P R O J E C T D O C U M E N T

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

1.1. 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.

일반적으로 자율주행에 대한 Dataset은 bdd100k를 이용하며 이미지 분류, 객체 탐지, 의미론적 분할에 대한 annotation 역시 존재한다. 단, Segmentation에 대한 annotation은 10K개만 존재하며 bdd100k를 대상으로 하여 MTL로 동시에 학습시키는 것을 시도한 논문이나 라이브러리를 찾지 못하였기에, Multi-Task Learning 학습 기법부터 Data Loader, 각 작업에 필요한 Decoder 전부 구현해야한다. 종합설계 강의 과정 내 이를 전부 구현하는 것은 현실적으로 힘들다 판단하여 본 실험에서는 최종적으로 PVY v2 프레임워크를 MTL로 확장하는 방식이 효용성이 있는지를 검증하고, 자율주행에 대한 MTL은 차후 실험 계획으로 넘겼다는 점 유의한다.

이에 본 프로젝트에서는 PVT v2 기반 프레임워크에 Multi-Task Learning을 접목한 모형을 만들어 성능을 검증하고, 자율주행 분야에서의 적용 가능성을 확인하고자 한다. 또한, NYUv2라는 실내 데이터를 이용하여 실내 로봇 비전 및 AR/VR 환경에 해당 모형을 접목시킬 수 있는지의 여부도 확인하도록 한다.

2.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, 일부 태스크를 조합한 MTL 모델
    - 프레임워크: PyTorch2.3.0+CUDA12.1
    - 주요 활용 라이브러리: mmcv(v2.2.0), mmengine, mmsegmentation, LibMTL
    - 사용 언어: python 3.10
    - GPU: NVIDIA RTX 3080 10GB

1.2 실험 방법

- PVT v2, ResNet-50, Swin Transformer백본을 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	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	94	0.2944	0.5558	0.5671	0.2410	30.9347	25.6424	0.2304	0.4478	0.5665	39.02M

Pretrain)											
ResNet-50 (No Pretrain)	99	0.2984	0.5620	0.5727	0.2403	32.6432	26.7498	0.2204	0.4321	0.5478	71.89M
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 Transformer (Pretrianed)	49	0.4891	0.7180	0.4157	0.4088	25.2206	18.9197	0.3120	0.5689	0.6838	112.46M
STL - Segmentation (Pretrained)	70	**0.5526**	**0.7605**	---	---	---	---	---	---	---	29.57M
STL - Depth (Pretrained)	66	---	---	0.3854	0.1562	---	---	---	---	---	29.57M
STL - Normal (Pretrained)	31	---	---	---	---	23.5531	**16.3254**	**0.3680**	**0.6149**	0.7147	29.57M
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

2.2 테스트 결과 상세 분석

- Segmentation 성능: PVTv2 Pretrained는 Swin Transformer 및 ResNet-50 기반 MTL 모형을 상회하는 수준의 mIoU (0.5463) 및 pixAcc (0.7585)를 보였다. 또한, Single Task Learning 모형과는 mIoU는 0.6%, pixAcc와는 0.2%정도의 차이를 보인다. 성능 하락이 아예 존재하지 않았다라고는 말하기는 힘들 수 있으나 그렇다고 성능이 크게 하락하지도 않았다 볼 수 있다. 즉, Single Task Learning 모형과 비교하여 성능 차이가 아주 미미하게 나며 다른 MTL 모형보다 높은 성능을 냄을 확인할 수 있다.
- Depth Estimation: abs_err/rel_err에서 PVTv2 Pretrained가 두 번째로 낮은 값을 보였다. 특기할만 사항으로 Depth Estimation에 대해 Single Task Learning을 진행한 모형보다 낮은 값

을 보여주었다. 이는 Segmentation과 Depth Estimation 간 Shared Representation이 존재하여 Multi-Task Learning 과정에서 이를 잘 학습하였다는 것을 함의하며 실제로 Segmentation과 Depth Estimation 두 가지 작업에 대한 MTL 모형의 abs_err/rel_err가 제일 낮게 나옴을 통해 확인할 수 있다.

- Surface Normal Prediction: Normal의 경우 측면에서 ResNet-50 기반 MTL이 PVT v2 기반 MTL보다 더 나은 결과를 보였다. 단, Normal에 대해 Single Task Learninig을 수행한 모형이 median값, <11.25° , 22.5° 인 normal의 비율면에서 더 높은 성능을 냄을 확인할 수 있다. 앞서 Depth Estimation을 통해 확인하였듯이 Segmentation과 Depth Estimation 사이의 Shared Representation이 존재함을 확인하였으므로 반대로 Normal과 다른 Task 간에는 이러한 Shared Representation이 존재하지 않고 오히려 Negative Transfer가 발생해 Normal Estimation에 대한 성능 하락을 일으킨 것이라 해석할 수 있다. Segmentatin & Normal MTL 모형의 경우와 Depth & Normal MTL 모형 모두에서 Normal Estimation에 대한 성능 하락이 두드러지게 나타남을 통해 이를 파악할 수 있다. 즉, 다른 Task들이 Surface Normal Prediction에 Negative Transfer를 일으킨다는 사실을 알 수 있다.
- Params:
    - PVTv2 MTL 모델은 39.02M 파라미터로 ResNet-50 대비 절반 수준(71.89M)이면서 유사한 성능을 달성하였다.
    - Swin Transformer MTL은 파라미터 수가 많음(112.5M)에도 불구하고 낮은 성능을 보여주었다.
    - PVT v2 백본 기반 STL 모형들은 29.57M으로 다중 작업을 수행할 시 PVT v2 MTL 모형이 파라미터 개수(39.02M) 측면에서 확실한 우위를 점할 수 있고(29.57*3=88.71) 다른 MTL 모형에 비해서도 큰 수준으로 파라미터 수가 차이가 남을 확인할 수 있다.

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

- 사전 학습된 PVT v2 기반 MTL 모델은 Segmentation, Depth Estimation 태스크에서 STL 모델과 비교해 경쟁력 있는 성능을 보여주었으며, 파라미터 수 대비 효율성이 높았다.
- Swin Transformer는 성능이 불안정하고 자원 소모가 크며, PVT v2와 ResNet-50 기반 MTL이 현실적 대안으로 부각된다. 특히, 파라미터 수 부분에서는 PVT v2 MTL이 압도적으로 유리하기에 제일 효용성이 높다 볼 수 있다.

- 단, Swin Transformer 백본은 Swin MTL 논문에서 제안한 디코더 구조를 사용한 것이 아니기 때문에 Swin Transformer에 덜 최적화된 학습 환경이었음을 고려해야하며, 반대로 PVT v2는 동일한 학습 환경에서 훨씬 나은 성능을 보여주었기에 PVT v2 MTL에 대한 적절한 최적화 디코더 구조를 고안하면 훨씬 유의미한 성능 향상이 이루어질 것이라 기대된다.
- Sementic Segmentation, Depth Estimation 사이에는 Shared Representation이 강하게 존재하며, 두 Task는 Surface Normal Prediction에 Negative Transfer를 일으킨다.

후속 연구 제안

- 야외/다중 환경 데이터셋에서의 일반화 테스트
- 태스크 간 손실 가중치 자동 조절 기법 적용
- bdd100k를 활용한 자율주행태스크에서의 PVT v2 성능 탐색
- M3ViT, MulT와 같은 MTL에 최적화된 모형과의 성능 비교
- PVT v2 MTL을 위한 최적화된 디코더 개발

활용 가능성

- 경량 MTL 모델이 요구되는 실내 로봇 비전
- 실내 AR/VR 환경
- 단일 모델로 다양한 태스크 처리가 필요한 임베디드 시스템

3.AI 도구 활용 정보

사용 도구	GPT-4, Gemini 2.5, Cursor AI
사용 목적	활용 가능성 아이디어 보조, 실험 코드 작성 보조, 분석 및 수정
프롬프트	● NYUv2같이 실내 데이터셋으로 학습된 MTL 모형이 어디에 활용될 수 있는지 예시를 들어줘 ● 현재 코드를 분석하고 수정해야할 부분을 확인해줘.
반영 위치	1. 활용 가능성 p.7 2. 실험 방법 p.4
수작업 수정	있음(실내 AR/VR 환경 채택, 코드 수정 및 디버깅)