Yolov8 - Efficient Object Detection for LiDAR Data

Figure 1:Exampel picture from the projects given NAPLab-LiDAR dataset

Emil Skogheim
Computer engineering
TDT4265 – Computer vision and deep learning
NTNU – Trondheim
emilsko@stud.ntnu.no

Abstract

This study presents the development and evaluation of YOLOv8, an advanced object detection model, tailored for LiDAR data and aimed at enhancing autonomous vehicle technology. The project focuses on optimizing real-time processing and accurate object detection within the complex urban environment of the Gløshaugen campus area, Trondheim. By leveraging a projection-based approach, where 3D LiDAR data is transformed into 2D grayscale images, the model effectively employs the YOLOv8 architecture's speed and accuracy capabilities. The model's backbone, featuring a Darknet variant, and an innovative detection head, eliminate the need for predefined anchor boxes, allowing direct prediction of object dimensions and classifications. Through extensive training on the NAPLab-LiDAR dataset and strategic hyperparameter tuning, the YOLOv8 model achieves a balance between detection precision and computational efficiency. Results indicate a promising mean Average Precision (mAP) of 0.312 at varying IoU thresholds, with particularly strong performance in detecting larger objects like cars and trucks, while highlighting areas for potential improvement in detecting smaller entities such as bicycles. The model's inference speed of 0.225 milliseconds per image exemplifies its capability for real-time application in autonomous driving, setting a benchmark for future enhancements in LiDAR-based object detection technology.

Content

Introduction

A. Problem statement

The project focuses on tackling the challenge of object detection using LiDAR data in the context of autonomous vehicles. Specifically, the task is to develop and implement an efficient object detection model capable of identifying and locating eight distinct object classes within LiDAR data collected from the Gløshaugen campus area in Trondheim.

B. Motivation and objectives

Accurate and efficient object detection using LiDAR is crucial for autonomous vehicles to perceive and understand their surrounding environment in real-time. This capability is essential for enabling safe and reliable navigation, obstacle avoidance, and path planning. The primary objective of this project is to explore and implement a YOLOv8 – based object detection model tailored for LiDAR data, aiming to achieve high accuracy while maintaining real-time processing speeds.

Related work

Object detection in the realm of autonomous vehicles has witnessed substantial progress with the advent of deep learning. Architectures like Faster R-CNN, SSD, and YOLO have become popular choices due to their ability to effectively utilize convolutional neural networks (CNNs) for feature extraction and subsequent object localization and classification.
YOLO (You Only Look Once) stands out for its remarkable balance between speed and accuracy. YOLOv8, the newest version, incorporates architectural refinements and innovative training methodologies, making it a strong candidate for real-time object detection tasks like those encountered in autonomous driving scenarios (Keylabs, 2024).

While these models typically operate on RGB images, my project centers on the utilization of 2D LiDAR data represented as grayscale images. This approach offers a unique perspective as it retains depth information while leveraging the efficiency of 2D CNNs. Several architectures have been explored for LiDAR object detection, each with its strengths and limitations.

One architecture was the combination off a faster R-CNN approach with a restnet 50 backbone. A well know architecture in object detection, with high accuracy performance. But with some slower results because off it is a two stage detector with its regional proposal and classification network (Rabbi, 2020).

Our project aligns with projection-based models, where the 3D LiDAR data is projected onto a 2D plane, generating a bird's-eye view or front-view image. This allows the application of efficient 2D CNN-based object detectors like the one stage detector YOLOv8 (Gallagher, 2023) while preserving crucial depth information encoded in the grayscale values.

Given the characteristics of our dataset and the need for real-time performance, I investigated the application of YOLOv8 with a projection-based approach. This method is expected to provide an optimal balance between accuracy and efficiency (Gallagher, 2023), therefor suitable for our LiDAR-based object detection task.

Methodology

A. Architecture

The chosen object detection model for this project is YOLOv8s, renowned for its speed and accuracy (Gallagher, 2023). YOLOv8s's architecture comprises two key components: backbone and head.

The backbone network responsible for feature extraction is a variant of the Darknet architecture. It consists of a series of convolutional layers, interspersed with batch normalization and SiLU activation functions, to extract meaningful features from the input images.

YOLOv8s also introduces the C2f module, which effectively combines high-level features with contextual information. This is achieved by concatenating the outputs of bottleneck blocks, each containing two 3x3 convolutional layers with residual connections. This design enhances feature representation and contributes to improved detection accuracy.

The architecture employs PANet (Path Aggregation Network) for efficient feature fusion. PANet combines features from different scales of the backbone network through a series of up

sampling and down sampling operations, allowing the model to capture both local and global contextual information.

YOLOv8s utilizes an anchor-free detection head, eliminating the need for pre-defined anchor boxes. The head directly predicts the center coordinates, width, and height of objects, along with their class probabilities. Making it suitable for detection tasks where object sizes may wary signifyingly.

The head is decoupled into three independent branches: one for objectness prediction, one for classification, and one for bounding box regression. This design allows each branch to specialize in its task, leading to improved performance.

YOLOv8s employs a combination of loss functions to optimize the model. The CIoU (Complete Intersection over Union) loss is used for bounding box regression, effectively considering the overlap area, center point distance, and aspect ratio between predicted and ground truth boxes. Binary Cross-Entropy loss is employed for objectness prediction, and Cross-Entropy loss is used for classification.

B. Training Techniques

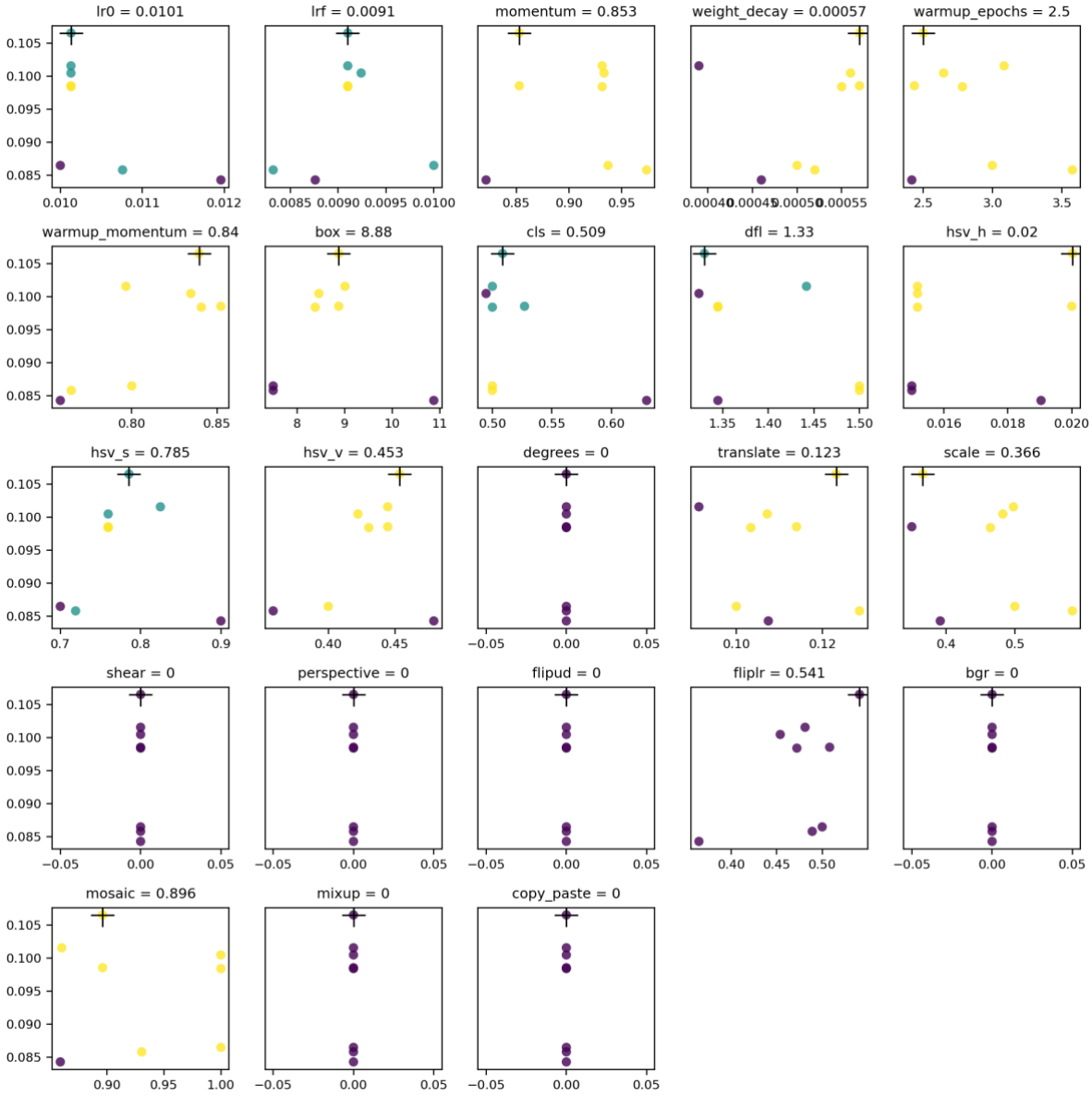

Figure 3: Results from the tuning process done by yolo.

I employed a hyperparameter tuning process using the yolos included tuning feature to optimize the model's performance. This involved exploring different configurations for epochs, batch size, and image size to identify the optimal settings for the training, thru 8 iterations with 3 epochs each the model served for the best parameters. As actional the tuning tries different argumentation techniques to see how the model preforms based on these. As example argumentation techniques like scaling and mosaic was found out preformed best on the dataset at values respectably at 0.366 and 0.896 each.

Considering the nature of object detection tasks, I utilized the predefined IoU function build into the yolo architecture. The function effectively penalizes both bounding box location and size inaccuracies, leading to improved detection performance.

AdamW optimizer was selected for its adaptive learning rate capabilities and effectiveness in handling sparse gradients. To mitigate overfitting, I incorporated regularization techniques like early stopping based on validation loss trends. If the training prosses did not improve within 15 epochs the training would been stopped.

C. Dataset

The project leverages the NAPLab-LiDAR dataset, which comprises LiDAR cloud data captured around the Gløshaugen campus in Trondheim. The data is pre-processed and provided as grayscale images in PNG format with corresponding annotations in YOLO v1.1 format, specifying bounding boxes for objects in the picture, and connected to one of the eight classes in the dataset. These classes are relevant to autonomous driving scenarios, such ass cars, trucks and persons. The dataset consists off a total 1904 pictures and corresponding labels.

D. Model Training

Training was conducted on the IDUN cluster using a single GPU. Adhered to the project guidelines. The total training time was 5 hours, encompassing both hyperparameter tuning and model training phases. A total of 100 epochs was run.

RESULTS AND DISCUSSION

Figure 3: Results from the trained model on one unseen training picture in the NAPLab-LiDAR dataset

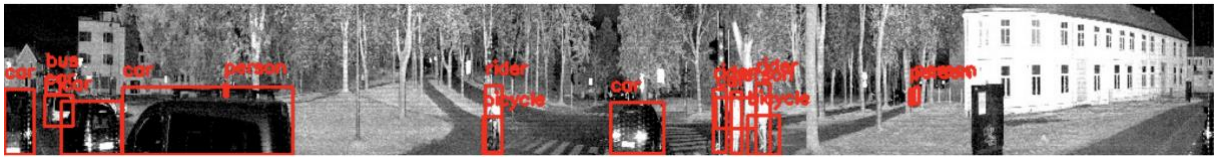

Figure 4: Shows the ground truth pictures on the same picture ass figure 2

A. Qualitative Analysis

Figure 3 showcases the results from the model on one unseen picture from the dataset. Showing a significant detection capability. The model detected 6 cars, 2 persons and one rider, with a 41.3 ms inference speed on a M1Pro cpu. Even though the model detects several main objects, it also showcases that its better on detecting cars at a high confidence level compered to persons. Compered to figure 4 and the ground truth boxes showcases that the model also missed smaller detections like bicycles and some persons.

B. Performance Metrics

The primary metric used to evaluate the model's performance is the COCO mean Average Precision (mAP) at IoU thresholds of 0.5 to 0.95 (mAP@0.5:0.95). This metric provides a comprehensive measure of the model's ability to accurately detect and localize objects across different sizes and levels of overlap.

C. Mean Average Precision

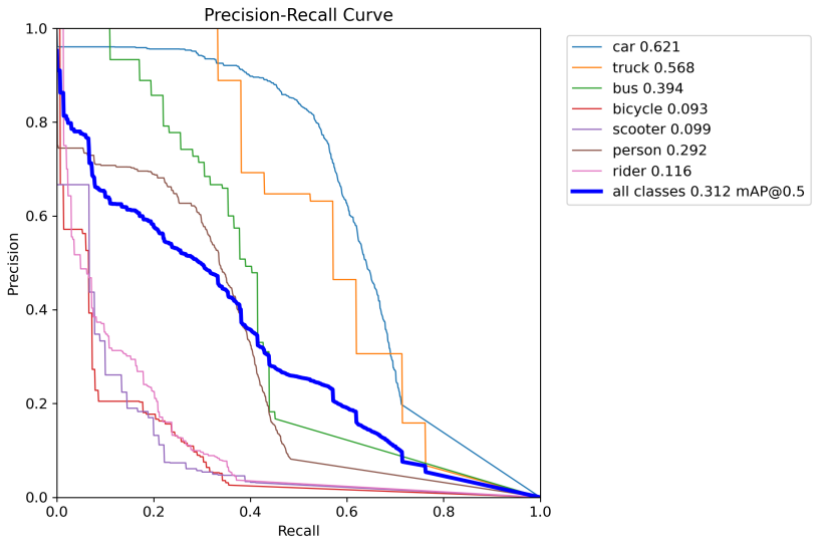

Figure 5: Shows the precision recall curves

The trained YOLOv8s model achieved a mAP@0.5:0.95 of 0.312 on the validation set. This indicates a moderate level of accuracy in detecting and localizing objects within the LiDAR data. The precision and recall curves show the tradeoff between precision and recall at different confidence thresholds. For instance, the model performs well on larger objects like cars and trucks, achieving high precision and recall. However, the performance is lower on smaller objects like bicycles and scooters, indication improvements areas.

D. Confusion matrix

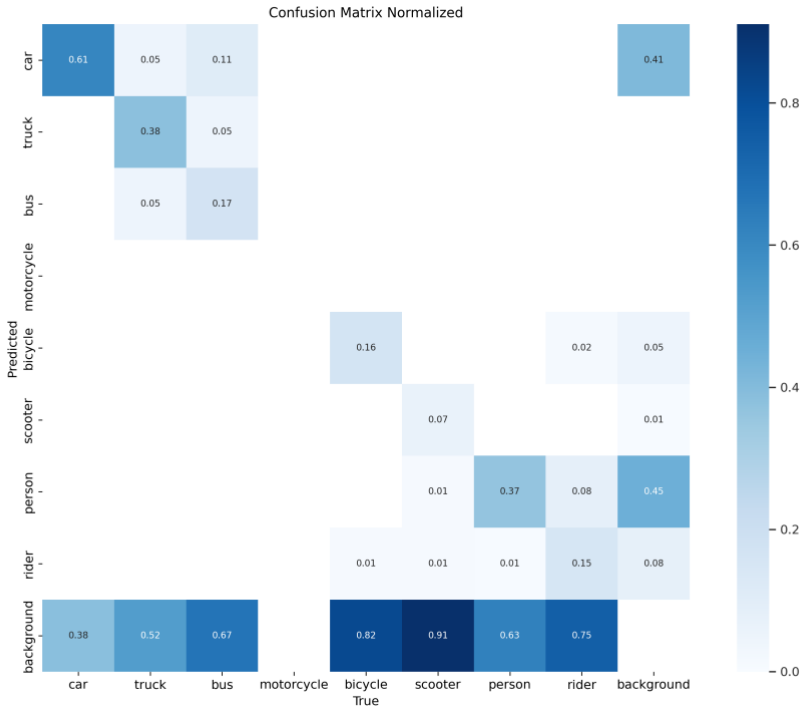

Figure 6: The confusion matrix normalized

Figure 6 visualizes the normalized confusion matrix. Showing good detections on objects like car, trucks, and persons. There seems to be some confusion, where the model struggles some with identifying the difference between a bus and a car, and rider and a person. Also here visualizing improvement potential.

E. Training loss

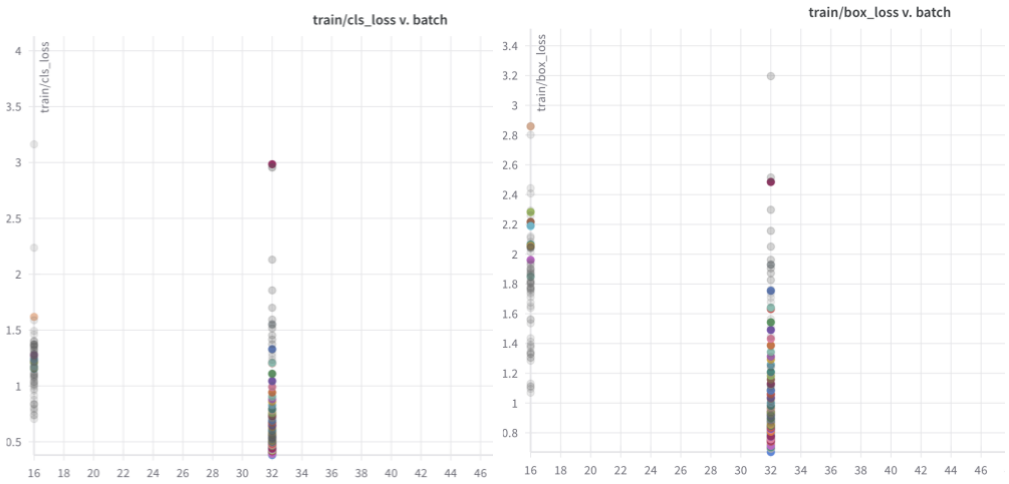

Figure 7: Shows the training loss on bounding boxes and classes thru training

The training loss on bounding boxes and classes was logged using weights and biases thru the epochs. Figure 7 shows the loss respectably getting better for each epoch and the best loss was at train box_loss: 0.6936 and train cls_loss: 0.39013.

F. Processing Time and model size

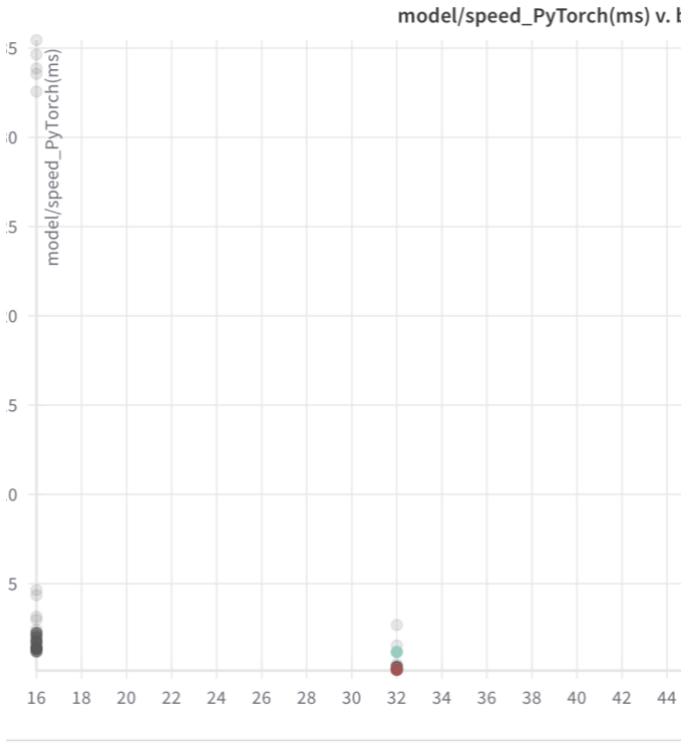

Figure 8: Shows the inference speed evaluation thru training

The model demonstrated an inference speed of 0.225 milliseconds per image and a size of 22,5 MB on the IDUN cluster with a single GPU. This processing time signifies the model's capability to perform real-time object detection, meeting the critical requirement for autonomous driving applications. Figure 8 showcases the inference speed getting better for each epoch.

G. Results

The YOLOv8s model trained on the NAPLab-LiDAR dataset demonstrated promising results for LiDAR-based object detection. The achieved mAP@0.5:0.95 indicates a reasonable level of accuracy, while the fast inference speed highlights its suitability for real-time applications such needed in autonomous vehicles.

Several avenues exist for potential improvement, for getting better mAP. Potential areas could be including more data augmentation. This could been done by extending the tuning process or exploring various data augmentation techniques self, such as random flipping, cropping, and rotation, could increase the dataset's diversity and improve the model's generalizability.

Antoher approach to better results is architecture modifications and utilizing a bigger version off the yolov8 models, such ass yolov8m or ass bigg ass yolov8x. Investigating deeper backbone networks or incorporating attention mechanisms could enhance feature extraction capabilities, potentially leading to better accuracy.

CONCLUSION

This project successfully explored the application of YOLOv8s for object detection using 2D LiDAR data in the context of autonomous driving. The developed model achieved a respectable mAP@0.5:0.95 of 0.312 and demonstrated real-time inference speed, highlighting its potential for practical deployment.

While the results are promising, several avenues for future improvement were identified, including data augmentation techniques and architecture modifications. Exploring these directions could lead to further enhancements in accuracy and robustness.

Overall, this project contributes to the advancement of LiDAR-based object detection for autonomous vehicles, showcasing the effectiveness of YOLOv8 as a viable solution for real-time perception tasks.

REFERENCES

Gallagher, J. (2023, December 6). How to detect objects with YOLOv8. Roboflow Blog. https://blog.roboflow.com/how-to-detect-objects-with-yolov8/

Keylabs. (2024, January 3). Mastering object detection with YOLOv8. Keylabs. https://keylabs.ai/blog/mastering-object-detection-with-yolov8/

Rabbi (2020, November). Tiny Object Detection in Remote Sensing Images: End-to-End Super-Resolution and Object Detection with Deep Learning. Researchgate. https://www.researchgate.net/figure/Faster-R-CNN-A-two-stage-detector-with-region-proposal-and-classification-network-60_fig4_346096343