Title

Enhanced Road Object Detection by Fine-Tuning You Only Look Once Version 8 (YOLOv8)

Author

Xie, Huarui

Publication Date

2024

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

Enhanced Road Object Detection by Fine-Tuning You Only Look Once Version 8(YOLOv8)

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Applied Statistics and Data Science

by

Huarui Xie

2024

ABSTRACT OF THE THESIS

Enhanced Road Object Detection by
Fine-Tuning You Only Look Once Version 8(YOLOv8)

by

Huarui Xie
Master of Applied Statistics and Data Science
University of California, Los Angeles, 2024
Professor Yingnian Wu, Chair

This research focuses on improving YOLOv8 for detecting road objects from a pedestrian's viewpoint. It involves training three pre-trained models (YOLOv8n, YOLOv8s, YOLOv8m) on over 10,000 images, which include both a self-collected dataset of road objects and a subset from the COCO dataset. The study employs transfer learning to maintain the models' proficiency in recognizing the original COCO dataset classes while integrating seven new categories. The models' effectiveness was gauged using metrics such as precision, recall, mAP, and processing speed to identify the most suitable model for real-time road detection. Ultimately, the YOLOv8m model showed superior accuracy and reasonable processing speed, though its performance still falls short of real-world detection requirements.

The thesis of Huarui Xie is approved.

Nicolas Christou

Qing Zhou

Yingnian Wu, Committee Chair

University of California, Los Angeles

2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

Introduction

In the era of rapid technological advancement, the way humans live has been significantly affected. With the help of thriving technologies, AI gradually becomes a reliable assistant to humans in various fields, largely impacting daily activities and tasks. Like automating complex tasks, AI's influence extends to enhancing personal convenience and safety. AI's potential in auto-assistance is a prime example of how technology can serve as an extension of human capabilities, allowing for more precise, efficient, and safer completion of tasks.

As AI continues to mature, its integration into support systems for visually impaired individuals marks a transformative leap forward. For instance, AI can power applications that translate visual information into audible descriptions, helping those who are visually impaired to navigate public spaces more independently and safely. By processing real-time data about their environment, AI can alert users to obstacles, provide route suggestions, or even assist in complex spatial interactions, thereby enhancing their ability to interact with the world around them. This pivotal shift not only broadens accessibility but also opens up new avenues for inclusion and independence for the visually impaired, illustrating AI's role as a crucial ally in improving human life. Sample images on the road from a pedestrian perspective are illustrated in Figure 1.1.

This dissertation focuses on advancing road object detection capabilities on mobile devices through the implementation of the cutting-edge YOLOv8 algorithm, known for its real-time object detection proficiency. The primary aim is to adapt a pre-trained YOLOv8 model to recognize seven additional road-related objects from a pedestrian perspective, en-

(a)

(b)

Figure 1.1: Sample images on road

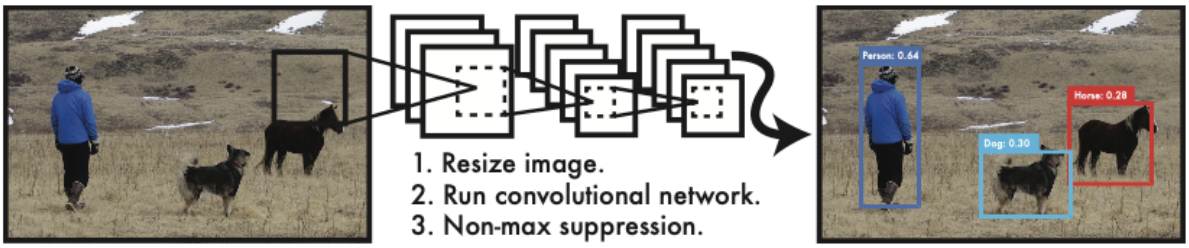


Figure 1.2: YOLO detection system.

hancing navigational aids for visually impaired individuals. The YOLOv8 model excels in quickly capturing and processing images to detect objects and estimate crowd density, which can be crucial for pedestrian safety.

YOLO (You Only Look Once), initially introduced by Joseph Redmon et al. in 2016, revolutionized object detection by enabling real-time detection through a single pass of the neural network. Unlike previous methods that relied on sliding windows or region proposals, YOLO processes the entire image at once, making it significantly faster and more efficient. The algorithm divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell. This approach allows YOLO to understand contextual information and improve detection accuracy. [5]. The working flow of the detection system is shown in Figure 1.2 [4].

To achieve this, the model will be fine-tuned on a curated dataset that includes these new objects, ensuring the model can accurately identify and localize them in various environmental conditions. This process involves adjusting the model's parameters to better suit the specific features and scales of road objects as seen from pedestrian viewpoints.

Additionally, integrating the output of the YOLOv8 model with language processing and text-to-speech technologies will enable the creation of real-time assistive messages. These messages will inform visually impaired users about their surroundings, helping them navigate public spaces safely and independently.

The subsequent chapters of this thesis will delve into the specifics of the YOLO algorithm, outlining the experimental setup, the fine-tuning process, and the results obtained. A detailed discussion will follow, analyzing how effectively the model performs and exploring potential limitations and improvements. Finally, the thesis will conclude with a consideration of the broader implications of deploying such advanced AI-driven technologies in assistive applications, emphasizing their potential to significantly enhance the quality of life for visually impaired individuals.

CHAPTER 2

Methodology

2.1 YOLO

2.1.1 YOLOv1

YOLO (You Only Look Once) is an innovative algorithm that employs a single convolutional neural network for fast and effective object detection. It processes the entire image at once for training and testing, allowing it to detect multiple objects simultaneously and provide bounding boxes and class probabilities. This processing approach helps YOLO capture contextual information about object classes more effectively than many other detectors [4]. YOLO is known for its good accuracy and superior speed, making it useful in real-time applications such as autonomous driving systems, face recognition, and surveillance systems, as shown in Figure 2.1 [5].

YOLO architecture includes 24 convolutional layers followed by 2 fully connected layers in Figure 2.2 [4]. Each convolutional layer uses 1 x 1 reduction layers followed by 3 x 3 convolutional layers, which was inspired by the GoogleNet model [4]. The model divides the input image into an S x S grid, in which multiple bounding boxes for objects and their confidence scores are calculated based on predefined anchor boxes. During training, an object may have multiple bounding boxes with different class probabilities, only the bounding predictor with an IoU greater than a pre-defined threshold will be kept [4].

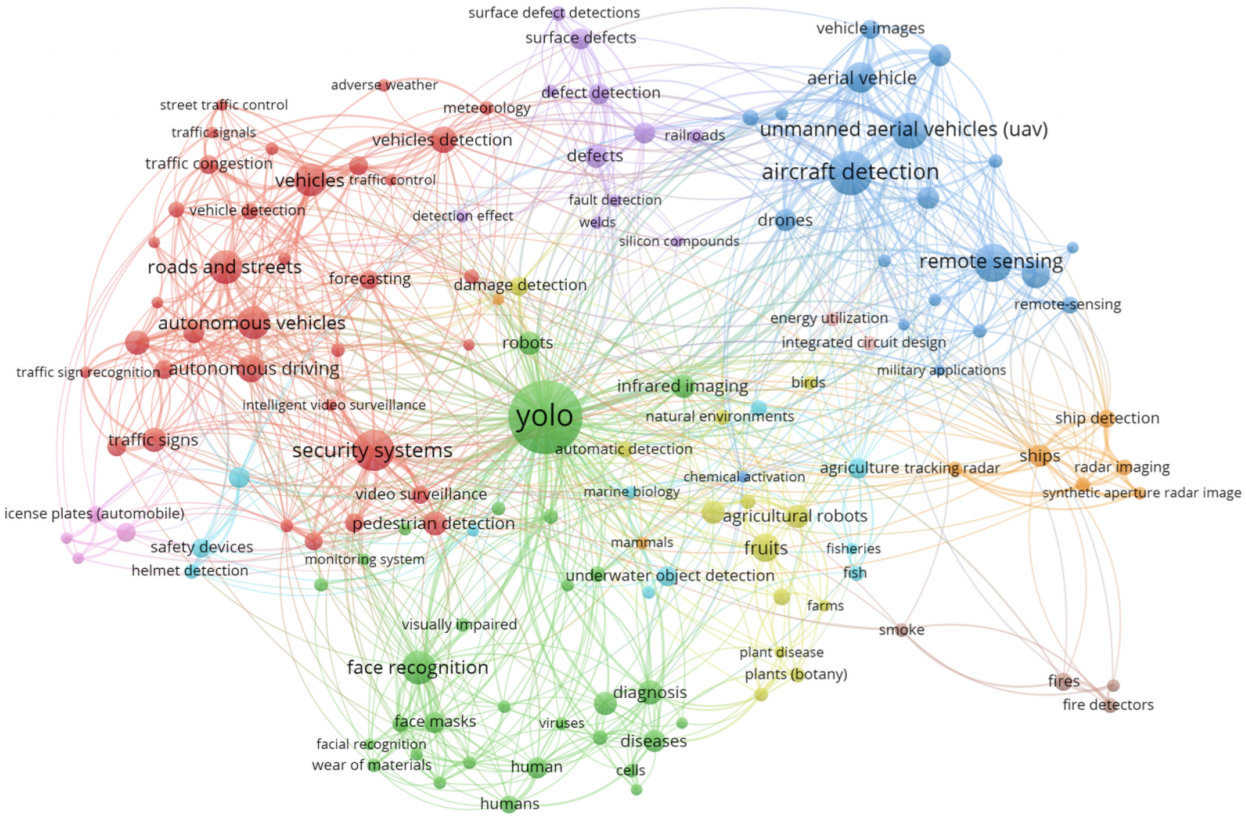


Figure 2.1: YOLO applications

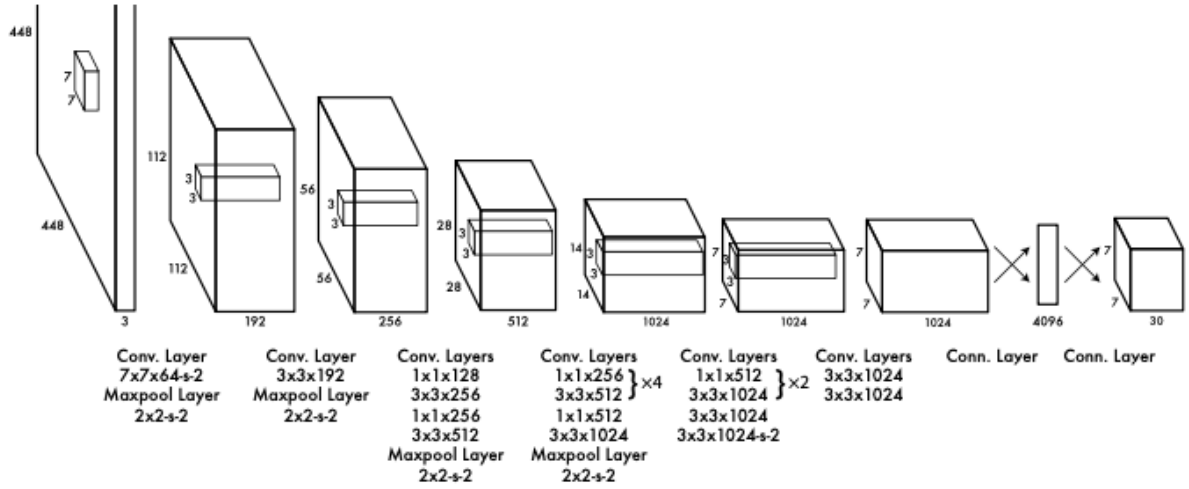


Figure 2.2: YOLOv1 architecture

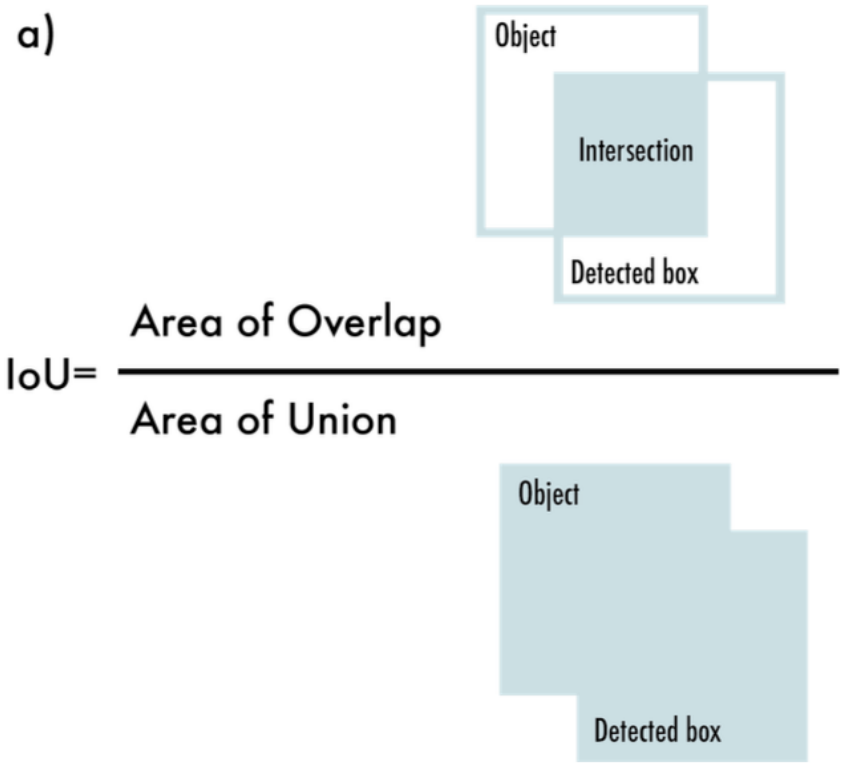
2.1.2 IoU

The IoU - Intersection Over Union between the predicted box and the ground truth is a crucial part of the Non-Max Suppression (NMS). Non-Max Suppression is a post-processing technique used to filter out redundant bounding boxes in object detection. When multiple bounding boxes are predicted for the same object, NMS helps in selecting the best one. [5]. IoU is defined as the area of overlap between the predicted and the actual bounding boxes divided by the area encompassing both boxes, effectively measuring their overlap, as shown in Figure 2.3 (a) [5]. The COCO benchmark evaluates object detection models by considering various IoU thresholds, thereby assessing the models' precision at different levels of localization accuracy [5].

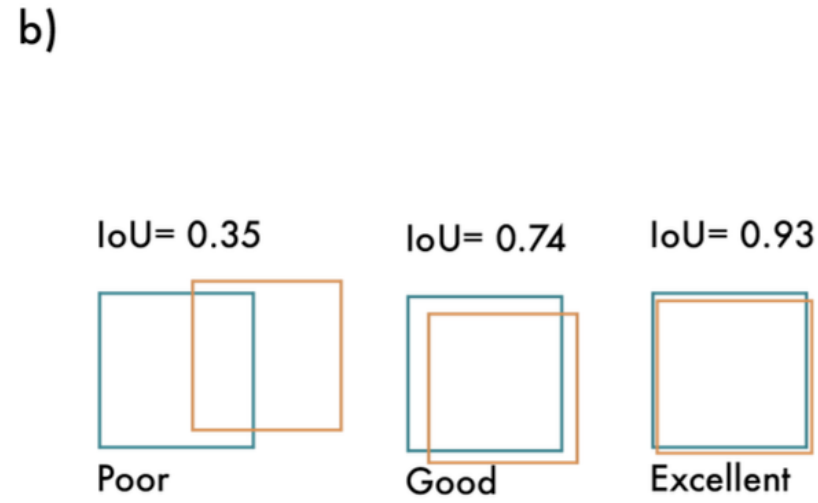
In the evaluation metrics, the mean Average Precision (mAP) at different levels of IoU will be calculated to test the robustness and accuracy of the object detection model. This comprehensive evaluation ensures that the model's performance is not only based on a single threshold but across a range of IoU values, providing a more detailed and nuanced understanding of its capability to localize and identify objects correctly. This method allows for a thorough assessment of how well the model can distinguish between objects that are close to one another and its ability to handle varying degrees of overlap between predicted and ground truth bounding boxes.

2.1.3 YOLOv8

YOLOv8 is one of the latest versions of the YOLO family. Over each iteration, the detection mechanism has changed and the capability becomes more stable and reliable to handle complicated tasks with faster speed and higher accuracy. Compared to the previous version, YOLOv8 utilizes a more complex and efficient architecture such as CSPDarknet53 backbone to enhance feature extraction as shown in Figure 2.4 [5]. By implementing an anchor-free approach, the model eliminates the need for predefined anchors, directly predicting bounding



(a) Calculation of IoU



(b) Examples of different IoU

Figure 2.3: Intersection over Union (IoU)

boxes based on the features extracted by the network. The usage of binary cross-entropy for classification loss and advanced loss functions like CIoU and DFL for bounding box loss improve the detection performance on small objects and classification tasks. Even though the previous YOLO versions are fast, their real-time processing capabilities are still limited by the hardware. The optimizations of YOLOv8 increase the computational efficiency and speed, making it suitable for real-time application on mobile devices [5].

For a real-time assistive application on a mobile device, YOLOv8 is suitable for its high accuracy, high speed, and low computational requirement. By fine-tuning the model, this application can largely assist visually impaired people.

A similar study was done by developing a visually impaired indoor navigation system integrated with the YOLO algorithm, which utilized real-time object detection and monocular depth estimation to assist visually impaired individuals in navigating indoor spaces safely. Combined with monocular depth estimation and spatial audio techniques, this approach provides accurate object' coordinates in audio format, enabling users to avoid obstacles and find safe paths, highlighting the potential for expanding such designs into wide usage outdoors for the visually impaired [1].

2.2 Transfer Learning

Since the pre-trained YOLOv8 can already detect several on-road object classes such as "person," "car," and "traffic light," it is unnecessary to retrain the model with those classes for computational and time concerns. Transfer learning offers an efficient solution for this study. It seeks to enhance the performance of models in specific target domains by leveraging knowledge acquired from different but related source domains [6]. By utilizing the trained weights from the pre-trained YOLOv8 model, the model retains its ability to detect existing classes. Figure 2.5 illustrates intuitive examples of transfer learning [6].

Transfer learning aims to minimize training time, costs, and the need for extensive

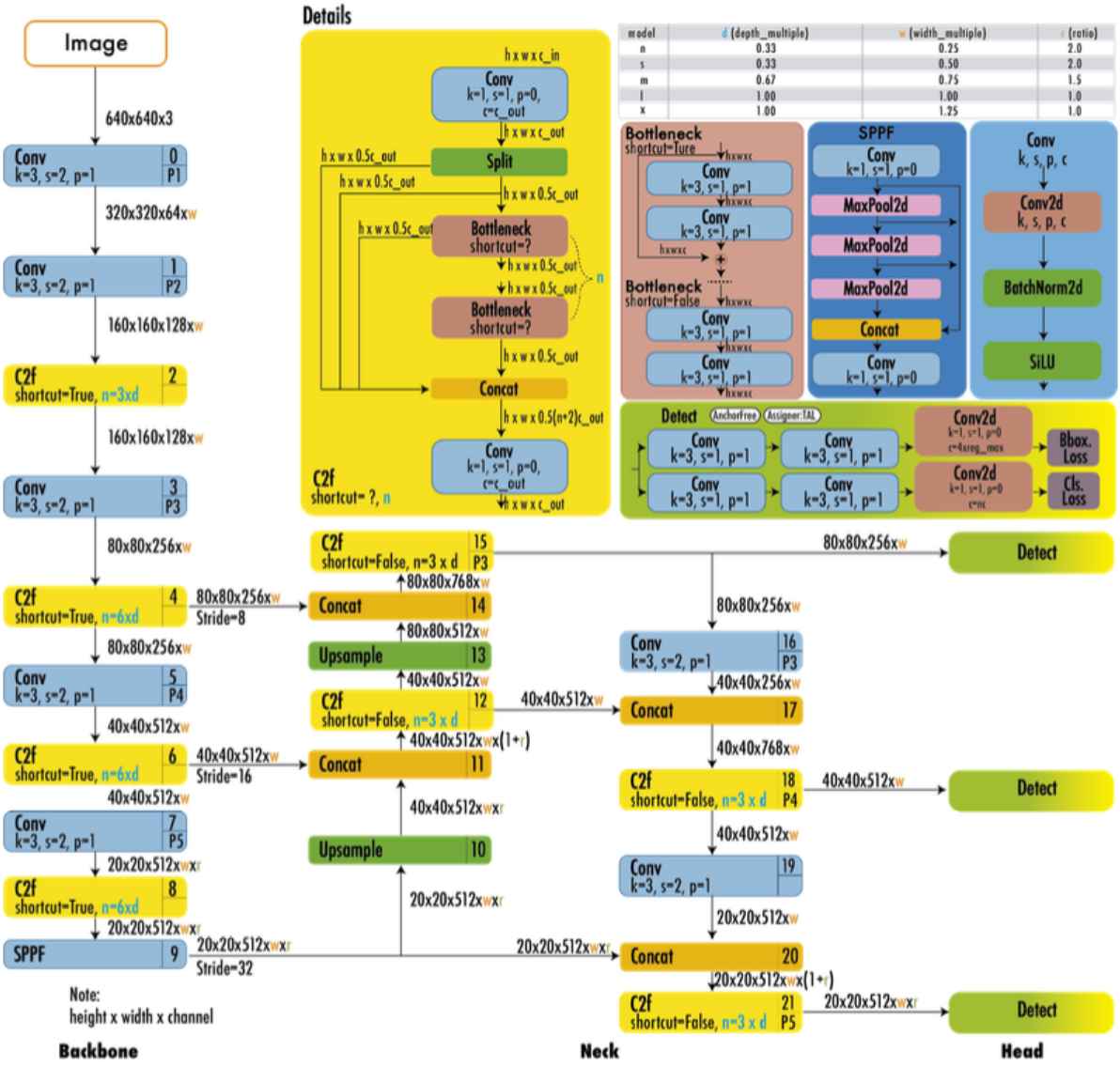


Figure 2.4: YOLOv8 architecture

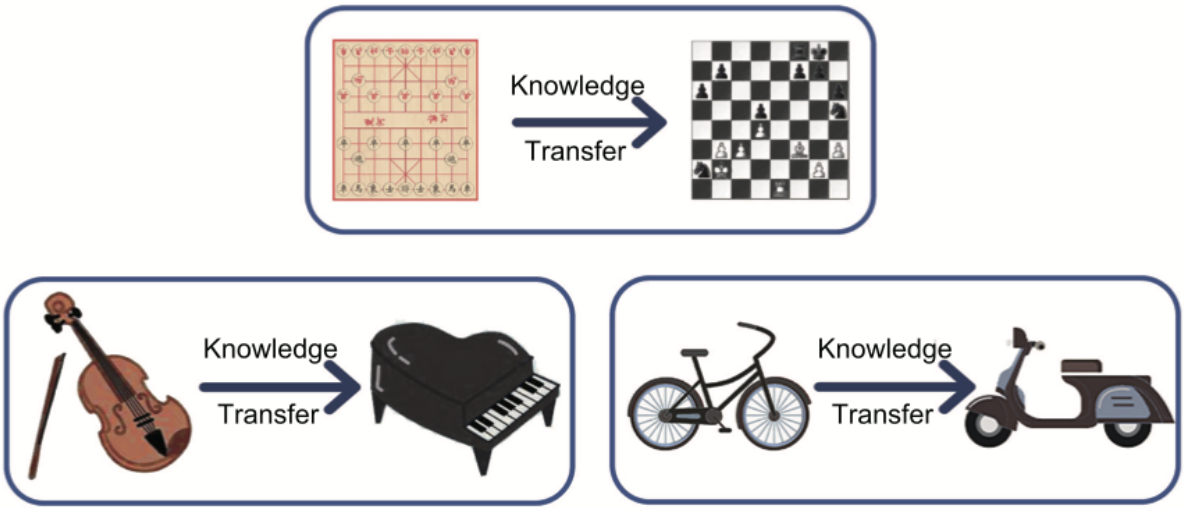


Figure 2.5: Transfer learning examples

datasets, which are often challenging to obtain, and pre-trained models can now be effectively run on edge devices like cell phones with limited processing power and training time. [2]. In this context, freezing the backbone layers of the YOLOv8 model—which consists of the initial layers responsible for feature extraction—allows the model to retain the learned features from the original dataset. Then, fine-tuning the model with a combination of the original dataset and new data specific to the additional road objects can enhance its detection capabilities without requiring extensive retraining from scratch, as shown in Figure 2.6 [2].

Incorporating transfer learning into this study can significantly reduce computational costs and training time. It allows for the reuse of established models, thereby accelerating the development of robust object detection systems for mobile devices. This approach not only conserves resources but also enables the model to maintain high performance in real-time detection tasks.

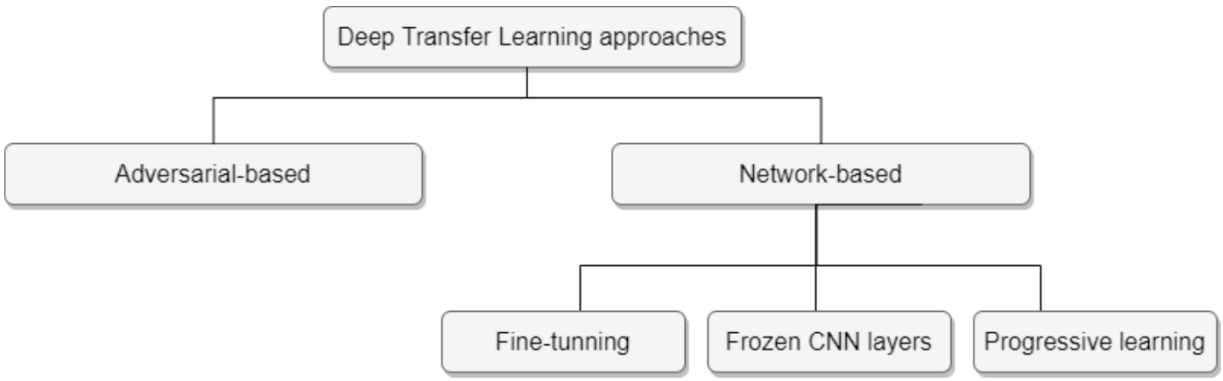


Figure 2.6: Deep transfer learning approaches

CHAPTER 3

Experiment

3.1 Dataset

The YOLOv8 developed by Ultralytics was trained on the public dataset COCO. More than 118,000 images of 80 classes were trained and 5,000 images were used to validate the model accuracy. Since the potential road objects are not fully covered in the COCO dataset, more data is needed. For this study, 7 additional classes of road objects will be added to the training along with a subset of the original COCO data.

3.1.1 Self-Collected Data

The new classes and the number of their labels are: "Tree": 11,633, "Building": 2204, "Stairs": 1418, "Street light": 1168, "House": 2048, "Waste container": 1153 from the Open Image Dataset V7, and "Crossing": 2935 from Zoned. All of them build up the training size of 8338, the testing size of 1368, and the validation size of 1266.

3.1.2 Subset of COCO

Retraining a subset of the original COCO data can retain the model's ability to detect the old classes, and avoid compromising the information of the other classes. The number of COCO data to be retrained is 2801, 600 for validation and 600 for testing separately, including all 80 original classes.

In total, the size of the training, testing, and validation images are 11,139, 1968, and 1896 respectively. Some of the training samples are in Figure 3.1.

3.2 Training and Evaluation

To meet the demands for speed in real-time applications, this study evaluates YOLOV8 Nano, Small, and Medium. Generally, larger models offer higher accuracy but at the cost of reduced processing time, which can be problematic for mobile device deployment. The selected models strike a balance, providing real-time detection capabilities with satisfactory accuracy. By analyzing both the prediction accuracy and processing time of those models, this study aims to determine which model outperforms in real-time detection tasks, considering the constraints of the mobile environments. The models and hyper-parameters selections were inspired by a pothole detection project, which is a similar real-time road detection task [3].

3.2.1 Hyper-Parameters

In this study, each input image has a size of 640 x 640 without any data augmentation, and the first 10 backbone layers were frozen to maintain the original weights. Model training was conducted on v100 GPU in Google Colab, the batch size was set to 64, the initial learning rate was set to 0.01, the IoU was set to 0.6, and the number of epochs was 50 due to the computational limit. The model testing process was conducted on the Apple M2 CPU.

3.2.2 Evaluation Metrics

The evaluation of the model focuses on several key metrics: precision, recall, mean Average Precision at IoU 0.5, mean Average Precision at IoU from 0.5 to o.95, and the processing time of each image (speed):

(a) Tree

(b) Stair

(c) Waste Container

Figure 3.1: Training Samples

- Speed: The total processing time per image, including the time for pre-processing, inference, and post-processing.

- Precision: This metric measures the proportion of correct positive predictions (true positive) among all positive predictions (including both true positive and false positive). A high precision indicates the model is accurate in detection. For instance, correctly detecting a stair rather than mistakenly detecting a crossing can prevent a pedestrian from potential danger.

- Recall: This metric measures the proportion of correct positive predictions (true positive) among all actual positives (combination of true positive and false negative). High recall ensures the model can correctly detect as many relevant objects as possible, promising the pedestrian to avoid obstacles.

The formulas for precision and recall are listed:

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

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

where TP is true positive, FP is false positive, and FN is false negative.

3.3 Results

3.3.1 Evaluation Metrics on Test Data

After training, the test results for all classes of the YOLOv8 Nano, Small, and Medium models are listed in Table 3.1. As expected, there is a clear trade-off between accuracy and speed: as the model size increases, the prediction becomes more accurate while the processing time increases. The mAP scores are consistent with precision and recall scores, indicating that the larger models can be reliable across different IoU thresholds. The YOLOv8 Medium model has the highest precision of 0.571 but the slowest processing time of 0.605 seconds per image. These evaluation metrics check the models' abilities in all 87 classes, while this

Table 3.1: Comparative Analysis of YOLO Models for All Classes

Model	**Precision**	**Recall**	**mAP:50**	**mAP50-95**	**Speed**
YOLOv8n	0.479	0.4	0.382	0.268	0.137s/image
YOLOv8s	0.552	0.474	0.468	0.333	0.296s/image
YOLOv8m	0.571	0.539	0.526	0.385	0.605s/image

study aims to identify the on-road objects. So the irrelevant classes may affect the overall accuracy since they were included in the training process.

Reviewing the evaluation metrics for specific on-road objects across the models in Tables 3.2, 3.3, and 3.4, it's evident that the class "Crossing" consistently outperforms well, achieving nearly perfect scores in precision and recall. In contrast, the classes "Traffic Light" and "Bicycle" display weaker performance metrics. This phenomenon can likely be attributed to the uneven distribution of class labels in the training dataset. Specifically, "Crossing" has significantly more labels compared to "Traffic Light" and "Bicycle," which are less represented in the COCO dataset subset. Surprisingly, despite having the largest number of labels among the newly added classes, "Tree" exhibits a low recall ratio, indicating challenges in effectively recognizing tree features within the collected data.

This observation underscores the significance of data augmentation as a strategy to enhance training data quality. By correcting class imbalances and enriching class diversity through training data, models are better equipped to accurately identify and classify a broader range of features.

3.3.2 Training Progress

The analysis of the YOLOv8 models (n, s, m) across precision-confidence, recall-confidence, and precision-recall curves in Figures 3.2, 3.3, and 3.4 demonstrates that YOLOv8m consistently outperforms the other variants in terms of precision and recall across varying con-

Table 3.2: Evaluation Metrics of YOLOv8n Significant Classes

Class	**Precision**	**Recall**	**mAP:50**	**mAP50-95**
Person	0.502	0.499	0.467	0.306
Bicycle	0.454	0.203	0.194	0.0884
Car	0.342	0.224	0.188	0.112
Traffic Light	0.355	0.0962	0.205	0.107
Tree	0.553	0.16	0.263	0.163
Building	0.594	0.157	0.271	0.198
Stair	0.591	0.458	0.514	0.283
Waste container	0.554	0.866	0.727	0.63
Crossing	0.825	0.877	0.925	0.665

Table 3.3: Evaluation Metrics of YOLOv8s Significant Classes

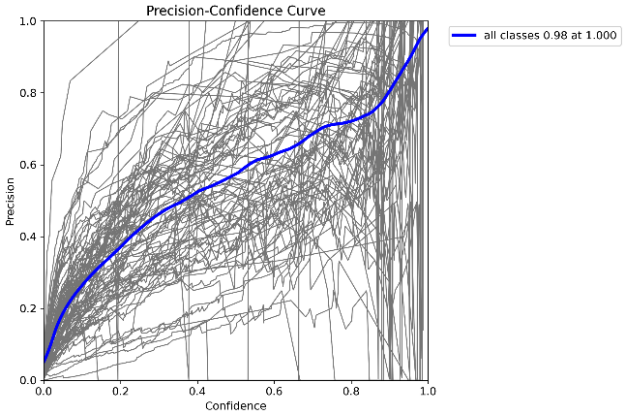
Class	**Precision**	**Recall**	**mAP:50**	**mAP50-95**
Person	0.535	0.585	0.537	0.364
Bicycle	0.237	0.119	0.136	0.0638
Car	0.514	0.314	0.311	0.192
Traffic Light	0.519	0.311	0.35	0.168
Tree	0.544	0.143	0.259	0.159
Building	0.606	0.166	0.252	0.173
Stair	0.566	0.438	0.447	0.26
Waste container	0.551	0.848	0.664	0.541
Crossing	0.852	0.864	0.932	0.683

Table 3.4: Evaluation Metrics of YOLOv8m Significant Classes

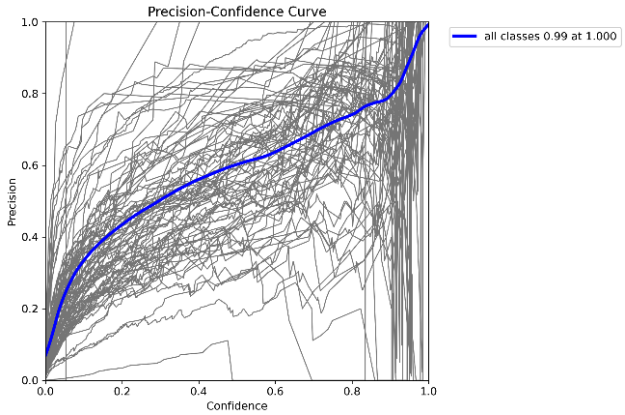
Class	**Precision**	**Recall**	**mAP:50**	**mAP50-95**
Person	0.577	0.63	0.594	0.412
Bicycle	0.33	0.288	0.216	0.106
Car	0.379	0.391	0.344	0.231
Traffic Light	0.443	0.423	0.385	0.211
Tree	0.593	0.177	0.297	0.192
Building	0.638	0.156	0.303	0.232
Stair	0.53	0.484	0.469	0.27
Waste container	0.625	0.866	0.763	0.654
Crossing	0.913	0.952	0.968	0.755

fidence thresholds. YOLOv8m maintains higher precision even at lower confidence levels and exhibits a slower decline in recall as confidence increases, indicating its robustness in identifying relevant objects without sacrificing accuracy. The precision-recall curves further illustrate that YOLOv8m has the best balance between precision and recall, with the highest area under the curve (AUC), suggesting it is the most reliable model for applications requiring high detection accuracy. While YOLOv8n and YOLOv8s show lower performance, they might still be suitable for scenarios where computational efficiency is more critical.

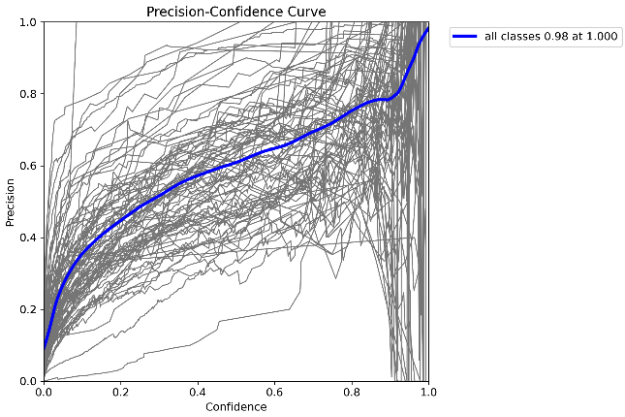
Figure 3.5 is the training results, including the loss of training and validation dataset. The loss curves of different models tell the story that all models show a decreasing trend in both training and validation losses over the epochs, indicating good model convergence. The loss curves are smooth without any fluctuations and flatten out as the epochs progress, suggesting that additional training beyond 40 epochs might yield diminishing returns in terms of loss reduction. Clearly, as the model complexity increases from n to m, there is a significant improvement in all evaluation metrics, meaning that the additional computational cost in the complex model is justified by the performance gains.



(a) YOLOv8n

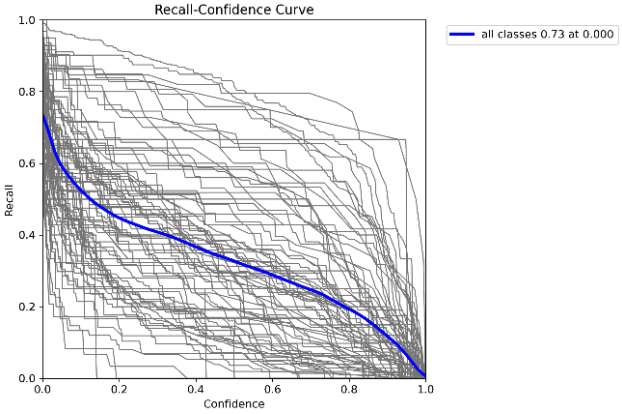


(b) YOLOv8s

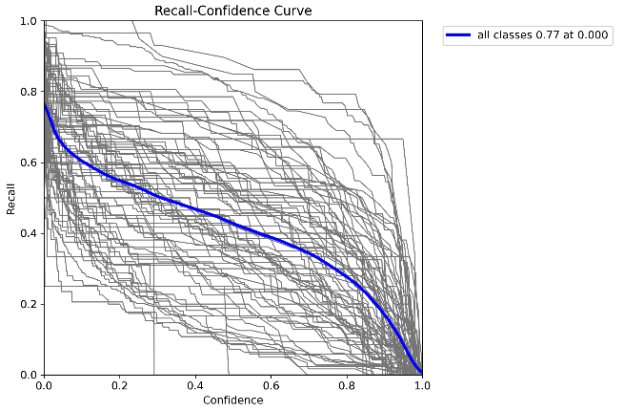


(c) YOLOv8m

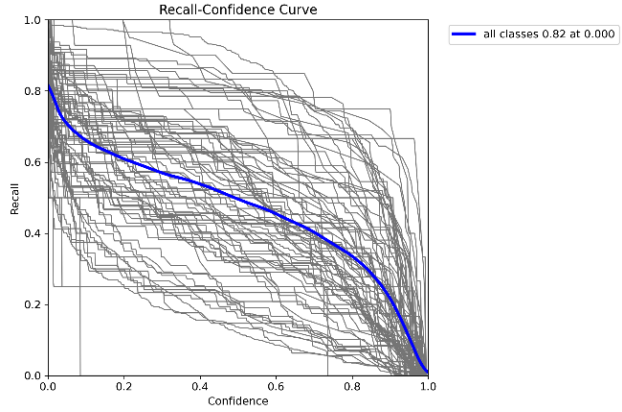
Figure 3.2: Precision curves of YOLOv8 models



(a) YOLOv8n

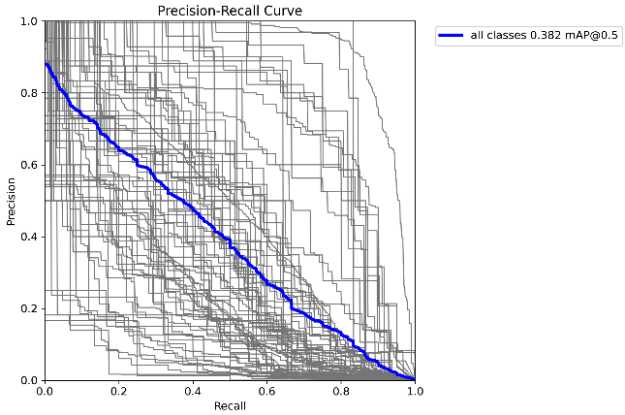


(b) YOLOv8s

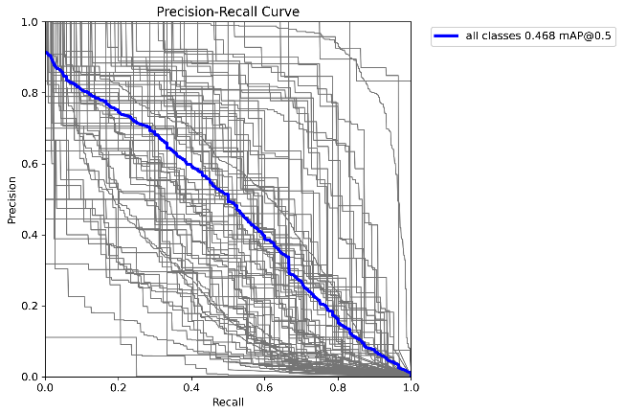


(c) YOLOv8m

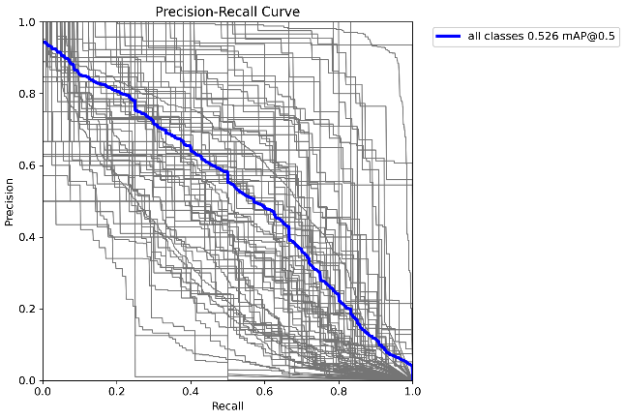
Figure 3.3: Recall curves of YOLOv8 models



(a) YOLOv8n

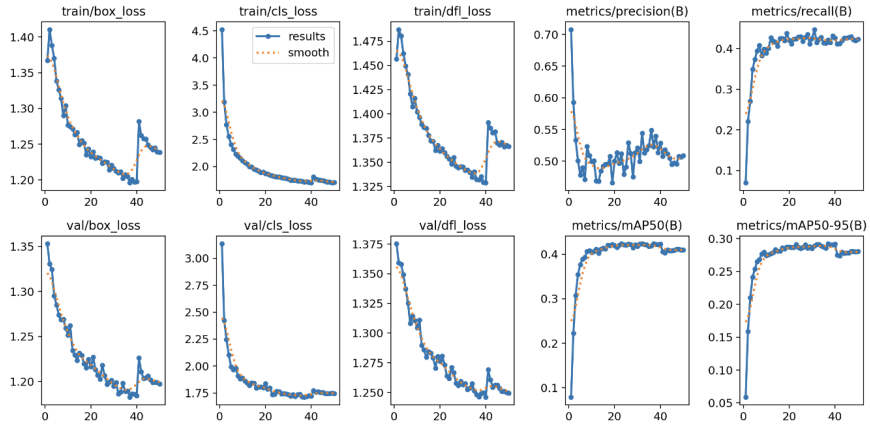


(b) YOLOv8s

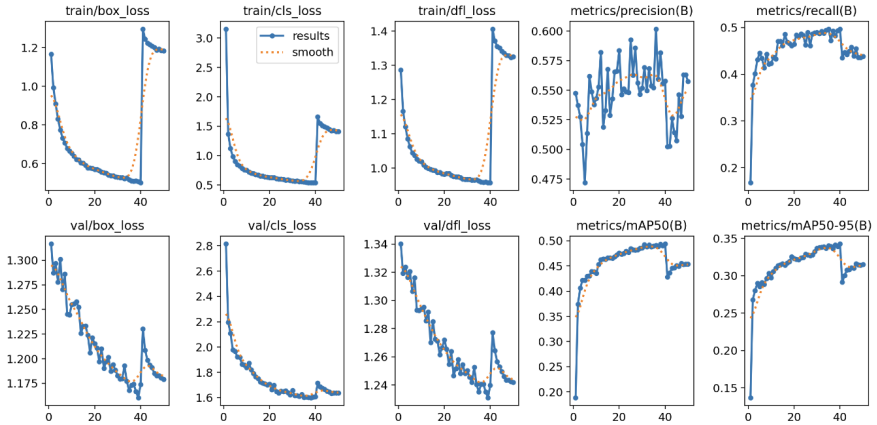


(c) YOLOv8m

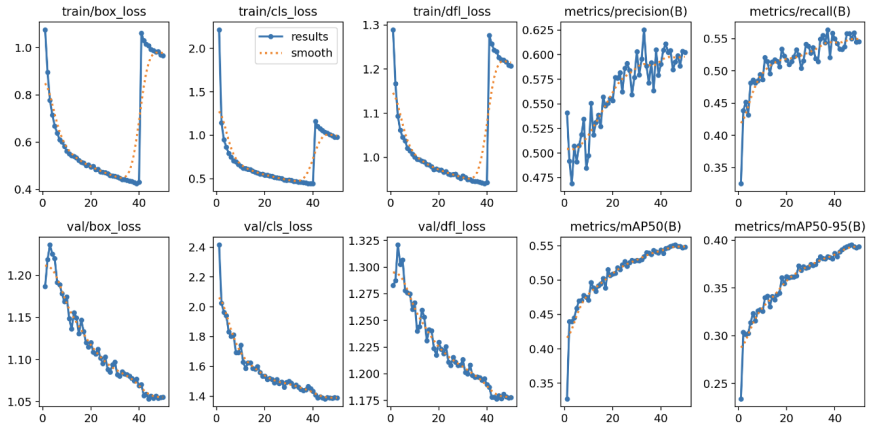
Figure 3.4: Precision-Recall curves of YOLOv8 models



(a) YOLOv8n



(b) YOLOv8s



(c) YOLOv8m

Figure 3.5: Training results of YOLOv8 models

3.3.3 Testing Examples

Based on the test images presented in Figures 3.6 and 3.7, several direct observations can be made about the performance of the YOLOv8 models:

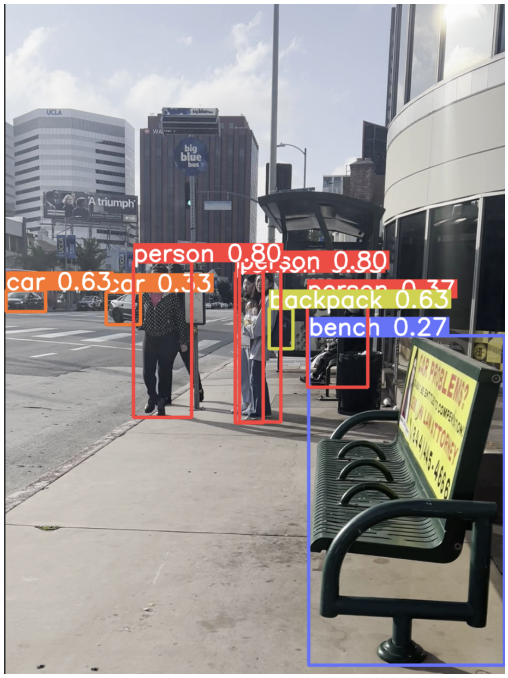
YOLOv8n: This model performs the fastest among the three but sacrifices some accuracy. For instance, in Figure 3.6(a), it correctly detects multiple "person" instances and a "bench" but misses other objects like "car" or "backpack". Similarly, in Figure 3.7(a), it identifies "person", "parking meter", and "car" but with lower confidence scores compared to the other models.

YOLOv8s: Striking a balance between speed and accuracy, YOLOv8s demonstrates a more consistent performance. As seen in Figure 3.6(b), it detects "person", "car", "backpack", and "bench" with higher confidence levels compared to YOLOv8n. Figure 3.7(b) also shows a similar trend, where YOLOv8s identifies "person", "car", "parking meter", and "street light" with moderate confidence, reflecting its ability to maintain a good trade-off between detection speed and accuracy.

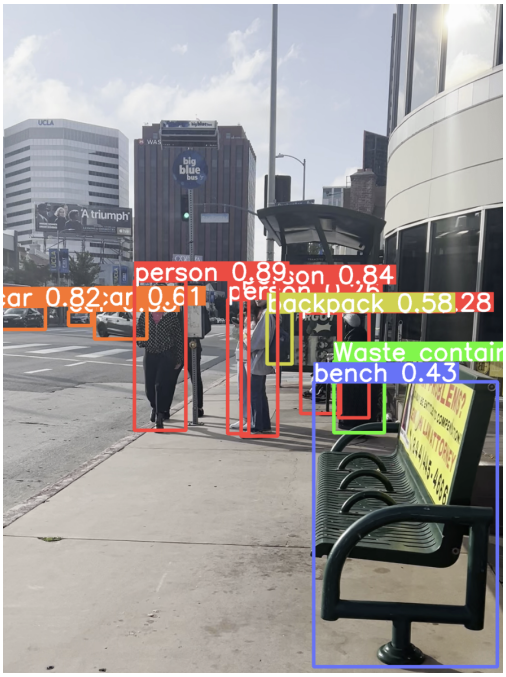
YOLOv8m: This model excels in detection accuracy but requires more processing time. In Figure 3.6(c), YOLOv8m not only detects all the objects identified by YOLOv8s but also recognizes additional items such as "waste containers" with higher confidence. Similarly, in Figure 3.7(c), YOLOv8m detects "person", "car", "parking meter", and "street light" with the highest confidence scores among the three models.

Overall, while YOLOv8m provides the most accurate detection results, its longer processing time might be a limitation for real-time applications. YOLOv8s offers a balanced approach, with a great trad-off between accuracy and speed. YOLOv8n, though the fastest, may miss some objects or detect them with lower confidence, making it less reliable for critical detection tasks.

(a) YOLOv8n



(b) YOLOv8s



(c) YOLOv8m

Figure 3.6: Test Images of YOLOv8 models

(a) YOLOv8n

(b) YOLOv8s

(c) YOLOv8m

Figure 3.7: Test Images of YOLOv8 models

CHAPTER 4

Discussion

4.1 Conclusion

Among the models evaluated, YOLOv8m stands out with the highest recall of 0.539 and a precision of 0.571, demonstrating superior performance in object detection. Despite a processing speed of 0.605 seconds per image, this rate remains within acceptable limits and suggests potential for improvement with more advanced processors. These results highlight YOLOv8m's efficiency and effectiveness, making it a reliable option for real-time applications that demand both high accuracy and reasonable processing speeds. Further enhancements in hardware could enhance its applicability in various practical scenarios, ensuring robust performance across diverse environments.

4.2 Limitation

Although YOLOv8m is the most capable of the three models in terms of detection, its precision and recall levels still fall short of practical, real-world detection tasks. There are several factors that could contribute to why deploying this model might be inadequate:

1. **Data Collection** Given that the study focuses on detecting on-road objects, the ability of the models to identify irrelevant items such as food is not essential and can be disregarded. Consequently, there is no necessity to retrain all classes from the original dataset to maintain the models' ability to recognize these unrelated objects, which

causes computational limits. Meanwhile, the amount of each new class is different, such as 11,633 trees but 1153 waste containers. The imbalance of classes will likely lead to biases in the model's predictions, where it may perform well in detecting and recognizing more frequent classes like trees, but poorly on rarer classes like waste containers.

2. **Computational Limits** Training three models on over 10,000 images across 87 classes requires substantial energy, and all training sessions were carried out on Google Colab, which limits GPU usage. As a result, the number of training epochs was reduced to 50 from the planned 100 and the hyper-parameters weren't tuned.

4.3 Future Work

To enhance the model's detection capabilities and address the limitations of training on a platform like Google Colab, the following strategies can be considered:

1. **Optimized Data Selection** Focus on a curated subset of relevant classes from the dataset, prioritizing on-road objects over irrelevant ones like food and animals. This targeted approach reduces computational demands and helps avoid over-fitting non-essential features.

2. **Class Balance** Addressing class imbalance by augmentation helps ensure the model learns to detect less frequent objects as effectively as more common ones. Augmentation techniques such as rotation, blur, and cutout on specific classes can bolster the robustness of the model, equipping the model to handle real-world variability and improving its overall detection accuracy across all classes.

3. **Hyper-Parameters Tuning** Optimizing hyper-parameters with adequate computational support significantly enhances the model's ability to accurately identify features across various classes. Fine-tuning elements such as the learning rate, the number of

epochs, and the batch size help in addressing issues like under-fitting or over-fitting. Additionally, making adjustments to the frozen layers in the context of this transfer-learning scenario allows the model to adapt and learn new features more effectively. The hyper-parameters tuning process is crucial for the model in specific tasks to improve its overall performance.

By optimizing the data selection and addressing the imbalance issue, the constraints of computational resources can be resolved, thus promising the following hyper-parameter tuning to maximize the model's effectiveness and enhance its practical deployment for real-world tasks.

4.4 Applications

Designed for real-time detection, YOLO opens new dimensions in how we interact with the world. With advancements in language models, a fine-tuned YOLO model can collaborate with these models to serve as an innovative road assistant. This system uses YOLO's output - coordinates and classes of objects - to inform a language model about the location of nearby objects, thus generating assistive messages to the users. Moreover, this message can be converted into spoken words through a text-to-speech model, enhancing situational awareness for the visually impaired. Such technology is particularly suited for devices like Vision Pro glasses, offering the visually impaired safety outdoors and freedom.

The success of the visually impaired indoor navigation system provides a promising concept that distance calculation can also be considered in outdoor applications. An indoor navigation example is illustrated in Figure 4.1 [1].

Utilizing the YOLO algorithm for object detection, combined with monocular depth estimation techniques, can significantly enhance outdoor navigation for visually impaired individuals. By implementing these technologies, the system can accurately detect and locate objects in the outdoor environment, calculate distances, and provide real-time audio

Figure 4.1: Example of indoor navigation system with distance

feedback to the user. This approach leverages YOLO's ability to quickly and efficiently identify objects, making it feasible to develop a robust and reliable navigation aid that ensures safety and independence for visually impaired people in diverse and dynamic outdoor settings.

REFERENCES

[1] Sukesh Davanthapuram, Xinrui Yu, and Jafar Saniie. Visually impaired indoor navigation using yolo based object recognition, monocular depth estimation and binaural sounds. *Embedded Computing and Signal Processing Research Laboratory*, 2022.

[2] Mohammadreza Iman, Hamid Reza Arabnia, and Khaled Rasheed. A review of deep transfer learning and recent advancements. *Technologies*, 11(2):40, 2023.

[3] Sovit Rath. Train yolov8 on custom dataset - a complete tutorial. LearnOpenCV, 2023.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, realrime object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[5] J. Terven and D.-M. Córdova-Esparza. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, 5(4):1680–1716, 2023.

[6] Fuzhen Zhuang, Zhiqiang Qi, Kai Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020.