

Multi-Task Learning 연구 계획

내가 지금 Multi-Task Learning에 대한 주제를 바탕으로 졸업프로젝트를 진행중이야. 학부생 수준에서 10주 안에 수행해야하기에 굳이 어렵게 하지 않으려 해. 현재 구체적인 연구 계획은 정해지지 않았고 연구 방향성은 Multi-Task Learning의 유효성 검증에 관한 것을 바탕으로 할 예정이야. 그래서 연구를 어떻게 진행해야할지 구체적인 문제 정의를 하고 싶어. 일단 내가 원하는 방향성은 기존에 알려져 있는 모형들을 Multi-Task Learning으로 학습시켜 본 후 성능향상이 실제로 이루어지는지, Multi-Task Learning의 실질적인 유효성을 검증해 보여주는 것이야. 이러한 방향성을 바탕으로 연구 목표를 더 구체화해줘. 예를 들어 최신 모형을 바탕으로 Multi-Task Learning을 수행한다 할 시 써먹을 논문을 구체적으로 정한다거나, 실제 이용되고 있는 AI서비스의 모형들을 Multi-Task Learning으로 수행해보는 등 말이야.

좋은 방향성이에요! 연구 목표를 더 구체화하기 위해 아래 사항들을 조금 더 알려줄 수 있을까요?

1. 어떤 종류의 태스크를 염두에 두고 있나요? 예: 자연어처리(NLP), 컴퓨터 비전(CV), 음성 인식 등
2. 단일 태스크 vs 멀티 태스크에서 비교해보고 싶은 성능 기준은 무엇인가요? 예: 정확도, F1 score, 일반화 능력 등
3. 사용할 수 있는 컴퓨팅 자원(GPU 유무, 제한 등)이나 데이터셋에 대한 제약이 있을까요?
4. 어떤 수준의 모델을 다루고 싶나요? 예: BERT나 ResNet 같이 잘 알려진 모델? 아니면 가벼운 모델 위주?

이 정보를 알려주시면, 학부 수준에서 10주 안에 수행 가능한 현실적인 주제와 실험 설계를 도와드릴 수 있을 것 같아요!

1. 컴퓨터비전과 관련된 태스크를 염두에 두고 있어
2. 기본적인 성능 기준은 정확도(Accuracy)야 하지만, 만약 성능이 비슷하거나 약간 안좋게 나온다고 하여도 파라미터의 수가 유의미하게 줄었으면 그것 또한 경량화가 잘 됐다는 증거로써 성능에 포함시킬 예정이야.
3. 컴퓨팅자원과 데이터셋에 대한 제약은 없어
4. 잘 알려진 모델을 사용하는 것이 좋겠지? 실제 상용화된 모형이 Multi-Task Learning으로 학습될 경우 그에 대한 성능이 실제로 향상되는 것이 목적이니까

좋아, 너의 연구 방향성과 조건에 맞춰 구체적인 연구 목표와 실험 설계를 구성해볼게. Multi-Task Learning이 컴퓨터 비전 분야에서 얼마나 유효한지, 정확도 향상 또는 파라미터 절감 측면에서 검증할 수

졸업 프로젝트 계획: 컴퓨터 비전에서의 다중 태스크 학습(MTL) 유효성 검증

1. 컴퓨터 비전 MTL의 대표 연구 사례 (논문 및 구현)

- **MultiNet: Real-time Joint Semantic Reasoning for Autonomous Driving** – Teichmann 등(2018)은 자율주행 환경에서 하나의 CNN으로 이미지 분류(도로 유형), 객체 탐지(차량), 도로 영역 세그멘테이션 3가지 태스크를 동시에 수행하는 모델을 제안했습니다
[AR5IV.ORG](#) . 단일 인코더-디코더 구조로 세 태스크의 특징을 공유하여 KITTI 데이터셋 기준 실시간(23 FPS) 추론이 가능하며, 도로 세그멘테이션에서는 당시 최신 단일태스크 성능을 능가했습니다 [AR5IV.ORG](#)
[AR5IV.ORG](#) . 저자들은 **전체 코드와 학습 스크립트**를 공개하여 재현성을 높였습니다 [AR5IV.ORG](#) [AR5IV.ORG](#)
(GitHub: [MarvinTeichmann/MultiNet](#)).
- **Multi-Task Learning Using Uncertainty to Weigh Losses** – Kendall 등(CVPR 2018)은 **장면 이해**를 위해 **세 가지 태스크(모노컬러 깊이 추정, 의미론적 세그멘테이션, 인스턴스 세그멘테이션)**를 결합한 MTL 모델을 선보였습니다
[OPENACCESS.THECVF.COM](#) [OPENACCESS.THECVF.COM](#) . 각 태스크 손실에 **불확실도 기반 가중치**를 자동 학습시키는 기법을 도입하여, **모든 태스크에서 개별 모델보다 높은 정확도**를 달성함을 보여주었습니다
[OPENACCESS.THECVF.COM](#) . 이 연구는 **태스크 간 손실 밸런싱의 중요성**을 부각했으며, PyTorch 구현 예제도 커뮤니티에 의해 제공되고 있습니다 [GITHUB.COM](#) .
- **MTI-Net: Multi-Scale Task Interaction Networks** – Vandenhende 등(ECCV 2020)은 **다중 밀도 예측 태스크**(예: 세그멘테이션+깊이)에 대해 **다중-스케일 특징 결합**을 활용하는 MTL 모델을 제안하여 NYUv2, CityScapes 등에서 단일태스크 대비 **유사 혹은 향상된 성능**을 보였습니다. 이 모델은 **공개 코드로** 구현되어 있으며
[GITHUB.COM](#) , 후속 설계인 **Attentive Single-Tasking** (CVPR 2019) 등과 함께 **MTL 간 간섭을 줄이고 태스크별 중요한 특징에 집중**하는 전략을 보여줍니다 [GITHUB.COM](#) .
– 그 외: 초기 방법인 **Cross-Stitch Networks**(CVPR 2016)는 서로 다른 태스크의 중간 표현을 교차 연결해 부분 공유하는 **소프트 파라미터 공유 MTL**을 도입했습니다 [GITHUB.COM](#) . 또한 **Pad-Net**(CVPR 2018), **AdaShare**(AAAI 2020) 등은 어떤 계층을 공유하고 분리할지 **학습으로 결정**하거나 **가중치 마**

스킨을 통해 하나의 네트워크로 다중 작업을 수행하는 등 다양한 접근이 제안되었습니다. 최신에는 Transformer 기반 통합 모델들(ICCV 2021 "UniT" 등)도 등장하여 이미지, 자막 등 멀티모달 MTL 까지 확장되고 있습니다.

2. 다중 태스크로 자주 조합되는 태스크 유형과 이유

- **객체 탐지 + 세그멘테이션** - 검출된 객체의 마스크까지 예측하는 인스턴스 세그멘테이션 문제로 많이 등장합니다. 검출과 픽셀 분할이 결합되면 하나의 모델로 ****물체의 위치(박스)****와 ****모양(마스크)****을 모두 얻을 수 있어 활용도가 높습니다. 대표적으로 Mask R-CNN 모델이 **Faster R-CNN 검출기에 세그멘테이션 분기**를 추가하여 두 작업을 동시에 수행했고, 마스크 예측 학습이 **검출 성능 향상**에도 도움이 될 수 있음이 보고되었습니다

[OPENACCESS.THECVF.COM](https://openaccess.thecvf.com) . 이러한 조합은 **자율주행** 등에서 객체 인지를 정교하게 하기 위해 널리 쓰입니다.

- **이미지 분류 + 객체 검출** - 한 이미지에 대한 ****전반적 분류(씬 분류 혹은 주요 객체 판별)****와 **개별 객체의 탐지**를 함께 다루는 경우입니다. 예를 들어 **자율주행 장면**에서 도로 종류나 날씨 등을 분류하면서 차량/보행자 검출을 동시에 수행할 수 있습니다. MultiNet 연구에서도 **전방 도로 유형 분류를 차량 검출 태스크와 함께 학습**시켜 높은 수준 맥락 정보가 객체 탐지에 도움이 되도록 했습니다

[ARXIV.ORG](https://arxiv.org) . 이 조합은 **전역 정보(분류)**와 **지역 정보(탐지)**를 공유하여, 별도 모델 둘을 운용할 때보다 **경량화**와 **실시간 처리** 측면에서 이점이 있습니다 [ARXIV.ORG](https://arxiv.org) .

- **세그멘테이션 + 깊이 추정(거리)** - ****장면의 의미론적 이해(무엇이 어디에 있는지)****와 ****기하학적 이해(얼마나 떨어져 있는지)****를 동시에 얻기 위해 자주 묶입니다. 두 태스크 모두 **픽셀 단위 예측**을 필요 하므로, 초반부 영상 특징을 크게 공유할 수 있습니다. 예를 들어 실내 RGB-D 데이터셋인 NYUv2에서는 **객체/공간의 클래스와 픽셀 단위 거리**를 한꺼번에 예측하도록 MTL을 구성하며, **물체 경계와 깊이 변화 경계**가 일치하는 등의 시너지를 통해 성능을 높입니다. 또한 한 모델에 통합하면 로보틱스 응용 등에서 **실시간 처리**에 유리합니다

[OPENACCESS.THECVF.COM](https://openaccess.thecvf.com) . 이러한 이유로 **자율주행**(도로 장면에서 세그멘테이션+깊이)이나 **증강현실**(장면 이해)에서 많이 연구됩니다 [ARXIV.ORG](https://arxiv.org) .

- **기타 조합: Semantic 세그멘테이션 + 경계 검출**(경계 정보로 세그멘테이션 정확도 향상), **사람 키포인트 검출 + 객체 탐지**(사람 객체 영역과 관절 키포인트를 한 번에 추론), **멀티 라벨 분류 태스크**(예: 얼굴 이미지에서 다수 속성 예측) 등도 MTL로 자주 다뤄집니다

[GITHUB.COM](https://github.com) [GITHUB.COM](https://github.com) . 실제 공개 데이터셋도 이러한 복합 태스크를 포함하도록 만들어지고 있습니다 (예: CityScapes는 **주행 영상당 차량/도로 세그멘테이션, 깊이추정 레이블** 제공 [GITHUB.COM](https://github.com) ; COCO는

3. MTL 성능 비교의 일반적인 기준

- **정확도(Accuracy)** - 분류 문제에서는 **Top-1 정확도** 등으로 성능을 비교합니다. 다중 클래스 분류의 경우 **정답 레이블과 예측이 일치하는 비율**로 정의되며, MTL에서는 **단일 태스크 모델 대비 정확도 저하 없이 유지되는지**를 주로 확인합니다

ARXIV.ORG . 경우에 따라 **Top-5 정확도**도 참고하지만, 학부 수준 프로젝트에서는 Top-1로 충분합니다.

- **평균 정밀도(mAP)** - 객체 검출 성능은 **mAP (mean Average Precision)** 지표가 표준적으로 사용됩니다. 각 클래스별 **Precision-Recall 곡선**에서 ****Average Precision(AP)****을 계산하고 평균낸 값으로, **값이 높을수록 탐지 성능이 좋음**을 의미합니다. 예를 들어 Pascal VOC에서는 IoU 0.5 기준 AP를 클래스별 계산해 mAP 산출하고, COCO에서는 여러 IoU 기준을 평균낸 mAP를 씁니다

ARXIV.ORG . MTL 실험에서는 **검출 태스크의 mAP가 단일 모델 대비 얼마나 유지 or 향상되는지**가 비교됩니다.

- **교차 분율(IoU) 및 mIoU** - 세그멘테이션에서는 **Intersection over Union (IoU)** 혹은 클래스별 IoU의 평균인 ****Mean IoU (mIoU)****가 주 척도입니다. **예측 마스크와 정답 마스크의 겹침 정도**를 나타내며, 값이 클수록 픽셀 단위 예측이 정확함을 의미합니다. 예컨대 CityScapes 벤치마크도 클래스별 IoU를 평균낸 **mIoU**로 순위를 매깁니다. MTL 모델에서 세그멘테이션 성능이 단일 모델 대비 **mIoU로 비교되며**, 유사한 수준이면 성공으로 봅니다.

- **모델 복잡도 지표** - 다중 태스크의 **효율성 평가**를 위해 **모델 파라미터 수, 메모리 사용량, 연산량 (FLOPs), 추론 속도(FPS)** 등을 함께 고려합니다. 특히 **모델의 총 파라미터 수**는 단일 태스크 모델 여러 개와 MTL 모델의 **크기 차이**를 정량화하는데 중요합니다. 예를 들어 MultiNet은 세 작업을 하나로 합쳐도 **45ms 이내 추론**이 가능하고, 작업별 개별 모델 대비 **메모리/연산 효율이 향상됨**을 강조했습니다

ARXIV.ORG . 일반적으로, **정확도가 비슷하면 파라미터/연산이 적은 쪽이 우수한 MTL 성과**로 간주됩니다. 실제 연구들도 MTL로 **계산량과 메모리 풋프린트 절감**을 보고하고 있습니다 ARXIV.ORG .

- **기타 지표**: 태스크에 따라 **정밀도-재현율(F1), ROC-AUC(분류), 평균 절대 오류(Depth 회귀)** 등 특화 지표를 병행하기도 합니다. 다만 프로젝트에서는 각 태스크별 **대표 성능지표와 **모델 경량화 효과(파라미터 수 감소)****에 초점을 맞추면 됩니다.

4. 학부 수준 구현을 위한 실험 세팅 예시

다중 태스크 학습을 위한 일반적인 모델 구조 예시 - 공유 합성곱 인코더와 태스크별 디코더/헤드로 구성된 다

AR5IV.ORG . 이러한 구조를 바탕으로 아래와 같은 실험을 설계할 수 있습니다:

- **데이터셋 선택 및 태스크 구성:** 가능한 한 동일한 데이터셋에서 여러 태스크 라벨을 제공하는 경우가 이상적입니다. 예를 들어:
 - **Pascal VOC:** 단일 이미지에 ****객체 검출(바운딩 박스)****과 **세그멘테이션(픽셀 마스크)** 어노테이션이 모두 있습니다. 하나의 모델에 ResNet-50 백본을 공유시키고, **검출용 헤드**(Faster R-CNN 스타일)와 **세그멘테이션용 헤드**(FCN 스타일)를 달아 **MTL 학습**을 할 수 있습니다. 이때 **VOC 클래스 존재 여부를 예측하는 이미지 분류 태스크**도 추가하면 3태스크 MTL도 가능합니다. Pascal VOC는 데이터 크기가 적당해 학부 수준에서 다루기 좋습니다.
 - **CityScapes:** 자율주행 영상에 대해 **차량/도로 세그멘테이션**과 **깊이(혹은 장애물 거리)** 정보를 활용한 MTL. CityScapes는 주로 세그멘테이션용이지만, **스테레오 영상으로부터 얻은 깊이지도**를 추가 태스크로 활용할 수 있습니다
GITHUB.COM . MobileNet 같은 **경량 백본**+디코더 두 개로 실험하면, **경량 MTL**의 정확도와 효율을 관찰할 수 있습니다. 자원이 충분하다면 CityScapes의 **인스턴스 세그멘테이션**까지 포함해 **3태스크 (분할+인스턴스+깊이)** 실험도 가능합니다.
 - **NYUv2 (Indoor):** 실내 RGB-D 데이터셋으로 **13-class 세그멘테이션**, **Depth 회귀**, **Surface Normal 예측** 총 3가지 태스크 라벨을 제공합니다
GITHUB.COM . 이 데이터셋은 규모가 크지 않아 **ResNet-18 공유 + 태스크별 작은 디코더**로도 학습이 가능하며, 여러 MTL 연구의 벤치마크로 쓰여 **참고 구현 코드**가 풍부합니다. 예를 들어, 앞서 언급한 MTAN의 공개코드에서는 NYUv2에 대한 세그멘테이션+Depth+노말 MTL 구현을 제공합니다.
 - **MS COCO:** 대규모이지만, **검출(Bounding box)**, **세그멘테이션(마스크)**, **키폰트(인간 포즈)** 라벨을 모두 갖춘 대표 데이터셋입니다
GITHUB.COM . 학부 프로젝트에서는 **전체 COCO**를 학습하기엔 무겁기 때문에, **미니 배치크기 조정**

이나 일부 클래스만 사용하는 서브셋으로 실험해볼 수 있습니다. COCO를 활용하면 실제 **멀티 태스크 모델 (예: CenterNet 기반)**이 어떻게 **포괄적 인지**를 하는지 경험할 수 있습니다.

- **모델 구조 설계:** 기본적으로 **하드 파라미터 공유** 전략을 사용합니다. ResNet, EfficientNet, MobileNet 등 검증된 분류 모델의 백본을 공유 인코더로 활용하고, **태스크별 출력 헤드**를 붙입니다 [AR5IV.ORG](#) . 예를 들어 EfficientNet-B0 백본 + (검출용 FPN+헤드) + (세그멘테이션용 업샘플링 헤드) 구조를 만들 수 있습니다. 혹은 ResNet 공유 후 **분류 헤드**(글로벌 풀링+FC), **세그멘테이션 헤드**(디코더 컨볼루션), **검출 헤드**(Region Proposal + 박스 회귀 분기) 등을 병렬로 달아줍니다. 초기 가중치는 **ImageNet pretrained**를 사용하면 수렴이 빠르고 성능이 높아집니다. 참고로 MultiNet도 **ImageNet 사전학습 가중치**로 인코더를 초기화했습니다 [AR5IV.ORG](#) [AR5IV.ORG](#) .
- 구현 난이도를 낮추기 위해, **Detectron2**(Facebook)나 **MMDetection**(OpenMMLab) 같은 오픈소스 검출 프레임워크를 활용해 **세그멘테이션 분기**를 추가하거나, **Mask R-CNN**구조를 변형해 실험할 수 있습니다. 이러한 프레임워크는 ResNet+FPN 백본 등 모듈화되어 있어 커스텀 MTL 구조를 구성하기에 용이합니다.
- 또는 **LibMTL** [GITHUB.COM](#) 과 **Multi-Task-Learning-PyTorch** [GITHUB.COM](#) [GITHUB.COM](#) 같은 **MTL 전용 라이브러리/코드베이스**를 활용하면, 여러 벤치마크 데이터셋에 대한 유명 MTL 알고리즘(예: 교차 스티치, 불확실도 가중, Gradient blending 등)을 쉽게 적용해볼 수 있습니다. 이러한 틀은 **여러 알고리즘의 공정한 비교 실험**도 지원하므로, 본 프로젝트에서도 기본 공유모델 대비 **향상된 기법**(예: 동적 손실 가중)도 시험해볼 수 있습니다.
- **실험 계획 및 평가:** 각 조합에 대해 **단일 태스크 모델**과 **MTL 모델**을 동일한 조건에서 학습시켜 **성능과 효율 비교**를 수행합니다. 예를 들어, Pascal VOC 검출 기준 **단일태스크 Faster R-CNN**의 mAP와 **MTL 모델**(검출+세그)의 mAP를 비교하고, 세그멘테이션 mIoU도 단일 모델(FCN 등)과 비교합니다. 이때 **MTL 모델의 파라미터 수는 단일 모델 대비 얼마나 절감되었는지** 산출하고, **정확도 손실이 없는 수준인지** 함께 분석합니다. 만약 정확도가 소폭 하락하더라도 파라미터/메모리/속도 면에서 **유의미한 이득**이 있다면 MTL의 **유효한 결과**로 판단합니다. 실제 한 연구에서는 **MTL 공유학습으로 각 태스크별 성능을 유지하면서도 개별 모델들보다 총 추론시간을 크게 단축**할 수 있음을 보였습니다 [ARXIV.ORG](#) [ARXIV.ORG](#) . 이러한 trade-off를 정량적으로 보고서에 정리하고, 어떤 태스크 조합에서 **상호 보완 효과**가 나타났는지도 논의합니다. 마지막으로, 최신 MTL 논문의 구조나 기법을 구현에 반영한 경우 그 효과도 추가 실험하여, 학부 수준에서 **MTL의 가능성과 한계**를 종합 평가합니다.

참고 자료 및 코드 레퍼런스: 위에서 언급한 논문들의 공식 구현이나 관련 오픈소스 프로젝트를 적극 활용하세요. MultiNet 등의 코드는 공개되어 있어 구조를 참고하기 좋고

[ARXIV.ORG](#) , LibMTL 라이브러리에는 다양한 MTL 알고리즘과 데이터셋 예제가 포함되어 있습니다

[GITHUB.COM](#) . 논문 제목(예: *Multi-Task Learning for Dense Prediction Tasks: A Survey* [ARXIV.ORG](#))과

GitHub 레포 링크는 원문 그대로 제공하였으니, 세부 구현시에 해당 레퍼런스를 확인하면 큰 도움이 될 것입니다.

     출처