

# 자율주행을 위한 다중작업학습에 관한 연구

전 우 민\*, 꺾 대 원\*, 이 성 진<sup>○</sup>

## Study on Multi-Task Learning for Autonomous Driving

Woomin Jun\*, Daewon Kwak\*, Sungjin Lee<sup>○</sup>

### 요 약

본 연구에서는 자율주행을 위해 해당 센서들로부터 수행되는 영상인식 작업들의 성능 (정확도, 처리속도)을 고려하여 주어진 하드웨어 조건에 따라 안전한 자율주행을 수행할 수 있는 방안에 대해 연구하였다. 특히, 여러 작업들을 동시에 처리할 수 있는 다중작업학습 (MTL: Multi-Task Learning)을 통한 다중 영상인식 작업들의 성능을 분석하고 이들을 개선 할 수 있는 방안인 MDE (Multi-task Decision and Enhancement) 알고리즘을 제안하였다. 이 MDE 알고리즘을 이용하면 안정성이 확보되고 전체 지연시간을 최소화 할 수 있는 다중 작업 셋, 공유 백본 모델과 관련 기증치를 결정할 수 있다. 실험 결과 해당 전략을 통하면 동일 수준의 정확도에 최대 약 15-54%의 수행시간 단축 성능을 도출할 수 있었다.

**키워드** : 자율주행, 다중작업학습, 다중작업 결정 및 개선, 객체인식, 차선인식, 주행영역인식

**Key Words** : Autonomous Driving, Multi-Task Learning, OD, LD, DAS

### ABSTRACT

For autonomous driving, we explored a method for safe autonomous driving based on the given hardware conditions, taking into account the performance (accuracy, processing speed) of image recognition tasks performed by the corresponding sensors. In particular, we analyzed the performance of multiple image recognition optimization tasks through multi-task learning (MTL), which can process several tasks simultaneously, and proposed a MDE (Multi-task Decision and Enhancement) algorithm for optimization. Using this MDE algorithm, it is possible to determine multiple working sets that can minimize the overall delay time while optimizing accuracy. As a result of the experiment, we achieved up to around 15-54% reduction in execution time with similar accuracy performance through this strategy.

### 1. 서 론

최근 딥러닝의 기술적 혁신은 로봇 기술 및 자율주행 기술에 비약적 발전을 가져왔다<sup>[1]</sup>. 특히 카메라, 라이더, 레이더, GPS, 초음파, IMU 등의 다양한 센서를 통해 얻어지는 자동차 상황 인지에 관한 정보는 딥러닝을

통해 주행 판단, 주행 제어에 유용한 정보를 제공한다<sup>[1-4]</sup>.

하지만, 이런 다양한 센서로부터 획득된 정보를 자율주행에 실시간으로 적용하기 위해서는 해당 연산들은 경량화 되어야 하며 가속화 되어야 한다<sup>[4-18]</sup>. 이런 센서들 중에 가장 고연산, 고지연율을 요구하는 작업들 중 하

※ 이 논문은 2023년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (2023 신산업특화선도전문대학 지원사업)

• First Author : Dong Seoul University Department of Electronic Engineering, [aplus912@naver.com](mailto:aplus912@naver.com), 학생회원

○ Corresponding Author : Dong Seoul University Department of Electronic Engineering, [sungjinlee@du.ac.kr](mailto:sungjinlee@du.ac.kr), 정회원

\* Dong Seoul University Department of Electronic Engineering, [kdw1181@naver.com](mailto:kdw1181@naver.com), 학생회원

논문번호 : 202305-105-C-RE, Received May 23, 2023; Revised July 23, 2023; Accepted July 29, 2023

나는 단연 카메라와 라이다, 레이더를 중심으로 하는 2D, 3D 영상인지 연산이다. 연구<sup>6,8,13-15</sup>에서는 카메라 기반의 2D 객체 탐지, 영상 분할 연산을 위한 네트워크 경량화 및 가속화 연구를 수행하였다. 연구<sup>5,16,17</sup>에서는 딥러닝 경량화를 위한 양자화, 프루닝, 지식 증류 방법들에 대해 연구하였다. 연구<sup>18</sup>에서는 카메라 기반 차선 인식을 연산을 위한 가속화 연구를 수행하였다.

하지만, 이런 연구들은 모두 단일 작업만을 위한 연구여서 이들을 모두 사용해야 하는 실제 환경에서는 해당 연산들이 모두 가속되어야 한다. 이런 이유로, [9-12,19-21] 연구에서는 다중작업학습 (MTL: Multi-Task Learning)에 대한 연구가 시작되었고, 위에 제시된 복수의 작업들을 가능한 동시에 수행할 수 있도록 하였다. 해당 연구에서는 자율주행에 필수적인 영상 인지 작업들, OD (Object Detection), LD (Lane Detection), DAS (Drivable Area Segmentation)을 다중작업학습으로 수행하였다<sup>19-21</sup>. 하지만, 이런 다중작업학습 연구들 중 여러 task들에 대해 확장 사용 가능성에 대해 다룬 연구들은 없었다.

본 연구에서는 자율주행을 위한 인지 센서들로부터 주변 상황인지의 역할과 성능 (정확도, 지연)을 분석해 보고, 이들을 다중작업학습을 통해 가속화 할 수 있는 전략들을 제시하였다. 특히 각 다중작업학습을 적용하는 task들의 정확도를 최대화하기 위한 백본 네트워크 모델선택, 손실 함수 선택, 훈련 전략들을 제시하였다.

## II. 관련 연구

자율주행 영상인지 분야에는 안전하고 신뢰성 있는 주행 판단을 위해 여러 센서들로부터 오는 정보들을 실시간으로 높은 정확도로 영상 인지 작업들을 수행해야 한다. 그중 카메라에 기반한 대표적인 영상인식 작업으로는 2D 객체 탐지, 2D 영상 분할, 차선 인식 등이 있으며, 3D 작업에는 3D 객체 탐지, 3D 영상 분할 등의 작업들이 있다. 우선 2D 객체 탐지로서 [8][22] 연구에서 1 stage 기반으로 52AP의 정확도에 30FPS 이상의 성능을 내는 기술을 발표하였다. 최근에는 Anchor Free 기반으로 동작하는 [23][24] 기술들이 정확도를 더욱 높이면서 280FPS 이상의 성능을 내는 기술을 발표하였다. 2D 영상 분할 분야에서는 [25][26] 연구가 82.4mAP의 정확도 성능을 내는 기술을 발표하였다. 3D 객체 탐지 연구에서는 카메라 기반의<sup>27</sup> 연구 (18.69% AP), 라이다 기반의<sup>28</sup> 연구 (81.8% AP), 카메라와 라이다의 센서퓨전 기반의<sup>29</sup> 연구 (82.4% AP)가 있다. 3D 영상 분할 연구에서는 라이다 기반의<sup>30</sup>

연구가 74% mIoU 성능을 보여준다.

하지만 위의 기술들은 개별 기술에 대한 연구이기 때문에 실제 자율주행 차량에서는 해당 인식 모델들 모두를 탑재, 실행할 경우 상호 기술간 동기화 문제, 시스템 과부하 문제 등이 발생하기에 해당 모델들을 서로 공유 할 수 있는 자율주행 MTL에 대한 연구가 시도되었다<sup>19-21</sup>. HybridNet에서는 3가지 작업 (주행 가능 영역, 차선 인식, 객체 탐지)에 대해 다중작업학습을 수행하였다<sup>19</sup>. 또한, YOLOPv1,2 에서도 위 3가지 작업에 기반한 다중작업학습에 대해 HybridNet의 성능을 개선하며 그 가능성을 보여주었다<sup>20,21</sup>.

위를 통해 각 다중작업학습은 백본 성능에 따라 그 정확도가 달라질 수 있다는 것을 알 수 있다. 하지만, [31][32] 연구에서 볼 수 있듯이, 반드시 복잡한 구조의 백본, 즉 ViT와 같은 구조를 사용하는 것이 모든 태스크에서 높은 정확도를 보장하는 것은 아니라는 것을 알 수 있고, 최종 주행 목적을 달성하기 위한 해당 기술간 상호 보완적 관계를 고려한 영상인식 기술설계가 중요하다는 것을 알 수 있다.

## III. 시스템 모델

그림 1은 자율주행을 위한 멀티태스크 러닝의 시스템 개요도를 나타낸다. 그림에서 보듯이, 주행 가능 영역, 차선 인식, 객체 탐지 의 태스크 들을 서로 동일한 백본 네트워크 모델 (SBM, Shared Backbone Model) B를 공유하며, 각 세부 태스크들을 위한 고유 서브넷 모델 (TSM, Task-specific Subnet Model) S를 거쳐 최종 작업들을 완성하게 된다.

SBM 후보로서 인코더-디코더 구조에 기반한 네트워크들 중 UNet [33], FPN [34], Bi-FPN [35], PFPN [27] 을 대상으로 하였다. 이들 네트워크 모두는 영상 분할을 기반으로 한 네트워크로서 대부분의 모델들이 객체 탐지, 영상 분할 등의 태스크에 공유될 수 있는 모델들이다.

위에서 알 수 있듯이, 다중작업학습의 핵심은 여러 영상인식을 수행함에도 이들을 위한 핵심 모듈인 백본 혹은 그 일부, 혹은 그 이상을 공유한다는데 있다. 이렇게 백본을 공유함으로써 각 다중작업들을 수행하는데 요구되는 일부 지연시간을 줄일 수 있으며, 각 다중작업의 다양한 데이터에 대한 학습을 백본이 경험함으로써 데이터 정확도 또한 향상시킬 수도 있다.

하지만, 이런 멀티 태스크들 중에서 너무 연관성이 적은 태스크들에 대해 동일한 백본을 공유하려고 한다면, 백본은 해당 태스크에 대한 최적화가 이루어질 수

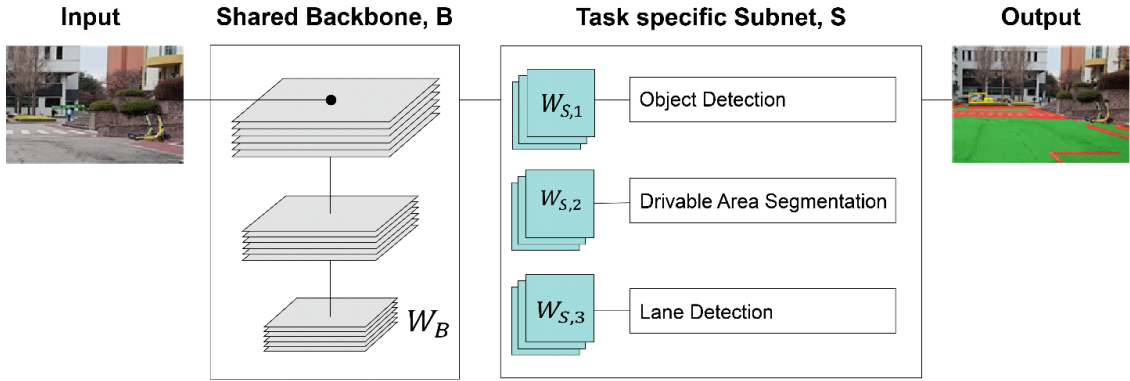


그림 1. 자율주행을 위한 다중작업학습의 시스템 개요도  
Fig. 1. System Architecture of Multi-Task Learning for Autonomous Driving

없기 때문에, 각각의 태스크들의 정확도가 떨어질 수도 있다. 특히 자율주행자동차와 같은 환경에서 미세한 영상인식 성능 차이는 운전자의 안전에 치명적인 영향을 미칠 수 있기 때문에 이런 다중작업학습의 적용은 신중히 접근해야 하는 문제라고 할 수 있다.

#### IV. MDE 알고리즘

본 논문은 각 태스크들의 목표 성능, 즉, 정확도 성능을 기준으로 전체 태스크들의 수행시간을 최소화하면서 정확도를 최대한 증가시킬 수 있도록 다음과 같이 MDE (Multi-task Decision and Enhancement) 알고리즘을 제안한다.

##### • MDE 알고리즘

**Step 1. 전체 지연시간을 최소화 할 수 있는 MTS(Multi-Task-Set)  $\mathbf{T}_m^*$ 과 SBM  $\mathbf{B}^*$  결정 :**

ImageNet에 사전학습된 가중치 값을 기반으로 수식 (1)에 기반하여 다중 작업 셋  $\mathbf{T}_m^*$ 을 판단하고 이를 위해 SBM  $\mathbf{B}^*$ 을 결정한다. 또한, TSM  $\mathbf{S}$ 는 MTS  $\mathbf{T}_m^*$ 이 결정되면서 이에 적합한 subnet model이 같이 결정된다고 설정한다. 여기서, ImageNet에 사전 훈련된 가중치 값만으로 MTS를 결정하는 이유는 이들 값 만으로도 충분히 MTS  $\mathbf{T}_m^*$ 을 결정할 수 있기 때문이며, 전체적인 알고리즘 동작 시간 또한 줄이기 위함이다.

$$\mathbf{T}_m^*, \mathbf{B}^* = \arg \min_{\mathbf{T}_m, \mathbf{B}, \mathbf{S}} \{ \sum LM(\mathbf{T}_m, \mathbf{B}) + \sum LI(\mathbf{T}_{m^c}) \}$$

$$\text{Subject To } ACC(t_i, \mathbf{B}) \geq \gamma_i, \quad t_i \in \mathbf{T}_t,$$

$$\mathbf{B} \in \{UNet, FPN, BiFPN, PFPN\}$$

(1)

위 수식 정의를 위해 아래 표와 같은 인자들을 정의한다.

$\mathbf{T}_t$  : 전체 영상인식 task set

$t_i$  : 영상인식 개별 task,  $t_i \in \mathbf{T}_t, \forall i \leq |\mathbf{T}_t|$ ,

$\mathbf{T}_m$  : MTL이 적용되는 task set,  $\mathbf{T}_m \subset \mathbf{T}_t$

$t_j$  : MTL이 적용되는 개별 task,  $t_j \in \mathbf{T}_m, \forall j \leq |\mathbf{T}_t|$ ,

$\mathbf{T}_{m^c}$  : MTL이 적용되지 않는 task set,

$LM(\mathbf{T}_m, \mathbf{B})$  : MTL이 적용되는 task set  $\mathbf{T}_m$ 이 SBM  $\mathbf{B}$ 를 통과하였을 때의 지연시간

$LI(\mathbf{T}_{m^c})$  : MTL이 적용되지 않는 task set  $\mathbf{T}_{m^c}$ 의 Latency,

$ACC(t_i, \mathbf{B})$  : 임의의 task  $t_i$ 의 정확도

$\gamma_i$  : 임의의 task  $t_i$ 의 기준 정확도 지표 값

위 수식 (1)에서 기준 정확도 지표 값  $\gamma_i$ 는 자율주행 안전성 및 지연시간에 밀접하게 연관되어 있는 지표로서, MTL의 전체 동작 방법론에 영향을 미칠 수 있다.

우선, MTL을 수행함으로써 정확도 지표의 상승을 얻고자 한다면, MTL을 적용하기 전의 기준 정확도 지표  $\gamma_i$ 에서  $\delta_i$  값을 더한 수치  $\gamma_i + \delta_i$ 를 기준 정확도 지표로 정할 수 있다. 물론, 이렇게 기준 정확도 지표 값을 올리면, 가용 MTL 적용 셋  $\mathbf{T}_m$ 의 feasible solution이 존재하지 않을 수 있다. 반면, 기준 정확도 지표를 다소 낮추어  $\gamma_i - \delta_i$ 로 정한다면, 더 많은 MTL 적용 셋  $\mathbf{T}_m$ 을 확보하여 전체 시스템 지연시간을 더욱 감소시킬 수 있지만, 안전성에 부정적 영향을 줄 수 있을 것이다. 이를 통해 위 수식에 기반한 가용 MTL 적용 셋  $\mathbf{T}_m$ 을 결정하는 일은 안정성과 속도에 상호 상충 관계를 유발한다 것을 알 수 있다.

또한, 정확도 지표  $\gamma_i$ 는 태스크 마다 다른 정확도 지

표를 둘 수 있다는 점에서 태스크 별 우선순위를 둘 수 있다. 즉, 자율주행 안전도에 즉각적인 영향을 미치는 영상인식 태스크에는 높은 정확도 기준을, 그 보다는 덜 한 태스크에는 다소 낮은 정확도 기준으로 설정함으로써 문제의 해법이 존재하도록 설정한다. 이런 다른 정확도 기준은 SBM  $\mathbf{B}^*$ 의 결정에도 영향을 미치게 한다. 본 연구에서는 차선인식과 주행가능영역 태스크의 경우 기준 정확도 지표  $\gamma_i$ 를 95%로 설정하고 객체인식의 경우 mAP의 기준 정확도 지표  $\gamma_i$ 를 0.80로 설정하였다.

**Step 2. 결정된 다중 작업 셋에 속한 각 태스크들의 정확도들을 더욱 극대화 할 수 있는  $W_B^*, W_S^*$  결정 :**

이제 결정된 다중 작업 셋  $\mathbf{T}_m^*$ 와 SBM  $\mathbf{B}^*$ 을 기반으로 최적의 백본 가중치  $W_B^*$ 와 각 태스크 별 서브 넷 가중치  $W_S^*$ 를 결정하여 각 태스크의 정확도를 극대화한다. 이들 정확도는 서로 간의 상대적 크기가 다르고 중요도가 다를 수 있기 때문에 이에 대한 가중치  $\alpha_i$ 를 고려하여 이들의 가중합을 최대화 시키도록 수식 (2)와 같이 재훈련 시킨다.

$$W_B^*, W_S^* = \arg \max_{W_B, W_S} \left\{ \sum_{t_i \in \mathbf{T}_m^*} \alpha_i \cdot Acc(t_i) \right\} \quad (2)$$

**V. 다중작업학습을 위한 subnet**

본 챕터에서는 다중작업학습을 위한 subnet, TSM S에 대해 좀 더 자세히 알아본다. 특히, 대상 태스크들을 정의하고 이들의 정확도를 더욱 올리기 위한 네트워크와 손실함수를 정의한다.

**5.1 OD (Object Detection)**

객체인식을 위한 subnet으로는 RetinaNet [8]의 subnet 구조를 사용한다. subnet의 두 subnet branch를 통과한 손실함수로는 Focal Loss와 Regression Loss를 사용한다.

**5.2 DAS (Drivable Area Segmentation)**

주행 가능 영역 작업을 위해서는 기본적으로 영상 분할 기반의 backbone을 사용한다. 손실함수로서는 Dice 손실함수와 BC (Binary Cross entropy) 손실함수를 선택적으로 사용한다.

**5.3 LD (Lane Detection)**

차선인식을 위해서는 DAS와 동일하게 영상 분할로서 접근하여 차선영역을 도출하였다.

**VI. 실험**

실험으로 다중작업학습의 다양한 작업에 대한 개별 성능 및 통합 성능을 확인하고 이를 기존 다중작업학습 방식들인 YOLOP (YP), HybridNet (HN), 그리고 기존 객체인식 방식인 RetinaNet, 기존 차선 인식 방식인 UFLD, CLRNet과 비교 할 수 있는 방향으로 진행되었다. 우선 제안하는 MDE알고리즘을 통한 1 - 3 task들에 대해 다중작업학습을 적용하였을 때의 성능과 기존 다중작업학습 방식들의 성능을 개별 태스크들, OD, DAS, LD의 정확도 및 지연시간 관점에서 성능비교를 수행하였다. 그리고 MDE를 적용한 결과, 즉, LD, DAS의 목표정확도 95%, OD의 목표 mAP 0.80를 상회하는 범위에서 최대 확장 가능한 다중 작업 셋  $\mathbf{T}_m^*$ 와 TSM S를 결정하여 성능을 확인하고, MTL의 적용가능성에 대해 생각한다.

우선 이를 위한 데이터셋은 BDD 100K[36]를 기반으로 하였다. 실험을 위한 기기 환경으로는 1 way RTX 4090 기반의 NVIDIA GPU에 tensorflow 기반으로 코딩하여 실험결과를 확인하였다. 학습률 조정 방식으로는 piecewise constant decay을 사용하였으며, 훈련 epoch 으로는 총 50 epoch 중 최고 성능 기준으로 도출하였다.

**6.1 MTL의 각 태스크 별 성능**

OD, LD, DAS 각 task의 실험을 위한 대조군으로 1 task 단독 모델과 2 task (LD+DAS), 3 task (OD+LD+DAS), YOLOP, HybridNet, RetinaNet, UFLD, CLRNet과 각 성능들을 비교하였다. 표 1은 OD 성능 비교를 위해 1 task, 3 task, YOLOP, HybridNet을 대상으로 하고, 표 2는 DAS성능 비교를 위해 1 task, 2 task, 3 task, YOLOP, HybridNet을 대상으로 하였으며, 표 3은 LD성능 비교를 위해 1 task, 2 task, 3 task, YOLOP, HybridNet을 대상으로 설정하였다. 표 4는 LD, DAS의 목표정확도 95%, OD의 목표 mAP 0.80를 상회하는 범위에서 모델들의 연산복잡도를 유추하기 위한 훈련 파라미터 수와 지연시간 성능을 나타내었다. 그림 2, 3은 해당 2 task 및 3 task 적용 예시를 나타내었다.

표 1. 객체인식 태스크의 성능  
Table 1. Performance of object detection task

	1 task	3 task (OD+LD+DAS)				YP	HN
	RN	UN	FPN	BiFPN	PFPN		
mAP	0.80	0.38	0.51	0.85	0.85	0.76	0.77
Lat (ms)	37.7	34.5	31.5	34.7	34.7	54.8	61.1

표 2. 주행가능 영역 인식 태스크의 성능  
Table 2. Performance of drivable area detection task

1 task						
model	UNet	FPN	BiFPN	PFPN		
Loss	Dice	Dice	Dice	Dice		
ACC	0.94	0.93	0.95	0.95		
Lat (ms)	22.5	22.7	23.1	23.3		
2 task						
model	UNet	FPN	BiFPN	PFPN		
Loss	Dice	Dice	Dice	Dice		
ACC	0.93	0.94	0.95	0.95		
Lat (ms)	26.2	26.7	27.7	27.9		
3 task						
model	UNet	FPN	BiFPN	PFPN	YP	HN
Loss	Dice	Dice	Dice	Dice	Tyversky	Tversky
ACC	0.45	0.46	0.44	0.40	0.97	0.91
Lat (ms)	37.7	34.56	31.5	32.7	54.8	61.1

표 3. 차선인식 태스크의 성능  
Table 3. Performance of lane detection task

1 task		2 task				
	UFLD	CLRNet	UNet	FPN	BiFPN	PFPN
Loss	LS	Line	Dice	Dice	Dice	Dice
ACC	0.98	0.99	0.95	0.97	0.98	0.99
Lat (ms)	10.7	11.2	26.2	26.7	27.7	27.9
3 task						
	UNet	FPN	BiFPN	PFPN	YP	HN
Loss	Dice	Dice	Dice	Dice	Tyversky	Tversky
ACC	0.44	0.45	0.42	0.38	0.70	0.85
Lat (ms)	54.56	51.56	52.79	54.76	54.8	61.1

표 4. 모델 파라미터에 따른 성능 (LD, DAS의 목표정확도 95%, OD의 목표 mAP 0.80)  
Table 4. Performance by Parameter

2 task (Requirement Satisfied)				3 task (Unsatisfied)		
	UNet	FPN	BiFPN	PFPN	YP	HN
Parameter	3.1M	2.5M	2.8M	3.3M	7.9M	12.8M
Lat (ms)	26.2	26.7	27.7	27.9	54.8	61.1
3 task (Unsatisfied)						
	UNet	FPN	BiFPN	PFPN		
Parameter	4.3M	3.8M	3.5M	4.9M		
Lat (ms)	54.5	51.5	52.7	54.7		



(a) UNet



(b) FPN



(c) BiFPN



(d) PFPN

그림 2. 2 task (LD+DAS) 수행 결과 예시  
Fig. 2. Examples of 2 task (LD+DAS)



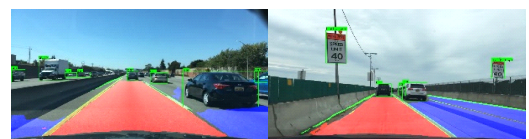
(a) UNet



(b) FPN



(c) BiFPN



(d) PFPN

그림 3. 3 task (OD+LD+DAS) 수행 결과 예시  
Fig. 3. Examples of 3 task (OD+LD+DAS)

## 6.2 다중작업학습의 성능분석 및 개선전략

표 1, 2, 3, 4의 결과에서 알 수 있듯이 3 task 모델 적용 결과, 자율주행을 위해서는 정확도 관련 성능이 제안하는 방식과 기존 다중작업학습 방식인 YoloP, HybridNet 모두에서 만족스럽지 못하다는 것을 알 수 있다. 특히 표 1에서도 알 수 있듯이 OD 성능이 목표 mAP 성능 0.80을 만족시키는 경우는 BiFPN, PFPN 뿐이며 표 2, 3의 3 task에서 목표 성능인 95%에 많이 미치지 못하는 DAS 및 LD 성능이 얻어진다는 것을 알 수 있다.

반면 2 task (DAS+LD) 모델 적용 결과는 DAS와 LD 모두 1 task, 3 task, 기존 다중작업학습 방식들에 비해서도 비견할 정도의 성능을 보이며 동작속도 역시 비슷함을 알 수 있다. 이를 통해 다중작업학습을 위한 태스크들은 서로 정확도에 도움이 되도록 쉽게 학습이 되고 그렇지 않은 MTS가 있음을 알 수 있으며 이런 태스크들 위주로 MTS를 구성하는 것이 자율주행의 안전성 확보에 도움이 된다는 것을 알 수 있다. 사실, DAS와 LD는 그 task의 성격이 유사하여 상호 보완적이며 OD의 경우 DAS와 LD와는 상호 연관성이 상대적으로 적기 때문에 이런 상호 열화된 성능이 도출된다. 그렇기 때문에, 본 연구의 다중작업학습 예시에서는 LD, DAS로 구성되는 2 task 만으로 운용되는 것이 좀 더 안전한 자율주행 영상인식 모델을 구현하는 방법일 것이다.

표 4는 LD, DAS의 목표정확도 95%, OD의 목표 mAP 0.80를 상회하는 범위에서 제안 기법들과 기존 기술인 YOLOP, HybridNet 기법의 지연시간 성능을 나타내었다. 그 결과 2 task 의 BiFPN, PFPN만이 성능 요구조건을 만족한다는 것을 알 수 있다. 사실, 해당 기술들을 모두 딥러닝 경량화 [16]를 적용하여 비교하는 것이 가장 공정한 비교일 수 있으나, 경량화는 적용 네트워크의 특성 (백본네트워크 종류/특성), 적용 경량화 방법 (양자화 종류, 프루닝 종류, 지식 증류 종류 등), 데이터의 특성 (클래스 개수, 데이터 분포도 등) 등의 조합 등에 따라 그 성능이 달라질 수 있어 또 다른 연구주제 일 수 있기 때문에, 해당 모델의 은닉층 수를 같이 기재하여 해당 기술의 연산 부담 정도를 유추할 수 있게 대신하였다. 또한, 정확도 측면에서는 경량화를 적용하더라도 성능 향상이 일어나기가 어렵기 때문에 정확도 요구조건을 만족하는 기술들과 그렇지 않은 기술들을 1차적으로 구분하고 이들의 훈련 파라미터 및 지연시간을 통해 복잡도를 2차적으로 비교해보도록 하였다.

위에서 언급한 것처럼 서로 학습에 상호 연관성이

높은 2 task로 MTS를 구성하는 MDE 기반 방식들이 YOLOP, HybridNet의 경우에 비해 지연시간 성능에서 약 54% 및 정확도 측면에서 DAS작업은 약 4%, LD작업은 약 41% 더 우수함을 알 수 있으며 사용 파라미터 수에서도 더 적게 사용함을 알 수 있다. LD, DAS의 목표정확도 95%, OD의 목표 mAP 0.80를 만족시키기 못하는 3 task 에 대한 경우를 비교하더라도 사용 파라미터 수에서 더 적은 수를 사용함을 알 수 있다.

또한, 표 2, 3의 DAS, LD 작업을 위해 사용된 손실 함수에 대해 분석해본다. 전통적으로 영상분할 문제에서 Binary Cross-entropy 함수가 주로 사용되지만 클래스 불균형 문제 해소를 위해서는 Dice 함수와 Tversky 함수가 사용된다 [31]. 그림 2, 3을 통해서도 알 수 있듯이 전경 영역의 크기가 전체 영상에 비해 그 비율이 작게 분포되어 있으므로 표 2, 3의 결과를 통해서 알 수 있듯이 제안하는 MDE기법에서는 BC보다는 Dice 함수가 더 좋은 성능을 나타낸다는 것을 알 수 있다. 기존 기법 YOLOP, HybridNet에서도 이와 유사한 Tversky 함수를 해당 손실함수로 사용하는 것으로 통해 볼때, 자율주행에서는 전경 크기들을 고려하는 Dice 혹은 Tyversky손실함수를 사용하는 것이 바람직하다 하겠다.

표 4를 보면 2 task 중에서 정확도 요구조건을 만족하는 공유 백본 모델은 BiFPN, PFPN 뿐이라는 것을 알 수 있다. 물론 표2,3 을 통해 PFPN이 다소 더 높은 정확도 성능을 가지고 있다는 것을 알 수 있지만 이들 중 BiFPN이 더 유리한 연산 복잡도를 가지고 있다. 그러므로 시스템 적용 하드웨어 조건과 정확도에 대한 민감도를 고려하여 공유 백본 모델을 선택 해야 할 것이다.

## VII. 결 론

자율주행을 위한 다중작업학습에 대해 정의하고 다중작업학습의 성능에 대해 고찰하였다. 특히, 자율주행 영상인지에 주로 사용되는 객체인식 OD, 주행영역인지 DAS, 차선인식 LD 모델을 다중작업학습에 적용해 보고, 이들의 실 적용을 위한 방법론인 MDE 알고리즘을 제안하였다. 여러 성능 평가를 통해 다중작업학습의 효율적인 사용을 위해서는 상호 연관성이 깊은 task들을 한정하여 수행해야 하며, 적용 데이터 셋의 특성을 파악해야 하고, 정확도 뿐 아니라 동작속도 성능을 고려하여 공유백본 모델을 선택해야 함을 알 수 있었다.

## References

- [1] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *J. Field Robotics*, vol. 37, no. 3, 2019. (<https://doi.org/10.1002/rob.21918>)
- [2] A. I. Károly, P. Galambos, J. Kuti, and I. J. Rudas, "Deep learning in robotics: Survey on model structures and training strategies," *IEEE Trans. Syst., Man, and Cybernetics*, vol. 51, no. 1, pp. 266-279, 2021. (<https://doi.org/10.1109/TSMC.2020.3018325>)
- [3] D. Kwak, J. Yoo, M. Son, D. Choi, and S. Lee, "Rethinking real-time lane detection technology for autonomous driving," *J. KICS*, vol. 48, no. 05, 2023. (<https://doi.org/10.7840/kics.2023.48.5>)
- [4] E. Bae and S. Lee, "Efficient training methodology in an image classification network," *J. KICS*, vol. 46, no. 6, 2021. (<https://doi.org/10.7840/kics.2023.48.5>)
- [5] 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, 2022. (<https://doi.org/10.3390/s22197344>)
- [6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," *Computer Vision - ECCV 2016 LNCS*, vol. 9905, 2016. (<https://doi.org/10.48550/arXiv.1512.02325>)
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CVPR 2016*, pp. 779-788, 2016. (<https://doi.org/10.48550/arXiv.1506.02640>)
- [8] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 42, no. 2, pp. 318-327, 2020. (<https://doi.org/10.48550/arXiv.1708.02002>)
- [9] D.-G. Lee, "Fast drivable areas estimation with multi-task learning for real-time autonomous driving assistant," *Applied Sciences*, vol. 11, no. 22, 10713, 2021. (<https://doi.org/10.3390/app112210713>)
- [10] K. Ishihara, A. Kanervisto, J. Miura, and V. Hautamäki, "Multi-task learning with attention for end-to-end autonomous driving," *2021 IEEE/CVF Conf. CVPRW*, pp. 2896-2905, 2021. (<https://doi.org/10.1109/CVPRW53098.2021.00325>)
- [11] M. Teichmann, M. Weber, M. Zöllner, R. Cipolla, and R. Urtasun, "MultiNet: Real-time joint semantic reasoning for autonomous driving," *2018 IEEE Intell. Veh. Symp. (IV)*, pp. 1013-1020, Changshu, China, 2018. (<https://doi.org/10.1109/IVS.2018.8500504>)
- [12] J. Guo, J. Wang, H. Wang, B. Xiao, Z. He, and L. Li, "Research on road scene understanding of autonomous vehicles based on multi-task learning," *Sensors*, vol. 23, no. 6238, 2023. (<https://doi.org/10.3390/s23136238>)
- [13] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017. (<https://doi.org/10.48550/arXiv.1704.04861>)
- [14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *IEEE/CVF Conf. CVPR*, pp. 4510-4520, Salt Lake City, UT, USA, 2018. (<https://doi.org/10.1109/CVPR.2018.00474>)
- [15] A. Howard, et al., "Searching for MobileNetV3," *IEEE/CVF ICCV*, pp. 1314-1324, Seoul, Korea, 2019. (<https://doi.org/10.1109/ICCV.2019.00140>)
- [16] Y. J. Lee, Y. H. Moon, J. Y. Park, and O. G. Min, "Recent R&D trends for lightweight deep learning," *Electr. and Telecommun. Trends*, vol. 34, no. 2, pp. 40-50, 2019. (<http://dx.doi.org/10.22648/ETRI.2019.J.340205>)

- [17] Z. Liu, et al., “Rethinking the value of network pruning,” *arXiv preprint arXiv:1810.05270*, 2018.  
(<https://doi.org/10.48550/arXiv.1810.05270>)
- [18] Z. Qin, H. Wang, and X. Li, “Ultra fast structure aware deep lane detection,” in *Proc. IEEE Trans. Pattern Anal. and Mach. Intell.*, pp. 1-14, 2022.  
(<https://doi.org/10.48550/arXiv.2004.11757>)
- [19] D. Vu, B. Ngo, and H. Phan, “HybridNets: End-to-end perception network,” *arXiv preprint arXiv:2203.09035*, 2022.  
(<https://doi.org/10.48550/arXiv.2203.09035>)
- [20] D. Wu, M. Liao, W. Zhang, X. Wang, X. Bai, W. Cheng, and W. Liu, “YOLOP: You only look once for panoptic driving perception,” *Mach. Intell. Res.*, vol. 19, pp. 550-562, 2022.  
(<https://doi.org/10.1007/s11633-022-1339-y>)
- [21] C. Han, Q. Zhao, S. Zhang, Y. Chen, Z. Zhang, and J. Yuan, “YOLOPv2: Better, faster, stronger for panoptic driving perception,” *arXiv preprint arXiv:2208.11434*, 2022.  
(<https://doi.org/10.48550/arXiv.2208.11434>)
- [22] X. Du, T. Y. Lin, P. Jin, G. Ghiasi, M. Tan, Y. Cui, Q. V. Le, and X. Song, “SpineNet: Learning scale-permuted backbone for recognition and localization,” *2020 IEEE/CVF Conf. CVPR*, pp. 11589-11598, 2020.  
(<https://doi.org/10.48550/arXiv.1912.05027>)
- [23] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021,” *arXiv preprint arXiv:2107.08430*, 2021.  
(<https://doi.org/10.48550/arXiv.2107.08430>)
- [24] J. Terven and D. Cordova-Esparaza, “A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond,” *arXiv preprint arXiv:2304.00501*, 2023.  
(<https://doi.org/10.48550/arXiv.2304.00501>)
- [25] A. Kirillov, R. Girshick, K. He, and P. Dollár, “Panoptic feature pyramid networks,” *CVPR*, 2019.  
(<https://doi.org/10.1145/3487983.3488301>)
- [26] 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>)
- [27] Y. Hong, H. Dai, and Y. Ding, “Cross-modality knowledge distillation network for monocular 3d object detection,” *Computer Vision - ECCV 2022*, vol. 13670, 2022.  
([https://doi.org/10.1007/978-3-031-20080-9\\_6](https://doi.org/10.1007/978-3-031-20080-9_6))
- [28] S. Shi, L. Jiang, J. Deng, Z. Wang, C. Guo, J. Shi, X. Wang, and H. Li, “PV-RCNN++: Point-voxel feature set abstraction with local vector representation for 3d object detection,” *Int. J. Computer Vision*, vol. 131, pp. 531-551, 2023.  
(<https://doi.org/10.48550/arXiv.2102.00463>)
- [29] Y. Kim, K. Park, M. Kim, D. Kum, and J. Choi, “3D Dual-Fusion: Dual-domain dual-query camera-lidar fusion for 3d object detection,” *arXiv Preprint arXiv:2211.13589*, 2022.  
(<https://doi.org/10.48550/arXiv.2211.13529>)
- [30] X. Lai, Y. Chen, F. Lu, J. Liu, and J. Jia, “Spherical transformer for LiDAR-based 3D recognition,” in *Proc. IEEE/CVF Conf. CVPR*, pp. 17545-17555, 2023.  
(<https://doi.org/10.48550/arXiv.2303.12766>)
- [31] D. Kwak, J. Choi, and S. Lee, “Rethinking breast cancer diagnosis through deep learning based image recognition,” *Sensors*, vol. 23, no. 4, 2307, 2023.  
(<https://doi.org/10.3390/s23042307>)
- [32] W. Wang, J. Dai, Z. Chen, Z. Huang, Z. Li, X. Zhu, X. Hu, T. Lu, L. Lu, H. Li, X. Wang, and Y. Qiao, “InternImage: Exploring large-scale vision foundation models with deformable convolutions,” in *Proc. IEEE/CVF Conf. CVPR*, pp. 14408-14419, 2023.  
(<https://doi.org/10.48550/arXiv.2211.05778>)
- [33] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” *MICCAI 2015*, vol. 9351, 2015.  
(<https://doi.org/10.48550/arXiv.1505.04597>)
- [34] T. Lin, P. Dollár, R. Girshick, K. He, B.



Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *2017 IEEE Conf. CVPR*, pp. 936-944, Honolulu, HI, USA, 2017.

(<https://doi.org/10.48550/arXiv.1612.03144>)

- [35] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and efficient object detection," *2020 IEEE/CVF Conf. CVPR*, pp. 10778-10787, 2020.

(<https://doi.org/10.48550/arXiv.1911.09070>)

- [36] F. Yu, "BDD100K: A large-scale diverse driving video database," *BAIR*, 2018.

(<https://bair.berkeley.edu/blog/2018/05/30/bdd/>)

**전 우 민 (Woomin Jun)**



2021년 3월~현재: 동서울대학교 전자공학과 재학중  
<관심분야> 딥러닝, 자율주행, 의료영상

**곽 대 원 (Daewon Kwak)**



2022년 2월: 동서울대학교 전자공학과 졸업  
<관심분야> 딥러닝, 의료영상, 영상인식

**이 성 진 (Sungjin Lee)**



2011년 8월: 연세대학교 전기전자공학과 박사 졸업  
2012년 9월~2016년 7월: 삼성전자 DMC연구소 책임연구원  
2016년 7월~현재: 동서울대학교 전자공학과 교수  
<관심분야> 딥러닝, 영상인식,

자율주행, 의료영상, 5G, 3D Reconstruction  
[ORCID: 0000-0003-3159-8394]