# Multi-Task Learning을 활용한 PVT v2 프레임워크 성능 개선

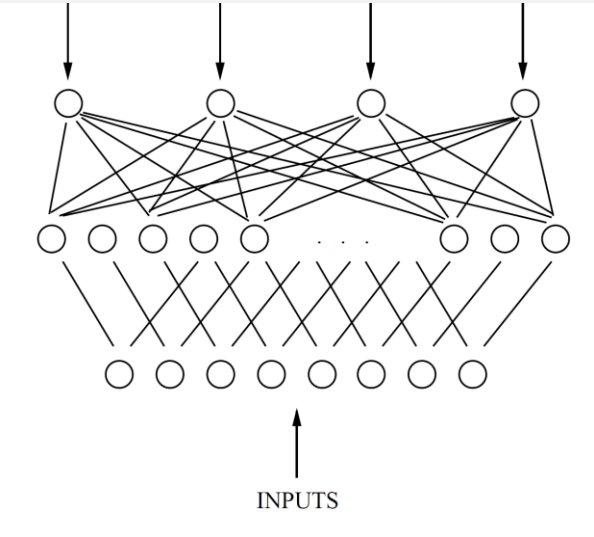
팀원: 김수영, 송재현

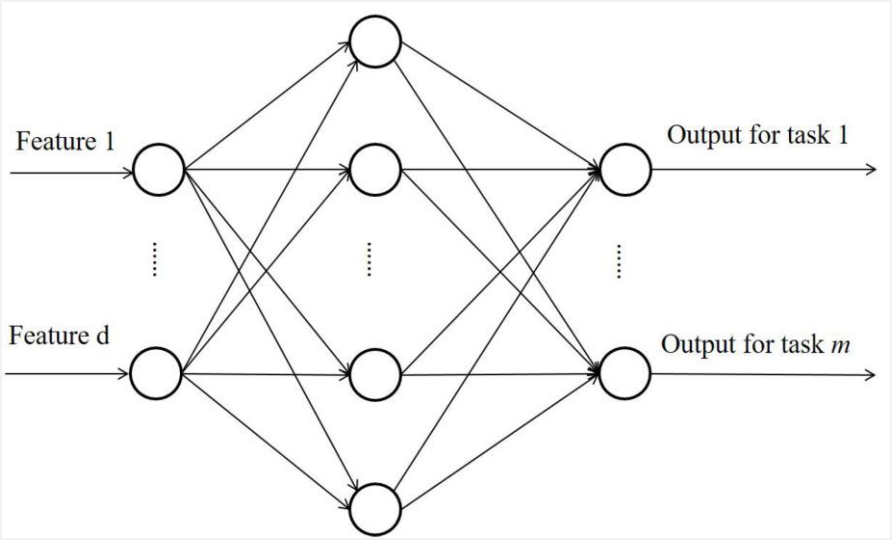
지도교수: 이종률

# 목차

# 프로젝트 개요

## 연구 배경




〈MTL의 기본 구조〉

# 프로젝트 개요

## MTL의 특징

- 경량화: 다양한 작업을 동시에 처리함으로써 모든 작업을 수행하는데 필요한 모형의 크기 경량화 가능
  - ✓ MTL: MTL 모형 하나가 필요
  - ✓ STL: 각 작업에 필요한 만큼의 모형이 필요
- 일반화: 여러 작업에 존재하는 공통된 표현(Shared Representation) 기반 학습을 통한 모형의 일반화 성능 확보 가능
  - ✓ Shared Representation: 여러 작업을 처리함에 있어 공통적으로 사용되는 지식, 특징 등을 의미

# 프로젝트 개요

## 연구 배경

〈자율주행에서 MTL의 동작 예시〉

# 프로젝트 개요

## MTL 연구 동향

- Multi-Task Learning에 최적화된 모형 개발
  - ✓ MulT, M3ViT, IPT
- 기존 Single-Task Learning 모형의 MTL로의 확장
  - ✓ Swin MTL

# 프로젝트 개요

## 연구 목적

- PVTv2 (Pyramid Vision Transformer v2)를 백본 네트워크로 활용한 MTL 모형 개발 PVTv2 프레임워크 확장 및 성능 개선
- 자율주행 환경에서 요구되는 이미지 분류, 객체 탐지, 의미론적 분할 작업을 하나의 모델로 처리
- STL 모형을 MTL모형으로 확장한 모형과의 비교를 통한 백본 네트워크 교체의 유효성 확인
- 경량화, 확장성, 정확도를 파악한 문제 해결 적용 가능성 확인

# 프로젝트 개요

## 팀원 소개 및 협업 내용

### 김수영

- ✓ PVTv2 백본 네트워크 구현
- ✓ LibMTL 오픈소스를 활용한 MTL 학습 및 실험
- ✓ 보고서 / 발표자료 제작

### 송재현

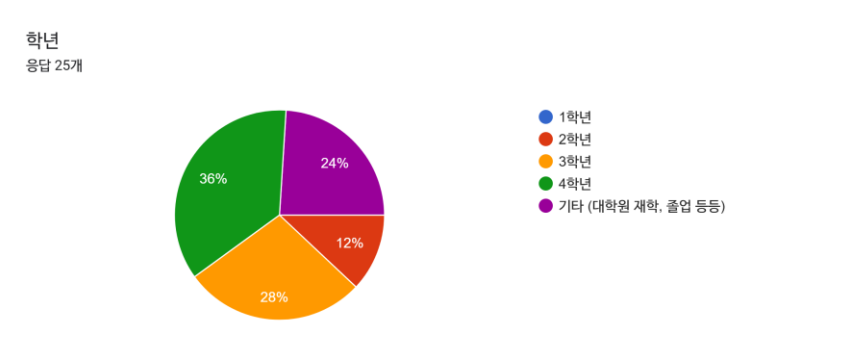
- ✓ 성능 평가 지표 탐색 및 구현
- ✓ 시각화 구현
- ✓ 보고서 / 발표자료 제작

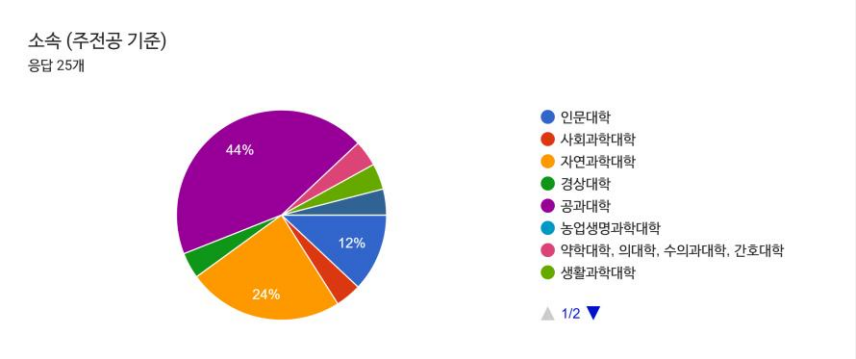
### GitHub, Zoom, 모바일 메신저, 대면 미팅을 활용한 협업 활동 진행

# 사용자 분석

## 이해관계자 설문조사

- 목적: MTL에 대한 대중 인식 파악 및 연구 목적의 타당성 확인
- 설문 대상: 조원 지인, 교내 커뮤니티 등
- 설문 방식: Google Form (총 25명)



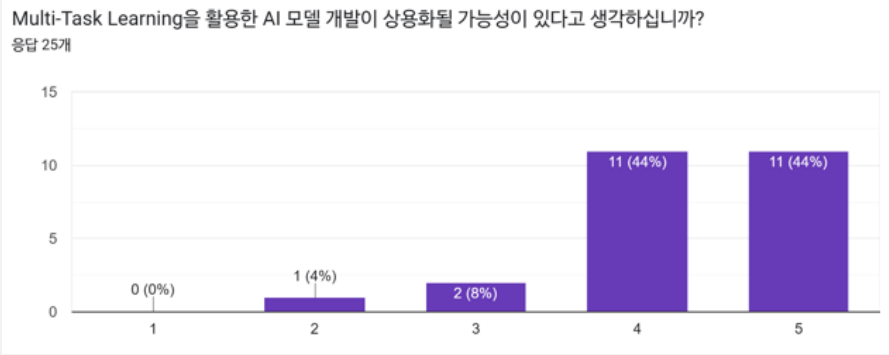
# 사용자 분석

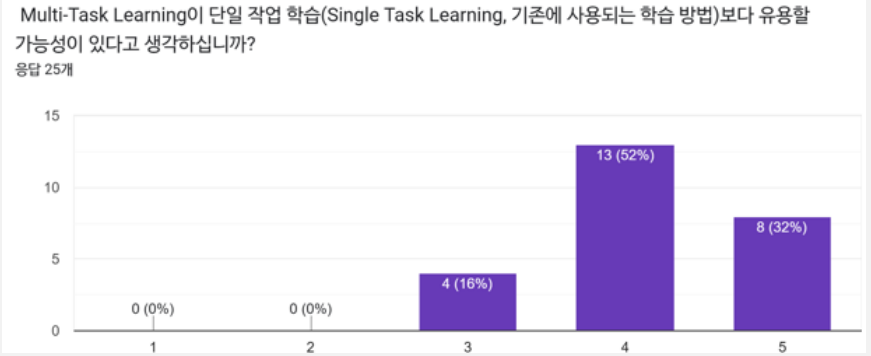
## 설문 문항 요약

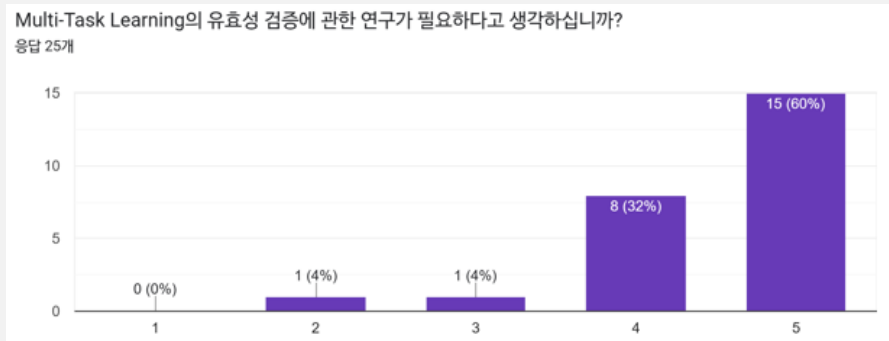
- MTL 상용화 가능성
- Multi-Task Learning vs Single Task Learning
- MTL 유효성 검증 필요성
- 향후 AI 성능 발전 기여 가능성

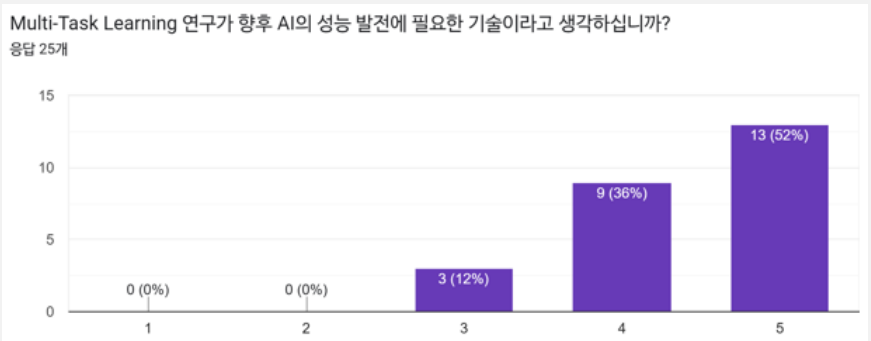
# 사용자 분석

## 설문 결과


Multi-Task Learning을 활용한 AI 모델 개발이 상용화될 가능성이 있다고 생각하십니까?
응답 25개
15
10
5
0
0 (0%)
1 (4%)
2 (8%)
11 (44%)
11 (44%)
1
2
3
4
5


Multi-Task Learning이 단일 작업 학습(Single Task Learning, 기존에 사용되는 학습 방법)보다 유용할 가능성이 있다고 생각하십니까?
응답 25개
15
10
5
0
0 (0%)
0 (0%)
4 (16%)
13 (52%)
8 (32%)
1
2
3
4
5


Multi-Task Learning의 유효성 검증에 관한 연구가 필요하다고 생각하십니까?
응답 25개
15
10
5
0
0 (0%)
1 (4%)
1 (4%)
8 (32%)
15 (60%)
1
2
3
4
5


Multi-Task Learning 연구가 향후 AI의 성능 발전에 필요한 기술이라고 생각하십니까?
응답 25개
15
10
5
0
0 (0%)
0 (0%)
3 (12%)
9 (36%)
13 (52%)
1
2
3
4
5

# 사용자 분석

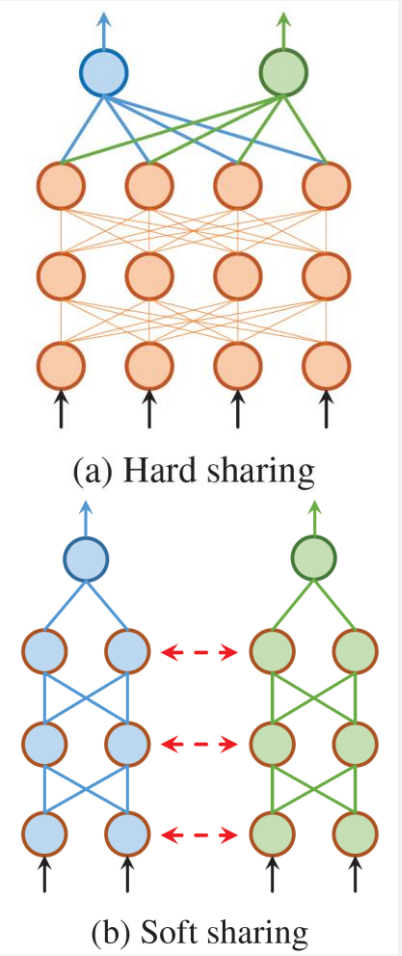
## 설문 결과 요약

- MTL 상용화 가능성 기대
- MTL이 기존 학습법보다 유용할 것이라는 의견 다수
- MTL 유효성 검증 필요성 공감대 형성
- 향후 AI 성능 발전에서 MTL의 기여 가능성 높을 것이란 의견 다수

# 핵심 아이디어

## 제안 방법

- 기존 PVT v2 모델을 MTL 구조로 확장
- Hard/Soft Parameter Sharing 구조 활용
- 세 가지 Task(분류/탐지/분할)에 대해 통합 학습 및 평가 진행
- 정량적 지표로 기존 STL/MTL 모형들과 성능 비교
  - ✓ Accuracy/AP/MIoU/#Param/Inference Time



# 핵심 아이디어

## 기존 해결방법 개선점

| 항목 | 기존 STL 구조 | 제안 방법 |
|---|---|---|
| 구조 | 각 Task마다 모델<br>별도 설계 필요 | PVT v2에 MTL을 적용,<br>공유 구조 설계 |
| 확장성 | 새로운 Task 추가 시<br>구조 재설계 필요 | 공유 백본으로<br>확장성 우수 |
| 성능 평가 | 단일 Task 성능만 평가 가능 | Accuracy, AP, mIoU,<br>#Param으로 다양한 Task<br>성능 평가 가능 |
| 학습 효율성 | Task별 학습으로<br>시간과 자원 소모 큼 | 공유 백본 사용, 연산 효율성 확보 |

# 핵심 아이디어

## 핵심 유스케이스

| | |
|---|---|
| **주요 Actor** | 자율주행 시스템 개발자, 운전자, MTL 연구자 |
| **주요 기능<br>구성 요소** | 주요 기능<br>- 자율주행 태스크 응용<br>- 객체 탐지, 의미론적 분할, 이미지 분류<br>- 멀티태스크 학습 모델 설계 및 학습<br>- 모델 평가 및 성능 분석<br>구성 요소<br>- PVT v2 프레임워크 및 MTL을 위한 디코더 |
| **입/출력 데이터** | 입력데이터(결과):이미지데이터셋(ImageNet,COCO,ADE20K)<br>출력 데이터(결과): 이미지 분류 레이블, 객체 탐지 박스, 의미론적 분할마스크 |
| **데이터 Flow** | 1) 이미지 데이터 전처리<br>2) MTL 모델 학습<br>3) 테스트 및 성능 분석 |
| **외부 시스템 연계** | 모델 비교 대상: Tesla HydraNet<br>평가 툴: PyTorch, sklearn을 활용<br>추가적인 모형 활용 시:Hugging Face |

# 데모

## 연구 질문 및 가설

- RQ1.
  PVT v2 프레임워크에 MTL을 적용한 모형은 단일 작업 학습 모형에 비해 작업(이미지 분류, 객체 탐지, 의미론적 분할)에 대한 유의미한 성능 향상이 이루어 지는가?

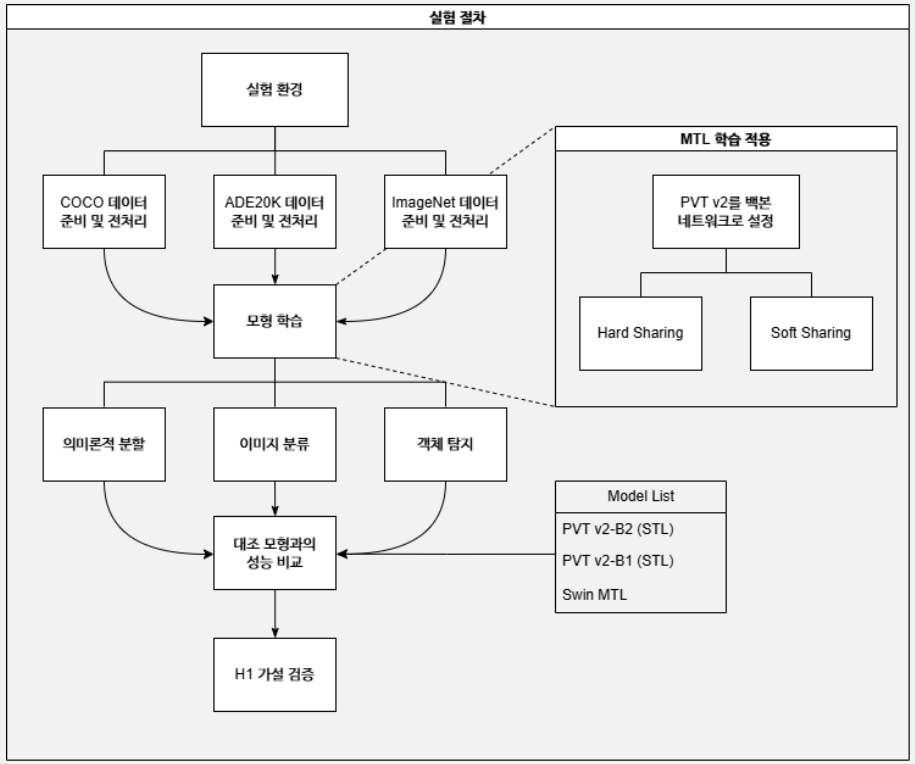
  - ✓ H1.
    MTL을 이용해 학습시킨 PVT v2 모형은 단일 작업 모형보다 정확도, AP, mIoU, #Param 등의 성능 지표에서 유의미한 개선을 보일 것이다.

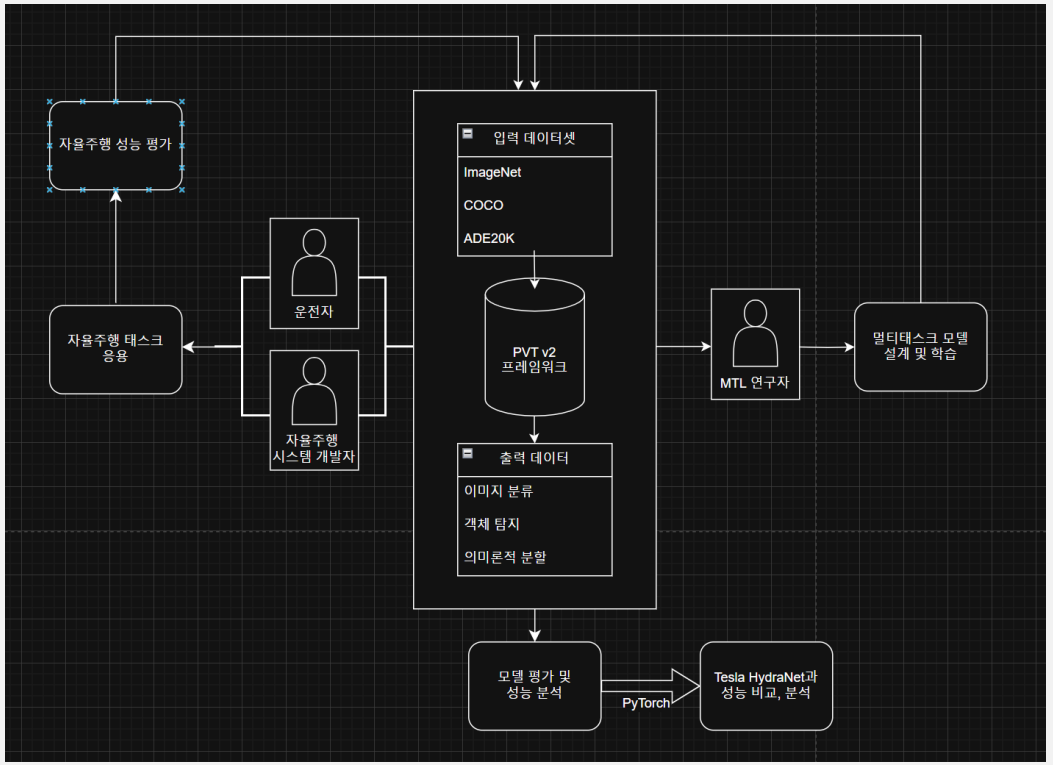
- RQ2.
  MTL을 이용해 학습시킨 모형은 자율주행 분야에서 기존 프레임워크 대비 어떤 장단점을 갖는가?

  - ✓ H2.
    MTL 기반 모델은 자율 주행 분야에서 연구되는 모형인 HydraNet 대비 정확도 측면에서 유의미한 성능 향상을 보여줄 것이다.

# 데모

## 시퀀스 다이어그램 / 실험 알고리즘 순서도



# 테스트

## 프로토타입 설계

### 활용 오픈소스

- ✓ LibMTL: MTL 오픈소스 라이브러리
- ✓ PVT: Pyramid Vision Transformer 오픈소스 라이브러리 *(Transformer 코드 활용)*
- ✓ Swin Transformer: Swin Transformer 오픈소스 라이브러리 *(Transformer 코드 활용)*

### 데이터셋

- ✓ NYUv2: 뉴욕 대학교에서 제작한 실내 공간에 대한 데이터셋
- ✓ Multi-Task Learning 평가에 주로 활용되는 벤치마크 데이터셋
- ✓ MTL 학습을 위해서는 각 작업의 평가에 활용하기 위한 **Annotation**이 필요
- ✓ 기존 ImageNet, COCO2017, ADE20K 2016은 단일 작업에 대한 Annotation만 제공되기에 **MTL에는 부적절하다 판단**

### 사용 장비 및 활용 프레임 워크

- ✓ 실험 환경: Linux 환경
- ✓ 사용 언어: python 3.10
- ✓ 프레임워크: **pytorch 2.3.0, CUDA12.1**
- ✓ 주요 라이브러리: **mmcv: 2.2.0**, *mmengine, mmsegmentation*, **LibMTL**
- ✓ 사용 GPU: RTX 3080 10GB

# 테스트

## 실험 계획

- Semantic Segmentation, Depth Estimation, Surface Normal Prediction 세 가지의 작업에 대한 학습 진행 → NYUv2가 지원하는 Annotation
- 독립 변수: 백본 네트워크(Swin Transformer, ResNet-50, PVTv2), 사전 학습 가중치 유무, 학습에 이용할 Task 조합
- 종속 변수:  mIoU, pixAcc, abs err, rel err, normal mean, normal median, normal⟨11.25, normal⟨22.5, normal⟨30, #Parameters

# 실험 결과

## PVTv2 / ResNet-50 (No Pretrained Learning Weight)

| Model | Best Epoch | Seg mIoU (↑) | Seg pixAcc (↑) | Depth abs_err (↓) | Depth rel_err (↓) | Normal mean (↓) | Normal median (↓) | Normal ⟨11.25 (↑) | Normal ⟨22.5 (↑) | Normal ⟨30 (↑) | Total Params |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PVTv2 (No Pretrained) | 94 | 0.2944 | 0.5558 | 0.5671 | 0.2410 | 30.9347 | 25.6424 | 0.2304 | 0.4478 | 0.5665 | 39.02M |
| ResNet-50 (No Pretrained) | 99 | 0.2984 | 0.5620 | 0.5727 | 0.2403 | 32.6432 | 26.7498 | 0.2204 | 0.4321 | 0.5478 | 71.89M |

# 실험 결과

## PVTv2 / ResNet-50 / SwinT(Pretrained Learning Weight)

| Model | Best Epoch | Seg mIoU (↑) | Seg pixAcc (↑) | Depth abs_err (↓) | Depth rel_err (↓) | Normal mean (↓) | Normal median (↓) | Normal ⟨11.25 (↑) | Normal ⟨22.5 (↑) | Normal ⟨30 (↑) | Total Params |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PVTv2 (Pretrained) | 81 | **0.5463** | **0.7585** | **0.3770** | **0.1544** | 25.2240 | 18.7480 | 0.3195 | 0.5703 | 0.6827 | 39.02M |
| ResNet-50 (Pretrained) | 98 | 0.5373 | 0.7570 | 0.3846 | 0.1618 | **23.5492** | **16.8995** | **0.3542** | **0.6115** | **0.7215** | 71.89M |
| SwinT (Pretrained) | 49 | 0.4891 | 0.7180 | 0.4157 | 0.4088 | 25.2206 | 18.9197 | 0.3120 | 0.5689 | 0.6838 | 112.47 M |

# 실험 결과

## PVTv2 MTL / PVTv2 STL

| Model | Best Epoch | Seg mIoU (↑) | Seg pixAcc (↑) | Depth abs_err (↓) | Depth rel_err (↓) | Normal mean (↓) | Normal median (↓) | Normal ⟨11.25 (↑) | Normal ⟨22.5 (↑) | Normal ⟨30 (↑) | Total Params |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PVTv2 - All (Pretrained) | 81 | **0.5463** | **0.7585** | **0.3770** | **0.1544** | 25.2240 | 18.7480 | 0.3195 | 0.5703 | 0.6827 | 39.02M |
| PVtv2 - Seg (Pretrained) | 70 | **0.5526** | **0.7605** | --- | --- | --- | --- | --- | --- | --- | 29.57M |
| Pvtv2 - Depth (Pretrained) | 66 | --- | --- | 0.3854 | 0.1562 | --- | --- | --- | --- | --- | 29.57M |
| Pvtv2 - Normal (Pretrained) | 31 | --- | --- | --- | --- | **23.5531** | **16.3254** | **0.3680** | **0.6149** | **0.7147** | 29.57M |

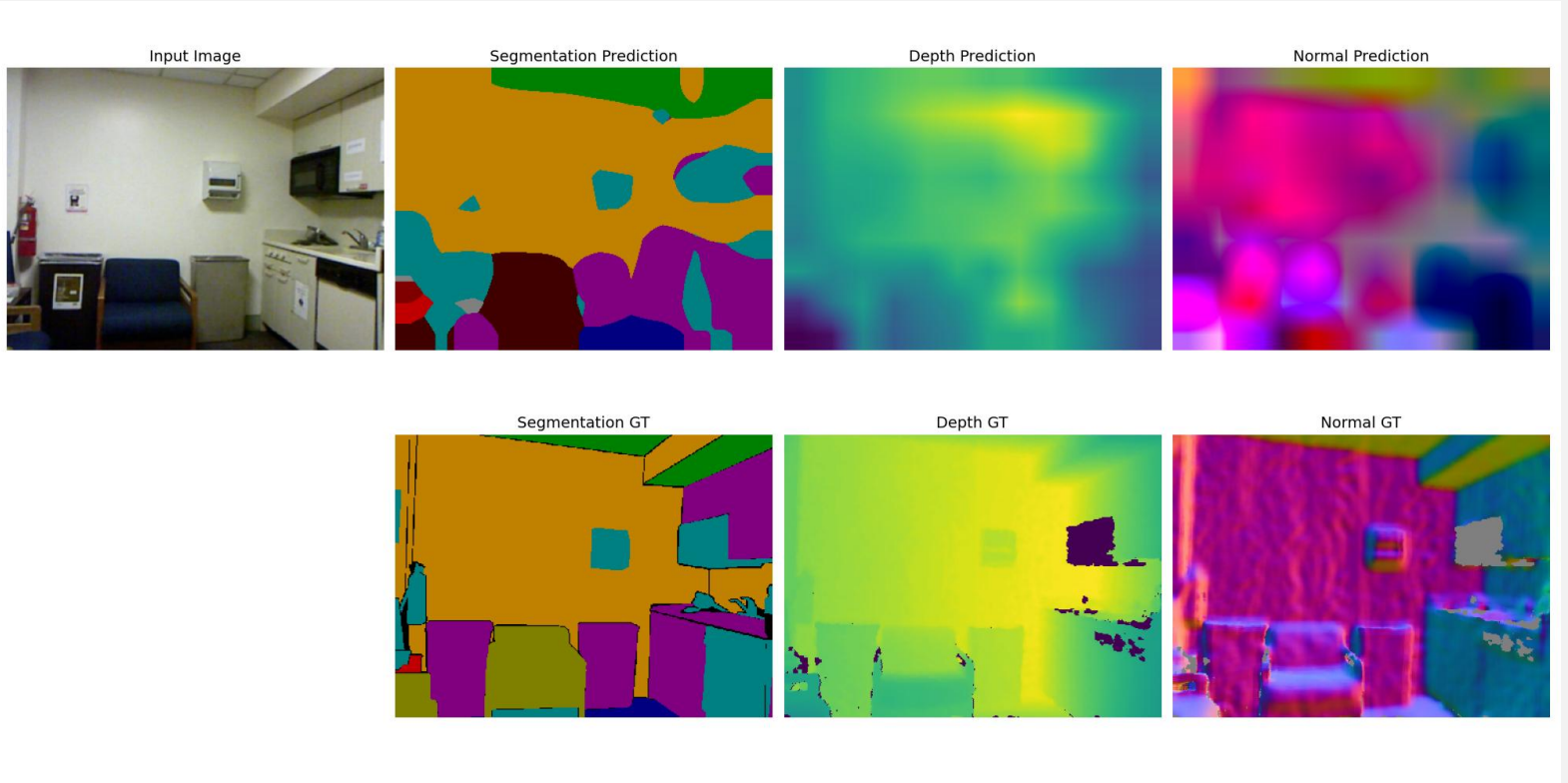
# 실험 결과

## Task Combination for PVTv2 MTL

| Model | Best Epoch | Seg mIoU (↑) | Seg pixAcc (↑) | Depth abs_err (↓) | Depth rel_err (↓) | Normal mean (↓) | Normal median (↓) | Normal ⟨11.25 (↑) | Normal ⟨22.5 (↑) | Normal ⟨30 (↑) | Total Params |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PVTv2- All (Pretrained) | 81 | 0.5463 | 0.7585 | 0.3770 | 0.1544 | 25.2240 | 18.7480 | 0.3195 | 0.5703 | 0.6827 | 39.02M |
| PVTv2 - Seg, Depth (Pretrained) | 84 | 0.5462 | 0.7577 | **0.3752** | **0.1524** | --- | --- | --- | --- | --- | 34.30M |
| PVTv2 - Seg, Norm (Pretrained) | 77 | **0.5471** | **0.7613** | --- | --- | 25.5821 | 19.1614 | 0.3129 | 0.5621 | 0.6755 | 34.30M |
| PVTv2 - Depth, Norm (Pretrained) | 54 | --- | --- | 0.3808 | 0.1534 | **24.9423** | **18.3239** | **0.3272** | **0.5778** | **0.6880** | 34.30M |

# 실험 결과

- 시각화


Input Image
Segmentation Prediction
Depth Prediction
Normal Prediction
Segmentation GT
Depth GT
Normal GT

# 실험 결과

## 한계점

- 데이터 제한성: NYUv2 데이터셋은 실내 환경에 최적화되어 있으며, 야외/도심 환경에 대한 일반화는 확인되지 않음
- 하이퍼파라미터 설정: 손실 가중치 및 학습률에 대한 최적화는 수동으로 설정되어 있으며 자동 튜닝은 반영되지 않음

# 실험 결과

## 핵심 인사이트

- PVT v2 기반 MTL 모델은 STL 대비 경쟁력 있는 성능을 보여주었으며, 파라미터 수 대비 효율성이 높음
- Swin Transformer MTL은 비교적 낮은 성능, PVT v2와 ResNet-50 기반 MTL 성능이 서로 비슷하며, 파라미터 수 부분에서는 PVT v2 MTL이 압도적으로 좋음
- Sementic Segmentation, Depth Estimation 사이에는 Shared Representation이 강하게 존재
- 두 Task는 Surface Normal Prediction에 Negative Transfer를 발생시킴

# 차후  실험 계획 및 기대 효과

## 차후 실험 계획

### 독립 변수의 다양화

- ✓ **MTL 구조**(Multi-Task Attention Network, Multi-gate Mixture-of-Experts etc.)
- ✓ **Weighting 전략**(Dynamic Weight Average, Nash MTL, Gradient Normalization etc.)
- ✓ **야외/다중 환경 데이터셋**에서의 일반화 테스트 수행 가능 (Cityscapes, Office-31 etc.)

### 최적화된 디코더 구조 개발

- ✓ Swin MTL에서 제안한 디코더 구조가 아닌 간단한 디코더 구조를 사용하였기에 Swin Transformer 백본 MTL 모형의 성능이 낮게 나왔다 생각
- ✓ 반대로 같은 학습 환경에서 높은 성능을 달성한 PVT v2 백본 MTL 모형에 추가적인 **디코더 최적화**를 수행할 시 높은 성능 향상 기대

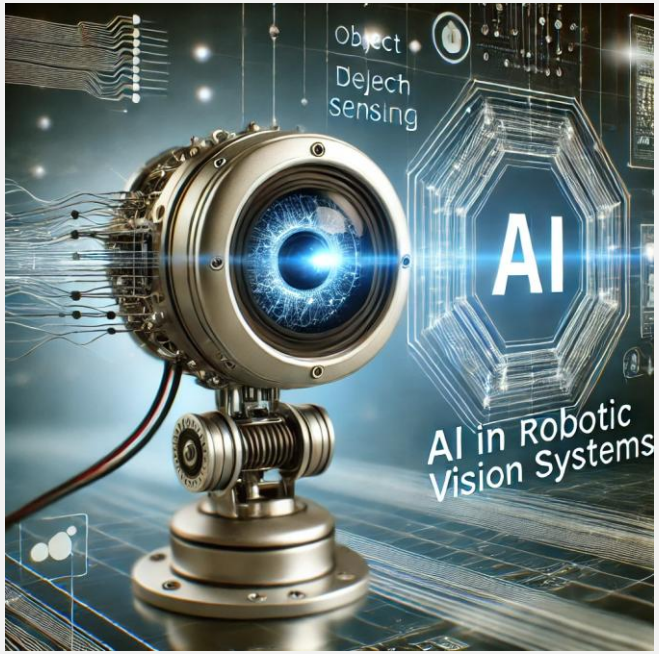
### bdd100k 데이터셋 활용을 통한 이미지 분류, 객체 탐지, 의미론적 분할에 대한  MTL 기법 구현

- ✓ NYUv2는 의미론적 분할, 깊이 추정, 법선 추정에 대한 annotation이 제공되었음
- ✓ bdd100k는 이미지 분류, 객체 탐지, 의미론적 분할에 대한 annotation이 존재
- ✓ bdd100k를 활용한 MTL 라이브러리가 없기에 기초적인 MTL 구조부터 구현이 필요
- ✓ 객체 탐지에 대한 decoder 구현이 까다롭기에 PVTv2 백본의 효용성을 알아본 후(현 실험) **자율주행 적용을 위한 MTL 구조 구현**

# 차후  실험 계획 및 기대 효과

## 기대 효과

- 단일 모형으로 다양한 태스크 처리가 필요한 임베디드 시스템
- 실내 환경에서 요구되는 실내 로봇 비전
- 실내 AR/VR 환경
- 자율 주행 자동차 (차후 실험을 통해 효용성 확인 예정)



# 참고 문헌


- ✓ https://github.com/median-research-group/LibMTL
- ✓ https://github.com/whai362/PVTv2-Seg
- ✓ https://github.com/whai362/PVT
- ✓ https://github.com/PardisTaghavi/SwinMTL
- ✓ https://github.com/microsoft/Swin-Transformer
- ✓ https://arxiv.org/abs/2103.14030
- ✓ https://arxiv.org/abs/2106.13797
- ✓ https://arxiv.org/abs/2203.14338
- ✓ https://arxiv.org/abs/2210.14793
- ✓ https://link.springer.com/article/10.1023/A:1007379606734
- ✓ https://arxiv.org/abs/2403.10662
- ✓ https://openaccess.thecvf.com/content/CVPR2022/html/Bhattacharjee_MulT_An_End-to-End_Multitask_Learning_Transformer_CVPR_2022_paper.html
- ✓ https://openaccess.thecvf.com/content/CVPR2022/html/Bhattacharjee_MulT_An_End-to-End_Multitask_Learning_Transformer_CVPR_2022_paper.html