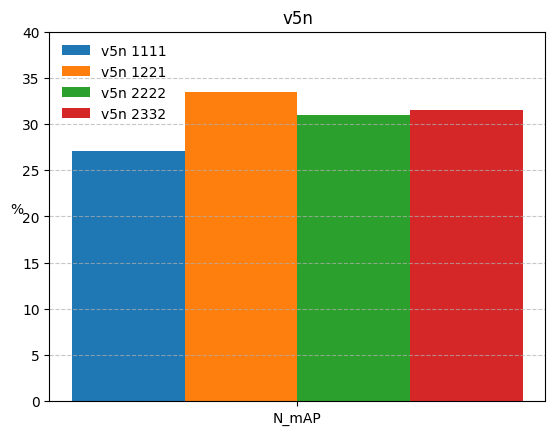
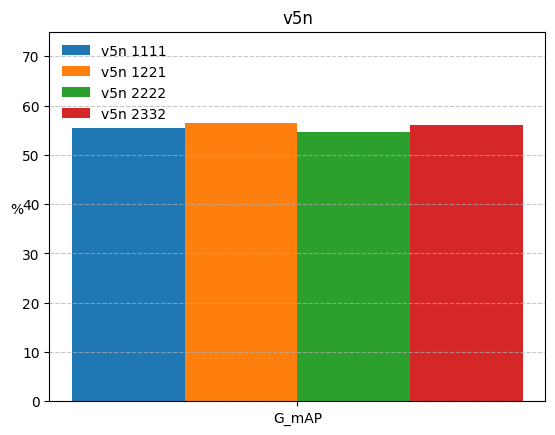
**Backbone 구조 변경 분석 보고서**

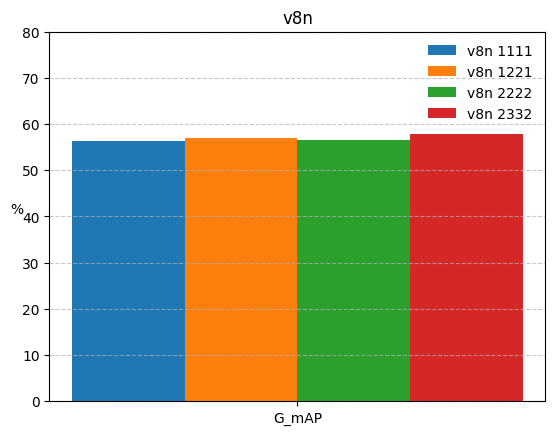
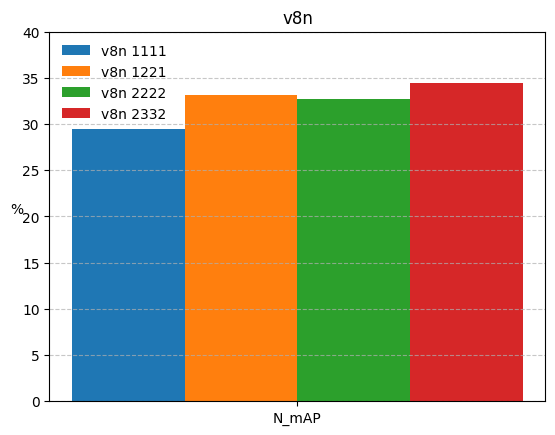
1. **서론**
   1. **배경**
      1. Backbone의 역할 및 yolo 구조 분석  
         : YOLO 모델은 크게 Backbone, Neck, Head 세 부분으로 이루어져 있다. Backbone은 이미지에서 특징 추출을 담당하는 부분이다. 이때 Backbone에서 v8 은 C2f 모듈 v5는 C3 모듈을 사용하는데, 두 모듈은 공통적으로 내부에서 Bottleneck 모듈을 사용하고 이는 Feature에 대해 더 많은 정보를 추출하는 역할을 한다. 따라서 본 분석 보고서는 Backbone에서의 Bottleneck 모듈 반복 횟수에 따른 성능 변화를 분석한다.
   2. **실험 준비**
      1. 실험 모델
         1. v5n\_1111
         2. v5n\_1221
         3. v5n\_2222
         4. v5n\_2332
         5. v8n\_1111
         6. v8n\_1221
         7. v8n\_2222
         8. v8n\_2332
   3. **목적**
      1. 실험 1 : v5모델 Backbone의 Bottleneck 반복 횟수에 따른 성능 분석  
         : [모델1] ~ [모델4]를 비교하여 v5n 모델에서 Backbone의 Bottleneck 반복 횟수를 조정했을 때 성능의 변화를 관찰한다. 이 때 GPU에서의 성능, NPU에서의 성능 두가지 관점으로 분석한다.
      2. 실험 2 : v8모델 Backbone의 Bottleneck 반복 횟수에 따른 성능 분석  
         : [모델5] ~ [모델8]를 비교하여 v8n 모델에 대해서도 마찬가지의 방법으로 비교한다. 이 때 GPU에서의 성능, NPU에서의 성능 두가지 관점으로 분석한다.
   4. **평가 지표**
      1. 모델 성능 : GPU mAP50, NPU mAP50
2. **본론**
   1. **실험 설계**
      1. 데이터셋
         1. 각 모델들이 학습에 사용된 데이터와 테스트에 사용된 데이터는 동일
            1. 학습 데이터 : 제공받은 train set과 자체적으로 수집한 data set을 병합
            2. 테스트 데이터 : 제공받은 test set
      2. 모델 설계

\* 1111 ~ 2332는 Backbone에서 C3/C2f 모듈 내의 Bottleneck 반복 횟수

* + - 1. v5n\_1111 : Backbone에서의 Bottleneck 반복 횟수를 (1, 1, 1, 1)로 설정한 모델
      2. v5n\_1221 : Backbone에서의 Bottleneck 반복 횟수를 (1, 2, 2, 1)로 설정한 모델
      3. v5n\_2222 : Backbone에서의 Bottleneck 반복 횟수를 (2, 2, 2, 2)로 설정한 모델
      4. v5n\_2332 : Backbone에서의 Bottleneck 반복 횟수를 (2, 3, 3, 2)로 설정한 모델
      5. v8n\_1111 : Backbone에서의 Bottleneck 반복 횟수를 (1, 1, 1, 1)로 설정한 모델
      6. v8n\_1221 : Backbone에서의 Bottleneck 반복 횟수를 (1, 2, 2, 1)로 설정한 모델
      7. v8n\_2222 : Backbone에서의 Bottleneck 반복 횟수를 (2, 2, 2, 2)로 설정한 모델
      8. v8n\_2332 : Backbone에서의 Bottleneck 반복 횟수를 (2, 3, 3, 2)로 설정한 모델
  1. **실험 결과 및 분석**
     1. 실험 1 v5모델 Backbone의 Bottleneck 반복 횟수에 따른 성능
        1. 실험 항목
           1. 대조군1 : 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. **가설 - Bottleneck 반복 횟수가 증가할수록 성능이 증가할 것이다.**: Residual 형식인 Bottleneck 구조는 Gradient Vanishing 문제를 방지하고, 학습을 안정시키는 역할을 한다. 이러한 Bottleneck의 반복 횟수를 증가시키면 이미지에서의 더 깊은 특징을 학습하게 되므로 모델의 성능이 증가할 것이다.
        3. 결과  
           - 평가 기준 : 모델 성능

[그래프 1] [그래프 2]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | v5n\_1111 | **v5n\_1221** | v5n\_2222 | v5n\_2332 |
| GPU mAP (%) | 55.40 | **56.46** | 54.56 | 56.14 |
| NPU mAP (%) | 27.04 | **33.43** | 31.00 | 31.56 |

* + - 1. 분석  
         : 위 그래프는 Bottleneck 반복 횟수 차이에 따른 모델 별 GPU mAP와 NPU mAP 값을 나타낸다.  
          실험 결과, 가설과는 다르게 GPU mAP와 NPU mAP 모두 v5n\_1221 모델이 가장 높은 성능을 보였다. 이를 통해 Backbone의 Bottleneck 반복 횟수를 늘리면 더 깊은 특징을 추출할 수는 있지만, v5n 모델에서 성능 개선의 한계점이 존재하기 때문에 적정선 이상으로 Bottleneck 모듈을 반복할 경우 성능이오히려 하락한다는 것을 확인했다.
    1. 실험 2 v8모델 Backbone의 Bottleneck 반복 횟수에 따른 성능
       1. 실험 항목
          1. 대조군1 : v8n\_1111  
             실험군1 : v8n\_1221, 실험군2 : v8n\_2222, 실험군2 : v8n\_2332  
             : 실험 1과 마찬가지로 v8n 모델에 대해 변인통제 하였다.
       2. **가설 - Bottleneck 반복 횟수가 증가할수록 성능이 증가할 것이다.**: [실험 1]과 동일
       3. 결과  
          - 평가 기준 : 모델 성능  
           [그래프 1] [그래프 2]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | v8n\_1111 | v8n\_1221 | v8n\_2222 | **v8n\_2332** |
| GPU mAP (%) | 56.40 | 56.88 | 56.58 | **57.76** |
| NPU mAP (%) | 29.42 | 33.20 | 32.66 | **34.42** |

* + - 1. 분석  
         : v5n 모델 실험 결과를 통해 도출한 “Bottleneck 모듈을 더 반복한다고 성능이 계속적으로 오르지 않을 것”이라는 분석과 다르게 v8n 모델은 Bottleneck 모듈의 가장 많은 반복 수를 가진 모델에서 좋은 성능을 보였다.  
          이에 대해 실험 2의 경우는 실험 1과는 다르게 Bottleneck의 반복 횟수를 늘렸을 때의 성능 개선 한계점에 도달하지 않았기 때문에 가장 많은 반복 수를 가진 v8n\_2332 모델의 성능이 가장 높다고 판단했다.

1. **결론**
   1. **주요 결론**
      1. 실험 1 / 2 **:** v5n, v8n모델 Backbone의 Bottleneck 반복 횟수에 따른 성능   
         : 사용할 모델의 종류와 Scale에 따라 Bottleneck 반복 횟수 변화에 따른 성능 개선의 한계점이 존재할 것이다. 따라서 Bottleneck의 반복 횟수 변경을 통해 정확도와 실행 속도 간의 trade off를 조절할 때, 사용할 모델에 따라 가장 적절한 값을 찾아야 한다는 결론을 얻었다.
   2. **향후 연구 방향**
      1. 실험 계획
         1. **[1.2 P2 기법 적용 분석 보고서]**에서 v8s\_P2 모델을 기반으로 실험을 진행하기로 했는데, 이때 향후 실험될 모델들에 대해 본 보고서의 결론에 따라 Bottleneck 반복 횟수를 변경하며 최적값을 탐색할 것이다.
      2. 필요 과제
         1. 본 보고서는 v5n, v8n 모델에 대해서 P2, P3, P4, P5 Layer의 반복 횟수를 순차적으로 증가시키며 각각의 영향을 분석하지 않고 있다. 따라서 각 Layer를 기준으로 다른 Layer들은 변인통제하며 반복 횟수를 증가시켜 분석하는 실험이 필요하다.
2. **참조 문헌**

[1] H. Zhou, Y. Yu, K. Wang, and Y. Hu, "A YOLOv8-Based Approach for Real-Time Lithium-Ion Battery Electrode Defect Detection with High Accuracy," *Electronics*, vol. 13, no. 173, 2024. DOI: [10.3390/electronics13010173](https://doi.org/10.3390/electronics13010173).

[2] https://ropiens.tistory.com/44