YOLOv8s 모델 실험 분석 보고서

1. 실험 개요
   1. 배경

Nextchip사로부터 받은 데이터셋으로 학습시킨 YOLOv8s 기본 모델로 evaluation을 해보았다. 이 과정에서 오토바이, 자전거, 사람과 같은 작은 크기의 객체에 대해 성능이 낮은 탐지 결과와 약 16ms 후반의 Inference time을 확인했다. 이러한 단점과 NPU에서의 inference time을 개선하기 위한 2가지 접근법을 고안했다.

첫번째는 추가 학습을 통해 소형 객체를 더욱 잘 탐지하도록 하기 위해 소형 객체의 비율을 높인 데이터셋을 구축하는 방법이다.

두번째는 소형 객체에 대한 feature map extraction을 강화하는 방법, 즉 모델 네트워크를 강화하는 방법이다.

이 실험에서는 여러 논문들을 참고하여 얻은 Insight를 통해 Ghost Convolution Layer를 활용하여 소형 객체에 대한 성능을 높이고, NPU에 탑재 가능한 처리 속도를 가진 모델 네트워크 구축을 목표로 한다.

- Reference

* + [Huangfu, Z., Li, S., & Yan, L. (2024). Ghost-YOLO v8: An Attention-Guided Enhanced Small Target Detection Algorithm for Floating Litter on Water Surfaces. *School of Information Engineering, North China University of Water Resources and Electric Power*, 80(3), 3714-3728.](https://www.techscience.com/cmc/v80n3/57888)
  + [Zhou, H., Yu, Y., Wang, K., & Hu, Y. (2024). A YOLOv8-Based Approach for Real-Time Lithium-Ion Battery Electrode Defect Detection with High Accuracy. *Electronics*, 13(1), 173.](https://doi.org/10.3390/electronics13010173)

1. 모델 수정 및 경량화 전략
   1. Ghost Convolution

기존의 Conv 모듈은 입력 채널에 대해 기본 Convolution 연산을 수행하며, 각 입력 채널에 대해 동일한 계산량이 소모된다. 예를 들어, 256개의 입력 채널이 있을 경우 256개의 필터를 이용하여 모든 채널에 대해 연산을 하므로 연산량이 많다. 이로 인해 계산량이 커지고, 모델의 처리 속도가 느려질 수 있는 반면, **Ghost Convolution**은 입력 채널의 절반 정도만 표준 Conv Layer를 사용하고, 나머지 절반은 연산량이 적은 추가 Conv Layer를 사용한다. 이 추가 Conv Layer는 주요 특징을 추출하는 데 필요한 연산을 최소화하고, 간단한 feature map을 생성하여 채널을 확장하는 역할을 한다. 이렇게 하면, 기존의 Conv 모듈에서는 모든 채널에 대해 계산을 수행했던 것에 비해, Ghost Conv는 적은 계산량으로 비슷한 성능을 유지할 수 있다.

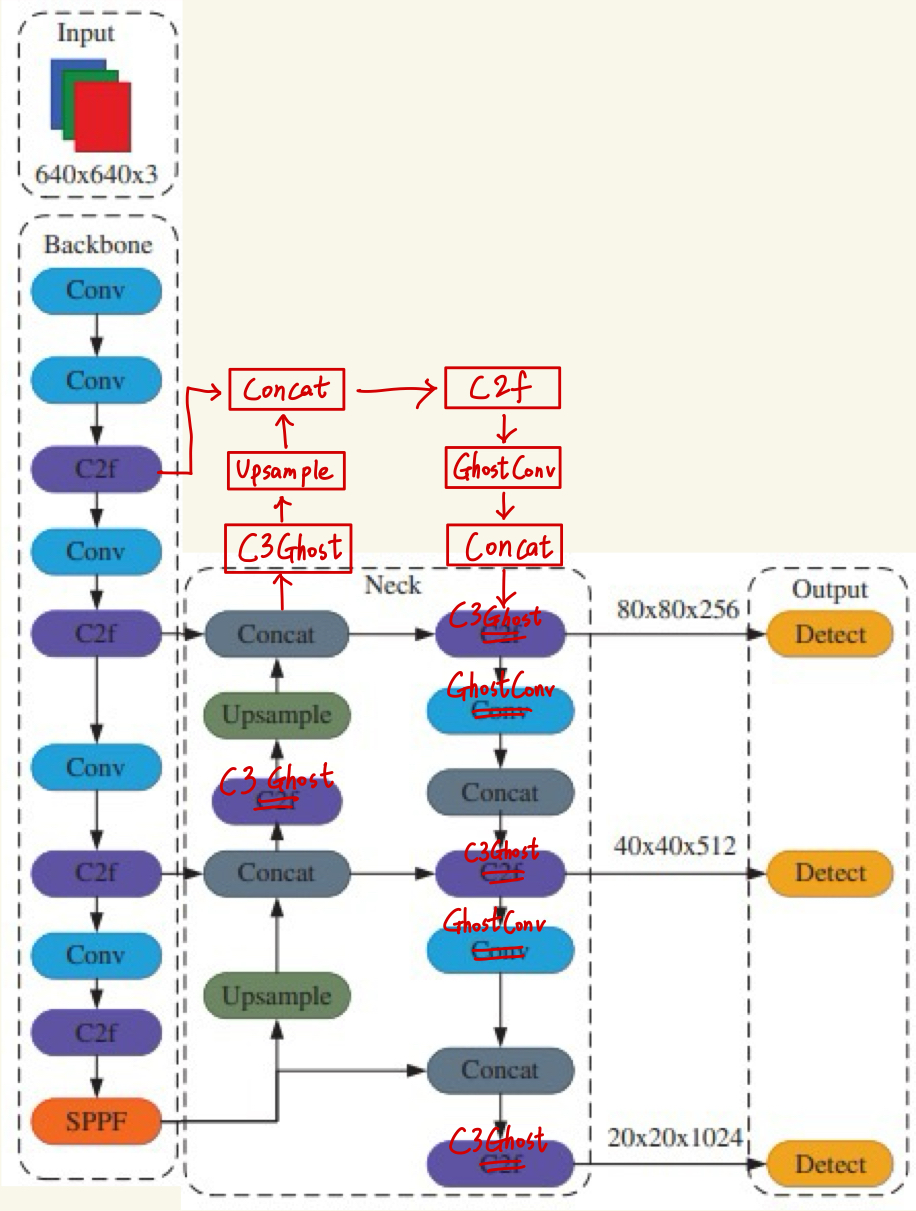
즉, **Ghost Conv**는 기존의 Conv보다 훨씬 적은 계산량으로 모델의 처리 속도를 향상시킬 수 있는 장점을 가진다. 이를 통해, 모델 네트워크에서 Neck 부분의 레이어를 추가하여 소형 객체에 대해 대응할 수 있으며, 줄어든 연산량으로 인해 모델이 모델이 무거워지지 않는다는 가설을 세우고 실험을 진행한다.

텍스트, 스크린샷, 도표, 디자인이(가) 표시된 사진

자동 생성된 설명

네트워크 구조는 위 사진과 같다. 기본 YOLOv8 네트워크 구조에 Backbone의 160 × 160 크기의 작은 소형 객체 레이어(빨간색으로 그려 놓은 곳)를 Neck에 추가하여 성능을 개선하며 연산량과 파라미터 수를 줄이기 위해 Neck부분의 Conv 레이어를 Ghost Conv 레이어로 수정하여 실험을 진행하는 전략을 세웠다.

1. 실험 세팅
   1. 모델 구조



* 1. 구조 설명

기본 YOLOv8s 모델의 Inference time이 16ms 후반이기 때문에 많은 경량화보다는 Neck에 Layer를 추가하고 C3Ghost와 Ghost Convolution을 진행하는 방식으로 네트워크를 수정했다.

Backbone에서는 Ghost Convolution은 Convolution보다 더 좋은 Feature 특성을 추출하는 것은 아니기 때문에 기존의 Convolution으로 유지한다. 기존 YOLOv8s 네트워크에서 소형 객체를 탐지하는 P3(80x80)부분의 특성을 보강하기 위하여 새로운 P2(160x160) 레이어를 쌓아 더 작은 소형 객체의 특성을 추가해준다. 즉, Neck에서 160x160 Layer까지 정보를 담아서 특성을 추출하기 때문에 160x160의 정보를 Ghost Convolution한 feature 특성을 P3 Layer에 함께 전달되어 소형 객체에서의 약한 탐지 기능을 보강한다.

1. 실험 결과
   1. Model information

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 모델명 | Layer 수 | Parameters | GFLOPs | mAP |
| v8s\_org\_100ep | 168 | 11,127,906 | 28.4 | 0.605 |

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 모델명 | Layer 수 | Parameters | GFLOPs | mAP |
| v8s\_ghost\_c3ghost | 269 | 9,443,842 | 27.7 | 0.649 |

텍스트, 스크린샷, 블랙, 폰트이(가) 표시된 사진

자동 생성된 설명

* 1. 성능 측정
     1. mAP
        1. Ghost Convolution과 C3Ghost block을 적용한 모델의 mAP(0.649)가 기본 YOLOv8s 모델의 mAP(0.605)보다 성능이 약 7% 증가했다..
     2. FPS 비교
        1. Ghost Convolution과 C3Ghost block을 적용한 모델의 FPS(약 50)가 기본 YOLOv8s 모델의 FPS(약 59)보다 약 16% 감소했다.

1. 결과 및 비교
   1. 결과
      1. 전체, 클래스별, confusion matrix
   2. 비교 : v8s\_org\_100ep랑 비교하기
2. 결론 및 분석
   1. 분석
      1. 문제점
   2. 향후 연구 방향
   3. 결론: 무슨 모델인지, 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