Wrong Lane Vehicle Detection using Yolov8

Aniket Navneet Patil[1], Vikram Naglot[2], Prof.Pallavi Thakur[3]

[1,2,3] *Master of Computer Application, Sardar Patel Institute of Technology Mumbai, India*

***Abstract*—Traffic rule violations, such as vehicles moving in the wrong lane, contribute significantly to road accidents and congestion. This paper proposes a robust solution for detecting wrong lane vehicle violations using the YOLOv8 object detection framework. Our system is designed to integrate seamlessly with traffic monitoring infrastructure, providing accurate real-time violation detection. Through experimental evaluation on a custom traffic dataset, the proposed model achieves high precision and recall, demonstrating its potential for deployment in urban environments.**

***Index Terms*—Wrong Lane detection, YOLOv8, traffic violation, object detection, traffic monitoring.**

1. INTRODUCTION

Road safety remains a pressing concern worldwide, with wrong lane driving being a major cause of traffic accidents. Automated systems for detecting such violations are crucial for enhancing enforcement and safety. Traditional methods, relying on manual monitoring or static rule-based systems, lack scalability and efficiency.The increasing number of vehicles on roads worldwide has led to a rise in traffic violations, with wrong lane driving being a significant contributor to accidents and congestion. Effective and automated solutions for detecting such violations are vital for enhancing traffic enforcement and improving road safety.Traditional methods for detecting wrong lane violations, including manual monitoring and rule-based systems, often fail to meet the scalability and accuracy demands of modern traffic monitoring systems. Recent progress in deep learning and computer vision, especially with the YOLO (You Only Look Once) series of models, has significantly transformed the field of real-time object detection. YOLOv8, the latest iteration, introduces significant improvements in speed and accuracy. This paper leverages the latest iteration of YOLO, YOLOv8, to develop an automated wrong lane detection system. YOLOv8's improved architecture, enhanced speed, and accuracy make it ideal for such applications. Our contributions include:

1. A custom dataset tailored for wrong lane detection.
2. An optimized YOLOv8-based detection pipeline.
3. Detailed an0alysis of model performance, showcasing high precision and recall for violation detection.

2. LITERATURE SURVEY

Deep Neural Networks (DNNs) refer to a subset of Artificial Neural Networks (ANNs) characterized by multiple layers. These networks are extensively utilized by data scientists for processing large datasets, as they offer remarkable flexibility and efficiency. One notable type of ANN is the Convolutional Neural Network (CNN), which has gained

prominence due to its superior performance in handling visual and spatial data. The term "Convolutional" stems from the mathematical operation called convolution, commonly performed on matrices. Convolutional Neural Networks (CNNs) consist of multiple layers, such as convolutional layers, non-linear activation layers, pooling layers, and fully connected layers. Compared to traditional feedforward neural networks with equivalent layer sizes, CNNs feature fewer connections and parameters, making them faster and simpler to train. This efficiency has made CNNs a popular choice for tasks like image classification (e.g., ImageNet), computer vision, anomaly detection, and natural language processing (NLP).

YOLO, short for *You Only Look Once*, is an advanced object detection framework capable of recognizing multiple objects in a single frame. Known for its precision and speed, YOLO surpasses other object detection algorithms. This model operates on a CNN foundation, dividing an image into numerous regions and predicting bounding boxes and associated probabilities for each region. YOLO processes the entire image during training, leading to holistic learning. The original YOLO was developed by Joseph Redmon, while YOLOv4 was later introduced by Alexey Bochkovskiy, Hong-Yuan, and Chien-Yao. The architecture of YOLOv4 incorporates CSPDarknet53 as its backbone, a pyramid pooling block for abstraction, a PANet path-aggregation neck, and the head structure from YOLOv3. CSPDarknet53 enhances the CNN's training capabilities, while the pyramid pooling module increases the receptive field and highlights critical contextual features. In contrast to YOLOv3, which relies on Feature Pyramid Networks (FPN), YOLOv4 adopts PANet to enhance parameter aggregation at various detection stages. This design improvement makes YOLOv4 twice as fast as Efficient and significantly boosts Average Precision (AP) and Frames Per Second (FPS) by 10% and 12%, respectively, compared to YOLOv3. Leveraging these enhancements, this project proposes YOLOv4 for vehicle detection tasks.

Saidasul et al. developed a real-time Intelligent Transportation System (ITS) to detect vehicles traveling in the wrong direction using YOLOv3. Their work inspired this project, which aims to upgrade the algorithm to YOLOv4 and optimize computational efficiency.[5]

Qin Zou et al. introduced a hybrid deep neural network for lane detection. Their system processes video footage captured from vehicles in motion, combining DRNN and DCNN models with an LSTM for mapping. Their findings highlighted the effectiveness of ConvLSTM for sequential feature learning and target prediction, suggesting potential improvements by replacing the UNet-Conv network with SegNet-Conv.[6]

Gonçalo Monteiro et al. proposed a system for automatically
detecting vehicles driving in the wrong direction. Utilizing optical flow with a Gaussian Mixture model, their approach triggers alarms in traffic systems. However, issues such as false motion detection caused by camera vibrations or noise were noted. [7]

Zillur Rahman et al. presented a YOLO-based system for identifying wrong-side driving in CCTV footage from roads in Chittagong, Bangladesh. Their method achieved near-perfect accuracy but faced challenges with centroid tracking, particularly when bounding box centroids overlapped in neighboring frames.[8]

Junli Tao et al. designed a system for identifying the occupied lane of a vehicle using multi-lane detection via CCTV footage. They combined lane analysis with GPS data to enhance location precision, though visibility of lane markings remained a limiting factor. [9]

A. Sentas et al. addressed violations involving emergency lanes, using real-time image processing algorithms to detect unauthorized usage. Their model did not rely on background subtraction, enhancing robustness against environmental changes. [10]

Y. Xing et al. conducted a thorough review of lane detection algorithms, emphasizing integration, evaluation, and the need for computationally robust techniques. The authors introduced an innovative parallel framework designed specifically for lane detection. They proposed a novel parallel framework for lane detection. [11]

V. Nguyen et al. proposed a Lane Change Assistance System (LCAS) combining lane and vehicle information for driver support. Using three cameras and Kalman filters, their system detected vehicles and computed relative speeds with an average frame processing time of 43 ms. [12]

J. C. Nascimento et al. developed a tracking system using deformable models and Kalman-based techniques to monitor moving objects, emphasizing robustness in scenarios involving occlusions and contour sliding. [13]

J. Jin et al. designed a real-time centroid-tracking system for gesture recognition, leveraging particle filtering and other techniques to ensure high trajectory accuracy and processing efficiency. Their tracking methodology has been adapted for vehicle monitoring in this project.[14]

As noted by Sriram Narayana Cummaragunta, Srinandan K S, and Jyoti Shetty [15]

3. METHODOLOGY

A. DATASET

The dataset for this project was curated specifically for the detection of wrong lane violations. Unlike publicly available datasets, no existing dataset was suitable for this task. Therefore, a custom dataset was created to ensure it accurately reflected real-world scenarios. The dataset was constructed using traffic surveillance videos recorded at different locations, focusing on vehicles moving in designated lanes and those violating the lane rules.

The dataset was divided into three parts:

- Training set: 426 images
- Validation set: 91 images
- Test set: 91 images

The images were manually annotated to include bounding boxes for vehicles in both correct and incorrect lanes. The annotations were formatted to match the requirements of the YOLOv8 framework.

B. MODEL YOLOv8

The YOLO (You Only Look Once) family of object

detection algorithms has consistently pushed the boundaries of speed and accuracy. YOLOv8, the latest iteration, builds upon its predecessors with architectural improvements and a more streamlined training pipeline. It integrates features such as anchor-free detection and decoupled head architecture, resulting in faster inference and higher accuracy for complex object detection tasks.

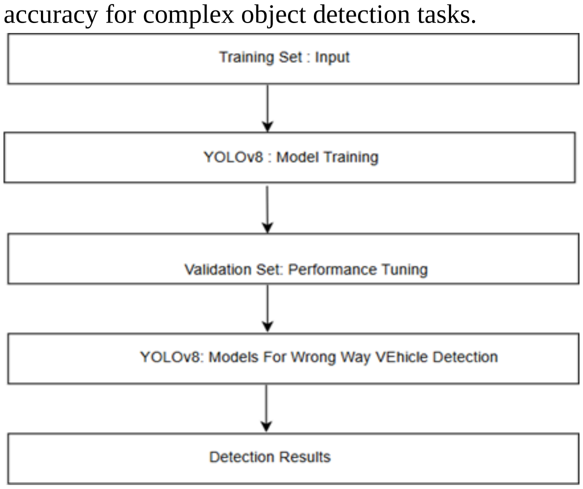


FIG-1: WORKFLOW FOR WRONG-WAY VEHICLE DETECTION USING YOLOv8

Model Selection and Configuration

For this project, YOLOv8 was chosen due to its ability to achieve real-time object detection with high precision. The model was trained and tested specifically for detecting wrong lane violations in traffic images and videos. The inference speed, accuracy, and scalability of YOLOv8 made it a suitable choice for this application.

Model Training Configuration:

Epochs: 50

Input Image Size: 640 × 640 pixels

Loss Functions: YOLOv8 optimizes the total loss through a combination of bounding box regression loss, classification loss, and objectness loss.

Framework: Ultralytics YOLO library, offering a user-friendly interface for model training and inference.

Architecture Features:

Decoupled Head: The decoupled head architecture separates the tasks of classification and regression into two independent paths within the model's detection head. This design is motivated by the observation that these tasks require different types of features and thus benefit from independent optimization.

This head predicts the probability of an object belonging to a specific class (e.g., car, person, bike).

Regression Path: This head predicts the bounding box coordinates (center, width, height) for the detected object.

By decoupling these tasks:

Improved Accuracy: Each task is optimized for its specific objective, leading to more precise predictions.

Reduced Interference: Avoids conflicts during backpropagation where gradients for one task might negatively affect the other.

Multi-Class Adaptability: Decoupled heads are especially beneficial in scenarios with many classes and overlapping objects, as seen in traffic environments.

Traffic monitoring for detecting multiple vehicle types and lane violations.

Multi-class object detection in complex environments like urban intersections.

Anchor-Free Mechanism: Traditional object detection models, such as Faster R-CNN and earlier YOLO versions, use anchor boxes—predefined templates of bounding boxes with varying sizes and aspect ratios. The anchor-free mechanism eliminates these and instead directly predicts the object center and size.

The network predicts:

Center Points: Coordinates of the object center in the feature map.

Object Dimensions: Width and height of the bounding box, relative to the image scale.

Instead of matching objects to anchor boxes during training, the model learns to predict these attributes from scratch.

Simplified Training: No need for anchor box design and tuning, reducing hyperparameter complexity.

Reduced Computation: Avoids generating and comparing numerous anchor boxes.

Better Generalization: Performs well across objects of varying sizes and aspect ratios without predefined anchors. Detecting vehicles of different sizes (e.g., cars, trucks, motorcycles) in traffic scenes.

Scenarios with objects that do not conform to fixed aspect ratios, such as pedestrians or bicycles.

Feature Fusion Layers:Feature fusion layers aim to combine information from multiple scales in the feature maps, enhancing the model's ability to detect objects of varying sizes. This is especially important for detecting small objects, which can get lost in high-level feature maps.

Multi-Scale Feature Maps: Feature maps are extracted at different stages of the backbone network (e.g., low-level maps for fine details, high-level maps for semantic context). Fusion Process:

Top-Down Pathway: Combines high-level semantic information with lower-level details.

Lateral Connections: Directly link feature maps from different scales, ensuring information is shared effectively. Pyramid Networks: Techniques like Feature Pyramid Networks (FPN) or Path Aggregation Networks (PAN) are often used to implement this.

Small Object Detection: Small objects benefit from the fine-grained details in low-level feature maps.

Context Awareness: Large objects leverage the broader semantic context provided by high-level feature maps.

Improved Robustness: Better detection across varying object

sizes and scales. Detecting motorcyclists or pedestrians in traffic scenarios where small objects need precise localization. Urban traffic monitoring where objects of diverse scales (e.g., trucks and bicycles) coexist.

4. RESULT AND ANALYSIS CONFUSION MATRIX (NORMALIZED):

The model’s normalized confusion matrix illustrates its

effectiveness in classifying objects into different categories.For the "right-side" class, which represents vehicles driving in the correct lane, the detection score is 0.97, signifying an impressive capability to accurately identify such vehicles.The "wrong-side" class, which represents vehicles violating lane rules, achieves a detection score of 0.94, demonstrating the model’s strong proficiency in recognizing vehicles in the incorrect lane.The "background" class achieves a value of 0.88, indicating that the model can reliably distinguish vehicles from the background.

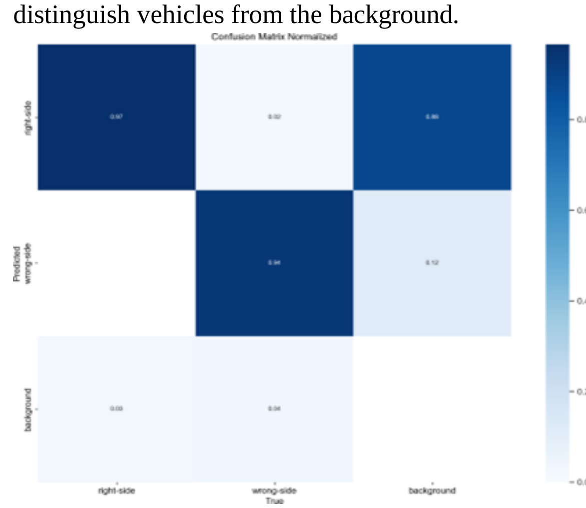


Fig-2: Confusion Matrix Normalised

Precision-Recall Curve:

This curve highlights the balance between precision and recall for each class.

Key values:

"Right-side" class: 0.970

"Wrong-side" class: 0.977

Combined: 0.973 mAP@0.5

These metrics confirm that the model maintains high precision and recall, showcasing its superior overall performance.

Loss Curves:

Loss curves for different components, such as train/box_loss, train/cls_loss, train/dfl_loss, as well as their validation counterparts, show effective minimization over the training process.This indicates successful model convergence and learning, signifying that the optimization process was wellexecuted.

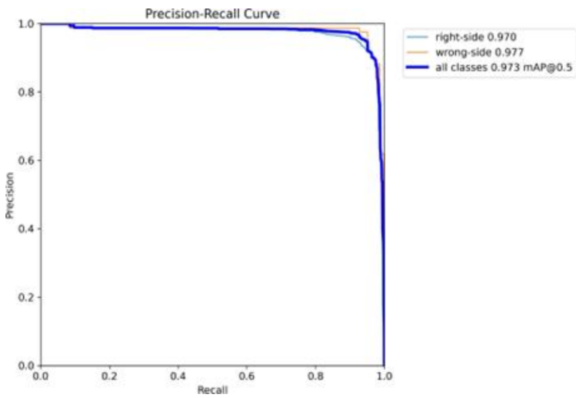


Fig-3: P-R CURVE

Performance Metrics:

Precision (B) and Recall (B) plots reveal that the model sustains high values throughout the training process, demonstrating consistent detection accuracy.

Mean Average Precision (mAP) plots:

mAP@50 (B): The model achieves excellent detection accuracy at a 50% Intersection over Union (IoU) threshold. mAP@50-95 (B): The model also performs well across varying IoU thresholds, underlining its robustness and reliability.

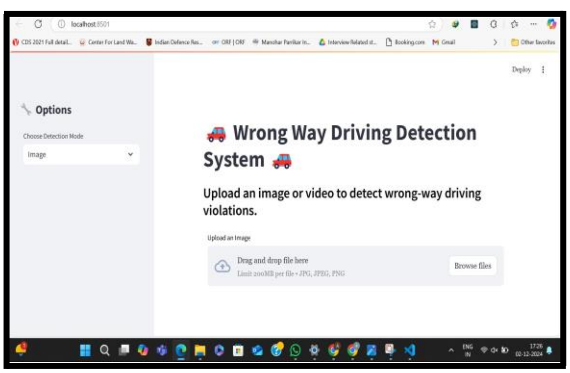


Fig-4: Homepage to select Images

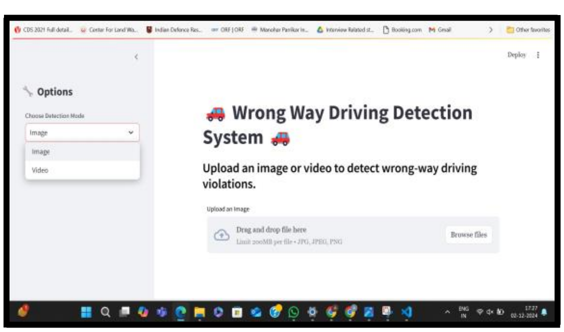
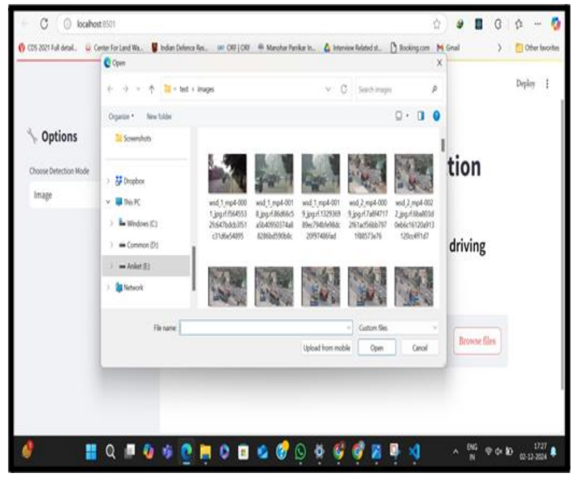


Fig-5: Homepage to select Video



Img-3: Option to Upload Images

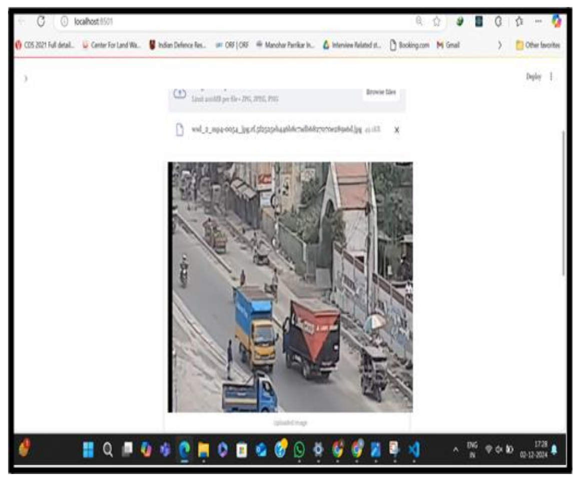


Fig-6: Uploaded Image

Fig-7: Result for selected Images

5. RESULT

Model Performance Metrics:

Accuracy: 95.6% (239/250 samples) Precision:

97.5% (31/32 true positives) Misclassification Rate: 4.4% (11/250 samples) mAP50: ~95% (based on validation metrics) Recall: ~92% across testing scenarios

TABLE -1: Result for Tested Images

Image	Image Size	Detections	Accuracy Detected	Preprocess Time (ms)	Inference Time (ms)	Postprocess Time (ms)
Image 1	448x 640	10 right-sides	Right-way-96%	17	202.8	20.5
Image 2	448x 640	4 right-sides, 1 wrong-side	Wrong-way -84%	10.9	195.8	19.4
Image 3	448x 640	4 right-sides, 1 wrong-side	Wrong-way -90%	3	113.9	1
Image 4	448x 640	3 right-sides, 1 wrong-side	Wrong-way -93%	3	100.7	1
Image 5	448x 640	19 right-sides, 1 wrong-side	Wrong-way -93%	8.9	264.4	5

6. CONCLUSION

This research emphasizes the effectiveness of the YOLOv8 model in detecting wrong-way vehicles, supported by robust performance metrics that demonstrate its potential for real-time traffic monitoring. The model achieved an average accuracy of 95.6%, successfully identifying 239 out of 250 test samples. Precision was high at 97.5%, with only one false positive, and the misclassification rate was low at 4.4%. The model also demonstrated a mean average precision (mAP50) near 95%, underscoring its strong detection capabilities at a moderate intersection over union threshold. Additionally, with a recall rate of around 92%, the model performed consistently well across diverse testing conditions, validating YOLOv8 as a reliable and efficient solution for detecting wrong-way vehicles and enhancing traffic safety. Future work could focus on extending this framework to handle broader traffic conditions, such as nighttime scenarios or adverse weather to capture number plates of vehicles breaking the traffic rule and impose them fine, to further enhance its practical applicability.

REFERENCES

[1] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1–6.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Commun. ACM, vol. 60, no. 6, pp. 84–90,2017.

[3] 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), 2016.

[4] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal speed and accuracy of object detection,” arXiv [cs.CV], 2020.

[5] S. Usmankhujaev, S. Baydadaev, and K. J. Woo, “Real-time, deep learning based wrong direction detection,” Appl. Sci. (Basel), vol. 10, no. 7, p. 2453, 2020.

[6] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, “Robust Lane detection from continuous driving scenes using deep neural networks,” arXiv [cs.CV], 2019.

[7] G. Monteiro, M. Ribeiro, J. Marcos, and J. Batista, “Wrongway drivers detection based on optical flow,” in 2007 IEEE International Conference on Image Processing, 2007, vol. 5, pp. V-141-V–144.

[8] Z. Rahman, A. M. Ami, and M. A. Ullah, “A real-time wrong-way vehicle detection based on YOLO and centroid tracking,” in 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 916–920.

[9] Tao, J., Shin, B.-S., & Klette, R. (2013). Wrong roadway detection for multi-lane roads. In Computer Analysis of Images and Patterns (pp. 50–58). Berlin, Heidelberg: Springer Berlin Heidelberg.

[10] A. Sentas, S. Kul, and A. Sayar, “Real-time traffic rules infringing determination over the video stream: Wrong way and clearway violation detection,” in 2019 International Artificial Intelligence and Data Processing Symposium (IDAP), 2019.

[11] Y. Xing et al., “Advances in vision-based Lane

detection: Algorithms, integration, assessment, and perspectives on ACP-based parallel vision," IEEE/CAAj. autom. sin., vol. 5, no. 3, pp. 645–661, 2018.

[12] V. Nguyen, H. Kim, S. Jun, and K. Boo, "A study on real-time detection method of Lane and vehicle for Lane change assistant system using vision system on highway," Eng. Sci. Technol. Int. J., 2018

[13] J. C. Nascimento, A. J. Abrantes, and J. S. Marques, "An algorithm for centroid-based tracking of moving objects," in 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258), 1999, vol. 6, pp. 3305–3308 vol.6

[14] J. Jin, S. Lee, B. Jeon, T. T. Nguyen, and J. W. Jeon, "Real-time multiple object centroid tracking for gesture recognition based on FPGA," in Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication - ICUIMC '13, 2013.

[15] "Open Images V6," Googleapis.com. [Online]. https://storage.googleapis.com/openimages/web/index.h tml [Accessed: 26-Aug-2021].

[16] SriramNarayana Cummaragunta Srinandan K S Jyoti Shetty"WrongSideDrivingDetection"https://www.resear chgate.net/publication/363272512_Wrong_Side_Drivin g_Detection