확한 Ground truth으로 발생한다. 더 정확한 Ground truth를 갖는 TAS500의 경우에서는 성능 개선이 이루어진 것이 확인 가능하다. 확장 비율 σ에 따른 성능 변화는 ν = 1/8의 경우에 변동이 더 큰 것을 확인할 수 있다. 네트워크 채널 용량을 ν = 1/8으로 경량화를 진행하여도 Head 모듈의 확장 성능을 4배로 증가 시키면 성능 저하를 낮춘 상태로 더 높은 fps를 확보할 수 있다. 특히 주행 가능 식물 클래스, 주행 불가능 식물 클래스에 대한 정확도에서는 ν = 1/4, σ = 2 구조보다 ν = 1/8, σ = 4 구조에서 더 좋은 성능을 보이는데 TAS500의 정교한 Ground truth로 학습이 잘 진행되어 강화된 Head 모듈의 네트워크 표현 능력 개선이 네트워크 경량화로 인한 성능 저하를 보완시킨 것으로 확인된다. Fig. 3의 결과는 ν = 1/4, σ = 2으로 네트워크를 구성한 RUGD validation 데이터 세트에 대한 실험 결과의 예시이다. Fig. 3의 상위 3줄은 'park-8'에 대한 실험 결과이고 나머지 하위 부분은 'trail-5'에 대한 결과이다. Prediction_600은 최대 epoch를 600으로 Prediction_1000은 최대 epoch를 1000으로 학습을 진행한 결과이다. 'park-8'에 대한 결과의 경우 풀과 수풀, 나무등과 같은 식물에 대해서 주행 가능 영역과 주행 불가능 클래스를 정확히 구분한 것을 알 수 있다. Fig. 3의 2번째 줄 결과를 보면 나무 기둥을 주행 불가능 식물 클래스 혹은 장애물 클래스 사이에서 혼동되는 경우가 있다. Table 7을 보면 지면 클래스의 정확도가 'trail-5' 보다 상대적으로 낮게 측정된다. 이는 Fig. 3을 보면 아스팔트 길은 잘 구분하지만 3번째 줄 결과에서 공원 내 흙길을 구분하는 경우 이동가능한 식물클래스와 혼동을 하는 경우가 존재하기 때문이다. 'trail-5'의 경우 Table 7과 Fig. 3의 마지막 줄 그림을 보면 지면 클래스와 주행불가능한 식물 클래스는 잘 구분 하지만 주행 가능한 식물 클래스의

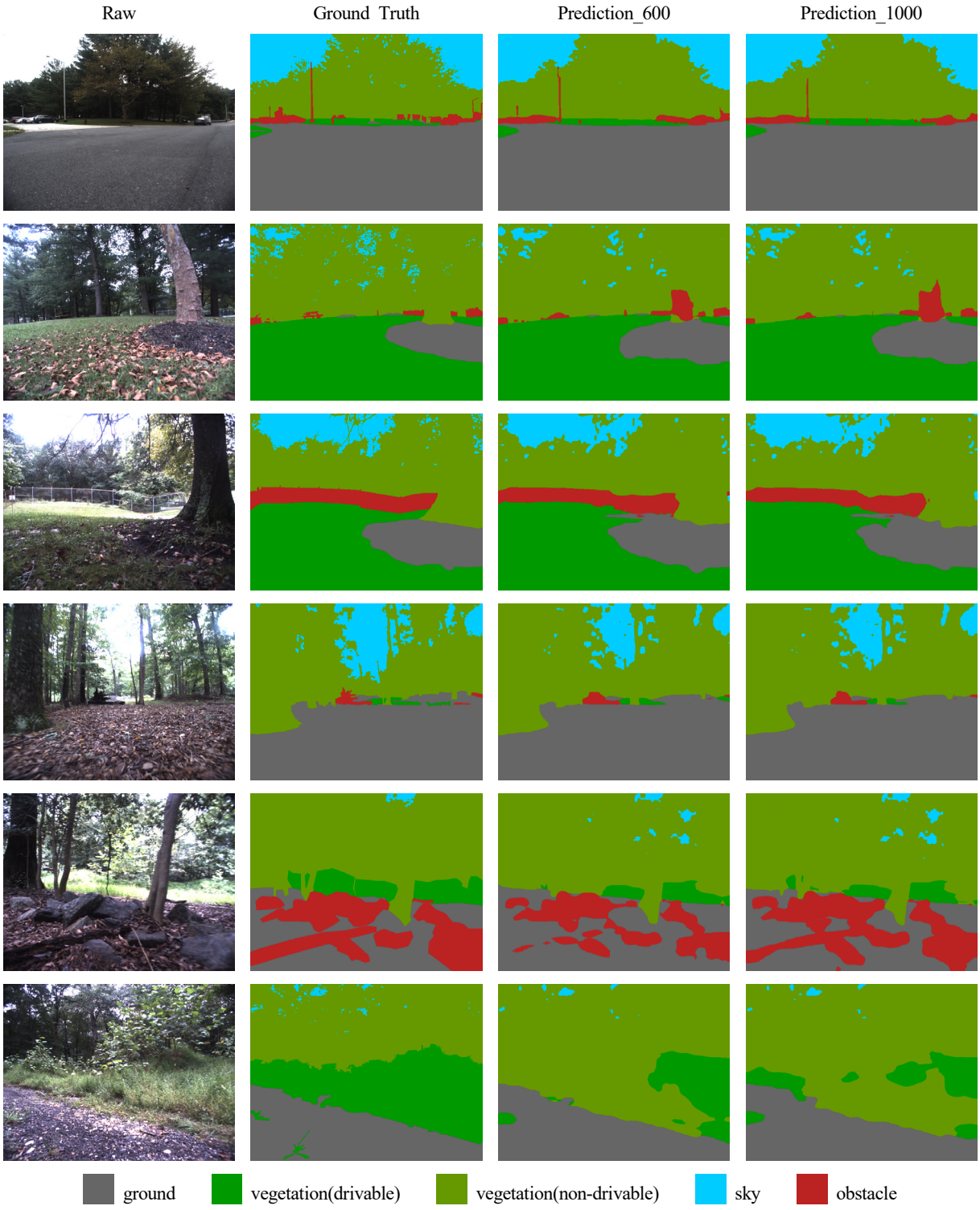

Fig. 3. Visualization examples on the RUGD validation set(park-8. trail-5)

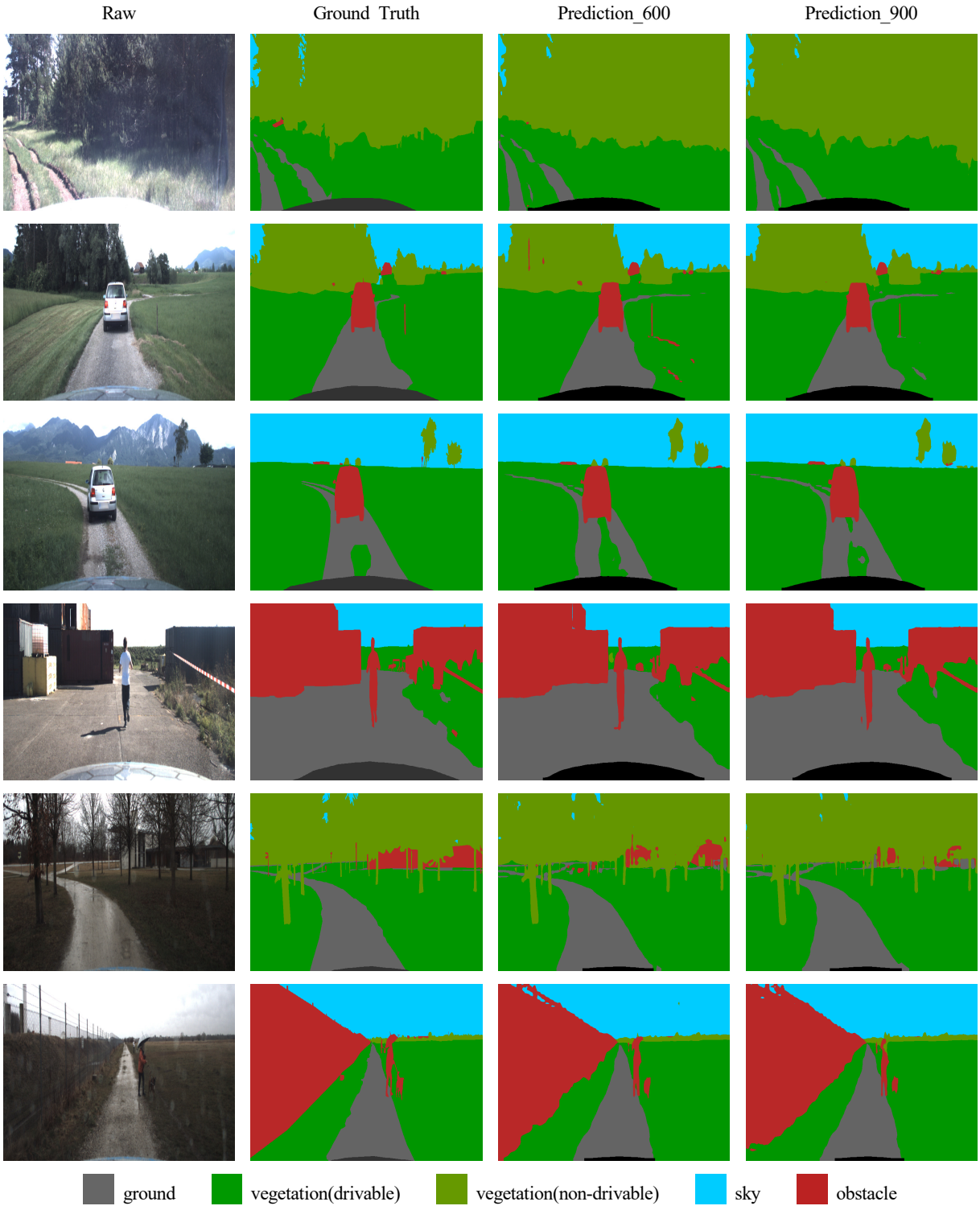

Fig. 4. Visualization examples on the TAS500 validation set

일정 높이 이상의 수풀 구분에서는 성능저하가 일어난다. Fig. 3의 5번째 줄을 보면 Ground truth의 하늘 클래스가 부정확한 것을 확인 할 수 있는데 이런 경우 네트워크에서 더 정확하게 하늘 클래스를 추정하여도 성능은 더 낮게 측정된다.

Fig. 4의 결과는 ν = 1/4, σ = 2으로 네트워크를 구성한 TAS500 validation 데이터 세트에 대한 실험 결과의 예시 이미지이다. Prediction_600은 최대 epoch를 600으로 Prediction_900은 최대 epoch를 900으로 학습을 진행한 결과이다. Fig. 4를 보면 주행 가능한 식물 클래스를 잘 구분하고 있으며 차량과 사람과 같은 장애물 클래스 또한 잘 구분하고 있다. 다만 Fig. 4의 마지막 줄에서 나무에 가려진 원거리 건물과 같은 경우 혹은 원거리의 사람과 같은 경우에는 정확한 클래스 구분에 어려움을 갖는다.

Fig. 3, 4에서 정성적 결과 평가를 위해 epoch를 달리하여 결과를 확인하였다. 일반적으로 eopch가 증가함에 따라 정량적으로 전체적인 성능은 더 오르지만 몇몇 구분이 어려운 일부 이미지에 있어서는 정성적으로 봤을 때 성능이 저하되는 경우를 확인하였다. Fig. 3을 보면 epoch에 증가함에 따라 전반적으로 성능이 개선된 것으로 보이지만 마지막 줄의 경우 식물의 세부 분류에 어려움이 있고 실제로 epoch가 더 진행될수록 성능이 저하된 것을 볼 수 있다. Fig. 4에서는 3번째 줄의 지면과 낮고 드물게 있는 식물과의 구분 그리고 마지막 줄의 멀리 있어 구분이 어려움이 있는 사람과 식물 간의 분류에서 epoch가 진행 될수록 성능 저하가 생기는 것을 확인 하였다.

4. 결 론

본 논문에서는 실시간 의미론적 영상 분할 알고리즘인 BiSeNetV2 알고리즘 기반으로 이동 가능 공간 탐지를 위한 딥러닝 네트워크를 제안하였다. 공개된 야지 환경 주행 데이터를 분석하여 클래스 불균형의 문제점을 파악하고 이를 해결하기 위해 클래스를 재정의하고 손실 함수를 변형하였다. 이에 맞게 네트워크 구조 또한 변경하여 약 1.6 %의 성능을 향상하였고 경량화를 통해 더 높은 실시간성 보장하였다. 추후 야지환경에 대한 추가 데이터 세트 확보와 주행 가능성 구분이 어려운 다양한 식물에 대한 레이블링 방법 및 정확도 확인을 위한 추가 연구가 필요하다.

후 기

본 논문은 딥러닝 기반 야지환경 영상인식 기술 연구의 일환으로 한화디펜스의 지원을 받아 수행된 연구입니다.

References

[1] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S and Schiele B, “The Cityscapes Dataset for Semantic Urban Scene Understanding,” In: Proc. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016.

[2] Geiger A, Lenz P and Urtasun R, “Are we ready for Autonomous Driving? the Kitti Vision Benchmark Suite,” In: Proc. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2012.

[3] A. Valada, G. Oliveira, T. Brox, and W. Burgard, “Deep Multispectral Semantic Scene Understanding of Forested Environments using Multimodal Fusion,” in International Symposium on Experimental Robotics (ISER), 2016.

[4] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, “Rellis-3d Dataset: Data, Benchmarks and Analysis,” arXiv, 2020.

[5] A. Valada, G. Oliveira, T. Brox, and W. Burgard, “Deep Multispectral Semantic Scene Understanding of Forested Environments using Multimodal Fusion,” in International Symposium on Experimental Robotics (ISER), 2016.

[6] K. A. Metzger, P. Mortimer, and H.-J. Wuensche, “A Fine-Grained Dataset and its Efficient Semantic Segmentation for Unstructured Driving Scenarios,” in International Conference on Pattern Recognition (ICPR), Milano, Italy, Jan. 2021.

[7] C. Yu, J. Wang et al., “BiSeNet: Bilateral Segmentation Network for Real-time Semantic Segmentation,” in European Conference on Computer Vision(ECCV), 2018.

[8] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang, “Bisenetv2: Bilateral Network with Guided

Aggregation for Real-time Semantic Segmentation," arXiv, 2020.

[9] He K, Zhang X, Ren S and Sun J, "Deep Residual Learning for Image Recognition," In: Proc. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016.

[10] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M and Adam H, "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv, 2017.

[11] Li G, Yun I, Kim J, and Kim J, "Dabnet: Depth-wise Asymmetric Bottleneck for Real-time Semantic Segmentation," In: Proc. British Machine Vision Conference(BMVC), 2019.

[12] Long J, Shelhamer E, Darrel T, "Fully Convolutional Networks for Semantic Segmentation," In: Proc. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2015.

[13] Ronneberger O, Fischer P, Brox T, "U-net: Convolutional Networks for Biomedical Image Segmentation," In: Proc. International Conference on Medical Image Computing and Computer-Assisted Intervention(MICCAI), 2015.

[14] Yu C, Wang J, Peng C, Gao C, Yu G, Sang N, "Learning a Discriminative Feature Network for Semantic Segmentation," In: Proc. IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018.

[15] Wang J, Sun K, Cheng T, Jiang B, Deng C, Zhao Y, Liu D, Mu Y, Tan M, Wang X, Liu W, Xiao B, "Deep High-resolution Representation Learning for Visual Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI), 2019.