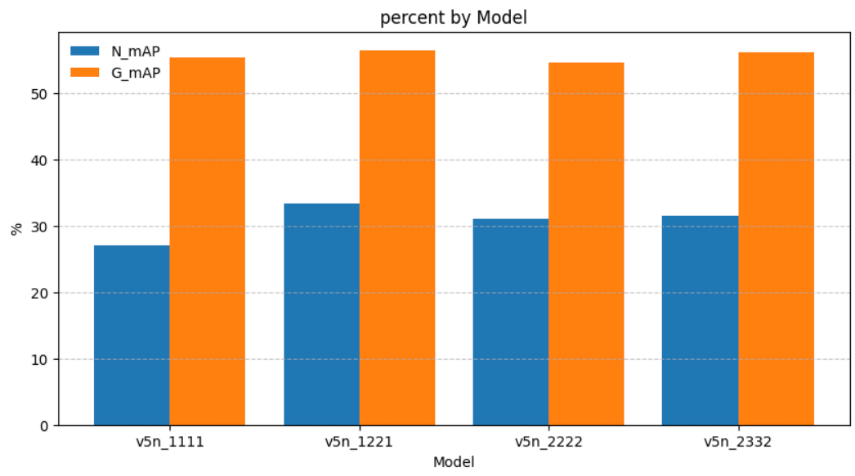
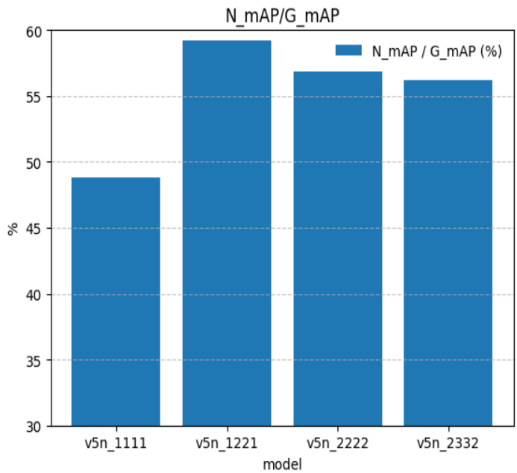
**Bottleneck 반복 수와 보존율 분석 보고서**

1. **서론**
   1. **배경**
      1. Yolo 모델의 구조 형성 방법  
         : 모델의 구조를 선언하는 yolo.yaml에서 모델의 사이즈를 결정하는 [depth, width, max\_channel] 인자가 있으며, 각 모듈이 어떻게 적용되는지에 대해 선언한 [from, number, module, args] 형태의 인자가 있다. 이 중 C2f, C3 모듈에 대해서 둘 모두 내부에 Bottleneck 모듈이 존재하는데,  
         [ Bottleneck 반복 수 = number \* depth(s, n 모델 기준: 0.33) ]  
         의 형태로 반복 수가 결정된다.
      2. C2f 모듈의 동작 방식  
         \* 보존율 = NmAP / GmAP ( GPU대비 NPU에서 성능이 재현되는 정도)  
         : C2f 모듈은 입력 채널을 두 개로 나눈 후 첫 번째는 shortcut을 통과하고 두 번째는 내부 Bottleneck 모듈에 의해 반복 수 만큼 새롭게 추출된 텐서 그룹들을 만들게 된다. 이 때 텐서 그룹수가 늘어날수록 min/max scale이 다양해지므로 양자화 과정에서 오차를 발생시킬 확률이 늘어난다.  
          따라서 C2f 모듈의 Bottleneck 모듈의 반복수가 늘어나면 최종 concat에서 텐서 그룹수가 늘어나 양자화 오차를 유발해 보존율에 영향을 미치게 된다.
      3. C3 모듈의 동작 방식  
         : C3 모듈은 먼저 입력 채널을 두 개로 나눈 후 첫번째는 cv1을 거친 후 shortcut을 통과한다. 두번째는 cv2를 통해 전달되며, 이후 내부 Bottleneck 모듈을 반복 수 만큼 통과하여 특징을 추출한다. 이 때 C2f 모듈과의 차이점은 Bottleneck 반복 수가 늘어나더라도 최종 concat이 받는 텐서 그룹수는 유지된다는 것이다.
      4. Concat 입력 텐서 그룹수의 양자화 오차 유발  
          결과적으로 v8 모델에서 C2f는 반복수가 늘어나면 concat이 입력받는 텐서 그룹수 증가함과 동시에 Bottleneck 모듈의 적용 횟수가 늘어나므로 변인통제가 되지 않아 입력 텐서 그룹 수만의 영향력을 판단하기 어렵다.  
          따라서 본 연구는 concat 입력 텐서 그룹수가 보존율에 끼치는 영향을 분석하기 위한 선행 연구로써 v5n 모델에서 Bottleneck 반복 수에 따른 보존율과의 상관관계를 먼저 분석한다.
   2. **실험 준비**
      1. 실험 모델
         1. v5n\_1111
         2. v5n\_1221
         3. v5n\_2222
         4. v5n\_2332
   3. **목적**
      1. 실험 1 : Bottleneck 반복 수와 보존율의 상관관계 분석  
         : [모델 1]을 기준으로 C3 내부의 Bottelneck 모듈의 반복수를 점진적으로 증가시킨 [모델 2], [모델 3], [모델 4]을 보존율의 관점에서 비교하여 두 요소의 상관관계를 분석한다.
   4. **평가 지표**
      1. Parameter 수
      2. NPU/GPU 비율
      3. NPU 성능
      4. GPU 성능
      5. FPS
2. **본론**
   1. **실험 설계**
      1. 실험 방법
         1. 동일한 데이터로 학습한 모델에 대해 동일한 입력 데이터를 사용하여 실험 모델 별 성능 측정.
            1. 학습 데이터 : 제공받은 train set과 자체적으로 수집한 data set을 병합
            2. 입력 데이터 : 제공받은 test set
      2. 모델 설계  
         \* 1111 ~ 2332는 Backbone에서 C3/C2f 모듈 내의 Bottleneck 반복 횟수  
         \* **[1.3 Backbone 구조 변경 분석 보고서]**의 [모델 1, 2, 3, 4]와 동일
         1. v5n\_1111
         2. v5n\_1221
         3. v5n\_2222
         4. v5n\_2332
   2. **실험 결과 및 분석**
      1. 실험 1 : Bottleneck 반복 수에 따른 보존율 분석
         1. 실험 항목
            1. 대조군 : v5n\_1111  
               실험군1 : v5n\_1221, 실험군2 : v5n\_2222, 실험군3 : v5n\_2332  
               : v5n 모델에 대해 동일 조건으로 Backbone의 반복 횟수만 변경하여 변인 통제, 가장 낮은 v5n\_1111을 대조군으로 사용한다.
         2. 모델 설명 :
            1. v5n\_1111: 기본 yolov5n 모델에 bottleneck 반복 수 1, 1, 1, 1로 설정한 모델
            2. v5n\_1221: 기본 yolov5n
            3. v5n\_2222: 기본 yolov5n 모델에 bottleneck 반복 수 2, 2, 2, 2로 설정한 모델
            4. v5n\_2332: 기본 yolov5n 모델에 bottleneck 반복 수 2, 3, 3, 2로 설정한 모델
         3. **가설** **– Bottleneck 반복 수가 늘어나면 보존율이 하락할 것이다.**  
             **Bottleneck** 반복 수가 증가하면 모델의 연산 복잡도가 늘어나고 레이어 간 연결과 정보의 흐름이 깊어진다. 이는 8bit 연산에서 양자화 손실을 증가시킬 가능성이 있다[1]. 따라서 1111 모델에서 Bottleneck 반복수가 증가한 모델일수록 보존율 하락폭이 클 것이다.
         4. 결과 :

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Parameter 수 | **NPU/GPU 비율** | NPU 성능 | GPU 성능 | FPS |
| v5n\_1111 | 2.41M | **48.81%** | 27.4% | 55.4% | 218.11 |
| v5n\_1221 | 2.46M | **59.21%** | 33.43% | 56.46% | 204.52 |
| v5n\_2222 | 2.31M | **56.82%** | 31% | 54.56% | 206.18 |
| v5n\_2332 | 2.68M | **56.22%** | 31.56% | 56.14% | 190.07 |

* + - 1. 분석 :
         1. **v5n\_1111 -> v5n\_1221**

보존율 기준 48.81%에서 59.21로 약 10.4% 상승폭을 보인다. NPU는 Bottleneck 반복 수가 늘어나면 더 효율적으로 병렬 연산을 수행하여 성능 개선에 크게 기여한다.

* + - * 1. **v5n\_1221 -> v5n\_2222**

보존율 기준 59.21%에서 56.82%로 약 2.4% 감소폭을 보인다. Bottleneck 반복수가 더 증가함에 따라 NPU에서의 연산 처리 효율이 감소한다.

* + - * 1. **v5n\_2222 -> v5n\_2332**

보존율 기준 56.82%에서 56.22%로 0.6%의 추가 감소폭을 보인다. 과도한 Bottleneck의 반복은 NPU에서의 적정 연산량보다 연산량이 포화되어 성능 개선이 거의 없거나 저하를 초래한다. Bottleneck 반복 수가 너무 많아지면 Bottleneck 블록 안에서 주요 정보가 희석될 수 있다.

1. **결론**
   1. **주요 결론**
      1. Bottleneck 반복수 증가에 따른 초기 성능 개선

Bottleneck 반복수가 1, 1, 1, 1에서 1, 2, 2, 1로 증가한 경우는 보존율이 약 10.4% 상승하였다. 이는 NPU가 Bottleneck 구조의 연산에 최적화되어 성능 향상에 기여할 수 있다고 판단한다.

* + 1. Bottleneck 반복수 증가의 한계

Bottleneck 반복수를 1, 2, 2, 1에서 2, 2, 2, 2로 증가한 경우는 보존율이 약 2.4%감소한 것을 볼 수 있다. 이는 Bottleneck 반복수가 증가하며 NPU에서의 연산량이 포화상태에 도달하여 연산 효율성이 저하된 것으로 보인다.

* + 1. Bottleneck 반복수의 과도한 증가의 부정적인 영향

Bottleneck 반복수가 2, 2, 2, 2에서 2, 3, 3, 2로 증가한 경우는 보존율이 약 0.6% 추가 감소한 것이 보인다. 이는 Bottleneck 블록의 과도한 반복이 NPU의 연산에서 연산 부담이 더 커지는 것을 의미한다. 또한 레이어와 레이어 사이에서 주요 정보가 희석되어 성능이 저하될 수 있다.

* 1. **향후 연구 방향**

레이어에서 Bottleneck외의 concat요소에 대한 분석과 다른 다양한 레이어(c2f, c3Ghost)들의 구조를 분석하여 보존율에 어떻게 영향을 끼치는지 알아볼 예정이다. 또한 conv레이어와 Ghostconv레이어를 비교하여 conv를 사용했을 때의 성능과 보존율, Ghostconv를 사용했을 때의 성능을 분석할 예정이다.

1. **참조 문헌**

[1] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," *arXiv preprint arXiv:1712.05877*, 2017.