**P2 적용 및 Backbone 구조 변경 분석 보고서**

1. **서론**
   1. **배경**
      1. 소형 객체 탐지 성능 향상  
         : **[1.2 P2 기법 적용 분석 보고서]**에서 얻은 결론에 따라 P2를 적용해 본 프로젝트의 주요 목표인 소형 객체에 대한 성능을 특히 향상시키고자 한다.
      2. Backbone 구조 변경을 통한 성능 향상  
         : **[1.3 Backbone 구조 변경 분석 보고서]**에서 얻은 결론에 따라 Backbone의 Bottleneck 반복 회수를 증가시켜 모델의 성능을 개선하고자 한다.
      3. P2를 적용한 Backbone의 구조 분석  
         : **[배경 i]**, **[배경 ii]**에 따라, 본 실험에서는 Yolo v8s 모델에 P2를 적용한 v8s\_P2 모델을 기반으로 Backbone에서 소형 객체의 정보를 주로 추출하는 P2와 P3단의 반복 횟수를 조절하여 모델의 성능을 향상시키려 한다. 따라서 본 실험에서는 이전 실험들에서 얻은 두가지 기법이 동시에 적용되었을 때에도 같은 경향성이 유지되는지 확인하는 동시에 모델의 성능을 최적화하고자 한다.
   2. **실험 준비**
      1. 실험 모델
         1. v8s\_P2
         2. v8s\_P2\_1321
         3. v8s\_P2\_2221
         4. v8s\_P2\_3221
         5. v8s\_P2\_3321
         6. v8s\_P2\_4221

* 1. **목적**
     1. 실험 1 : P2단 반복 횟수에 따른 성능 변화 분석  
        : [모델 1]을 기준으로 P2단의 반복 횟수만 증가시킨 [모델 3], [모델 4], [모델 6]을 비교한 결과를 통해 P2단의 반복 횟수에 따른 모델의 성능을 관찰한다. 이 때 결과는 본 실험의 목적에 부합하게 모델 성능과 객체 크기별 정확도 두가지 관점에서 분석한다.
     2. 실험 2 : P3단 반복 횟수에 따른 성능 변화 분석  
        : [모델 1]을 기준으로 P3단의 반복 횟수만 증가시킨 [모델 2]를 비교한 결과와 [모델 4]를 기준으로 마찬가지인 [모델 5]를 비교한 결과를 종합해 P3 단의 반복 횟수에 따른 모델의 성능을 관찰한다. 이때 결과는 마찬가지로 모델 성능과 객체 크기별 정확도 두가지 관점에서 분석한다.
  2. **평가 지표**
     1. 모델 성능 : GPU mAP, NPU mAP, FPS
     2. 객체 크기별 정확도

1. **본론**
   1. **실험 설계**
      1. 데이터셋
         1. 각 모델들이 학습에 사용된 데이터와 테스트에 사용된 데이터는 동일
            1. 학습 데이터 : 제공받은 train set과 validation set을 병합
            2. 테스트 데이터 : 제공받은 test set
      2. 모델 설계

\* 1321 ~ 3321는 Backbone에서 C2f 모듈 내의 Bottleneck 반복 횟수  
 본 실험에 사용된 모델은 모두 P2 적용

* + - 1. v8s\_P2 : YOLOv8s 기본 모델에 P2 Layer를 추가한 모델
      2. v8s\_P2\_1321 : P2 Layer를 추가한 후, Bottleneck 반복 횟수를 (1, 3, 2, 1)로 설정한 모델
      3. v8s\_P2\_2221 : P2 Layer를 추가한 후, Bottleneck 반복 횟수를 (2, 2, 2, 1)로 설정한 모델
      4. v8s\_P2\_3221 : P2 Layer를 추가한 후, Bottleneck 반복 횟수를 (3, 2, 2, 1)로 설정한 모델
      5. v8s\_P2\_3321 : P2 Layer를 추가한 후, Bottleneck 반복 횟수를 (3, 3, 2, 1)로 설정한 모델
  1. **실험 결과 및 분석**
     1. 실험 1 P2를 추가한 후 Backbone에서의 Bottleneck 반복 횟수 조정에 따른 성능 변화 분석
        1. 실험 항목
           1. 대조군1 : v8s\_P2 / 실험군1 : v8s\_P2\_1321, v8s\_P2\_2221
           2. 대조군1 : v8s\_P2\_2221 / 실험군1 : v8s\_P2\_3221
           3. 대조군1 : v8s\_P2\_3221 / 실험군1 : v8s\_P2\_3321
        2. **가설 – P2를 추가한 모델에서 Backbone의 Bottleneck 반복 횟수가 증가할수록 성능이 증가할 것이다.**: [분석 보고서1.2]에서 P2 Layer를 추가했을 때 모델에서 성능이 증가함을 확인했다. [분석 보고서1.3]의 내용과 조합하여 P2 Layer를 추가한 모델에서 Backbone의 Bottleneck 반복 횟수를 늘리면 성능이 더 증가할 것이다.
        3. 결과
           1. v8s\_P2 vs (v8s\_P2\_1321, v8s\_P2\_2221)  
              텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

              자동 생성된 설명

|  |  |  |  |
| --- | --- | --- | --- |
|  | v8s\_P2 | v8s\_P2\_1321 | **v8s\_P2\_2221** |
| GPU mAP (%) | 67.82 | 68.02 | **68.74** |
| NPU mAP (%) | 48.34 | 47.49 | **49.94** |

* + - * 1. v8s\_P2\_2221 vs v8s\_P2\_3221  
           텍스트, 스크린샷, 도표, 그래프이(가) 표시된 사진

           자동 생성된 설명

|  |  |  |
| --- | --- | --- |
|  | **v8s\_P2\_2221** | v8s\_P2\_3221 |
| GPU mAP (%) | **68.75** | 68.88 |
| NPU mAP (%) | **49.94** | 47.24 |

* + - * 1. v8s\_P2\_3221 vs v8s\_P2\_3321  
           텍스트, 스크린샷, 도표, 번호이(가) 표시된 사진

           자동 생성된 설명

|  |  |  |
| --- | --- | --- |
|  | v8s\_P2\_3221 | **v8s\_P2\_3321** |
| GPU mAP (%) | 68.88 | **69.55** |
| NPU mAP (%) | 47.24 | **49.04** |

* + - 1. 분석
         1. v8s\_P2 vs (v8s\_P2\_1321, v8s\_P2\_2221)  
            : [실험항목 A]에서 v8s\_P2 모델을 기준으로 v8s\_P2\_1321 모델의 성능을 비교한 결과, GPU mAP는 0.2 소폭 증가하였지만 NPU mAP는 0.85 감소했다. 반면, v8s\_P2\_2221 모델의 GPU mAP는 0.92 상승하였고 NPU mAP 또한 1.6 상승했다.
         2. v8s\_P2\_2221 vs v8s\_P2\_3221  
            : [실험항목 B]에서 v8s\_P2\_2221 모델을 기준으로 v8s\_P2\_3221 모델의 성능을 비교한 결과, GPU mAP는 0.13 소폭 증가하였지만 NPU mAP는 -2.7 감소했다.
         3. v8s\_P2\_3221 vs v8s\_P2\_3321  
            : [실험항목 C]에서 v8s\_P2\_3221 모델을 기준으로 v8s\_P2\_3321 모델의 성능을 비교한 결과, GPU mAP는 0.67 증가하였고 NPU mAP 또한 1.8 증가했다.  
              
            [실험항목 A]와 [실험항목 B]를 통해 단순히 Bottleneck의 반복 횟수를 늘림으로 인해 NPU 성능이 증가할 것이라는 가설의 오류 가능성을 확인했다. 다만, 모듈 간의 Bottleneck 반복 횟수의 Balance를 적절하게 맞춰주어 조정했을 때 NPU 성능이 증가함을 확인했다.  
            따라서, P2~P5 단의 반복 횟수 간 편차를 크지 않게 Balance를 적절히 맞춰야 하는 인사이트를 도출했다.  
              
            [실험항목 C]에서는 두 모델 모두 Bottleneck 반복 횟수의 편차가 고르지 않지만 v8s\_P2\_3321 모델이 P2, P3 단의 반복 횟수를 균일하게 맞춤으로써 v8s\_P2\_3221 모델보다 편차가 고르기 때문에 NPU 성능이 좋게 나왔을 것이다.

1. **결론**
   1. **주요 결론**
      1. 실험 1 P2를 추가한 후 Backbone에서의 Bottleneck 반복 횟수 조정에 따른 성능 변화 분석  
         : 소형 객체 탐지에 유리한 P3 단의 Bottleneck 반복 횟수만 늘릴 경우 오히려 NPU 성능이 감소했다. P2~P5 단의 Bottleneck 반복 횟수를 균형 있게 조정하였을 때 NPU 성능이 가장 높았다.  
         따라서, [분석 보고서 1.3] 결론에 언급한 바와 같이 사용할 모델의 종류와 Scale에 따라 Bottleneck 반복 횟수를 균형 있게 조합해야 높은 성능을 얻을 수 있다.
   2. **향후 연구 방향**
      1. 향후 실험  
         추가 실험을 통해 v8s 모델에서 Bottleneck 반복 횟수를 균형 있게 맞추어 최적의 모델을 탐색한다.  
         또한, 균형 있는 Bottleneck 반복 횟수를 가지더라도 오히려 과도하게 늘릴 경우 발생할 수 있는 성능 저하에 대한 연구 방향을 설정한다.