Microsoft Word Author Guidelines for CVPR Proceedings

Cyclists Detection Using Custom YOLOv8 Model and Pre-Trained SSD MobileNetV3 Model

Liyuan Chen liyuan@stanford.edu

Abstract

This paper investigates the problem of identifying cyclists on the road and proposes a solution using computer vision techniques. With the increasing popularity of active transportation and the growing number of bike lanes, it becomes crucial to detect cyclists for the safety of drivers, passengers, and cyclists themselves. The paper focuses on utilizing customized YOLOv8 and pre-trained SSD MobileNetV3 models for cyclist detection. The video frames of road conditions serve as input, while the output comprises the coordinates of cyclists in each frame. The related work section discusses the challenges in cyclist detection and highlights the effectiveness of machine learning techniques such as ACF, DPM, R-CNN, and YOLO algorithms. The methods section outlines the use of YOLOv8 and SSD MobileNetV3 models, explaining their architectures and differences. The dataset and features section describes the dataset obtained from Kaggle, including the YOLO format labels. The experiments and results section presents the training and testing processes of the custom YOLOv8 model, achieving a precision of 0.57. Additionally, a pre-trained SSD MobileNetV3 model is employed, yielding precisions of 0.446 and 0.518 with different buffers. A comparison of the bounding boxes generated by both models is illustrated, highlighting the effectiveness of the custom YOLOv8 model in accurate cyclist detection. Overall, this research provides insights into cyclist detection using computer vision techniques and offers a promising approach for enhancing road safety.

1. Introduction

The problem under investigation involves the identification of cyclists on the road. As the government promotes active transportation and individuals increasingly opt for walking or cycling instead of driving, the number of bike lanes on roads has significantly increased. While some bike lanes have physical barriers like bollards for cyclist protection, others lack such measures. Although drivers can typically spot cyclists in front of them, it can be challenging to anticipate cyclists approaching from behind. This issue becomes particularly prominent when passengers need to exit vehicles, as open doors may obstruct cyclists. Consequently, ensuring the safety of drivers, passengers, and cyclists necessitates the ability of vehicles to detect cyclists and provide timely warnings.

To address this challenge, this paper focuses on the implementation of computer vision techniques for cyclist detection on the road. Specifically, two approaches are explored: a customized YOLOv8 (You Only Look Once Version 8) model and a pre-trained SSD (Single Shot Detector) MobileNetV3. The proposed system takes video frames of road conditions as input and produces the coordinates of cyclists present in each frame as output. By leveraging advanced computer vision algorithms, this research aims to enhance road safety by enabling vehicles to detect and respond to cyclists in real-time scenarios.

2. Related Work

Cyclist detection has been a challenge in the field of computer vision due to the lack of datasets [1]. After certain amount of data is collected, machine learning techniques such as Aggregated Channel Features (ACF), Deformable Part Models (DPM), and Region-based Convolutional Neural Networks (R-CNN) have been proved effective in detecting cyclists on road [1]. During cyclist detection, YOLO algorithm, which consists of convolutional and Maxpool layers, has been found efficient with high precision and accuracy. Due to the architecture of the YOLO algorithm, it is also fast to complete cyclist recognition [2]. In recent years, researchers have been investigating the different variations of YOLO algorithm by modifying the number and type of convolutional layers. Some of the examples are WOG-YOLO, Tiny-YOLO, and Complex-YOLO models [3-4]. Different from YOLO's DarkNet, SSD which utilizes MobileNet is another popular algorithm in object detection field [5]. SSD MobileNet has also found to be able to detect cyclists accurately in a short period of time [6]. In some cases, SSD algorithm has proven to be able to detect

cyclists with best accuracy than other mechanisms in both image and real-time detection [7].

3. Methods

This paper intends to use both YOLOv8n and SSD MobileNetV3to achieve the object detection. Being the latest version of YOLO, YOLOv8 has a larger feature map and improved convolutional network, making it more effective than previous versions of YOLO, which has also been proved in other experiments [8]. However, different from YOLO models from earlier stages, YOLOv8 utilizes anchor free detection, which predicts the centre of the object instead of the bounding box around the object, thus making the model more efficient [9]. The convolution architecture of YOLOv8 is illustrated in Figure 1.

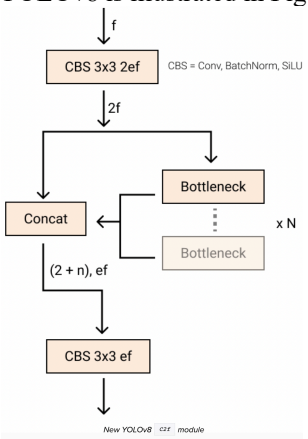

Figure 1: Convolution architecture of YOLOv8 [9]

In terms of the backbones which is the most important component of the model, YOLOv8 supports multiple different backbones including EfficientNet, ResNet, and CSPDarknet [10]. Among all the different backbones, Darknet is one of the fastest neural network frameworks for real time object detection [11]. Therefore, Darknet is chosen as the YOLOv8's backbone for this experiment.

On the other hand, SSD MobileNet, which combines the Single Shot Detector (SSD) framework with the MobileNet architecture, is a widely used object detection model that offers a balance between accuracy and computational efficiency. The SSD framework, introduced by Liu et al. in 2016 [12], is renowned for its real-time object detection capabilities. It leverages a series of convolutional layers with different scales to detect objects at various sizes and aspect ratios. The MobileNet architecture, proposed by Howard et al. in 2017 [13], is designed specifically for mobile and embedded devices, offering lightweight and efficient convolutional operations.

SSD MobileNet takes advantage of the MobileNet backbone to reduce the computational complexity of the model, making it well-suited for resource-constrained environments. The MobileNet architecture replaces the standard convolutional layers in a network with depth-wise separable convolutions, which significantly reduces the number of parameters and computations required for each convolutional operation.

Several studies have demonstrated the effectiveness of SSD MobileNet in object detection tasks. Sandler et al. applied SSD MobileNet to object detection in a real-time video stream, achieving impressive results in terms of both accuracy and speed [14]. The lightweight nature of MobileNet enables fast inference times, making it suitable for real-time applications.

In summary, SSD MobileNet combines the efficient MobileNet architecture with the SSD framework to deliver a powerful object detection model. It offers a favorable trade-off between accuracy and computational efficiency, making it particularly well-suited for real-time applications and deployment on mobile and embedded devices.

A comparison of YOLO and SSD MobileNet's architecture is illustrated in Figure 2 [15].

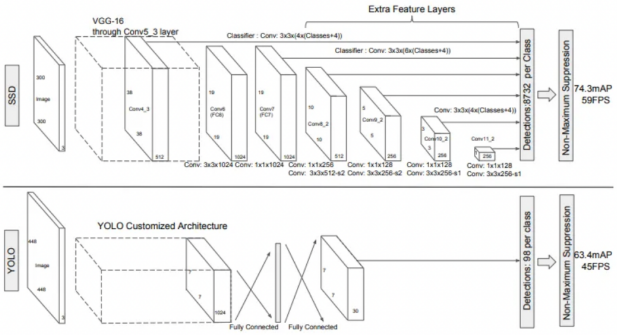

Figure 2: Model architecture for YOLO and SSD [10]

When comparing SSD MobileNet and YOLOv8, SSD MobileNet incorporates the lightweight MobileNet architecture with the SSD framework, offering a balance between accuracy and computational efficiency. On the other hand, YOLOv8 introduces architectural enhancements, including a larger feature map and anchor-free detection, to achieve improved detection performance. The choice between SSD MobileNet and YOLOv8 depends on specific requirements such as computational resources, speed, and detection accuracy, and it is crucial to consider these factors when selecting the most suitable model for a given application.

In terms of the evaluation metrics, the dataset will be run against both models and compare its detection results. The result will be measured using precision, which is defined in equation below.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

4. Dataset and Features

The dataset that will be used for this project is obtained from Kaggle [16] where video frames were taken on road conditions. Cyclists were identified and their coordinates in each frame were provided as labels. The labels were in YOLO format which consists of object id (object class),

relative coordinate of centre of x with respect to the picture width, relative coordinate of centre of y with respect to the picture height, relative width of the bounding box with respect to the picture width, and relative height of the bounding box with respect to the picture height.

Due to the GPU restriction, 4000 pictures and their labels were used as training set to train the model, and another 1000 pictures and their labels were used as test set.

5. Experiments and Results

A custom YOLOv8 model was trained and validated using the dataset. 80% of the data was used to train the YOLO model whereas 20% of the data was used to test the model's accuracy. Due to the constraint on GPU and RAM, the model was only trained for two epochs, which took approximately two and half hours, where the inputs of the model are the images and the label of cyclist location in the image in YOLO format. The images in the test set were then fed into the custom model and the output of the model was the location of cyclists in the image in YOLO format. It took the model a few seconds to take the input images and to detect the cyclists. The two labels were compared to obtain the accuracy of the model. The confusion matrix of the YOLOv8 model result on test set is presented in Figure 3.

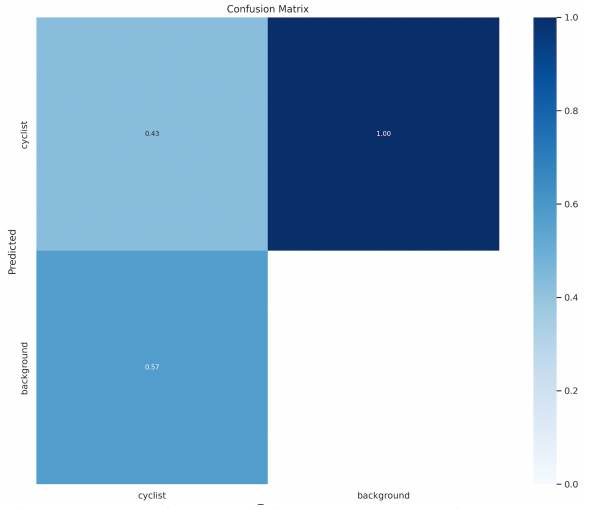

Figure 3: Confusion matrix for YOLOv8 (epoch = 2)

From the confusion matrix, it can be seen that the custom YOLOv8 model has a precision of 0.57.

To compare, a pre-trained SSD MobileNet model, trained on a large COCO dataset, was also utilized for cyclist detection. Since the pre-trained model did not include the specific class "cyclist," the class "person" was used instead. The labels produced by the SSD MobileNet model were in the format of minimum and maximum coordinates of the bounding boxes. To align with the YOLO format, the bounding box coordinates were converted by calculating the relative coordinates of the center and dimensions with respect to the image size.

To assess the precision of the SSD MobileNet model, the coordinates of the bounding box centers generated by the model were compared with the centers indicated in the ground truth labels, using buffers of 0.05 and 0.1. If the difference between the estimated and actual bounding box centers fell within the buffer, the detection was considered accurate. Table **1** summarizes the precision results for both models.

Table 1: Precision of custom YOLOv8 model and the pre-trained SSD MobileNetV3 model

Model	Precision
YOLOv8	0.570
Pre-Trained SSD MobileNetV3 (Buffer = 5%)	0.446
Pre-Trained SSD MobileNetV3 (Buffer = 10%)	0.518

From Table **1**, it can be seen that YOLOv8 model has the highest precision of 0.570, followed by Pre-Trained SSD MobileNetV3 with buffer = 0.1 and buffer = 0.05. This shows that the custom YOLOv8 model is able to produce the most accurate detection.

Figure 4 illustrates the bounding boxes produced by the custom YOLOv8 model and the pre-trained SSD MobileNetV3 model overlaid on the same image. The green bounding box represents the box generated by the SSD MobileNetV3 model, which focuses solely on the person. In contrast, the blue bounding box generated by YOLOv8 includes the cyclist and the bicycle. Despite the size difference between the bounding boxes, both models accurately detect the presence of cyclists.

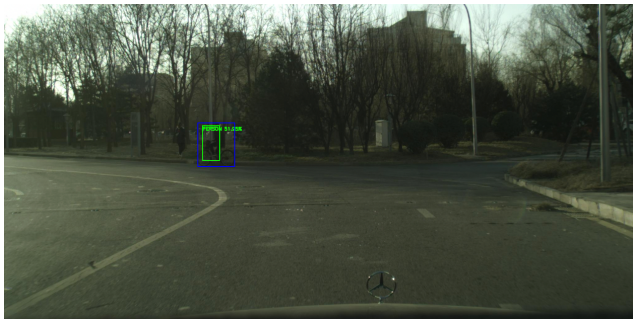

Figure 4: Bounding boxes produced by custom YOLOv8 model and pre-trained SSD MobileNetV3 model

These results highlight the superior performance of the custom YOLOv8 model in accurately detecting cyclists, as

it recognizes and includes both the person and the bicycle in the bounding box, while the pre-trained SSD MobileNetV3 model solely focuses on human detection.

6. Conclusions

In this study, we have addressed the problem of cyclist detection on the road using computer vision techniques. With the increasing adoption of active transportation and the need for improved road safety, accurately identifying cyclists becomes crucial. Through the utilization of customized YOLOv8 and pre-trained SSD MobileNetV3 models, we have demonstrated the effectiveness of these algorithms in detecting cyclists in video frames.

The experiments have shown that the custom YOLOv8 model outperformed the pre-trained SSD MobileNetV3 model in terms of precision, achieving a precision of 0.57. The YOLOv8 model successfully detected both the cyclist and the bicycle, whereas the SSD MobileNetV3 model primarily focused on identifying humans. Despite some differences in the bounding box sizes, both models demonstrated the capability to accurately detect cyclists in the given context.

While this study has made significant progress in cyclist detection using computer vision techniques, there are several avenues for future exploration and improvement. Some potential areas of future work include expanding the dataset used for training the models could enhance their performance and generalization capabilities. Incorporating a wider range of road conditions, lighting conditions, and cyclist behaviors would enable the models to better adapt to real-world scenarios. The models can be further optimized to improve accuracy and efficiency. Exploring different architectural variations, incorporating advanced feature extraction techniques, or fine-tuning the models on specific cyclist detection tasks could yield better results.

The research can also be extended to real-time implementation would be valuable for developing practical applications. Real-time detection could be achieved by leveraging hardware acceleration techniques, optimizing model inference speed, or exploring more lightweight architectures. Most importantly, the cyclist detection models could be integrated into the vehicle systems as a test. Integrating the cyclist detection models with vehicle systems, such as advanced driver assistance systems (ADAS) or autonomous driving systems, could enhance safety on the road. By providing real-time warnings or alerts to drivers and autonomous vehicles, potential collisions or conflicts with cyclists could be avoided.

In conclusion, this study has demonstrated the effectiveness of customized YOLOv8 and pre-trained SSD MobileNetV3 models for cyclist detection. Future work includes dataset expansion, model optimization, real-time implementation, and integration with vehicle systems would contribute to further advancements in cyclist detection and improve road safety for all road users.

References

[1] Xiaofei Li et al., "A new benchmark for vision-based cyclist detection," 2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg, Sweden, 2016, pp. 1028-1033, doi: 10.1109/IVS.2016.7535515.

[2] Saranya, K. C., Thangavelu, A., Chidambaram, A., Arumugam, S., & Govindraj, S. (2019). Cyclist detection using Tiny Yolo V2. Advances in Intelligent Systems and Computing, 969–979. https://doi.org/10.1007/978-981-15-0184-5_82

[3] Xu, L., Yan, W., & Ji, J. (2023). The research of a novel Wog-Yolo algorithm for autonomous driving object detection. Scientific Reports, 13(1). https://doi.org/10.1038/s41598-023-30409-1

[4] Dazlee, N. M., Khalil, S. A., Abdul-Rahman, S., & Mutalib, S. (2022). Object detection for autonomous vehicles with sensor-based technology using Yolo. International Journal of Intelligent Systems and Applications in Engineering, 10(1), 129–134. https://doi.org/10.18201/ijisae.2022.276

[5] García-Venegas, M., Mercado-Ravell, D. A., Pinedo-Sánchez, L. A., & Carballo-Monsivais, C. A. (2021). On the safety of vulnerable road users by cyclist detection and tracking. Machine Vision and Applications, 32(5). https://doi.org/10.1007/s00138-021-01231-4

[6] Tang, C. (2023, January 16). A semi-trailer truck right-hook turn blind spot alert system for detecting vulnerable road users using transfer learning. arXiv.org. https://arxiv.org/abs/2303.11223

[7] Mulyanto, A., Borman, R. I., Prasetyawan, P., Jatmiko, W., & Mursanto, P. (2019). Real-time human detection and tracking using two sequential frames for Advanced Driver Assistance System. 2019 3rd International Conference on Informatics and Computational Sciences (ICICoS). https://doi.org/10.1109/icicos48119.2019.8982396

[8] Aboah, A., Wang, B., Bagci, U., & Adu-Gyamfi, Y. (2023). Real-Time Multi-Class Helmet Violation Detection Using Few-Shot Data Sampling Technique and YOLOv8. https://doi.org/ https://doi.org/10.48550/arXiv.2304.08256

[9] Solawetz, J. (2023, January 25). What is Yolov8? the ultimate guide. Roboflow Blog. https://blog.roboflow.com/whats-new-in-yolov8/

[10] Mehra, A. (2023, May 26). Understanding yolov8 architecture, applications & features. Labellerr. https://www.labellerr.com/blog/understanding-yolov8-architecture-applications-features/

[11] Khare, T. (2020, June 10). Custom object detection using darknet. Medium. https://towardsdatascience.com/custom-object-detection-using-darknet-9779170faca2#:~:text=What%20is%20Darknet%3F,can%20be%20used%20for%20images).

[12] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single Shot

MultiBox Detector. In European Conference on Computer Vision (pp. 21-37). Springer.

[13] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv preprint arXiv:1704.04861.

[14] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4510-4520).

[15] Cochard, D. (2021, May 25). MobilenetSSD : A machine learning model for fast object detection. Medium. https://medium.com/axinc-ai/mobilenetssd-a-machine-learning-model-for-fast-object-detection-37352ce6da7d

[16] SemiEmptyGlass. (2022, March 15). Cyclist dataset for object detection. Kaggle. https://www.kaggle.com/datasets/semiemptyglass/cyclist-dataset