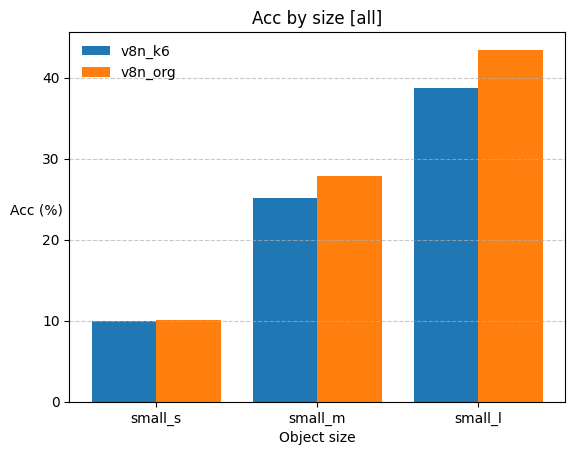
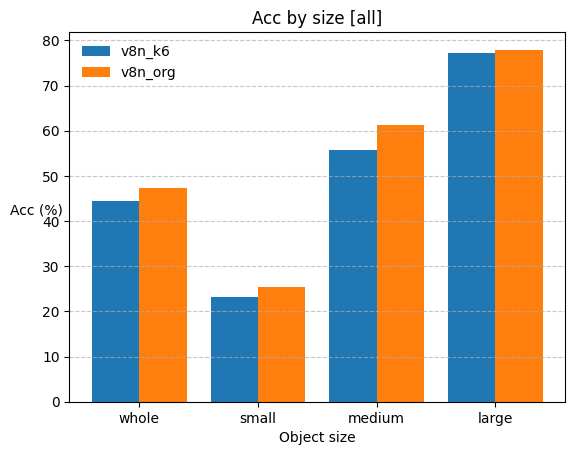
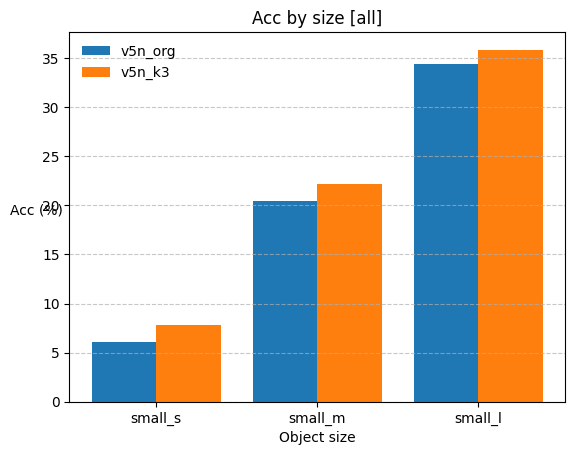
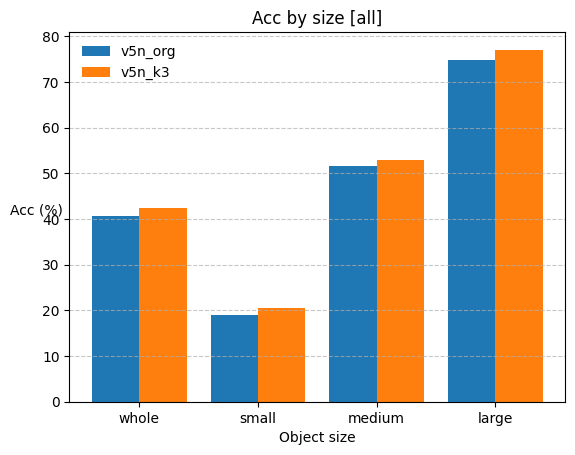
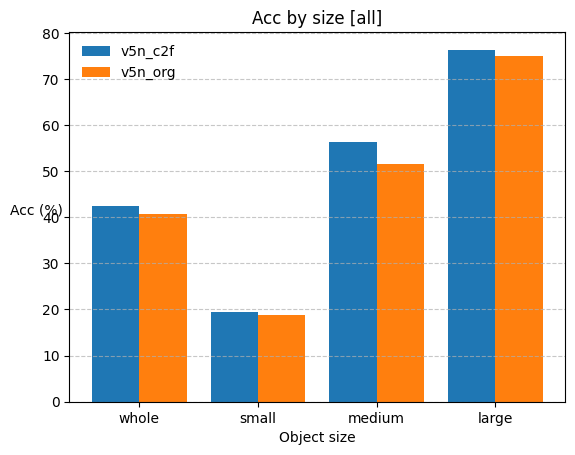
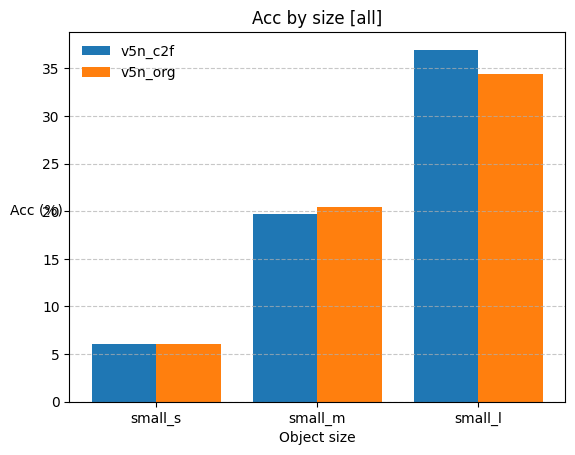
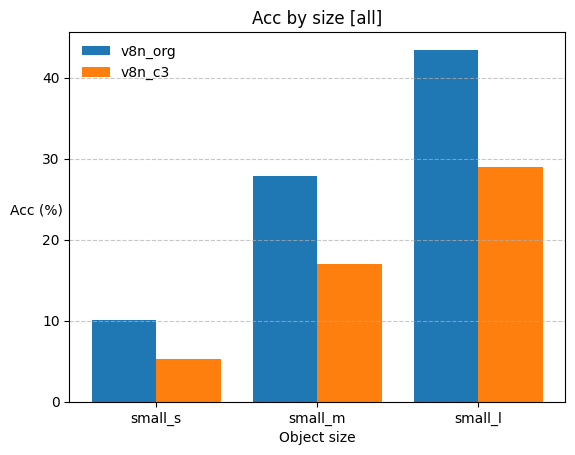
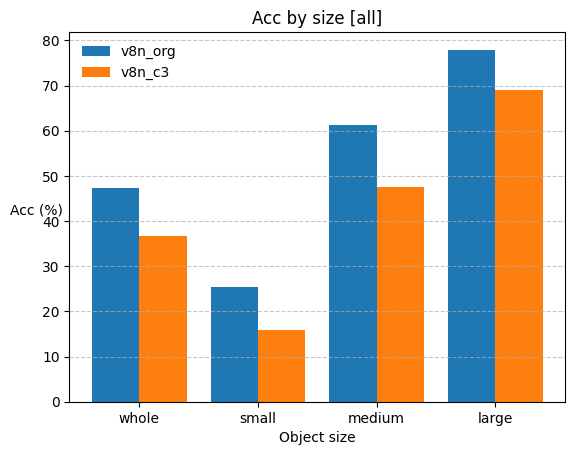
**v5n, v8n 분석 보고서**

1. **서론**
   1. **배경**
      1. 커널 사이즈와 소형 객체 검출 성능의 상관관계 분석  
         : 일반적으로 특징 추출을 위한 Convolution 연산에서 커널 사이즈는 객체의 크기와 밀접하게 연관되어 있다. 예를 들어 크기가 작은 커널(ex. 3x3)은 소형 객체(이미지에서 차지하는 영역이 작은 객체)의 특징들을 추출하는데 유리하며, 크기가 큰 커널(ex. 7x7)은 대형 객체의 특징을 추출하는데 유리하다.
      2. yolov5와 v8의 구조적 차이 및 성능 차이  
         : v5와 v8의 구조적 차이점은 두 가지로 요약할 수 있다. 첫째, 초기 Conv 레이어에서 v5는 kernel\_size=6을 사용하는 반면 v8은 kernel\_size=3을 사용한다. 둘째, v5는 C3 모듈을 사용하지만 v8은 C2f 모듈을 사용한다.  
          그런데 [1.0 데이터셋과 모델 성능 상관관계 분석 보고서]에서 v8n 모델이 v5n 모델과 비교했을 때 전체적인 성능이 높았으며, 소형 객체에 대한 성능 역시 높은 것을 확인할 수 있었다.  
          따라서 이러한 두 가지 차이점이 모델 성능에 미치는 영향을 분석하기 위해 본 실험을 진행한다.
   2. **실험 준비**
      1. 실험 모델
         1. v8n\_org\_tv
         2. v8n\_k6\_tv
         3. v8n\_c3\_tv
         4. v5n\_org\_tv
         5. v5n\_k3\_tv
         6. v5n\_c2f\_tv
   3. **목적 : v5n과 v8n의 성능 차이 원인 분석**
      1. 실험 1 : 커널 사이즈에 따른 성능 차이  
         : [모델1]을 기준으로 커널 사이즈를 6으로 증가시킨 [모델2]를 비교한 결과와 [모델4]를 기준으로 커널 사이즈를 3으로 낮춘 [모델5]를 비교한 결과를 종합해 커널 사이즈에 따른 성능 변화를 관찰한다. 이 때의 결과를 전체 정확도와 객체의 크기별 정확도 두가지 관점에서 분석한다.
      2. 실험 2 : C2f 모듈과 C3 모듈이 성능에 미치는 영향 분석  
         : C2f 모듈이 있는 [모델1]을 기준으로 C3 모듈로 변경한 [모델3]을 비교한 결과와 내부에 C3 모듈이 있는 [모델4]를 기준으로 C2f 모듈로 변경한 [모델6]을 비교한 결과를 통해 각 모듈이 성능에 미치는 영향을 관찰한다. 이 때 전체 정확도와 객체의 크기별 정확도 두가지 관점에서 분석한다.
   4. **평가 지표**
      1. 객체 크기별 정확도
2. **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

\* org는 기본 Yolo모델(original)  
\* k3, k6는 각각 첫번째 Conv의 커널 사이즈가 3, 6인 모델  
\* c3와 c2f는 내부 모듈이 변경된 경우

* + - 1. v8n\_org : v8n 기본 모델
      2. v8n\_k6 : v8n 기본 모델의 첫 Conv 레이어에 k = 6 적용
      3. v8n\_c3 : v8n 기본 모델의 C2f 모듈을 모두 C3로 교체
      4. v5n\_org : v5n 기본 모델 (anchor free)
      5. v5n\_k3 : v5n 기본 모델의 첫 Conv 레이어에 k = 3 적용
      6. v5n\_c2f : v5n 기본 모델의 C3 모듈을 모두 C2f로 교체
  1. **실험 결과 및 분석**
     1. 실험 1 Conv 첫 레이어의 kernel size에 따른 성능
        1. 실험 항목
           1. 대조군1 : v8n\_org, 실험군1 : v8n\_k6
           2. 대조군2 : v5n\_org, 실험군2 : v5n\_k3
        2. **가설 - 커널 사이즈가 작은 경우에 소형 객체에 정확도가 높을 것이다.**  
           : 배경에서 언급했듯이 작은 크기의 커널일수록 소형 객체의 특징을 추출하는데 유리하기 때문에, 동일한 네트워크 구조를 가진 모델끼리 커널 사이즈만 변경하여 정확도를 비교한다.
        3. 결과
           1. v8n\_org vs v8n\_k6
           2. v5n\_org vs v5n\_k3
        4. 분석  
           : 실험 A에서 커널 사이즈를 3으로 사용한 v8n\_org 모델이 small 객체와 medium 객체에 대한 검출 성능이 증가하는 것을 확인할 수 있었다. 특히 소형 객체를 크기별로 다시 세단계로 분류한 small\_s, small\_m, small\_l 객체들에 대해서 다시 조사해보면 small\_m, small\_l 객체에서 성능이 증가하는 것을 확인할 수 있었다. 실험 B에서는 커널 사이즈를 3으로 사용한 v5n\_k3 모델이 전체적인 성능이 좋은 것을 확인할 수 있었는데, 마찬가지로 small\_s, small\_m, small\_l 객체들에 대해 살펴봤을 때 모두 성능이 증가하는 것을 확인할 수 있었다.
     2. 실험 2 적용 모듈(C2f, C3)에 따른 성능
        1. 실험 항목
           1. 대조군1 : v8n\_org, 실험군1 : v8n\_c3
           2. 대조군2 : v5n\_org, 실험군2 : v5n\_c2f
        2. 결과
           1. v8n\_org vs v8n\_c3
           2. v5n\_org vs v5n\_c2f
        3. 분석  
           : 실험 A에서 c2f 모듈을 사용한 v5n\_c2f 모델의 전체 성능이 소폭 높았으며, small 객체에 대해 자세히 살펴봤을 때에는 small\_l 객체에서 약간의 성능 차이만 있었다.  
            실험 B의 경우 마찬가지로 c2f 모듈을 사용한 v8n\_org 모델이 성능이 좋았는데, 실험 A와 비교했을 때 성능의 차이가 훨씬 큰 것을 확인할 수 있었다. 또한 small 객체에 대해 자세히 확인했을 때에도 small\_s, small\_m, small\_l 세 경우 모두 큰 폭으로 성능이 차이나는 것을 확인했다.

1. **결론**
   1. 주요 결론
      1. 실험 1 : kernel size에 따른 모델 성능  
         : 실험 1을
      2. 실험 2 : 모듈 사용에 따른 모델 성능  
         : ㅁㄴㅇ
   2. 향후 연구 방향
      1. 향후 실험
      2. 필요 과제
2. **참조 문헌**

[1] J. Fernandez-Marques, P. N. Whatmough, A. Mundy, and M. Mattina, "Searching for Winograd-aware Quantized Networks," *Proc. of the 3rd MLSys Conf.*, arXiv:2002.10711, 2020.

[2] 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.

[3] <https://github.com/orgs/ultralytics/discussions/15762>

[4] https://pytorch.org/docs/stable/quantization-accuracy-debugging.html