https://doi.org/10.7840/kics.2024.49.10.1345

자율주행을 위한 BEV Segmentation에 관한 연구

전 우 민◆, 이 성 진°

Research on BEV Segmentation for Autonomous Driving

Woomin Jun◆, Sungjin Lee°

요 약

자율주행 기술에서 Ego 차량 주변의 3D 환경에 대한 2D 조향도 작성은 차량 조향 및 속도 제어를 용이하게 한다. 특히, 주변 도로 환경의 객체, 그 위치와 크기를 실시간으로 완벽하게 2D 조향도로 표현하는 BEV Segmentation 기술은 안전한 주행을 위해서 필수적이다. 본 연구는 자율주행 임베디드 환경에 실시간으로 동작할 수 있으며 적은 용량을 가지고도 높은 정확도를 가지는 BEV Segmentation 기술과 해당 모델 최적화 기술에 대해 다루었다. 정확도 개선을 위해 BEV에 사용되는 image encoder에 다양한 backbone을 사용하여 해당 BEV Segmentation 모델에서 기존 mIoU 성능을 능가하는 기술 조합을 도출하였다. 모델 크기 및 동작 시간 축소를 위해서, Quantization (FP16)을 진행하였다. 실험 결과 최대 44.9 mIoU 를 달성하여, 기존 기술 대비 17.8%의 mIoU 개선 효과를 달성하였다. 반면, Quantization을 통해 제안하는 향상된 정확도 모델의 동작시간을 3.6% 감소시켰고, 모델 크기는 약 50% 축소 성과를 도출할 수 있었다. 해당 기법을 NVIDIA AGX Orin 기반의 On-Device에 탑재하여 전력 수급에 따른 성능 분석 결과, 충분한 전력 공급이 지연시간 감소에 중요하게 작용함을 알 수 있었다.

키워드 : 자율주행, 조감도, 이미지 분할
Key Words : Autonomous Driving, BEV, Segmentation

ABSTRACT

In autonomous driving technology, creating a 2D bird's eye view (BEV) map of the 3D environment surrounding the Ego vehicle facilitates vehicle steering and speed control. Particularly, the BEV Segmentation technology, which perfectly represents the objects in the surrounding road environment, including their positions and sizes, in real-time on a 2D map, is essential for safe driving. This study addresses the optimization of a BEV Segmentation model that operates in real-time in autonomous driving embedded environments and achieves high accuracy with a small footprint. To improve accuracy, various backbones were employed in the image encoder used for BEV segmentation, leading to a combination of techniques that outperformed the previous mIoU performance of the model. For model size and operation time reduction, Quantization was conducted. Experimental results achieved an mean Intersection over Union (mIoU) of 44.9, showing a 17.8% improvement in mIoU over existing technologies. On the other hand, through Quantization, the proposed enhanced accuracy model achieved a 3.6% reduction in latency and approximately 50% reduction in model size. By implementing this technique on an NVIDIA AGX Orin-based on-device system and analyzing performance in relation to power supply, it was found that sufficient power supply plays a crucial role in reducing latency.

◆ First Author : Dong-Seoul University Department of Electric Engineering, aplus912@naver.com, 학생회원
° Corresponding Author : Dong Seoul University Department of Electronic Engineering, sungjinlee@du.ac.kr, 정회원
논문번호 : 202405-105-A-RN, Received May 16, 2024; Revised June 28, 2024; Accepted July 3, 2024

Ⅰ. 서 론

인공지능 기술의 발전으로 최근 자율주행 및 로봇 기술은 그 상용화로 향해 가속화되고 있다[1,2]. 인공지능 발전의 초반에는 영상인식 기술로부터 시작하여 자율주행의 초기 상황인지를 위한 영상분류, 객체 인식, 객체 분할 등의 기술이 주로 발전하였으나, 이제는 강화학습, Transformer 기반의 시계열 처리, Multi-modal Learning을 기반으로 Sensor Fusion 기반 상황인지, 경로 계획, End-to-End Learning으로까지 확장되어 가는 추세이다[1,2]. 특히, 실제적인 자율주행 결정을 위해서는 카메라 입력 프레임 별 2D Object Detection, 2D Segmentation 기술들 보다는 BEV (Bird's Eye View) 관점에서 입력 프레임을 재해석하여 변환하는 기술이 중요하며, 이와 관련된 정확도 지표 향상 및 임베디드 시스템에서의 실시간 처리 기술이 중요해지고 있다[3-5]. 이처럼 자율주행 기술에서 실시간 딥러닝 경량화 기술 역시 임베디드 시스템에 중요한 요소로 작용하며 관련 기술들, Light Convolution Module, Factorization, Quantization, Pruning, Knolwdge Distillation 등이 연구되어 자율주행 영역에서 관련 기술들이 중요하게 활용 되고 있는 중이다[1,2]. 본 연구에서는 실제적인 자율주행을 위해 중요하게 다루어지는 BEV 변환 기술과 해당 BEV Segmentation 기술의 경량 최적화 기술을 다루었다. 이를 위해 자율주행 안전성 확보를 BEV 변환 정확도, 변환 시간을 측정하며 이에 대한 모델 크기 역시 분석한다.

Ⅱ. 관련 연구

자율주행에서 주변 3D 환경의 2D BEV 변환 기술은 IPM (Inverse Perspective Mapping) 기술을 중심으로 구현되어 왔다[5]. 이는 카메라가 설치되는 내, 외부 매개 변수들과 관련 변환 행렬 연산들을 통해 2D 입력 이미지들을 3D 공간좌표로 변환 후 다시 2D 평면 투영을 통해 BEV 영상을 얻어내는 구조이다. 하지만, 카메라가 설치되는 주행 환경 요인, 기상 조건, 동적 객체 등의 다양한 변이조건들로 인해 그 투영 정확도에 한계가 발생할 수 있다. 하지만 최근 딥러닝 및 컴퓨터 비전 기술의 발전으로 이런 내재적 한계를 지닌 IPM 기반의 BEV 변환 기술에 딥러닝 기술을 적용한 기술적 흐름이 나타났다. 이러한 BEV 변환 기술은 카메라나 라이다와 같은 센서를 중심으로 하는 자신의 차량 (Ego)을 좌표에 맞게 3D 공간을 구현하여 위에서 아래로 보는 시점으로 변환하는 기술이다. 그래서 조감도 (Bird's Eye View) 라고 표현하기도 하며, 이런 시점 변환 특성들을 이용하여 3D Object Detection, Segmentation 등의 작업들로 확장 할 수도 있다. BEV 변환 기술에는 Depth 기반 방식, MLP 기반 방식, Transformer 기반 방식으로 분류될 수 있다.

Depth 기반 방식은 Depth 추론 기술에 기반하여 2D 이미지를 3D 공간상의 좌표로 맵핑 후 이를 BEV 시점으로 변환하는 기술이다. Depth 기반[6-9] 연구에 두 가지 기술 OFT[6] (Orthographic Feature Transform), LSS[7] (Lift-Splat-Shoot)가 있다. OFT는 2D 이미지데이터에서 정사영 맵으로 변환하는 과정이다. LSS은 3단계로 나눌 수 있다. Lift 단계에서는 입력 데이터를 고차원 특징 공간으로 변환하는 단계이고 Splat은 시공간 그리드에 맵핑 하는 단계이고 Shoot은 미래의 상태를 예측하여 경로 계획을 한다. 이러한 LSS은 OFT를 기반으로 발전하여 만들어졌다.

MLP 기반 방식은 2D 이미지에서 추출된 특징을 각 픽셀 위치의 특징 벡터 간 복잡한 관계를 학습하여 BEV 시점에 맞게 재배치합니다. MLP 기반[10-13] 연구에 두 가지 기술 VPN[10] (View Parsing Network), HDMapNet[11]가 있다. VPN은 주변 이미지 특징 맵을 BEV 특징 맵으로 변환하기 위해 두 개의 레이어로 구성된 MLP를 사용한다. 이후 다른 카메라에서 나온 모든 특징 맵을 추가하여 다중 뷰 퓨전을 수행한다. HDMapNet은 이미지 특징 맵와 라이다 특징맵을 퓨전하여 두 개의 레이어로 구성된 MLP로 BEV 특징맵으로 활용할 수 있다.

Transformer 기반 방식은 Transformer 구조를 활용하거나 모방하여 다양한 방법으로 데이터를 이용할 수 있다. Transformer 기반[14-17]연구로는 PolarFormer[15], DETR3D[14]가 있다. DETR3D은 Depth 추론을 사용하지 않고 3D 공간에서 직접 인식을 한다.시스템 구조로는 여러 카메라 이미지에서 특징맵을 얻고 3D 객체 쿼리를 카메라의 변환 행렬을 사용하여 얻어 3D 위치를 2D 이미지들에 연결하여 BEV 특징맵을 얻을 수 있다. PolarFormer은 극좌표가 3D 인식에서는 직관적으로 표현할 수 있기에 극좌표 변환을 적용하였고 성능을 높였다. 모델 구조로는 Cross-Attention 메커니즘을 이용하여 여러 이미지 특징맵을 통합하여 BEV 특징맵을 얻는다.

Ⅲ. 논문 기여사항

본 논문의 주요 기여 사항으로 다음의 3가지를 정리한다.

1) BEV 변환 정확도 개선: IntenImage, Vision-Transformer 기반의 Mix-Transformer 구조, EfficientNet 구조의 비교를 통해 최적의 Image Encoder 구조 결정. 원본 논문보다 더 높은 정확도 수치 달성.
2) 모델 경량 최적화: 모델에 양자화 기술을 적용하여 해당 모델의 정확도 및 지연시간을 측정하여 실시간 자율주행 임베디드 시스템에 경량 최적화하여 적용
3) On-Device에서의 BEV Segmentation 성능 분석: NVIDIA AGX Orin 기기 상에서 해당 모델을 탑재하여 실제 동작 시, 가용성에 대해 분석하였다.

Ⅳ. 시스템 모델

본 논문에서 사용된 모델 구조로는 Lift-Splat-Shoot, HDMapNet를 사용하였다. 그림 1에서는 각각의 시스템 구조를 나타낸다. 그림 1을 보듯이 두 모델은 Encoder, Neural View Transformation, BEV-Decoder 3가지 구조로 나눠진다. 4.1부터 4.3까지 시스템 구조마다 설명한다.

4.1 Encoder

Encoder는 기본적인 구조를 사용하며, 6장의 이미지의 특징을 추출하는 구조로 두 시스템 모두 같은 구조를 채택하였고, 사용한 모델로 EfficientNet[19], Internimage[20], Mix-Transformer[21], ResNeXt[25]를 사용했다.

4.2 Neural View Transformation

Encoder에서 추출된 2D 이미지 특징맵을 BEV로 시점 변환하는 네트워크이다.

LSS에 NVT (Neural View Transformation) 구조는 그림 1을 보듯이 특징맵이 Depth-Conv를 지나 Soft-Max를 통과한 행렬을 Depth 차원으로 사용되고 통과하지 않은 행렬과 Outer Product 하여 Depth 차원을 추가한다. 추가된 Depth 차원을 Voxel 데이터처럼 활용하여 Point Pillars의 Voxel Pooling을 통하여 BEV 시점으로 변환하여 BEV의 특징을 만든다. 위와 같은 구조는 OFT 구조를 기반하여 만들어졌다.

HDMapNet에 NVT 구조는 그림 2를 보듯이 Encoder에서 추출한 특징을 Multi- Layer Perceptron을 사용하여 원근법 시점과 카메라 좌표계 사이의 픽셀 간의 관계를 추론한다. 추론된 특징을 IPM 기술을 사용하여 BEV 시점으로 변환하여 BEV의 특징맵을 만든다.

4.3 BEV-Decoder

BEV-Decoder은 Fully Convolutional Network[22]으로 구성되었고 두 개의 분기로 나눠진다. 각각의 분기에서 Semantic Segmentation 진행하여 두 가지의 예측을 한다. 두 시스템 구조는 같은 BEV-Decoder 구조를 사용하였다.

Ⅴ. 실 험

이미지 크기는 128x352로 설정하였다. 실험 컴퓨팅 환경은 2 way RTX 4090을 사용하여 훈련하고, 1 way RTX 4090을 통해 추론하였으며 PyTorch 기반으로 프로그래밍하였다. 학습을 위해 Cross Entropy Loss와 AdamW Optimizer를 사용하였으며, 학습률 조정은 Step LR 방식을 적용하였다. 각 모델의 성능은 20 Epochs 동안의 최고 성능의 IoU로 평가하였다. BEV Segmentation 성능 분석을 위해 Car, Lane, Crosswalk, Boundary 클래스의 IoU 성능과 이들의 평균인 mIoU 성능을 모델 구조 및 인코더 구조에 따라 표 1, 2, 3에서

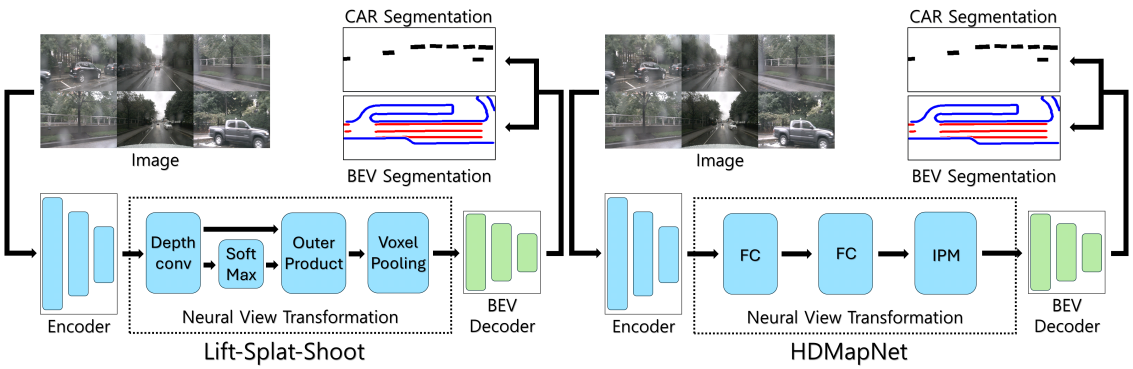

그림 1. Lift-Splat-Shoot와 HDMapNet 시스템 구조도
Fig. 1. System Architecture of HDMapNet and Lift-Splat-Shoot

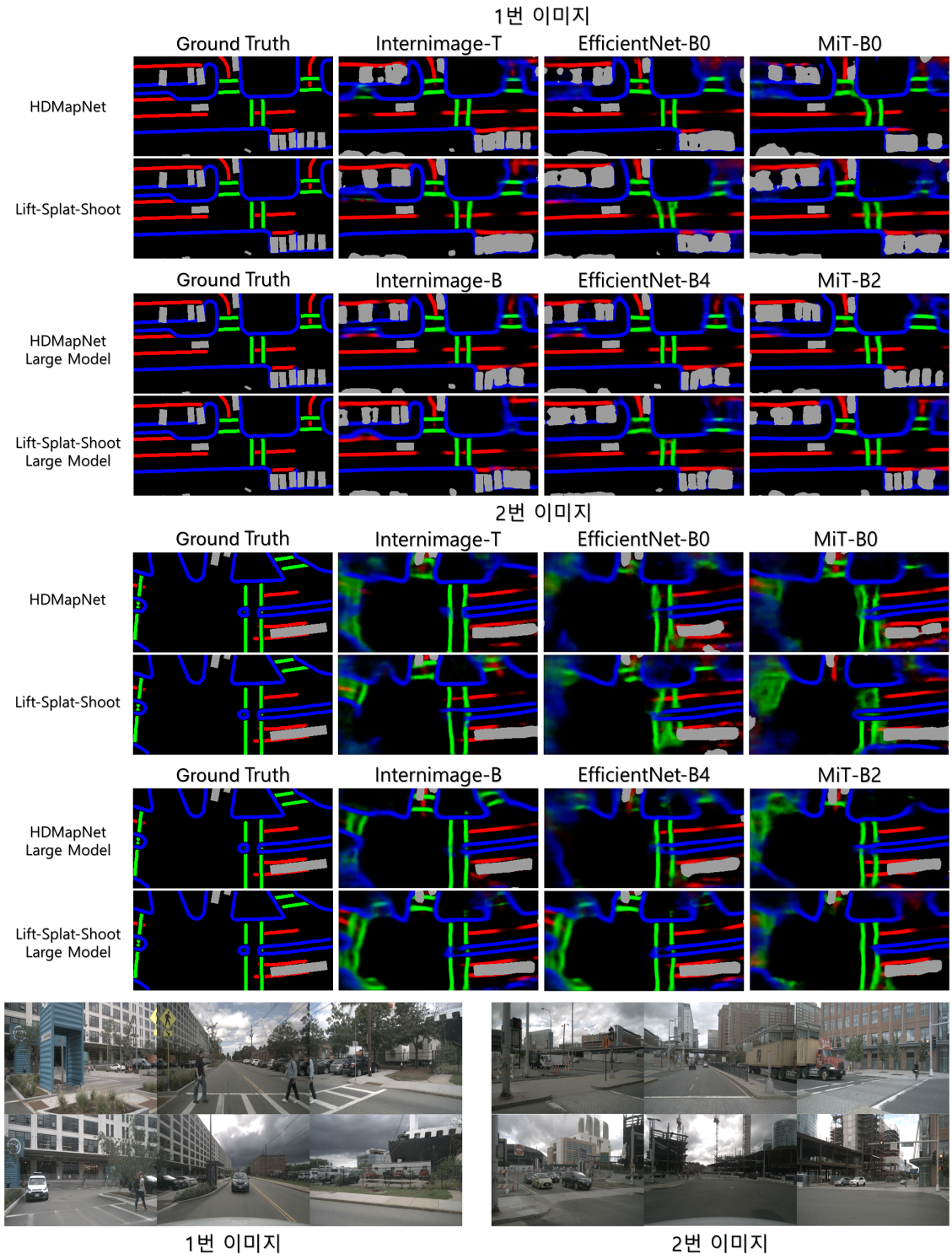

그림 2. 각 모델에 대한 수행 결과 이미지
Fig. 2. Image of performance results for each model

표 1. 지연시간 및 평균 성능 (Latency: ms, Size: MB)
Table 1. Latency and average performance (Latency: ms, Size: MB)

Model	HDMapNet							Lift-Splat-Shoot						
	Small to Medium Model					Large Model		Small to Medium Model					Large Model	
	EffNet B0	EffNet B4	MiT B0	MiT B2	ResXt	InternT	InternB	EffNet B0	EffNet B4	MiT B0	MiT B2	ResXt	InternT	InternB
mIoU	34.6	37.4	32.6	36.6	36.1	41.4	43.2	36.0	38.7	35.1	39.6	37.2	43.3	45.1
Latency	18.5	28.1	15.4	22.3	29.4	28.4	30.8	21.0	30.7	17.6	23.6	31.2	30.9	33.2
Size	292.7	347.4	285.0	365.8	380.2	387.4	646.0	61.1	118.3	53.1	140.1	189.1	161.0	428.5

표 2. 소-중규모 인코더 구조에 따른 성능
Table 2. Performance of BEV Seg in S2M IE models.

Model		Lane	Cross walk	Boundary	Car
HMN	EffNet B0	38.2	17.7	40.7	42.1
	EffNet B4	41.6	20.0	43.2	44.9
	MiT B0	37.1	16.1	38.4	39.1
	MiT B2	40.8	19.5	42.4	44.0
	ResXt	40.1	18.6	42.4	43.6
LSS	EffNet B0	39.7	17.4	42.0	45.8
	EffNet B4	41.7	20.8	43.9	48.6
	MiT B0	38.3	16.1	41.0	45.1
	MiT B2	42.8	23.0	44.5	49.4
	ResXt	40.7	20.5	43.0	44.6

실험을 진행하였다. 비교하는 모델 구조로는 LSS (Lift-Splat-Shoot)와 HMN (HDMapNet) 구조를 사용하였고, 비교하는 인코더 구조로는 EffiNet (EfficientNet), ResNeXt50 (ResXt), Intern (Internimage), MiT (Mix-Transformer)를 사용하였으며 이들의 ImageNet에서의 정확도 및 파라미터 수를 표 6에서 제시하였다. 또한 이 조합들의 성능으로 IoU와 Latency 값을 사용하였다.

모델이 인식하는 대상 클래스 4개의 BEV 변환 및 BEV Segmentation 적용 예시는 그림 2에 제시하였다. 그림 2의 빨간색 선은 Lane, 파란색 선은 Boundary, 초록색 선은 Cross walk, 회색 영역은 Car의 인식 결과이다. 또한 성능 파라미터로서 mIoU 는 모든 클래스들, 즉, Lane, Crosswalk, Boundary, Car에 IoU 성능에 대한 평균 값으로 측정하였고, 지연시간 Lat (Latency)은 ms (mili second)를 단위로 측정하였으며 모델 크기 (size)는 MB (Mega Byte)를 단위로 측정하였다.

최적화 성능은 FP16 Quantization을 적용하여 성능 평가를 하였다[23]. 마지막으로 On-Device 성능 비교는 NVIDIA AGX Orin 기기에 GPU 소비전력 30W, 50W로 제한하여 비교하였다[24].

표 3. 대규모 인코더 구조에 따른 성능
Table 3. Performance of BEV Seg in Large IE models.

Model		Lane	Cross walk	Boundary	Car
HMN	InternT	45.0	25.4	46.7	48.4
	InternB	45.7	28.9	48.1	50.4
LSS	InternT	46.3	27.1	48.0	52.1
	InternB	47.8	30.6	49.1	53.2

표 4. FP16 Quantization을 통한 최적화 성능 (Latency: ms, Size: MB)
Table 4. Optimization Performance through FP16 Quantization (Latency: ms, Size: MB)

Model	Quantized HDMapNet							Quantized Lift-Splat-Shoot						
	EffNet B0	EffNet B4	MiT B0	MiT B2	ResXt	InternT	InternB	EffNet B0	EffNet B4	MiT B0	MiT B2	ResXt	InternT	InternB
mIoU	34.6	37.4	32.6	36.6	36.1	41.4	43.2	36.2	38.1	34.2	39.6	37.2	43.0	44.9
Lat	17.5	28.1	15.3	20.7	27.8	26.9	29.9	19.6	30.3	17.8	23.2	30.7	29.0	32.0
Size	146.4	173.7	142.5	182.9	160.1	193.8	323.3	30.6	59.3	26.6	70.1	94.5	80.6	214.5

표 5. 전력 제한에 따른 On-Device 성능 (Latency: ms)
Table 5. On-Device performance according to power limitation (Latency: ms)

Model	Quantized HDMapNet							Quantized Lift-Splat-Shoot						
	EffNet B0	EffNet B4	MiT B0	MiT B2	ResXt	InternT	InternB	EffNet B0	EffNet B4	MiT B0	MiT B2	ResXt	InternT	InternB
mIoU	34.6	37.4	32.6	36.6	38.6	41.4	43.2	36.2	38.1	34.2	39.6	39.7	43.0	44.9
Lat (30W)	59.32	115.1	49.6	83.9	109.4	103.5	93.5	90.1	130.5	82.5	118.6	126.2	123.5	98.5
Lat (50W)	52.9	98.8	44.9	66.5	94.5	92.8	85.3	74.9	117.5	62.8	94.2	109.2	100.9	82.5

표 6. 인코더마다 ImageNet Top-1 성능
Table 6. ImageNet Top-1 performance per Encoder

Model		Top-1 Acc	Params (M)
S2M (Small to Medium) Model	EfficientNet-B0	77.1	5.3
	EfficientNet-B4	82.9	19.0
	Mix Transformer B0	70.5	3.7
	Mix Transformer B2	81.6	25.4
	ResNext50	77.8	25.0
Large Model	InternImage-T	83.5	30.0
	InternImage-B	84.9	97.0

5.1 BEV 변환모델 및 인코더 모델에 따른 성능

표 1은 표 2, 3에 4개의 클래스 인식 결과 IoU 값의 평균 값인 mIoU와 지연시간 (Latency)를 나타낸다. 표 2는 소-중규모 인코더 모델들 (EfficientNet-B0, EfficientNet-B4, MiT-B0, MiT-B2, ResNeXt50)를 사용한 결과를, 표 3은 대규모 인코더 모델들 (InterImage-T, InternImage-B)를 사용한 결과를 보여준다. 표 1, 2, 3의 BEV Segmentation 모델들, HDMapNet과 LSS 방식을 비교하면, 전체적으로 MLP 기반 HDMapNet 구조보다 Depth 기반 LSS 구조가 성능이 높다는 것을 알 수 있다. 이는 이미지에서 BEV로 변환할 때 2D 이미지에서 FC기반의 View Transformation 방식보다 2D-3D 변환 기반 Voxel 추출 기반의 View Transformation 방식이 mIoU 성능에 더 유리하게 작용하기 때문이다.

클래스별 성능을 보면, Lane, Car, Boundary 클래스에서는 높은 정확도를 보였으나, Crosswalk 클래스에서는 정확도가 낮아 전체 mIoU 성능의 Bottleneck으로 나타난다는 것을 알 수 있다. 이는 Crosswalk의 구조적 특성 상, Lane 과 그 형태가 유사하지만 좀 더 복잡한 구조를 가지고 있어 이미지 해상도에 따른 성능 의존성이 존재하기 때문이다. 이는 그림 2의 2번 예시 이미지 예측 결과들을 통해 확인할 수 있다. 즉, 가까운 거리에서 해당 Crosswalk 클래스가 정확히 인식되지만, 먼 거리에서는 획득되는 이미지가 적은 해상도로 인해 Texture 정보가 부족하게 되기 때문에 해당 클래스의 IoU 성능이 저하되는 것을 확인할 수 있다. 따라서 해당 Crosswalk과 같은 클래스를 정확하게 인식하기 위해서는 기존 연구에서 제안하는 EfficientNet 모델 보다는 좀 더 대규모의 정교한 구조를 가진 InternImage 모델을 인코더 구조로 사용하는 것이 좀 더 안전한 자율주행을 가능하게 할 것으로 판단된다.

표 1에 HDMapNet 성능은 표 6 에 인코더의 성능에 비례하여 성능이 향상된다. 반면, LSS의 성능은 표 6의 EfficientNet-B4와 MiT-B2의 성능 순위와는 다르게 MiT-B2가 EfficientNet-B4 보다 1.2% 더 우수한 성능을 보여준다. 이는 Transformer 구조가 2D 이미지에서 3D 정보 변환 및 BEV 변환 시 더 유리하다는 것을 알 수 있다.

5.2 최적화 적용에 따른 On-Device에서의 성능

표 4는 표 1의 실험군들을 FP16 Quantization을 적용한 결과이며, 표 5은 표 4의 FP16 Quantization 적용 모델들을 NVIDIA AGX Orin 기기에 탑재하여 두 가지 전력 모드, 즉 30W, 50W에 대한 성능을 측정한 결과이다.

FP16 Quantization 적용 후 데이터 타입이 FP32에서 FP16으로 줄어들어 모델 Size가 약 50%정도 감소하고 지연시간은 평균적으로 1ms 정도 감소하게 된다. 반면, mIoU 성능은 HDMapNet과 LSS의 경우 모두 동일하거나 유사한 성능을 보이는 것을 알 수 있다. 그러므로 실 제품 탑재 시, FP16 Quantization 과 같은 최적화 기법 적용이 추천된다.

표 5의 On-Device 탑재 결과를 보면 표 4와 동일한 mIoU을 보이지만 전력 공급이 제한됨에 따라 더 늘어난 지연시간 성능을 보이게 된다. 특히, 두 가지 모델 중 더 낮은 mIoU 성능을 갖는 HDMapNet은 더 높은

mIoU 성능을 갖는 LSS에 비해 평균적으로 30W의 경우 21%, 50W의 경우 17% 정도 감소된 지연시간을 보여주어, HDMapNet이 실시간 시스템에 더 유리하다는 것을 알 수 있다.

또한, 전력 제한 30W와 50W를 비교하면, 50W 전력 조건에서 30W 전력 조건 대비 평균적으로 약 15% 줄어든 지연시간 성능을 보인다는 것을 알 수 있다. 이는 전력 공급량이 지연시간 감소에 중요하게 작용 할 수 있음을 보여주는 것으로, 안전한 자율주행을 위해서는 대용량의 배터리로 충분한 전력을 공급해주는 것이 실시간 인지 판단에 필수적임을 알 수 있다.

이를 통해, 인지 정확도가 중요한 자율주행 시스템에서는 안전성을 위해 LSS 기법을 사용하되 전력을 충분히 공급하여 해당 지연시간을 최소화하여 사용하는 것이 권장된다 할 수 있겠다. 또한, Voxel 기반 View Transformation 모듈에 경량 가속화 기법을 적용하여 LSS 지연시간을 최소화하는 것이 추후 권장되는 연구 방향이라고 할 수 있겠다.

Ⅵ. 결 론

자율주행을 위한 BEV Segmentation 기법들의 동작 방식과 그 정확도, 지연, 메모리 크기 성능에 대해 고찰하고 NVIDIA AGX Orin 기반의 On-Device에서의 전력 사용량에 따른 가용성 성능을 분석하였다. 전체적인 성능에서 Bottleneck으로 나타난 Crosswalk 클래스에 대한 문제를 해결하기 위한 높은 해상도의 이미지를 획득하고, 대규모의 정교한 InternImage와 같은 모델을 사용하는 것이 유리할 것으로 판단된다. 또한 Lift-Splat-Shoot 방식이 HDMapNet 방식 보다 더 정확한 mIoU 성능을 가지며 더 작은 모델 크기를 갖는 장점이 있는 반면 더 느린 지연시간 성능을 갖는 단점이 존재한다는 것을 확인할 수 있었다. 이런 Lift-Splat-Shoot 기법의 단점들을 극복하고 더욱 최적화하기 위해 FP16 Quantization 기법을 적용하면 동일하거나 유사한 mIoU 성능으로 50% 줄어든 메모리 크기 성능과 감소된 지연시간 결과를 얻을 수 있었다. 또한 이를 전력 제한이 있는 NVIDIA AGX Orin 기반의 On-Device에 탑재하면 전력 수급에 따라 불리한 지연시간 성능을 얻을 수 있으므로, 이를 해결하기 위해 충분한 배터리 전력 공급과 해당 BEV Segmentation 모델의 경량 가속화 기법이 더욱 연구되어야 함을 알 수 있었다.

References

[1] E. Yurtserver, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” *IEEE Access*, vol. 8, 2020. (https://doi.org/10.1109/ACCESS.2020.2983149)

[2] J. Zhao, W. Zhao, B. Deng, Z. Wang, F. Zhang, W. Zheng, W. Cao, J. Nan, Y. Lian, and A. F. Burke, “Autonomous driving system: A comprehensive survey,” *Expert Syst. with Appl.*, vol. 242, 2024. (https://doi.org/10.1016/j.eswa.2023.122836)

[3] A. Tampuu, M. Semikin, N, Muhammad, D. Fishman, and T. Matiisen, “A survey of end-to-end driving: Architectures and training methods,” *IEEE Trans. Neural Netw. and Learn. Syst.,* vol. 33, no. 4, Apr. 2022. (https://doi.org/10.1109/TNNLS.2020.3043505)

[4] Z. Yang, X. Jia H. Li, and J. Yan, “LLM4Drive: A survey of large language models for autonomous driving,” *arXiv preprint, axXiv:2311.01043,* 2023. (https://doi.org/10.48550/arXiv.2311.01043)

[5] H. Xu, J. Chen, S. Meng, Y. Wang, and L.-P. Chau, “A survey on occupancy perception for autonomous driving: The information fusion perspective,” *arXiv preprint, axXiv:2405.05173*, 2024. (https://doi.org/10.48550/arXiv.2311.01043)

[6] T. Roddick, A. Kendall, and R. Cipolla, “Orthographic feature transform for monocular 3D object detection,” *arXiv preprint, axXiv:1811.08188,* 2018. (https://doi.org/10.48550/arXiv.1811.08188)

[7] J. Philion and S. Fidler, “Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D,” *ECCV 2020 LNIP*, vol. 12359, pp. 194-210, 2020. (https://doi.org/10.1007/978-3-030-58568-6_12)

[8] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, “Pseudo-LiDAR from visual depth estimation: Bridging the gap in 3D object detection for

autonomous driving," *IEEE/CVF Conf. CVPR*, pp. 8445-8453, 2019.
(https://doi.org/10.1109/CVPR.2019.00864)

[9] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, "BEVDet: High-performance multi-camera 3D object detection in bird-eye-view," *arXiv preprint, axXiv:2112.11790*, 2022.
(https://doi.org/10.48550/arXiv.2112.11790)

[10] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robotics and Automat. Lett.*, vol. 5, no. 3, pp. 4867-4873, 2020.
(https://doi.org/10.1109/LRA.2020.3004325)

[11] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "HdMapNet: An online HD map construction and evaluation framework," *IEEE ICRA*, 2022.
(https://doi.org/10.1109/ICRA46639.2022.9812383)

[12] C. Lu, M. J. G. ven de Molengraft, and G. Dubblman, "Monocular semantic occupancy grid mapping with convolutional variational encoder-decoder networks," *IEEE Robot. and Automat. Lett.*, vol. 4, no. 2, 2019.
(https://doi.org/10.1109/LRA.2019.2891028)

[13] T. Roddick and R. Cipolla "Predicting semantic map representations from images using pyramid occupancy networks," *IEEE/CVF Conf. CVPR*, pp. 11138-11147, 2020.
(https://doi.org/10.1109/CVPR42600.2020.01115)

[14] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "DETR3D: 3D object detection from multi-view images via 3D-to-2D queries," in *Proc. 5th Conf. Robot Learn.*, *PMLR* vol. 164, pp. 180-191, 2022.
(https://doi.org/10.48550/arXiv.2110.06922)

[15] Y. Jiang, L. Zhang, Z. Miao, X. Zhu, J. Gao, W. Hu, and Y-G. Jiang "PolarFormer: Multi-camera 3D object detection with polar transformer," *AAAI Conf.* vol. 37 no. 1, pp. 1042-1050, 2023.
(https://doi.org/10.1609/aaai.v37i1.25185)

[16] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," *ECCV 2022 LNCS*, vol. 13669, pp. 1-18, 2022.
(https://doi.org/10.1007/978-3-031-20077-9_1)

[17] Y. Liu, T. Wang, X. Zhang, and J. sun, "PETR: Position embedding transformation for multi-view 3D object detection," *ECCV 2022 LNCS*, vol. 13687, pp. 531-548, 2022.
(https://doi.org/10.1007/978-3-031-19812-0_31)

[18] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," *IEEE/CVF Conf. CVPR*, pp. 11621-11631, 2020.
(https://doi.org/10.1109/CVPR42600.2020.01164)

[19] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *ICML, 2019*, vol. 97, pp. 6105-6114, 2019.
(https://doi.org/10.48550/arXiv.1905.11946)

[20] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, H. Li, X. Wang, and Y. Qiao, "InternImage: Exploring large-scale vision foundation models with deformable convolutions," *IEEE/CVF Conf. CVPR*, pp. 14408-14419, 2023.
(https://doi.org/10.1109/CVPR52729.2023.01385)

[21] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic Segmentation with transformers," *Advances in NeurIPS*, vol. 34, pp. 12077-12090, 2021.
(https://doi.org/10.48550/arXiv.2105.15203)

[22] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE/CVF Conf. CVPR*, pp. 3431-3440, 2015.
(https://doi.org/10.1109/CVPR.2015.7298965)

[23] H. Lee, N. Lee, and S. Lee, "A method of deep learning model optimization for image

classification on edge device," *Sensors*, vol. 22, no. 19, 7344, 2022. (https://doi.org/10.3390/s22197344)

[24] https://www.nvidia.com/ko-kr/autonomous-machines/embedded-systems/jetson-orin

[25] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *IEEE/CVF Conf. CVPR*, pp. 1492-1500, 2017. (https://doi.org/10.1109/CVPR.2017.634)

전 우 민 (Woomin Jun)

2023년 2월 : 동서울대학교 전자 공학과 졸업

2024년 3월~현재 : 동서울대학교 전자공학과 학사과정

<관심분야> 딥러닝, 영상인식, 자율주행, 조감도 인식

이 성 진 (Sungjin Lee)

2011년 8월 : 연세대학교 전자공학과 박사 졸업

2012년 9월~2016년 7월 : 삼성전자 DMC연구소 책임연구원

2016년 7월~현재 : 동서울대학교 전자공학과 교수

<관심분야> 딥러닝, 영상처리, 자율주행, 이동통신