##### Project Document

Test Result Document

|  |  |
| --- | --- |
| Project Name | Multi-Task Learning을 활용한 PVT v2 프레임워크 성능 개선 |

08 조

202001156 김수영

202002510 송재현

지도교수: 이종률 교수님 (서명)

Table of Contents

[1. Introduction 3](#_Toc197987914)

[1.1. Objective 3](#_Toc197987915)

[2. Experiment Result Report 4](#_Toc197987916)

[3. AI 도구 활용 정보 5](#_Toc197987917)

# Introduction

## Objective

이 프로젝트의 연구 목표는 MTL의 유효성을 검증하는 것으로, Multi-Task Learning이라는 학습 방법론에 대한 성능을 검증하는 것이다. 여기서 성능이란, 수행 능력으로써의 성능을 포함해, MTL의 특성상 경량화와 확장성 역시 매우 중요하다고 할 수 있다.

PVT v2는 비전 태스크에서 우수한 특징 추출 능력을 보여주고 있으며 연구 대조군으로 사용한 Swin Transformer보다 높은 성능을 보여주는 Transformer이다. 그렇기에 이를 MTL로 확장할 경우 Swin MTL과 기존 PVT v2 모형보다 높은 성능을 보여줄 것이라 예상되며 자율주행 환경에서 요구되는 객체 탐지, 의미론적 분할, 이미지 분류 등 다양한 작업을 동시에 효율적으로 처리할 수 있을 것으로 기대된다. 테슬라의 HydraNet 사례에서 볼 수 있듯이, 실제 자율주행 환경에서는 여러 비전 태스크들의 통합적 처리가 필수적이며, MTL을 활용하는 우리의 연구는 더 효율적인 모델 구조를 제안함으로써 실제 적용 가능성을 높이는 데 기여할 수 있다.

이에 본 프로젝트에서는 PVT v2 기반 프레임워크에 Multi-Task Learning을 접목한 모형을 만들어 성능을 검증하고, 자율주행 분야에서의 적용 가능성을 구체적인 실험을 통해 확인하고자 한다.

# Experiment Result Report

|  |
| --- |
| **1. 서론** |
| 1.1 실험 개요 |
| * 본 실험의 목적은 STL 기반의 PVT v2 모델을 MTL 구조로 확장했을 때, Semantic Segmentation, Depth Estimation, Surface Normal Prediction 성능에 어떤 영향을 주는지 정량적으로 평가하기 위한 것이다. * 가설: 사전 학습(pretraining)된 PVT v2 모델은 MTL 환경에서 STL보다 경쟁력 있는 성능을 유지하면서도, 파라미터 수 및 처리 효율 측면에서 우수한 결과를 도출할 수 있다. * 입력 데이터: NYUv2 * 태스크: Semantic Segmentation, Depth Estimation, Surface Normal Prediction * 실험 환경: * 모델: PVT v2, ResNet-50, Swin Transformer (사전 학습 포함/미포함), STL 기반 단일 태스크 모델 * 프레임워크: PyTorch * GPU: NVIDIA RTX 3080 10GB |
| 1.2 실험 방법 |
| * PVT v2 및 ResNet-50 백본을 사용하여 MTL 구조 구성 (사전 학습 포함/미포함) * STL 구조는 각 태스크별로 단일 모델로 학습 * 평가 지표: * Segmentation: mIoU, Pixel Accuracy * Depth: abs\_err, rel\_err * Normal: mean, median, <11.25°, <22.5°, <30° * 공통: 총 파라미터 수 |
| **2. 테스트 결과 상세** |
| 2.1 테스트 결과 개요 |
| |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **Model Configuration** | **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 Pretrain) | 94 | 0.2944 | 0.5558 | 0.5671 | 0.2410 | 30.9347 | 25.6424 | 0.2304 | 0.4478 | 0.5665 | 39,020,753 | | ResNet-50  (No Pretrain) | 99 | 0.2984 | 0.5620 | 0.5727 | 0.2403 | 32.6432 | 26.7498 | 0.2204 | 0.4321 | 0.5478 | 71,888,721 | | PVTv2 (Pretrained) | 81 | **0.5463** | **0.7585** | **0.3770** | **0.1544** | 25.2240 | 18.7480 | 0.3195 | 0.5703 | 0.6827 | 39,020,753 | | ResNet-50 (Pretrained) | 98 | 0.5373 | 0.7570 | 0.3846 | 0.1618 | **23.5492** | **16.8995** | **0.3542** | **0.6115** | **0.7215** | 81,888,721 | | STL – Segmentation (Pretrained) | 70 | **0.5526** | **0.7605** | --- | --- | --- | --- | --- | --- | --- | 29,575,373 | | STL – Depth (Pretrained) | 66 | --- | --- | 0.3854 | 0.1562 | --- | --- | --- | --- | --- | 29,572,289 | | STL – Normal  (Pretrained) | 31 | --- | --- | --- | --- | **23.5531** | 16.3254 | **0.3680** | **0.6149** | 0.7147 | 29,572,803 | | SwinT  (Pretrained) | 49 | 0.4891 | 0.7180 | 0.4157 | 0.4088 | 25.2206 | 18.9197 | 0.3120 | 0.5689 | 0.6838 | 112,467,977 | |
| 2.2 테스트 결과 상세 분석 |
| * 예외 사례, 비정상 결과, 의외의 결과 해석 * 왜 이 지점에서 성능이 낮았는가? (데이터 문제인지, 모델 한계인지) * Segmentation 성능: PVTv2 Pretrained는 SwinT 및 ResNet-50을 상회하거나 유사한 수준의 mIoU (0.5463) 및 pixAcc (0.7585)를 보였다. 이는 MTL 구조에서도 표현 손실 없이 유지 가능함을 시사한다. * Depth Estimation: abs\_err/rel\_err에서 PVTv2 Pretrained가 가장 낮은 값을 보였다. * Surface Normal Prediction: 평균 오차(25.22도), <11.25 정확도(31.95%) 측면에서 ResNet-50이 더 나은 결과를 보였지만, PVTv2 역시 경쟁력 있는 수치를 보였다. * Params: * PVTv2 MTL 모델은 약 39M 파라미터로 ResNet-50 대비 절반 수준이면서 유사한 성능을 달성하였다. * SwinT는 파라미터 수가 많음(112.5M)에도 불구하고 성능은 우수하지 않았다. |
| 2.3 실험 결과의 한계와 위협 요인 |
| - 데이터 제한성: NYUv2 데이터셋은 실내 환경에 최적화되어 있으며, 야외/도심 환경에 대한 일반화는 확인되지 않았다.  - 하이퍼파라미터 설정: 손실 가중치 및 학습률에 대한 최적화는 수동으로 설정되어 있으며 자동 튜닝은 반영되지 않았다. |
| **3. 결론** |
| - 핵심 발견 요약 : 무엇을 알게 되었는가?  - 후속 연구/ 실제 적용/ 시스템 개선을 위한 제안  - 활용 가능성 등  핵심 발견 요약   * 사전 학습된 PVT v2 기반 MTL 모델은 Segmentation, Depth, Normal 태스크 모두에서 STL 모델과 비교해 경쟁력 있는 성능을 보여주었으며, 파라미터 수 대비 효율성이 높았다. * SwinT는 성능이 불안정하고 자원 소모가 크며, PVT v2가 현실적 대안으로 부각된다   후속 연구 제안   * 야외/다중 환경 데이터셋에서의 일반화 테스트 * 태스크 간 손실 가중치 자동 조절 기법 적용   활용 가능성   * 경량 MTL 모델이 요구되는 실내 로봇 비전 * 실내 AR/VR 환경 * 단일 모델로 다양한 태스크 처리가 필요한 임베디드 시스템 |

# AI 도구 활용 정보

|  |  |
| --- | --- |
| 사용 도구 | GPT-4 |
| 사용 목적 | 활용 가능성 아이디어 보조 |
| 프롬프트 | * NYUv2같이 실내 데이터셋으로 학습된 MTL 모형이 어디에 활용될 수 있는지 예시를 들어줘 |
| 반영 위치 | 1. 활용 가능성 p.6 |
| 수작업  수정 | 있음(실내 AR/VR 환경 채택) |