A Comparative Analysis of KITTI and nuImages Datasets for Robust YOLOv8 Object Detection with Artificial Lens Flare

FRANCESCO OLIVIERI DAMIANO BASTIANELLI
fraoli@kth.se | damianob@kth.se

January 21, 2025

Abstract

This research investigates the performances of You Only Look Once (YOLO) framework, trained on two autonomous driving datasets, KITTI and nuImages, and tested under varied artificial light conditions, specifically lens flare. The study aims to identify the dataset better suited to provide a more robust foundation for object detection in scenarios under varied light conditions. More specifically, we evaluate and compare the performance of YOLOv8 models trained on subsets of KITTI and nuImages datasets. Then, we tested on a modified version of BDD100K, where artificial glare was introduced in the images to simulate challenging lighting conditions. Performance metrics, including Precision-Recall curve, F1 score, and mean Average Precision (mAP), were used to evaluate and compare the models' effectiveness. The results demonstrate that the YOLOv8 model trained on nuImages outperformed the KITTI-trained model in terms of robustness and generalization under challenging conditions, specifically artificial light disturbances. This is reflected in higher performance across various metrics, while the KITTI-trained model struggled to adapt to diverse scenarios, highlighting its vulnerability in noisy conditions. Our findings underscore the critical role of diverse training datasets in enhancing the robustness of object detection models, particularly in challenging lighting conditions. By leveraging the nuImages dataset, we demonstrate that YOLOv8 can achieve superior performance, leading to safer and more reliable object detection in autonomous driving systems.

Contents

1 Introduction

Thanks to the recent developments in Graphical Processing Unit (GPU) technologies and machine learning paradigms, AI and autonomous systems have rapidly developed. With an effervescent market and promising future expansions, this sector has spread into several different subfields: from medical applications to agriculture as well as finance and autonomous vehicles. In many of these applications, computer vision, a technology that through machine learning models extracts information from images, has gained a relevant position among the most impacting and disruptive technologies.

Going deeper, object detection models have become a crucial stream of research in enhancing this innovation. In the field of autonomous vehicles, deep learning models like You Only Look Once (YOLO) have been widely used in real-time road image detection due to the impressive results obtained in object detection tasks [1]. Nevertheless, these models can be deficient in challenging real-world situations, making the detection process still a big concern today.

One of the factors that can make an object detection system perform poorly is when camera lenses are affected by flare, a condition that can easily happen in the real world and a day-to-day scenario. This can be caused, for example, by direct sunlight, headlights from another vehicle, and refractions of light from the surrounding environment, leading to a potentially strong decrease in detection performance and increasing risk of incidents. Since autonomous vehicles must operate safely in varied and unpredictable environments, where an indefinite number of situations may happen, they must use models robust to the most common environmental perturbations. This can be achieved by training the model with the most suitable dataset, which can provide challenging conditions to satisfy the previous requirements.

In this research we are going to use subsets of three different widely used datasets in autonomous driving research: KITTI [2], nuImages (NuScenes) [3], and BDD100K [4]. We will train separate YOLOv8 models on KITTI and nuImages. To evaluate the robustness of these models, we will introduce controlled noise into the BDD100K dataset, simulating challenging lighting conditions.

1.1 Theoretical Framework

Object detection, as explained in [5], is a widely used technology, whose use cases vary from medical image analysis to self-driving vehicles, and its importance is continuously growing. In the new generation of cars, this technology is used to detect and localize objects such as vehicles, traffic signs, and people. It can be seen as a safety and support tool for the driver, enabling the car to emergency braking or audio alarms for possible hazards. The necessity of ensuring the ability to perform all of these tasks led us to choose YOLOv8 for our evaluation.

1.1.1 YOLO Characteristics

YOLOv8 [1] is a computer vision model developed by Ultralytics [6], considered state of the art among the algorithms for object detection, in particular real-time object detection. The acronym YOLO reflects its primary characteristic of predicting all possible boxes through a single-shot object detector, dividing images into grids, and predicting bounding boxes and class probabilities from the image in one pass. This makes the algorithm outperform its peers, who have to pass through several steps before being able to predict classes. It is the significant successor of YOLOv5, another model provided by the same company that had an impactful contribution to the field. However, thanks to the introduction of an anchor-free detection system, able to detect the center of an object without the use of a known reference box, the new model can perform incredibly well in object recognition, classification, and segmentation.

1.1.2 YOLO Architecture

The architecture of the model, as described in [1], consists of three parts: the backbone, used to extract features from the input image through deep learning techniques; the neck, which acquires and combines the information produced by the backbone layers; the head, where all the previous manipulations are used to predict the bounding-boxes and the labels.

1.2 Literature Study

Several studies have explored the impact of noise on object detection models. For example, the research that led to the creation of the GLARE dataset [7] analyzed the performance degradation of popular object detection models like YOLO and Faster R-CNN when trained on datasets without glare. Those results were then enhanced by training their models on the GLARE dataset, showing the importance of the dataset used for training. Further research [8] delves into how different light conditions affect object detection models. Their work showed that by understanding and addressing light parameters such as intensity and contrast, accuracy in both controlled and real-world environments can be enhanced. These studies highlight the importance of training datasets that capture diverse environmental conditions, including glare. However, a gap exists in the literature about practical guidance for choosing between popular datasets for training. This report addresses this gap by providing a data-driven comparison of two datasets widely used in autonomous systems development: KITTI and nuImages. We evaluate which dataset retrieves a YOLOv8 model that demonstrates better robustness to challenging environmental conditions, particularly those related to the issues discussed above.

2 Research Question and Hypotheses

As discussed in the introduction, datasets meant to be used in the autonomous vehicles field should enable the development of robust models that can operate in a wide variety of external conditions. One significant challenge that the models should handle is the presence of sunlight or reflections toward the camera, making it harder for the models to detect objects. Therefore, in this report we aim to answer the research question, which dataset, among KITTI and nuImages, provides a more robust model for object detection in presence of noise due to light conditions?

Based on observations of the datasets, we expected YOLOv8 to do better when trained on nuImages dataset compared to the model trained on KITTI. The hypothesis is based on the fact that nuImages contains a broader range of scenarios and light conditions, making the latter dataset more representative of real-world environment. While KITTI is also meant for autonomous driving, its limited range of scenarios may result in comparatively lower performance.

3 Research Methods

This section describes the research approach, data collection methods, and analysis techniques used in the study. The study, in order to answer the research question and prove the hypothesis in section 2, adopts an experimental approach to extrapolate quantitative evaluations. This is done by analyzing the difference in performances of YOLOv8 trained separately with the KITTI dataset and nuImages dataset and then tested on an unseen dataset, BDD100k, with artificially varied light conditions.

3.1 Model

As mentioned in the section 1.1 we choose YOLOv8 by Ultralytics as the state-of-the-art model in object detection. The model is available in different versions: n, s, m, l, xl, and the difference between these variants lies in the trade-off between accuracy, speed, and model size. In particular, if we use n, the smallest model, we will see the fastest inference but the lowest accuracy while with xl we will have the opposite performance. We chose the s version, to guarantee a balance between accuracy and speed during the process.

3.2 Datasets

In this project, we took into consideration established datasets commonly employed in the development and assessment of autonomous vehicle technologies. However, due to constraints related to computational resources and processing efficiency, we opted to utilize smaller versions of the original datasets. These subsets maintain the original dataset's essential characteristics and diversity. The class distribution of the datasets used for training can be seen in the Appendix A.1. Next follows a description of the original datasets from which these versions were derived.

- **KITTI** dataset [2] comes from the effort to produce a dataset to progress towards fully autonomous driving cars. It combines the use of different sensors such as perspective stereo cameras, two fisheye cameras, a Velodyne, and a SICK laser scanning unit to be able to perceive information from a 360-degree perspective. In addition, annotations are defined in both 3D and 2D bounding boxes. A subset of this dataset will be used for training.
- **NuImages** dataset is part of nuScenes [3]. Taking advantage of the fast rise of autonomous vehicles, the project aims to create image-based benchmark datasets to foster the development of computer vision and object detection systems. Using a wide range of sensors such as 1 lidar, 6 cameras, and 5 radars it deploys dynamic scenes with 3D and 2D bounding boxes. The original dataset is comprehensive of 23 classes and 93,000 images. The subset used for training was taken from [9].
- **BDD100k** [4] is a dataset for instance segmentation, semantic segmentation, object detection, and identification tasks used in the automotive industry. The dataset provides a comprehensive evaluation platform for image recognition algorithms since it includes geographic, environmental, and weather diversity. A subset of BDD100K taken from [10] is used for the model evaluation.

Due to computational constraints, the datasets were reduced to approximately 6000 images each. The specific subsets used for training and testing are detailed in Appendix A.2.

3.3 Metrics

We used three metrics to evaluate the robustness of the model. The theory behind each metric is derived from [11].

Firstly, the precision-recall (PR) curve is introduced, this graph shows the tradeoff between precision and recall for different classification thresholds. A large area under the curve represents both high recall and high precision which generally means a better model. To calculate the curve the following metrics are used:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Moreover TP denotes true positives (correctly predicted as positives), FP represents false positives (erroneously predicted as positives), and lastly FN corresponds to false negatives (incorrectly predicted as negatives). The precision and recall values are calculated for different thresholds and then plotted on a graph. Then, the F1 Score metric is used, which represents the harmonic mean of precision and recall. That is, the F1 Score is calculated as:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Finally, the Mean Average Precision ($mAP@50$) is calculated. This metric describes the average precision (AP) for each class averaged across all classes, using an Intersection over Union (IoU) threshold of 0.5 to determine a correct detection. That is, the $mAP@50$ is calculated as:

$$\text{mAP@50} = \frac{1}{N} \sum_{i=1}^{N} \text{AP}_i$$

where N is the number of classes and AP_i represents the average precision of the class taken into consideration. The average precision of class i can be calculated as:

Table 1: Google Colab Specifications.

Component	Specification
Processor	Intel Xeon CPU
GPU	16GB GPU Tesla T4
GPU accelerator	CUDA 12.2

Table 2: Local PC Specifications.

Component	Specification
Processor	Intel Core i7
RAM	16GB
Storage	512GB SSD

$$\mathrm{AP}_i = \frac{1}{m}\sum_{j=1}^{m} \mathrm{Precision}(j)$$

Here m denotes for the number of recall thresholds considered.

3.4 Experimental Methodology

Here the experimental setup and procedures followed in our experiment are outlined. The environment used for training and testing, the training process for model development, and the steps taken to prepare and modify the test dataset are described in the following paragraphs. Finally, the evaluation approach is detailed, ensuring a systematic comparison of the models under consistent conditions.

3.4.1 Experimental Setup

The training and testing procedures were conducted in two different computational environments. The YOLOv8s training was performed using the free version of the Google Colab platform, which was chosen for its accessibility and computational capabilities. However, the main constraint of the free version is the limited access to the platform's GPU. The environment's specifications are shown in Table 1. The testing dataset preparation and the evaluation of the models were conducted on a personal computer. The specifications of the local PC used for this purpose are provided in Table 2.

3.4.2 Training Procedure

After collecting two training datasets, Ultralytics' YOLOv8s model was used for object detection training. The training was conducted on Google Colab, leveraging the platform's resources listed in the table 1. The training hyperparameters are the standard one for Ultralytics train function, for clarity some of them are listed in Table 3.

Two distinct YOLOv8s models were retrieved after training, each reshaped to the specific characteristics of its dataset. These models will be evaluated using the test dataset.

Table 3: YOLOv8s Training Hyperparameters.

Hyperparameter	Value
Batch size	16
Image size	640x640 px
Epochs	100
Learning rate	0.01
Pre-trained	True

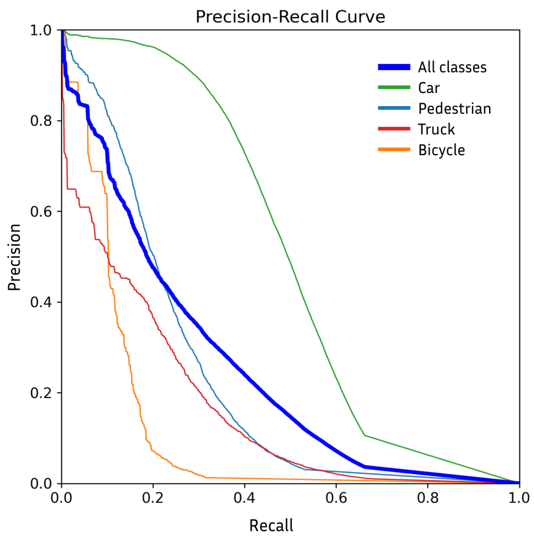

Figure 1: Precision-Recall Curve of nuImages model.

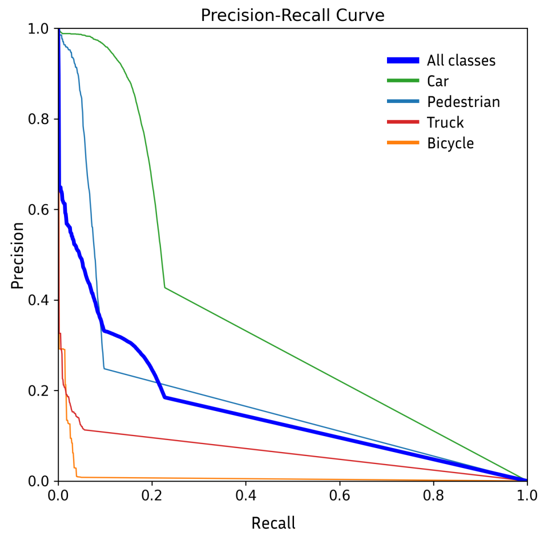

Figure 2: Precision-Recall Curve of KITTI model.

3.4.3 Testing Dataset Preparation and Evaluation

The test dataset [10], a subset of the BDD100k dataset, was initially manipulated to align with the evaluation requirements. Firstly, heavily underrepresented classes were removed from the dataset, then images with no valid annotations after filtering were discarded, afterwards labels were adjusted to match the class definitions used in the two trained models, and finally, artificial light disturbances were added to the images using the open-source python package OpenCV [12]. The implementation details and an example of processed image can be seen in the code available at the GitHub link in the Appendix B.

Once preprocessing was completed, the test dataset was used to evaluate the KITTI-trained and nuImages-trained models. The evaluation metrics used can be found in the previous subsection 3.3.

4 Results and Analysis

This section evaluates the robustness of the two YOLOv8s models trained on different datasets by testing their performance on a third dataset modified with artificial lens flare, as described in 3.4. The goal was to analyze how well each model reacts to challenging conditions, particularly in detecting objects under visual disturbances. The PR curve for the nuImages-trained model demonstrates a more gradual drop in precision as recall increases, generally indicative of better generalization capabilities. On the other hand, the PR curve for the KITTI-trained model showed a sharper decline, especially in the underrepresented and challenging classes.

An important observation is the difference in the areas under the two PR curves, with nuImages showing a noticeably larger area meaning better overall performance. Despite this, it is worth noting that at high recall values, the KITTI model has slightly higher precision. This phenomenon could reflect a more conservative approach by the KITTI model, avoiding false positives.

Another interesting aspect of those two PR curves is the performance of individual object classes. The car class consistently achieves higher precision due to its prominence in the training datasets which causes the model to be more confident even in challenging conditions. Inversely, less-represented classes, such as bicycles, reveal weaknesses in detecting smaller and less frequently occurring objects.

Table 4 summarizes key performance metrics explained in section 3.3. The nuImages model steadily outperforms the KITTI model across all metrics, achieving a mAP@50 of 0.250 compared to 0.157 for the KITTI model. This difference in the mAP metric underscores the nuImages model's superior ability to handle diverse and noisy scenarios. Even for the car class, which is more present in the KITTI dataset, the nuImages model accomplished an AP of 0.496, significantly higher than the 0.364 achieved by the

Table 4: Performance metrics of models on testing dataset.

Model	**Car AP**	**Pedestrian AP**	**Truck AP**	**Bicycle AP**	**mAP@50**	**F1 Score**
NuImages-based	0.496	0.229	0.162	0.114	0.250	0.31@0.21
KITTI-based	0.364	0.185	0.067	0.014	0.157	0.13@0.00

KITTI model. The maximum F1 Score for the nuImages model further confirms our results, reaching a value of 0.31, while the KITTI model's best F1 Score is at just 0.13 which is achieved at nearly 0 confidence threshold. This result may indicate high uncertainty for the model in classifying objects due to the complexity of the dataset.

An interesting and valuable insight can be retrieved when we compare the mAP@50 values achieved by the two models on their respective training datasets. Indeed, the KITTI model scored an impressive mAP@50 of 0.921, while the nuImages model achieved a fair lower value of 0.590. These data suggest that the KITTI model may overfit its training data, excelling in scenarios similar to the original dataset but struggling heavily in more generalized and noisy conditions. Furthermore, the low mAP@50 score achieved by the nuImages model on the original dataset confirms our belief of nuImages being a more variable and difficult dataset to learn, but as was shown that led to better generalization and robustness.

5 Discussion and Future Research

The research project results show how using the nuImages dataset to train an object detection YOLO model is more robust towards light-condition variations compared with the same model trained with the KITTI dataset. We can also deduce from these results, how the nuImages dataset comprehends diverse scenarios and more complex environmental conditions relative to KITTI.

Given the good results obtained, it is important to discuss and consider some aspects of our research project: partial datasets were used for the project due to limited resources and the datasets' model-readiness characteristics, which facilitated their integration into the workflow, avoiding further tuning to adapt to YOLOv8. The hyperparameters were not optimized, leaving room for potential performance improvements through a dedicated hyperparameter optimization process. Additionally, as highlighted in [13], the two datasets are not perfectly balanced, with some classes containing a larger number of instances in one dataset compared to the other. Specifically, KITTI is heavily biased toward cars, while nuImages offers relatively balanced classes. Despite this imbalance, the results demonstrate the robustness of the approach, as the model trained with nuImages outperformed the KITTI-trained model, even in car detection. Lastly, it is worth noting that the training process involved recognizing a broader range of classes than those selected during the testing phase.

The results obtained align with the research papers mentioned in the Literature Study. Following the trend described by the results in [7], nuImages proved to be a dataset that displays a higher variability of environmental conditions, producing a more robust model compared to the KITTI-trained one. In addition, as stated in [8], it is confirmed how the importance of considering the possible light conditions is crucial in order to produce a reliable model for real-world applications.

Future research could expand this study firstly by exploiting larger resources, enabling this study to be conducted over the entireness of the original dataset and with optimized training, with the ultimate goal of extracting the best possible model. Then it is also possible to explore and develop simulations which investigate different perturbations in external conditions, such as fog, rain and dirty cameras. This can be done to gain a more comprehensive overview of the robustness of diverse object detection datasets. In addition, it is possible to evaluate the robustness under varied light conditions using different models, such as Faster R-CNN [14], SSD [15], and DETR [16].

6 Conclusion

In conclusion, this research demonstrates the superior performance of nuImages dataset over KITTI for training object detection models in autonomous driving applications, specifically when facing challenging lighting conditions such as lens flare. The practical implications of our findings suggest that developers using YOLOv8 for object detection in autonomous driving scenarios, where the problem of lens glares and light disturbances is present, should prioritize the nuImages dataset over KITTI for improved performance.

This study provides a stronger foundation for developing object detection models with higher reliability and adaptability to real-world scenarios. By highlighting the robustness of nuImages dataset, this study contributes to the ongoing efforts toward creating safer and more dependable autonomous driving technologies, ultimately paving the way for a future defined by increased trust in technology and safer roads.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[2] Y. Liao, J. Xie, and A. Geiger, “Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, Mar. 2023, doi: 10.1109/TPAMI.2022.3179507.

[3] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Aug. 2020, pp. 11 618–11 628, doi: 10.1109/CVPR42600.2020.01164.

[4] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, “Bdd100k: A diverse driving dataset for heterogeneous multitask learning,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, Jun 2020, pp. 2633–2642, doi: 10.1109/CVPR42600.2020.00271.

[5] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, “A review and comparative study on probabilistic object detection in autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 9961–9980, Aug. 2022, doi: 10.1109/TITS.2021.3096854.

[6] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLOv8,” 2023, Ultralytics. [Online]. Available: https://github.com/ultralytics/ultralytics

[7] N. Gray, M. Moraes, J. Bian, A. Wang, A. Tian, K. Wilson, Y. Huang, H. Xiong, and Z. Guo, “GLARE: A Dataset for Traffic Sign Detection in Sun Glare,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 12 323–12 330, Nov. 2023, doi: 10.1109/TITS.2023.3294411.

[8] R. Huber, K. Schneider, A. Wetzel, E. Neitzel, and T. Brandmeier, “Light Analysis for Optimized Object Detection with Cameras for Integrated Safety Systems,” in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, Prague, Czech Republic, Jul. 2022, pp. 1–6, doi: 10.1109/ICECET55527.2022.9872761.

[9] new-workspace hbhr7, “Nuimages 10k dataset,” Roboflow, Mar. 2022. [Online]. Available: https://universe.roboflow.com/new-workspace-hbhr7/nuimages-10k-dataset

[10] eastsky, “bdd100k dataset,” Roboflow, Aug. 2023. [Online]. Available: https://universe.roboflow.com/eastsky/bdd100k-e3s18

[11] A. Zheng, *Evaluating Machine Learning Models*, 1st ed. Sebastopol, CA, United States: O'Reilly Media, Inc., 2015.

[12] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[13] D. Fernández Llorca, P. Frau, I. Parra, R. Izquierdo, and E. Gómez, "Attribute annotation and bias evaluation in visual datasets for autonomous driving," *Journal of Big Data*, vol. 11, no. 1, p. 137, Sep. 2024, doi: 10.1186/s40537-024-00976-9.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," in *Computer Vision – ECCV 2016*, Cham, Switzerland, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End–to–End Object Detection with Transformers," in *Computer Vision – ECCV 2020*, Cham, Switzerland, 2020, pp. 213–229, doi: 10.1007/978-3-030-58452-8_13.

A Datasets Details

A.1 Dataset Class Distribution

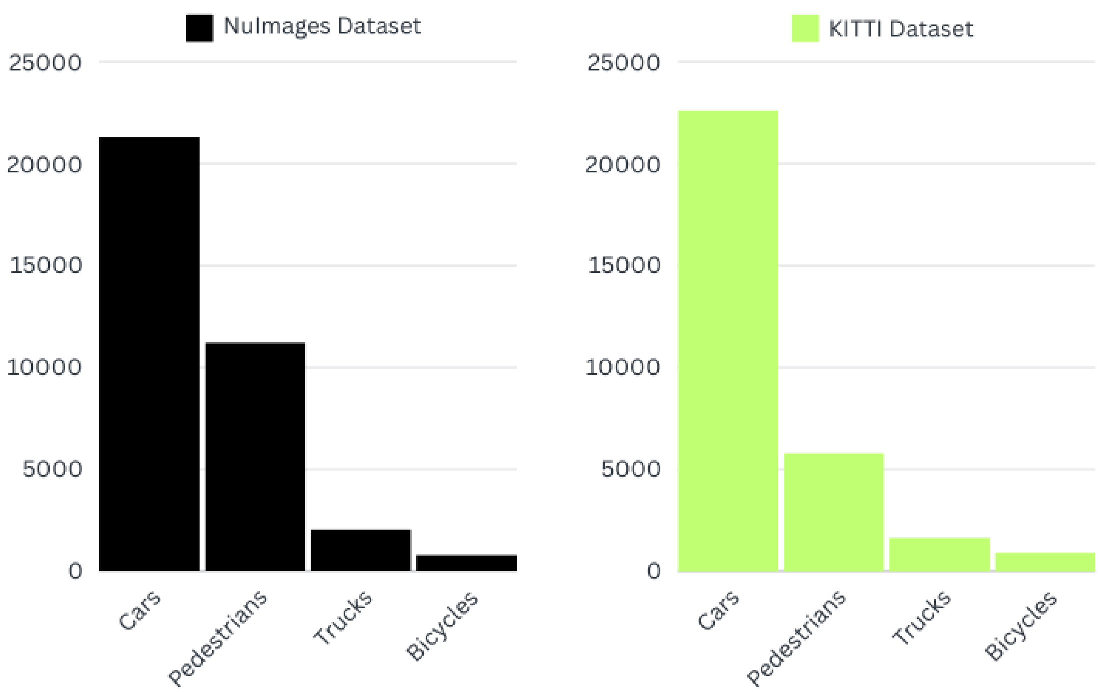

Figure 3: Class distribution plot for the training datasets.

A.2 Datasets Link

The datasets used in this experiment can be found here:
`https://kth-my.sharepoint.com/:f:/g/personal/fraoli_ug_kth_se/Evbz5EyoQHtOqjBmTPofaFUB0i7ouY9fq-kSBWhuY5N8qQ?e=86f9Qd`

B Code

The code for image processing can be found here:
`https://github.com/francescoolivieri/II2202-Code`.

Figure 4: Original image

Figure 5: Image processed with light disturbance