YOLOv8n 모델 실험 분석 보고서

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명앞으로 여러 모델 네트워크 구조를 실험 및 분석하기에 앞서, 기본 모델부터 분석을 하는 것이 앞으로의 방향성을 정하는 데에 있어 기준이 된다고 생각한다. 따라서 우선적으로 YOLOv8n 모델에 대해 분석한다.

|  |  |
| --- | --- |
| 모델 이름 | v8n\_org\_100best |
| 모델 사양 | 168 layers  3.0M parameters  8.1 GFLOPs |
| mAP50 | 0.524 |
| FPS | : 151.5(NPU), |

위의 사진은 yolov8n 모델을 nextchipt dataset으로 100epochs 학습 시킨 후의 결과이다.

Ultralytics 공식 문서에 따르면 모델 학습을 할 때, 만족할만한 결과를 얻으려면 데이터셋에서 class 별로 1500장 이상의 이미지를, class 별로 10000개 이상의 instances를 확보해야 한다고 한다.( <https://docs.ultralytics.com/yolov5/tutorials/tips_for_best_training_results/>)

결과를 보면 알 수 있듯이, 버스, 트럭, 자전거, 오토바이의 데이터 수가 적다는 것을 확인할 수 있다. 특히, 자전거의 경우 데이터의 수가 가장 적기 때문에 결과 값이 가장 안 좋은 것으로 판단된다.

따라서, 모델의 성능을 얻기 위해서는 **필수적으로 데이터셋을 확보 및 추가해야 한다.** 현재 이를 위해 데이터를 수집하고 추가하고 있는 과정에 있다.

여러 reference에 따르면, 객체 감지 분야에서 소형 객체를 감지하는 것이 까다롭다고 한다. 객체의 크기가 작고, 큰 객체에 가려지는 상황이나 복잡한 환경에 의해 소형 객체 탐지에 대한 성능이 떨어진다. 따라서, 실험의 첫 방향을 소형 객체 탐지의 성능을 높이는 것을 목표로 YOLOv8n의 네트워크 구조를 변경할 계획이다.

참조 : Wang, H., et al. (2023). *SOD-YOLOv8: Enhancing YOLOv8 for Small Object Detection in Traffic Scenes*. arXiv preprint arXiv:2408.04786.

**첫 번째 실험 : v8n\_p2\_100best**

1. 실험 개요
   1. 배경 : 이전 실험에서 객체 탐지 분야에서 소형 객체 탐지가 어렵기 때문에 실험의 첫 방향으로 소형 객체 탐지의 성능을 높이는 것으로 하였다. 소형 객체 탐지의 성능을 높이면 전체 성능이 올라갈 것이기 때문에 소형 객체 탐지를 늘릴 수 있도록 이번 실험에서는 기존 P3, P4, P5 detect에서 P2 detect을 추가한다.
2. 모델 수정 및 중량화 전략
   1. P2 : 모델의 Backbone은 다양한 특징을 추출하여 feature map을 생성한다. Neck에서는 생성된 여러 feature map을 FPN과 같은 구조를 통해 효과적으로 통합한다. 이러한 과정을 통해 P3(80, 80), P4(40, 40), P5(40, 80)의 feature map 결정되며, 이를 바탕으로 detect를 수행한다. Feature map의 크기가 크면 해상도가 좋기 때문에 소형 객체의 정보가 뚜렷하고, 큰 객체의 경우 convolution 연산이 커버할 수 있는 영역이 좁기 때문에 큰 객체를 한번에 포괄하기 어렵다. 따라서 P3의 해상도는 상대적으로 크므로 소형 객체를 탐지하는데 유리하고, P5의 해상도는 반대로 낮기 때문에 큰 객체를 탐지하는데 유리하다. Feature map의 크기가 클수록 소형 객체를 탐지하는 것에 유리하므로, 이번 실험에서는 feature map에 P2(160, 160)를 추가하여 소형 객체 탐지의 성능이 올라갈 것으로 예상한다.ㄴ
3. 실험 세팅
   1. 모델 구조 : 사진
   2. 구조 설명 :
4. 실험 결과
   1. Model.summary (params, layers, GFLOPs)

|  |  |
| --- | --- |
| 모델 이름 | v8n\_p2\_100best |
| 모델 사양 | 207 layers  2.9M parameters  12.2 GFLOPs |
| mAP50 | 0.617 |
| FPS | : 151.5 |

* 1. 텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

     자동 생성된 설명성능 측정 (Performance Metrics):
     1. mAP : 각 모델의 mAP (Mean Average Precision) 측정.
     2. FPS(NPU 기준)

1. 결과 및 비교
   1. 결과: 전체, 클래스별, confusionmatrix
   2. 비교 : v8\_org\_100ep랑 비교
2. 분석 및 결론
   1. 분석
      1. 문제점
   2. 향후 연구 방향 (Future Work):
   3. 결론 : 무슨 모델, 성능 값, 문제점
3. 참고 문헌
   1. YOLOv8, GhostConv, CBAM 관련 논문 및 자료.
   2. APACHE5 NPU 기술 문서 및 관련 자료.
   3. 기타 경량화 관련 연구들