# 모듈명: 인공지능실습 팀 이름: 아무래도(C)

**날짜**: 2024.11.23(금)

**팀원**: 이재욱, 김민성, 이하윤, 손관우, 오준서

# 학습목표

**1. 데이터 획득 및 관리**

* 모델 학습에 필요한 데이터셋을 확보하고, 라벨링 품질을 높이기 위한 효율적인 방법을 학습하여 데이터 정확도를 향상시킨다.

**2. 데이터 활용 및 학습 효율화**

* 학습 데이터 부족 문제를 해결하고 데이터 증강을 통해 데이터셋을 보강하며, 학습 시간 단축과 컴퓨팅 자원 절약을 위한 최적화 기법을 익힌다.

**3.** **모델 최적화 및 성능 개선**

* 하이퍼 파라미터 조정과 학습률 스케줄링을 통해 최적의 학습 환경을 마련하고, 모델의 성능을 검증하며 개선하는 방법을 체계적으로 학습한다

**가설**/**해결안**(Idea)

* **가설:**

YOLO v5 모델의 성능은 데이터의 양과 질을 개선하고 학습률과 같은 주요 하이퍼 파라미터를 최적화함으로써 향상될 수 있다. 데이터 증강을 통해 데이터셋의 다양성을 높이고, Pre-trained YOLO 모델을 활용한 전이 학습과 Fine-tuning을 통해 특정 데이터셋에 적합한 성능을 끌어낼 수 있을 것으로 예상된다. 또한, 학습률 스케줄링과 같은 최적화 기법을 적용하면 학습 안정성과 정확도를 더욱 향상시킬 수 있을 것이다.

* **해결안:**

양질의 데이터를 확보하고 데이터 증강 기법을 통해 데이터셋의 다양성을 증가시킨다. 학습률, 배치 크기 등의 하이퍼 파라미터를 조정해 최적의 학습 환경을 마련하며, 학습률 스케줄링 기법을 활용해 학습 초반에는 빠르게 수렴하고 이후에는 세밀한 학습이 이루어지도록 한다. 또한, Pre-trained YOLO 모델을 기반으로 전이 학습을 활용해 초기 학습을 가속화하고, Fine-tuning을 통해 성능을 최적화한다.

**알고 있는 사실**(Facts)

1. **데이터셋의 중요성**: YOLO 모델은 객체 탐지의 특성상 대규모 데이터셋을 필요로 한다. 데이터셋의 크기와 다양한 객체 특성의 포함 여부가 모델 성능에 큰 영향을 준다.
2. **컴퓨팅 자원의 요구**: YOLO는 대규모 데이터셋 학습 시 많은 컴퓨팅 자원이 필요하다. 이를 위해 GPU 클러스터나 클라우드 컴퓨팅 자원을 적극 활용하는 것이 좋다.
3. **데이터 라벨링의 중요성**: 데이터 라벨링의 질이 모델의 성능에 미치는 영향은 매우 크다. 잘못된 라벨링이 포함된 데이터는 모델 성능을 저하시킬 수 있다.
4. **전이 학습의 효과**: 기존에 학습된 YOLO 모델을 활용하는 전이 학습은 데이터를 효율적으로 사용하면서도 높은 성능을 낼 수 있는 방법이다.

**더 알아야 할 사항** (Learning Issue)

1. **데이터 증강 기법의 최적화**: 어떤 데이터 증강 기법이 YOLO 모델의 성능 향상에 가장 효과적인지 파악하기 위한 실험이 필요하다. 회전, 크기 조절, 색상 변형 등 다양한 증강 기법을 적용해 보는 것이 좋다.
2. **하이퍼 파라미터 튜닝**: 학습률, 배치 크기, 최적화 알고리즘 등 여러 하이퍼 파라미터가 모델 성능에 큰 영향을 미친다. 최적의 값을 찾기 위해 여러 시도를 해봐야 한다. 아직까지 구체적인 수정 방안을 알지 못하기 때문에 파라미터 파일에 대한 이해와 공부가 수반되어야한다.
3. **효율적인 라벨링 도구 탐색**: 데이터 라벨링을 더 효율적으로 할 수 있는 최신 도구나 기술을 조사할 필요가 있다. 특히 경계 박스를 정밀하게 설정할 수 있는 도구가 필요하다. 수업시간에 알려준 어플도 괜찮지만 팀원들이 이전에 사용해본 MakeSense 라벨링 사이트를 사용하는 것도 좋은 방법일 것 같다.
4. **평가 지표의 심화 이해**: YOLO v5의 성능을 평가하는 지표(AP, IoU 등)에 대해 더 깊이 이해하고, 모델 성능을 다양한 상황에서 분석할 수 있어야 한다.

**앞으로의 계획 (Action Plan)**

**1. 데이터셋 확장 및 질적 개선**

* 다양한 환경과 객체를 포함한 데이터셋을 확보하여 모델 학습에 사용할 데이터를 확대한다.
* 데이터 증강 기법(회전, 크기 조절, 색상 변형 등)을 적용해 데이터셋의 다양성을 높이며, PADDING 기법을 추가로 도입하여 데이터 처리 방식을 개선한다.

**2. 라벨링 도구 활용**

* 효율적인 데이터 수집과 라벨링 작업을 위해 LabelImg 및 MakeSense 라벨링 사이트를 활용하며, 팀원들이 일관성 있는 라벨링을 수행할 수 있도록 프로세스를 표준화하고 작업 가이드를 설계한다.

**3. 로컬 및 클라우드 환경 병행 학습**

* CUDA를 지원하는 로컬 환경에서 학습을 수행하며 GPU 성능을 활용한 실험을 진행한다.
* Google Colab과 같은 클라우드 환경에서도 병행 학습을 수행하여 대규모 데이터셋 학습 및 다양한 하이퍼 파라미터 실험을 효율적으로 진행한다.

**4. 전이 학습 및 Fine-tuning**

* 기존에 학습된 Pre-trained YOLO 모델을 활용하여 초기 학습 속도를 가속화하고, 특정 도메인 데이터셋에 맞춘 Fine-tuning을 통해 성능을 최적화한다.

**5. 학습 최적화 및 반복 개선**

* 학습률, 배치 크기 등 주요 하이퍼 파라미터를 조정하고 학습률 스케줄링을 도입하여 학습 초반에는 빠르게 수렴하고 이후에는 정밀한 학습이 이루어지도록 한다.
* 반복 실험을 통해 학습 결과를 검토하며 데이터셋 및 학습 과정에서의 개선점을 지속적으로 반영한다.

**6. 모델 성능 평가 및 피드백**

* 학습된 모델의 성능을 mAP(mean Average Precision), IoU(Intersection over Union)와 같은 주요 지표를 활용하여 평가한다.
* 테스트 데이터에서의 모델 성능을 분석하고, 모델의 약점을 파악하여 데이터 보완 및 하이퍼 파라미터 조정을 통해 성능을 지속적으로 개선한다.
* 필요에 따라 k-fold 교차 검증을 활용해 모델의 일반화 성능을 검증하고, 실험 결과를 기반으로 피드백 루프를 구축하여 반복적인 개선을 진행한다.