서울대학교 중급 프로젝트

결과 보고서

팀명 : Sapphire Vision

팀원 : 정현택, 김유로, 잘리너바 아이가늠

목차

**Warboy Preparation**

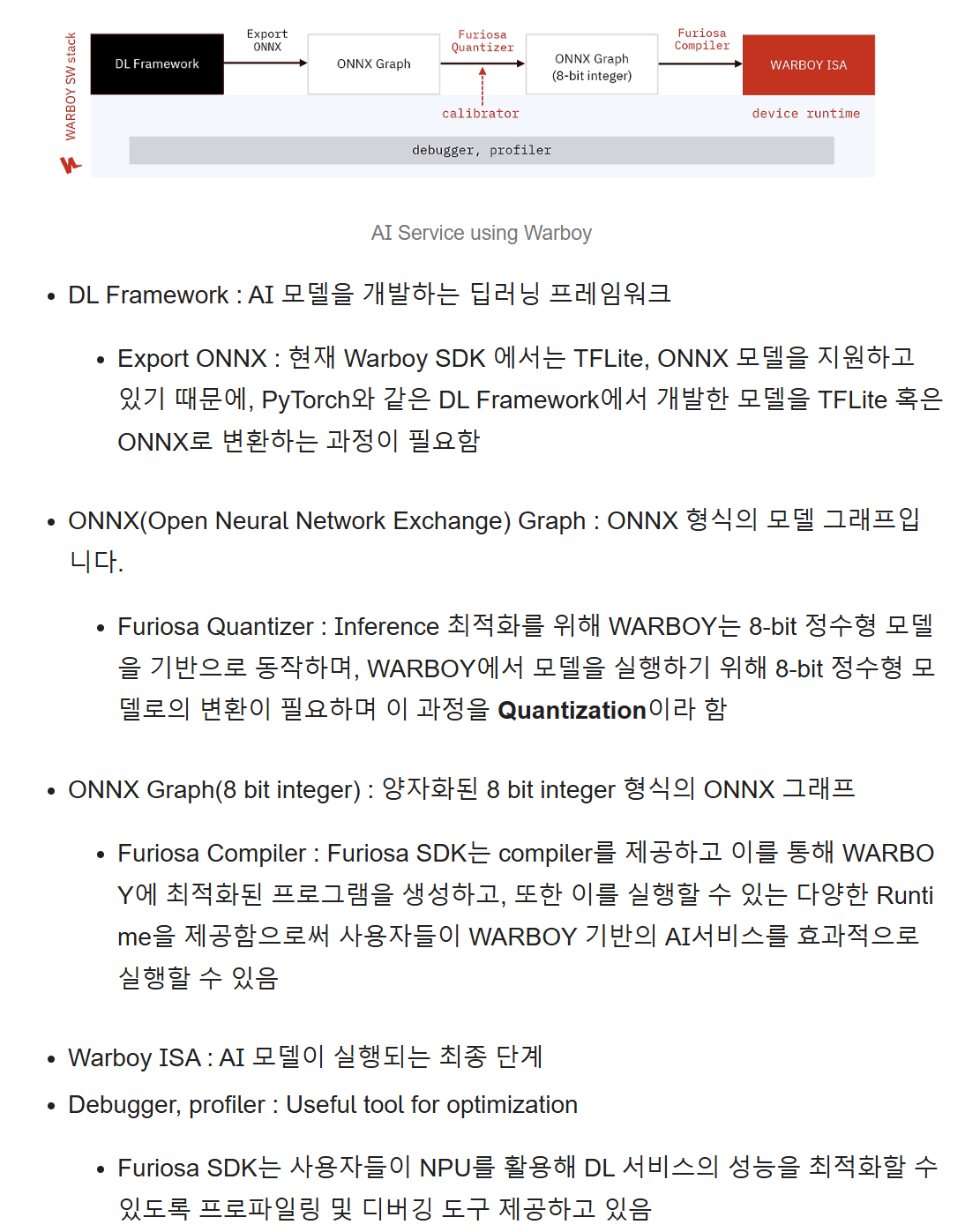
**Object detection**

1. **Model Quantization**
2. **Measuring model inference performance**
3. **Developing Vision AI application(Object Detection)**

**Instance Segmentation**

1. **Model Quantization**
2. **Measuring model inference performance**
3. **Developing Vision AI application(Object Detection)**

Warboy Preparation



아래의 명령어를 통해 기본 설정을 마무리

**Object detection**

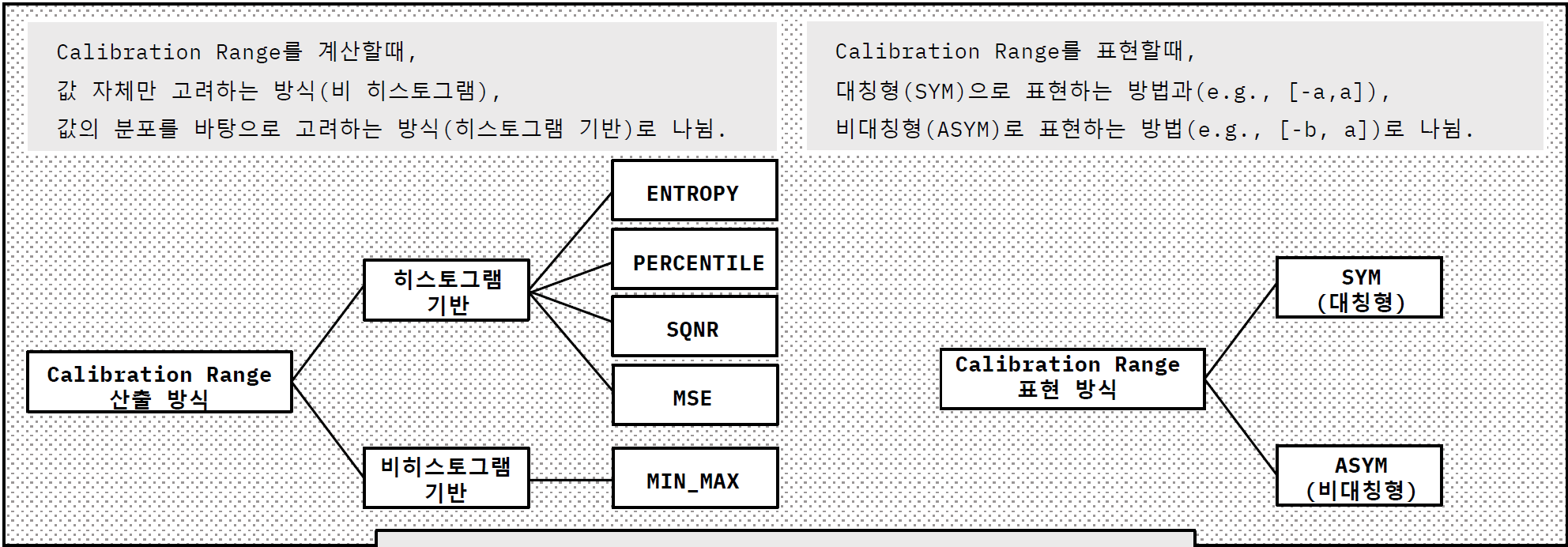
1. **Model Quantization**

2-1. Pytorch 모델을 onnx 타입으로 변환

위 명령어를 통해 yolo8n.onnx 파일이 만들어지며 이를 [Netron](https://netron.app/)을 통해 시각화한 결과 다음과 같은 결과를 얻을 수 있습니다.

2-2. onnx 파일을 양자화하기

저희는 Object Detection을 목표로 하기에 이에 적합한 calibration method를 찾아보았습니다.



총 10개의 calibration method에 대한 특성을 파악 후 특성 비교 후 적절한 calibration method를 선정 후, 각각의 calibration method 성능 평가를 통해 예측이 맞는지 확인해 볼 것입니다.

**(1) Entropy**

* **특성**:
  + **데이터의 분포를 고려**하여 양자화 범위를 설정합니다
  + 양자화 전후 분포 차이를 줄입니다
  + Outlier에 강하고, 데이터 분포를 잘 반영하지만 계산 비용이 높습니다

**적합성**:

* + 정확도 손실이 적고 다양한 데이터셋에 안정적

**(2) Percentile**

* **특성**:
  + 데이터의 특정 상위 백분위수(Percentile)를 기준으로 양자화 범위를 설정합니다.
  + 이상치(outlier)에 민감하지 않으며, 분포가 비대칭적인 데이터에 적합
* **적합성**:
  + Outlier가 많거나 데이터의 동적 범위(dynamic range)가 큰 경우 유용

**(3) MinMax**

* **특성**:
  + 데이터의 최소값과 최대값을 기준으로 양자화 범위를 설정합니다
  + 계산이 빠르지만 Outlier에 매우 민감하여 양자화 범위가 비효율적일 수 있음
* **적합성**:
  + 간단한 계산이 필요한 상황에서 유용하지만 정확도 손실 가능성 있음

**(4) MSE (Mean Squared Error)**

* **특성**:
  + 양자화 전후 값 간의 평균 제곱 오차를 최소화하도록 범위를 설정
  + 데이터의 분포를 잘 반영하며 정확도 유지에 유리
* **적합성**:
  + 정확도가 중요한 모델에 적합

**(5) SQNR (Signal-to-Quantization Noise Ratio)**

* **특성**:
  + 양자화로 인한 신호 대 잡음 비율(SQNR)을 최적화하여 스케일을 설정
  + 데이터의 손실을 최소화하면서 양자화 범위를 지정
  + **SQNR - Asym**: 제로포인트를 포함하여 비대칭 분포에 적합
  + **SQNR - Sym**: 대칭 범위를 사용하여 효율성을 강화
* **적합성**:
  + 정확도와 효율성을 모두 고려해야 할 때 유리

**ASYM(비대칭형)** : 제로포인트를 포함하여 비대칭 분포에 적합

**SYM(대칭형)** : 계산 효율성을 강화

위의 특성을 고려하여 저희는 Object Detection application을 만들기 위해 Entropy – ASYM을 사용하기로 하였습니다.

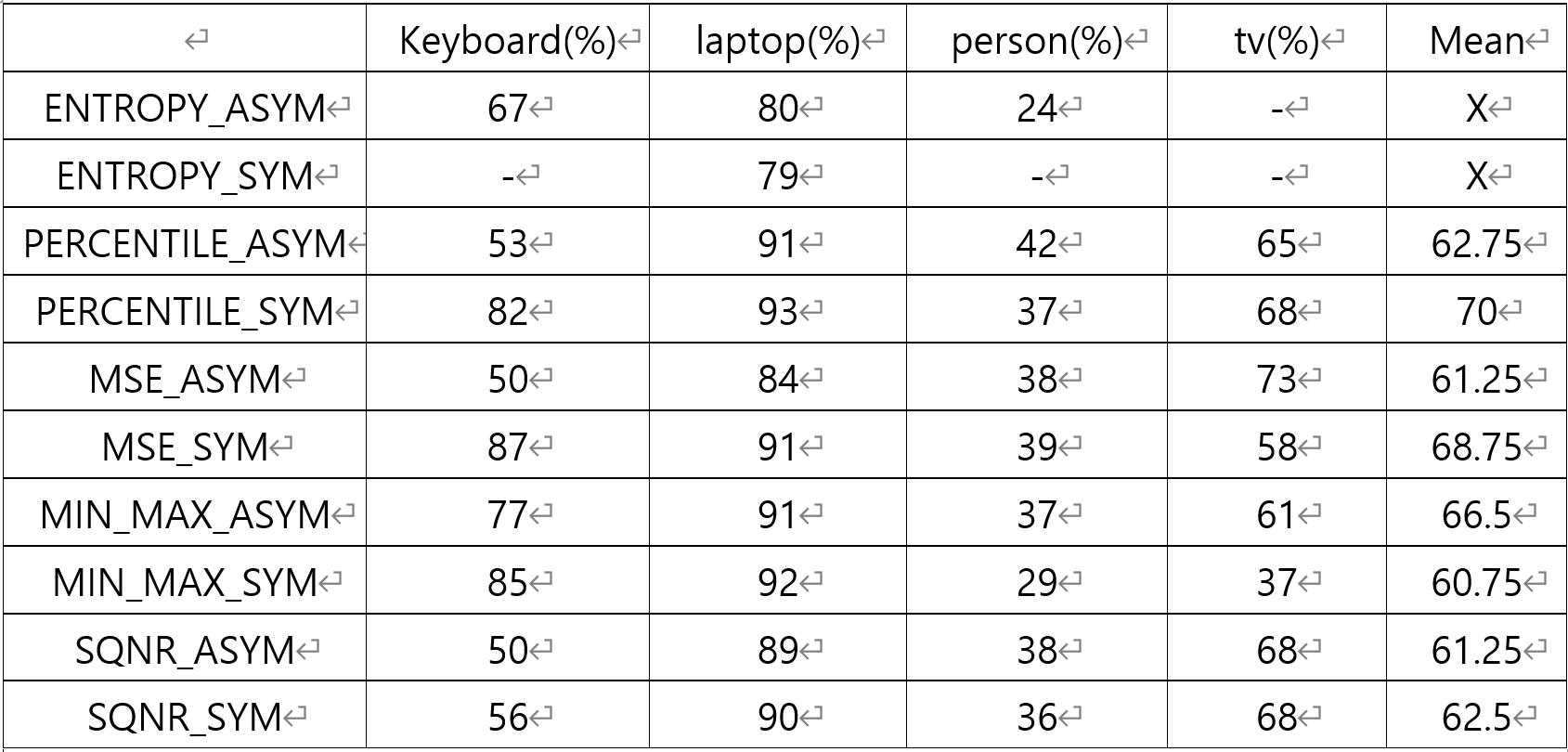
다만, 더 근거있는 calibration method를 정하기 위해 calibration method로 quantize를 진행한 후 정확도를 비교하였습니다.

Input에는 대략 4초의 input video를 넣어 진행하였습니다.

Inference를 진행할 때 30프레임으로 즉, 전체 114개의 bmp파일이 output으로 출력됩니다.

그 중 57번째 bmp 파일에서 모델이 laptop, keyboard, person, tv를 식별할 수 있어 그 bmp파일을 기준으로 정확도를 판단하였습니다.

아래의 표는 그 정확도를 판단한 것입니다.

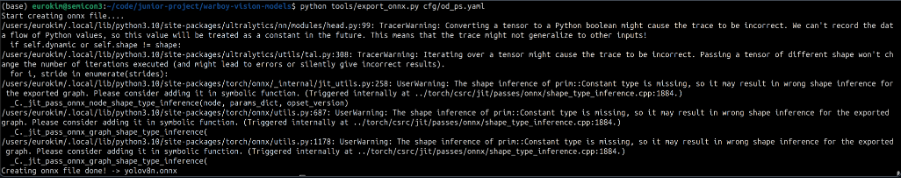


따라서 가장 정확도가 높은 PERCENTILE\_SYM을 사용하기로 했습니다.

1. **Measuring model inference performance**

이렇게 저희는 PERCENTILE\_SYM을 사용하여 onnx 추출, 양자화, 추론을 진행하고 성능 비교, latency와 throughput를 하였습니다.

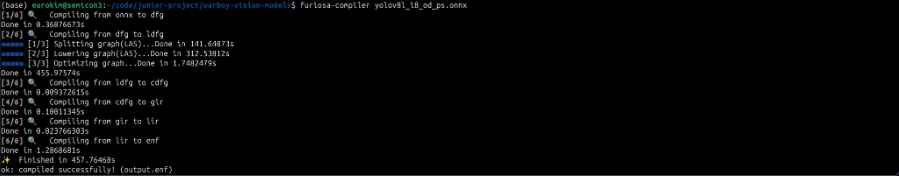
2-1)

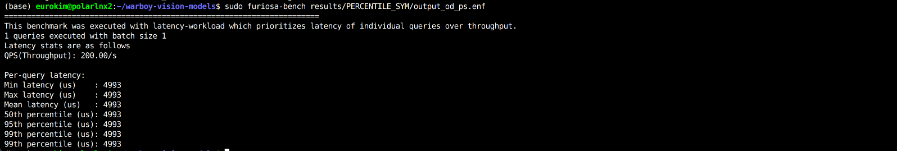


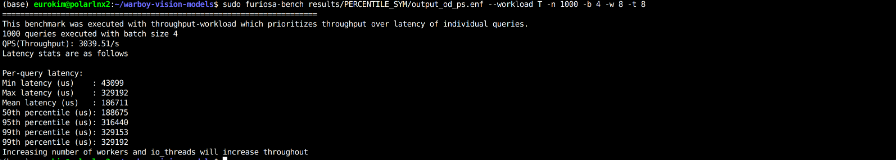
2-2)



3-1)



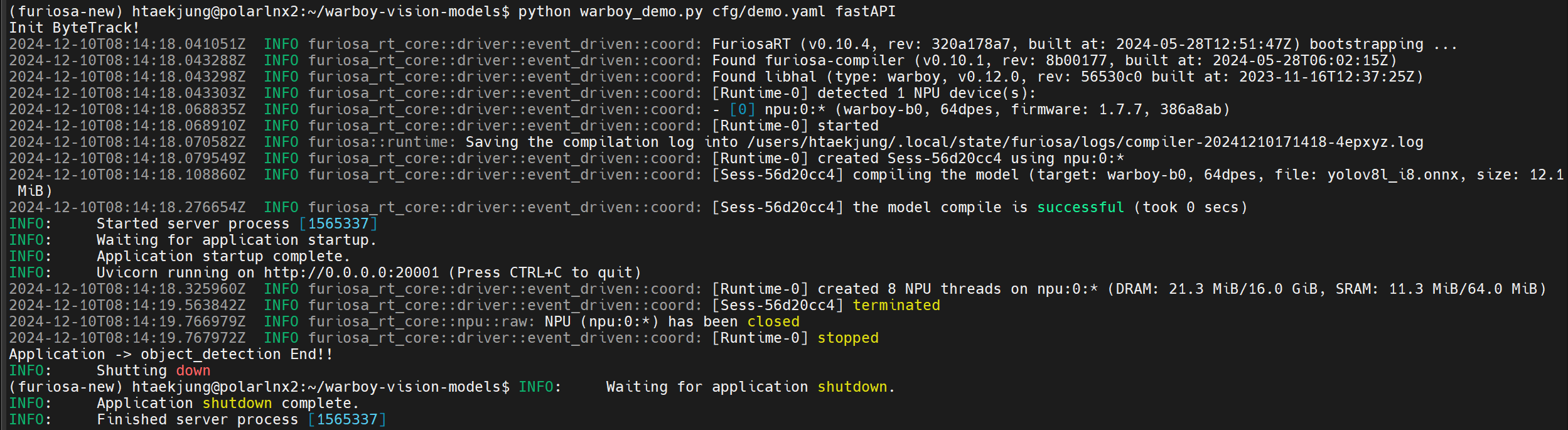
3-2) 

3-3) 

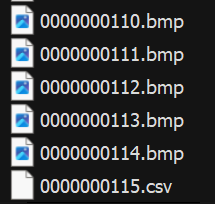
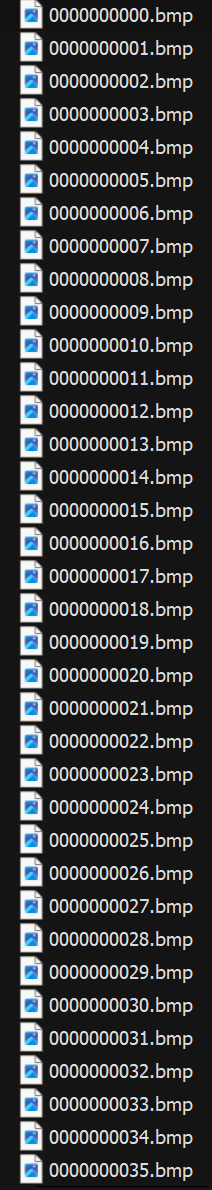
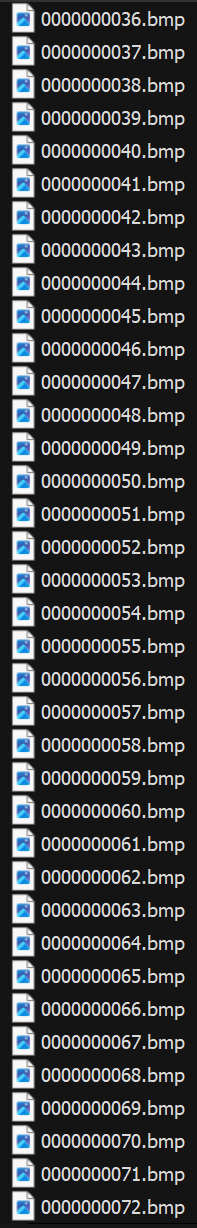
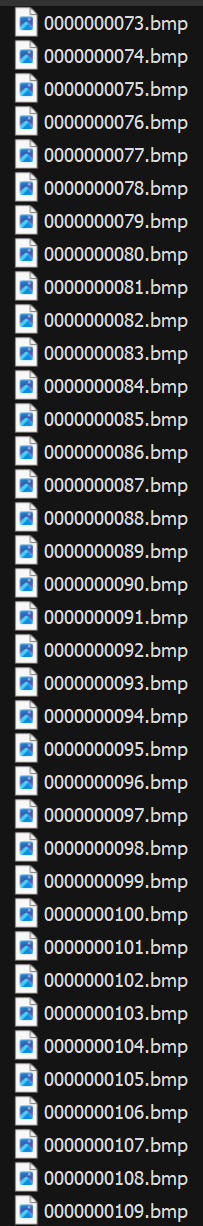
**Result**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | QPS(Throughput) | Min latency | Max latency | Mean latency |
| PERCENTILE\_ASYM | 3039.51 | 43099 | 329192 | 186711 |

1. **Developing Vision AI application(Object Detection)**



위 명령어를 통해 input video에 대해 object detection이 된 bmp파일을 출력하였습니다.





위 명령어를 통해 출력된 bmp파일을 하나의 동영상으로 만들어 input video에 대해 object detection을 진행한 output video를 출력하였습니다.



**Instance Segmentation**

1. **Model Quantization**

|  |  |
| --- | --- |
|  |  |
| ENTROPY\_ASYM |  |
| ENTROPY\_SYM |  |
| PERCENTILE\_ASYM |  |
| PERCENTILE\_SYM |  |
| MSE\_ASYM |  |
| MSE\_SYM |  |
| MIN\_MAX\_ASYM |  |
| MIN\_MAX\_SYM |  |
| SQNR\_ASYM |  |
| SQNR\_SYM |  |

따라서 저희는 calibration method로 MSE\_SYM 방식을 선택하였습니다.

1. **Measuring model inference performance**

이렇게 저희는 MSE\_SYM을 사용하여 onnx 추출, 양자화, 추론을 진행하고 성능 비교, latency와 throughput를 하였습니다.

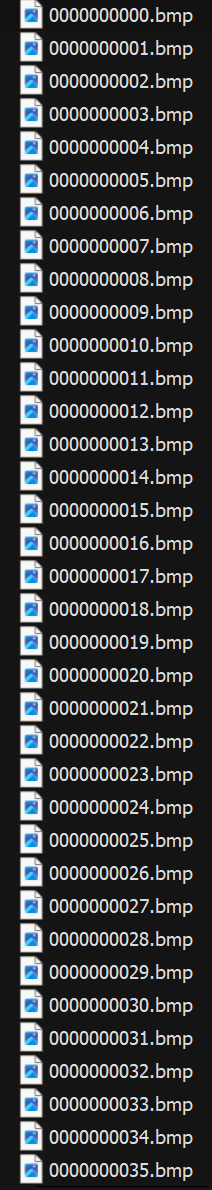
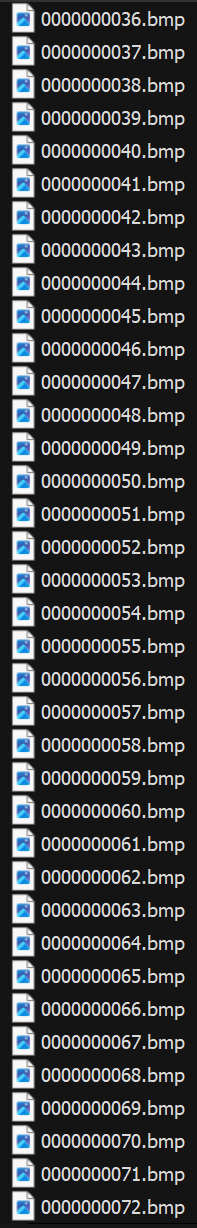
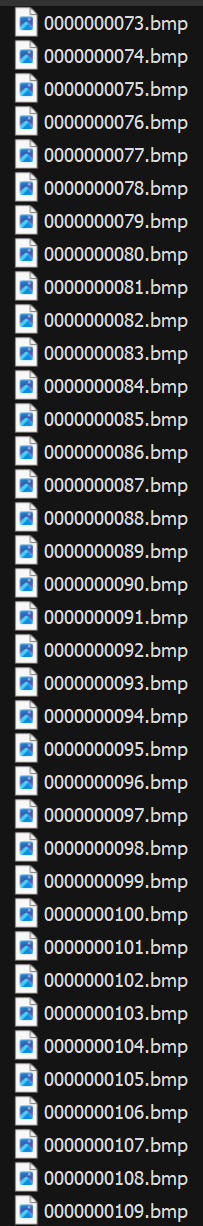
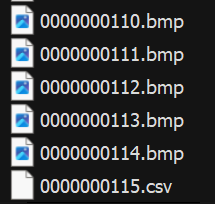
2-1)

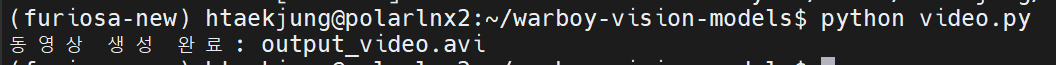
2-2)

3-1)

3-2)

3-3)

1. **Developing Vision AI application(Object Detection)**

****Bmp 파일로 변환 후 video.py를 실행시켜 output을 비디오로 만들었습니다.