

---

# Test Plan / Test Cases Design 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
1.1.	연구 질문/ 가설 .....	3
2.	TEST PLAN.....	4
3.	TEST CASES.....	6
4.	AI 도구 활용 정보 .....	7

## 1.

# Introduction

## 1.1. 연구 질문/ 가설

본 연구는 다음과 같은 연구 질문에 답하고자 한다:

- **RQ1**  
PVT v2 프레임워크에 MIL을 적용한 모델은 단일 작업 학습(SIL) 모델과 기존에 알려져 있는 MIL 모형에 비해 학습한 작업(이미지 분류, 객체 탐지, 의미론적 분할)에 대한 유의미한 성능 향상이 이루어지는가?
- **RQ2**  
Multi-Task Learning 기반 PVT v2 프레임워크는 자율주행 환경에서 기존 프레임워크(HydraNet 등) 대비 효율성 및 정확도 측면에서 어떤 차이를 보이는가?

본 연구는 다음과 같은 가설을 설정할 수 있다:

- **H1**  
MIL을 적용한 PVT v2 모델은 SIL 기반 PVT v2 및 기존 MIL 모델(Swin MIL)보다 정확도, AP, mIoU, #Param 등의 성능 지표에서 유의미한 향상을 보일 것이다.
- **H2**  
제안하는 MIL 기반 PVT v2 프레임워크는 자율 주행 분야에서 연구되는 모형인 HydraNet 대비 정확도 측면에서 유의미한 성능 향상을 보여줄 것이다.

## 2. Test Plan

<b>1. 배경과 목적</b>
1.1 배경
<ul style="list-style-type: none"> <li>- MTL은 여러 작업을 동시에 학습해 일반화 성능과 모델 경량화를 도모하지만, 부정적 전이, 작업 선정 기준 부재, 평가 불확실성 등의 한계가 있다. STL 모델을 MTL로 확장하는 것은 구조적 당위성이 있고, 성능 개선 사례(Swin MTL)도 있으며, 검증된 STL 모델(PVT v2)을 기준으로 MTL의 효과를 정량 평가할 수 있다는 장점이 있다.</li> </ul>
<b>2. 테스트 상세</b>
2.1 독립 / 종속 변수 정의
<ul style="list-style-type: none"> <li>- 독립 변수: <ul style="list-style-type: none"> <li>● 학습 구조 (STL vs MTL, Hard Sharing, Soft Sharing)</li> <li>● 백본 모델 종류</li> </ul> </li> <li>- 종속 변수: <ul style="list-style-type: none"> <li>● 성능 지표: Accuracy, AP (Average Precision), mIoU (mean Intersection over Union), #Parameters, 연산 속도</li> </ul> </li> </ul>
2.2 실험 대상 / 환경
<ul style="list-style-type: none"> <li>- 참가자: <ul style="list-style-type: none"> <li>● 김수영</li> <li>● 송재현</li> </ul> </li> <li>- 데이터셋: <ul style="list-style-type: none"> <li>● ImageNet (이미지 분류)</li> <li>● COCO (객체 탐지)</li> <li>● ADE20K (의미론적 분할)</li> </ul> </li> <li>- 사용 장비/도구 <ul style="list-style-type: none"> <li>● GPU: NVIDIA GTX 3080 10GB</li> <li>● 프레임워크: PyTorch</li> <li>● 모델: PVT v2 기반 구조 + MTL 확장 (Hard/Soft Sharing)</li> </ul> </li> </ul>
<b>3. 테스트 관리</b>
3.1 실험 절차 요약
<ul style="list-style-type: none"> <li>- 실험의 흐름</li> </ul> <ol style="list-style-type: none"> <li>1. 데이터셋 준비 및 전처리</li> <li>2. PVT v2 모델을 Hard/Soft MTL 구조로 확장</li> </ol>

3. 세 가지 태스크(의미론적 분할, 객체 탐지, 이미지 분류)에 대해 각각 학습
4. 기존 모델(Swin MTL, PVT v2-B1, B2)과 비교
5. 성능 지표를 기반으로 유의미성 검증

### 3.2 측정 지표 및 도구

- 정량 평가 지표, 정성 평가 지표, 사용 도구 등을 명시

정량 지표:

- Accuracy (이미지 분류)
- AP (객체 탐지)
- mIoU (의미론적 분할)
- #Parameters
- Inference Time

도구: COCO API, mIoU 평가 스크립트, FLOPs 계산기 등

## 3. Test Cases

1. 테스트 케이스					
1.1 테스트 케이스 명세					
Id	대상(모델/조건)	실험 조건	테스트 데이터	평가지표	예상 결과
TC-1.1	PVT v2-B0 (STL)	단일 작업 학습	ImageNet	Accuracy	Accuracy 70%
TC-1.2	PVT v2-B0 (STL)	단일 작업 학습	COCO	AP	AP 37.1
TC-1.3	PVT v2-B0 (STL)	단일 작업 학습	ADE20K	mIoU	mIoU 37.2%
TC-2	Swin MTL	MTL 구조 (기존 구조)	3개 데이터셋	Accuracy, AP, mIoU	Acc: 81.3% AP: 50.5 mIoU: 50.35%
TC-3	PVT v2-B0 + MTL (Hard Sharing)	다중 작업, 공유 구조	3개 데이터셋	Accuracy, AP, mIoU	Acc: 75% AP: 43 mIoU: 43%
TC-4	PVT v2-B0 + MTL (Soft Sharing)	Soft 공유 구조	3개 데이터셋	Accuracy, AP, mIoU	Acc: 72% AP: 40 mIoU: 39%
1.2 검증 기준(metric)					
<ul style="list-style-type: none"> <li>- Accuracy: 이미지 분류 정확도</li> <li>- AP: 객체 탐지에서의 평균 정밀도</li> <li>- mIoU: 의미론적 분할에서의 평균 IoU</li> <li>- #Param: 모델의 전체 파라미터 수</li> <li>- Inference Time: 테스트 1장당 소요 시간(실시간 분석용)</li> </ul> <p>PVT v2에 대한 성능 예상 결과는 해당 논문으로부터 발췌하였으며, Swin MTL에 대한 부분은 해당 논문에서 Swin Transformer와 비교한 결과를 바탕으로 설정하였다. mIoU에 대한 성능에 대해서는 논문에서 PVT v2와 비교하지 않았기에 Swin Transformer 논문을 참고하여 작성하였다. PVT v2-B0는 가장 작은 모형이기에 해당 모형을 백본으로 활용하는 TC-3에서는 Swin MTL보단 낮고 기존 모형보다는 높을 것이라 예상하고 있다. 또한, Soft Sharing의 경우는 각 Task가 서로 이미지에 관한 작업들이기에 Task 간 Shared Representation이 높을 것이라 예상하여, Hard Sharing보다는 낮은 성능이 나올 것이라는 예상을 하고 있다.</p>					

## 4. AI 도구 활용 정보

사용 도구	GPT-4
사용 목적	테스트 케이스 측정 지표 설명, 실험 설계 도움
프롬프트	<ul style="list-style-type: none"> <li>● 성능 평가 지표(Accuracy, AP mIoU, #Param, Inference Time)에 대해 설명해줘</li> <li>● 실험 설계 시 고려할 만한 사항들을 알려줘</li> </ul>
반영 위치	<ol style="list-style-type: none"> <li>1. 테스트 케이스 검증 기준 (p.6)</li> <li>2. 테스트 계획 부분 (p.4)</li> </ol>
수작업	있음 (필요 부분 추출 및 논리 보강)
수정	