|  |
| --- |
| 넥스트칩(컬러01) |
|  |
| **AI 기술인재 선발대회 결과 보고서** |
|  |
| Date: 2024.12.01 |
|  |

**Team : 자비스**

내용

[1. Introduction 3](#_Toc183602586)

[1.1. Abstract 3](#_Toc183602587)

[1.2. Goal 3](#_Toc183602588)

[1.3. Constraints 3](#_Toc183602589)

[1.4. Abbreviation 3](#_Toc183602590)

[1.5. Reference 4](#_Toc183602591)

[2. Details 5](#_Toc183602592)

[2.1. Members 5](#_Toc183602593)

[2.2. Timline 5](#_Toc183602594)

[2.3. Environments 5](#_Toc183602595)

[3. Model 5](#_Toc183602596)

[3.1. Overview 6](#_Toc183602597)

[3.2. Data Analysis 6](#_Toc183602598)

[3.3. Architecture 6](#_Toc183602599)

[3.3.1. Backbone 6](#_Toc183602600)

[3.3.2. Neck 6](#_Toc183602601)

[3.3.3. Head 6](#_Toc183602602)

[3.4. Training 7](#_Toc183602603)

[3.5. Example(추가로 진행한 방법론이 있다면 작성) 7](#_Toc183602604)

[4. Experiments 7](#_Toc183602605)

[5. Conclusion 7](#_Toc183602606)

[6. Appendix 7](#_Toc183602607)

# Introduction

## Abstract

- 소형 객체 탐지 성능 향상 : 기본 v8s 모델에 P2를 적용  
- 모델의 백본 구조 변경을 통한 성능 향상 : 백본의 반복수를 변경  
- 경량화 : Neck에서 사용되는 모듈을 교체하여 성능 손실을 최소화 하면서 추론 속도를 향상

## Goal

- 모델 성능 향상  
: NPU에서의 모델 성능을 향상하되, 특히 기존 YOLO의 문제점인 소형 객체에 대한 성능을 향상  
  
- 보존율(GPU에서의 성능 대비 NPU 에서의 성능) 향상  
: 모델의 구조를 NPU 친화적이게 변경함과 동시에 다양한 양자화 기법을 적용하여 NPU에서의 성능을 최대한 GPU에서의 성능에 근접하도록 구현  
  
- 경량화  
: 백본과 넥의 구조를 변경하여 성능 손실을 최소화 하며 모델의 경량화를 진행

## Constraints

- 데이터셋의 한계  
: 제공받은 데이터셋과 모델의 테스트가 NPU에서 진행되는 특성상 일반적인 시점의 데이터를 추가하여 학습을 진행한 경우 GPU에서는 성능이 향상되지만 NPU에서의 성능은 오히려 감소하는 것을 확인, 따라서 기존에 제공받은 데이터셋만 활용하여 모델의 학습을 진행  
  
- Detect단 구조 고정의 한계  
: P2를 추가한 이후 Detect단에 입력되는 채널의 사이즈를 변경하여 모델의 성능 향상을 확인하려 했으나, 기술상의 문제로 시도할 수 없었음

## Reference

|  |  |  |
| --- | --- | --- |
| **문서번호(저자)** | **제목** | **출처** |
| |  | | --- | | Li et al. |  |  | | --- | |  | | |  | | --- | | A YOLOv8-Based Approach for Real-Time Lithium-Ion Battery Electrode Defect Detection with High Accuracy |  |  | | --- | |  | | https://github.com/ ultralytics/yolov5 |
| J. Fernandez-Marques et al. | Searching for Winograd-aware Quantized Networks | https://arxiv.org/abs/2002.10711 |
| B. Jacob et al. | Quantization and training of neural networks for efficient integer-arithmetic-only inference | https://arxiv.org/abs/1712.05877 |
| PyTorch Team | Quantization Accuracy Debugging Documentation | https://pytorch.org/docs/stable/quantization-accuracy-debugging.html |
| H. Zhou et al. | A YOLOv8-Based Approach for Real-Time Lithium-Ion Battery Electrode Defect Detection with High Accuracy | https://doi.org/10.3390/electronics13010173 |
| Ropiens Blog | YOLO and its Applications in Image Processing | https://ropiens.tistory.com/44 |

[표 ] 참조 문서

# Details

## Members

김인호 (팀장)  
: 일정 관리, 팀원 업무 수행 방향 지시, 네트워크 설계, 실험 계획 및 관리, 데이터 관리, 보고서 검수, 업무 자동화, 결과 분석  
  
김기현 (팀원)  
 : 모델 양자화  
  
김세찬 (부팀장)  
: 네트워크 설계, 기존 기술 리서치 및 분석, 실험 설계 및 세팅, 결과 분석  
  
이찬혁 (팀원)  
: 모델 변환 방법론 분석, 모델 양자화, 네트워크 분석 및 설계

## Timline

YOLO 관련 리서치 : 2024.09.12 ~ 2024.09.26

NPU 디바이스 활성화 : 2024.09.26 ~ 2024.10.3  
YOLO v5, v8 비교 분석 : 2024.10.4 ~ 2024.10.14  
백본 P2 Layer 적용 실험 : 2024.10.15 ~ 2024.10.29  
백본 구조 분석 및 재설계 : 2024.10.30 ~ 2024.11.3  
모델 경량화 실험 : 2024.11.3 ~ 2024.11.9  
모델의 보존율 향상 실험 : 2024.11.10 ~ 2024.11.26

양자화 방법론 연구 : 2024.11.10 ~ 2024.11.26  
진행 내용 문서화 : 2024.11.27 ~ 2024.12.1

## Environments

Platform: Colab

Data: 제공받은 nextchip dataset 사용

Table Task's Classes

|  |  |
| --- | --- |
|  | Detect |
| Classes Number | 6 |
| Classes Details | person  car  bus  truck  cycle  motorcycle |
| Scene | Rearview |

# Model

## Overview

# 전반적으로 수행한 방법론에 대한 전반적인 내용 작성  
- 소형 객체 탐지 성능을 위한 구조 변경 : Neck의 구조를 변경하여 P2 Layer를 적용  
  
- Backbone 구조 변경 : P2 Layer가 적용된 모델을 기반으로 Backbone의 C2f 모듈의 반복수를 조절해가며 성능을 향상  
  
- 경량화 : Neck에 사용되는 모듈을 GhostConv와 C3Ghost 모듈로 교체하여 실행 속도의 손실을 최소화 하면서 모델의 성능을 향상시킴  
  
- PTQ : Aiware Stuio를 활용하여 모델이 양자화 될 때의 오차율을 줄이기 위한 방법론을 모색, nnef와 stats에 접근하여 가중치 값의 분포에 따라 적절히 rescaling 시도, 이후 Aiware Studio에서 제공하는 method들을 v5 anchor-free, anchor-based 두 경우에 모두 시도

## Data Analysis

**[1.0 데이터셋과 모델 성능 상관관계 분석 보고서]**에 서술

## Architecture

v5n과 v8n을 비교 분석한 후 yolov8s를 Base Model로 설정하여 경량화와 성능 향상을 동시에 진행  
  
**[1.1 v5n, v8n 분석 보고서]** 에 서술 및 이후 보고서들 참조

.

### Backbone

제출물 **[1.3 Backbone 구조 변경 분석 보고서]  
[1.4 P2 적용 및 Backbone 구조 변경 분석 보고서]**

**[2.1.1 Bottleneck 반복수와 보존율 분석 보고서]  
[2.1.2 Concat 텐서 그룹수와 보존율 분석 보고서]**내용 참조

### Neck

제출물

**[1.2 P2 기법 적용 분석 보고서]**

**[1.4 P2 적용 및 Backbone 구조 변경 분석 보고서]  
[2.2.1 GhostConv, Conv 분석 보고서]  
[2.2.2 C2f, C3, C3Ghost 분석 보고서]**  
내용 참조

## Training

**[JARVIS MODEL 산출 보고서]** 내용 서론부에 서술

# Experiments

제출물 - **documents** 폴더 내에 순차적으로 정리

# Conclusion

정량적 결과는 제출물 **[JARVIS MODEL 산출 보고서]** 및 **[exp\_list.csv]** 참조  
  
본 실험에서는 YOLO 모델을 기반으로 소형 객체 탐지 성능을 높이고, GPU 대비 NPU 성능 재현율을 개선하기 위해 P2 레이어 적용, Bottleneck 반복수 증가, 및 Neck 경량화 전략을 적용하였다.

1. **인과적 추론**:

- P2 레이어 및 Bottleneck 반복수 증가로 인해 소형 객체의 세밀한 특성 학습이 가능해졌고, 이는 탐지 정확도의 주요 향상 요인으로 작용했다.

- Neck 경량화를 통해 실행 속도를 향상하고, NPU에서의 성능 효율성을 최대화하였다. 특히 GhostConv와 C3Ghost를 함께 사용한 구조는 기존 대비 계산량을 줄이면서도 모델의 정확도를 유지하는 데 효과적임을 확인했다.

- 전반적인 성능 개선은 Backbone에서의 적절한 특징 학습과 Neck에서의 효율적인 최적화를 통해 이루어 냈다.

1. **종합적 통찰:**

- Bottleneck 반복 수를 증가시켜 성능 향상을 시켰으나 그에 따른 효과로 실행 속도가 감소했다. 이후에 Neck 경량화를 통해 이를 상쇄시켰다. 따라서 정확도를 향상하면서 실행 속도의 감소는 최소화 할 수 있었다.

- GhostConv와 C3Ghost를 함께 사용하는 경량화 기법은 정확도의 감소를 최소화 하면서 모델의 실행 속도를 높이게 되었는데, 특히 NPU 환경에서 실행했을 때 그 효과가 더 크게 나타났다. 따라서 엣지 디바이스에서의 적합하도록 모델의 구조를 변경하는데 성공했다고 생각한다.