Traffic Violence Detection Using Deep Learning

Phan Phuoc Loc Ngoc [1,2], Le Tan Loc [1,2],
[1]Faculty of Information Science and Engineering
[2]University of Information Technology
Vietnam National University, Ho Chi Minh City, Vietnam

Abstract

Automatic traffic monitoring systems in our country are rapidly evolving and becoming an important trend for enhancing traffic safety and managing vehicles more effectively in crowded cities. Common traffic violations, such as not wearing helmets, overloading, or running red lights, remain persistent issues that demand accurate and timely detection and resolution.

In this study, we develop a traffic observation system using deep learning models to detect and classify traffic violations. The YOLOv8 model is deployed to identify vehicles and recognize license plates from real-time video data. Additionally, the system integrates the ConvNeXt model to classify two-wheeler violations, including not wearing helmets and overloading.

Furthermore, the system employs a tracking algorithm named DeepSORT to label and track vehicles across video frames, ensuring the highest accuracy in processing data streams. Experimental results on real-world datasets demonstrate the system's outstanding performance in object detection, traffic violation recognition, and license plate processing. This research contributes significantly to the development of modern smart cities while improving traffic law compliance in critical scenarios.

1 Introduction

Traffic violations are one of the primary causes of severe accidents worldwide. Common violations, such as not wearing helmets, running red lights, and overloading, not only breach traffic laws but also negatively impact road safety and public traffic culture. In rapidly developing countries like Vietnam, implementing effective traffic management and monitoring solutions faces challenges due to resource limitations and the accuracy of traditional systems.

Currently, most traffic monitoring systems rely heavily on human intervention, leading to inconsistent detection and delayed handling of violations. Additionally, sensor-based and traditional camera-based technologies encounter similar challenges when processing complex traffic data, especially in real-time object detection.

The primary goal of this research is to develop an automatic traffic monitoring system that leverages modern deep learning technologies to overcome the limitations of traditional systems. The system is designed to address the following tasks:

1. Detect vehicles and recognize license plates using the YOLOv8 model.
2. Classify two-wheeler traffic violations, focusing on not wearing helmets and overloading.
3. Integrate tracking algorithms to analyze real-time video data streams.

2 Related Work

Research on vehicle detection and license plate recognition systems is a significant focus within the application of deep learning in computer vision. Several studies, such as Tonge et al. (2020), have employed the YOLO model combined with OCR to detect vehicles and recognize license plates from video data. This system automates traffic monitoring processes and records traffic violations, especially in complex traffic scenarios. Ellahyani et al. (2016) proposed a method based on image features combined with Random Forest (RF) to detect and recognize traffic signs. Although this approach achieved high accuracy in experimental environments, it is not optimized for real-time applications due to processing speed limitations. Asadianfam et al. (2020) introduced a system leveraging MapReduce and deep learning to handle large-scale traffic data, improving vehicle detection performance in densely populated cities. The YOLO model, particularly YOLOv8, used in this study, has demonstrated superior performance in real-time traffic object detection with high accuracy. By integrating OCR and YOLO, the system achieves precise license plate extraction from video data.

Classifying traffic violations, especially those involving two-wheelers, has become a prominent topic in recent research. Sharma and Jamwal (2023) presented a study utilizing CNNs to detect traffic violations such as not wearing helmets and running red lights. Their model achieved high accuracy in small-scale experimental datasets but has yet to be integrated into real-time monitoring systems. Dokhe et al. (2023) implemented the YOLOv5 model to detect vehicles using GIS and GTSRB datasets, focusing on vehicle classification and license plate recognition, proving effective in real-world traffic conditions. In this research, we utilize ConvNeXt to accurately classify common violations such as not wearing helmets and overloading. The model is trained on extensive traffic datasets to ensure generalization and scalability.

Real-time tracking and analysis of traffic violations pose significant challenges, especially in congested environments. Sargar et al. (2023) integrated YOLO and DeepSORT to track vehicles in real-time video streams. Their system assigns unique IDs to each vehicle, tracking their location and speed to detect abnormal behaviors. Asadianfam et al. (2020) successfully developed a traffic data analysis system using MapReduce to process large-scale data streams from surveillance cameras, enhancing urban traffic management capabilities. In this research, DeepSORT is employed to label and track vehicles frame-by-frame, ensuring thorough analysis and improving system accuracy.

License plate recognition plays a critical role in automatic traffic monitoring systems. Qin (2020) applied deep learning techniques to recognize license plates in crowded traffic environments, achieving high accuracy and processing speeds. Fleyeh and Dougherty (2005) proposed a method to recognize license plates from images, but its processing limitations made it unsuitable for real-time applications. In this study, Tesseract OCR is used to extract license plate information after detection by YOLOv8, facilitating storage and automated violation notifications.

After reviewing numerous studies and scientific papers, which often focus on a single aspect such as object detection or license plate recognition, this research integrates state-of-the-art technologies, including YOLOv8, ConvNeXt, and DeepSORT, to develop a comprehensive automatic traffic monitoring system. This system not only detects violations but also tracks behaviors and processes real-time traffic data, paving the way for deployment in smart cities with dense traffic.

3 Proposed Method

The proposed method includes the following main modules:

- Vehicle detection module.
- License plate recognition module.
- Two-wheeler violation detection module.
- Violation counting module.

As shown in Figure 1, the workflow begins with a video input. Bounding boxes of vehicles are detected and extracted using the YOLOv8 model. If the vehicles contain license plates, the bounding boxes of the license plates are further extracted using YOLOv8. For violation detection, the study focuses on two-wheelers and classifies violations, including wearing helmets, not wearing helmets, or overloading. Finally, a tracking algorithm is applied to the video to detect and count violations, providing the final output.

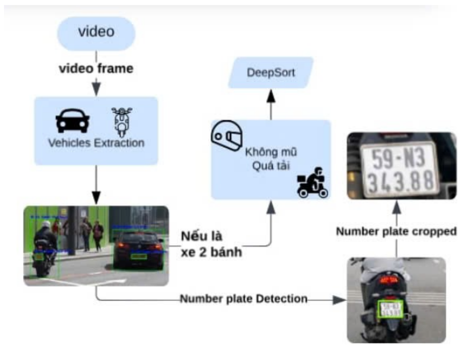

Figure 1: Processing workflow including all modules.

3.1 Vehicle Detection

To detect vehicles in videos, the YOLO model, trained on our collected dataset, is utilized. YOLO (You Only Look Once) is an object detection algorithm that directly predicts the coordinates of bounding boxes using fully connected layers built on convolutional feature extractors. By predicting offsets instead of coordinates, the algorithm simplifies the problem, making the network easier to train. YOLO outperforms other object detection algorithms like R-CNN, Fast R-CNN, and Faster R-CNN, which involve multi-step pipelines for object detection.

YOLOv8, the latest version in the YOLO series, is used in this study due to its significant improvements in performance and accuracy. YOLOv8 incorporates advanced techniques, such as a lightweight architecture, faster training, and better generalization across diverse datasets. This version integrates improvements in feature pyramid processing and uses modern optimization strategies to enhance prediction capabilities, especially for small or overlapping objects. These enhancements make YOLOv8 an optimal choice for high-performance real-time object detection and tracking applications. The model outputs bounding boxes and labels for detected vehicles in the images.

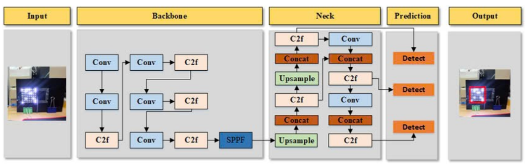

Figure 2: YOLOv8 architecture.

3.2 License Plate Recognition

For license plate recognition, the YOLOv8 model is retrained on a custom dataset. The bounding boxes generated by the vehicle detection module are used as input to train the model. The output includes bounding boxes corresponding to license plate regions of the detected vehicles.

3.3 Two-Wheeler Violation Detection

To detect violations involving two-wheelers, the bounding boxes generated by the vehicle detection module are filtered to include only two-wheeler labels, excluding other objects. These bounding boxes are then input into the ConvNeXt classification model. ConvNeXt is a modern Convolutional Neural Network (CNN) architecture designed to harness the potential of CNNs in the context of Vision Transformer (ViT)-dominant tasks. ConvNeXt inherits core ideas from ResNet architecture while incorporating technical improvements inspired by Transformers, thereby enhancing performance for complex tasks.

Key features of ConvNeXt:

- **Simple and efficient architecture:** ConvNeXt retains the simplicity of traditional convolutional networks but refines components, such as using layer normalization instead of batch normalization, removing bottleneck layers, and adjusting kernel sizes.

- **Optimized performance:** The model integrates modern techniques, including stochastic depth, GELU activation function, and drop path, to enhance training and overall performance.

- **Scalability:** ConvNeXt scales well to suit different computational requirements, from real-time applications to complex tasks demanding high accuracy.

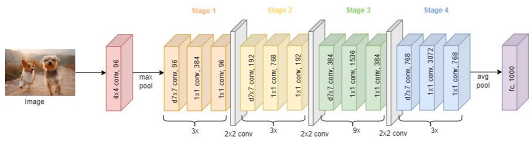

Figure 3: ConvNeXt architecture.

3.4 Violation Counting

After using ConvNeXt to detect and classify violations, such as not wearing helmets or overloading, DeepSORT is applied to track and count the number of violating vehicles. DeepSORT (Deep Simple Online and Realtime Tracking) extends the SORT algorithm by leveraging deep learning to improve object recognition and tracking, especially in scenarios involving overlapping objects or complex motion patterns.

The main components of DeepSORT include:

- **Kalman Filter:** Predicts object positions in subsequent frames based on information from previous frames.

- **Hungarian Algorithm:** Solves the optimization problem of matching predictions with detected objects in the current frame.

- **Appearance Descriptor:** Unlike SORT, DeepSORT uses a deep learning-based feature extractor to improve the accuracy of object matching, even when objects leave the frame temporarily.

The process of applying DeepSORT to count violations is as follows: bounding boxes of vehicles detected by YOLOv8 are passed to the ConvNeXt model for violation classification. Once violating vehicles are identified, DeepSORT assigns unique IDs to each object across video frames, ensuring each vehicle is counted only once. The number of violations is displayed in the processed video output as the final result.

Figure 4: Final output using DeepSORT.

4 Dataset

4.1 Dataset for Vehicle Detection

The dataset used to train the YOLOv8 model was aggregated from three different datasets available on Kaggle. These datasets contain diverse images of vehicles captured from various angles, including close-up images taken by mobile phones and overhead views captured by traffic cameras. This ensures that the YOLOv8 model is trained to detect vehicles in a wide variety of scenarios. Each image is accompanied by a '.txt' file containing labels and bounding box coordinates of vehicles. The dataset structure was organized to be compatible with YOLOv8 training requirements. The final dataset includes a total of 9 classes: auto, bus, car, lcv, motorcycle, multiaxle, tractor, truck, and ambulance, with a total of 2500 images.

Figure 5: The dataset includes various angles of vehicles.

4.2 Dataset for License Plate Recognition

The dataset for license plate recognition was also sourced from Kaggle. It includes two types of license plates commonly found in Vietnam: single-line and double-line text formats. The dataset contains 5000 images, each with bounding box annotations for license plate regions. The images include both clear and blurry cases to enhance the model's generalization and effectiveness.

Figure 6: The dataset includes various types of license plates and angles.

4.3 Dataset for Two-Wheeler Violation Classification

The dataset for classifying two-wheeler violations was collected from Kaggle. It includes images of motorcycles and riders captured from various angles. The dataset covers three violation categories: wearing helmets, not wearing helmets, and overloading. It contains a total of 2500 images, each annotated with its respective labels.

Figure 7: The dataset for two-wheeler violations collected from Kaggle.

5 Experiments

5.1 Experimental Settings

5.1.1 YOLOv8

For training the YOLOv8 model for vehicle detection, we configured basic parameters such as epoch = 20, batch size = 16, and image size normalized to 640x640 to align with the YOLOv8 requirements. To enhance input diversity, data augmentation techniques were applied with different parameter settings during the training process for both vehicle detection and license plate recognition tasks.

5.1.2 ConvNeXt

In this experiment, we utilized the ConvNeXt-Small model with pretrained weights from the ConvNeXt_Small_Weights.DEFAULT. This advanced architecture is efficient for feature extraction and was customized to suit a three-class classification task. Specifically, the output classification layer of the model was modified from 768 dimensions (default) to 3 dimensions. The loss function used in the experiment was CrossEntropyLoss, which is appropriate for multi-class classification problems. The chosen optimization algorithm was AdamW, with a learning rate set to 2e-4 and a weight decay coefficient of 0.05.

5.2 Results

5.3 Vehicle Detection and License Plate Recognition

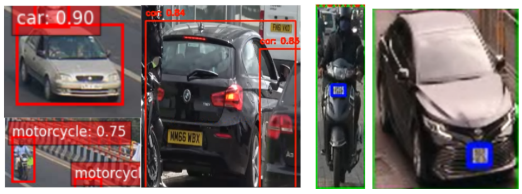

Figure 8: Results of vehicle detection (left) and license plate recognition (right).

The results of the detection module shown in Table 2 indicate that the accuracy for vehicle detection is approximately 0.74, while license plate recognition achieves 0.98. The relatively lower accuracy for vehicle detection can be attributed to the dataset used for training, which was aggregated from three sources with varying angles and vehicle types, leading to a lack of consistency and imbalance in the classes. However, the model performs acceptably well on clear images. On the other hand, license plate recognition achieves high accuracy due to the balanced dataset and clearer image quality.

Model	P	R	mAP50	mAP50-95
Vehicle Detection	0.744	0.746	0.793	0.551
Plate Recognition	0.982	0.988	0.994	0.901

Bảng 2: Performance metrics for Vehicle Detection and Plate Recognition.

5.4 Two-Wheeler Violation Detection

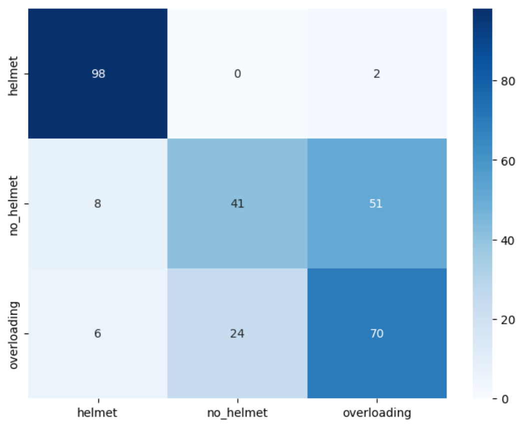

Figure 9: Confusion matrix for the classification task.

The performance metrics for the ConvNeXt model after training on the dataset are shown in Table 3.

Hyperparameter	Vehicle Detection	Plate Recognition
Mosaic	1.0	1.0
HSV Hue (hsv_h)	0.015	0.015
HSV Saturation (hsv_s)	0.7	0.7
HSV Brightness (hsv_v)	0.4	0.4
Degrees (Image Rotation)	10.0	10.0
Translate	0.1	0.1
Scale	0.5	0.5
Shear	0.1	0.1

Bảng 1: Hyperparameters for Vehicle Detection and Plate Recognition tasks.

The results, with an average accuracy of approximately 0.69, indicate that the model performs at an acceptable level. The confusion matrix (Figure 9) reveals that for riders wearing helmets, the model achieves high prediction accuracy. However, the accuracy for the ‘no_helmet‘ and ‘overloading‘ classes is relatively low. This can be attributed to the dataset’s limited diversity in these two classes and potential ambiguity in labeling. For instance, riders who are both not wearing helmets and overloading may be randomly labeled as either ‘no_helmet‘ or ‘overloading‘, leading to confusion during training and prediction.

Model	Accuracy	P	R	F1-score
ConvNeXt	0.6967	0.6916	0.6967	0.6831

Bảng 3: Performance metrics for ConvNeXt.

5.5 Violation Counting

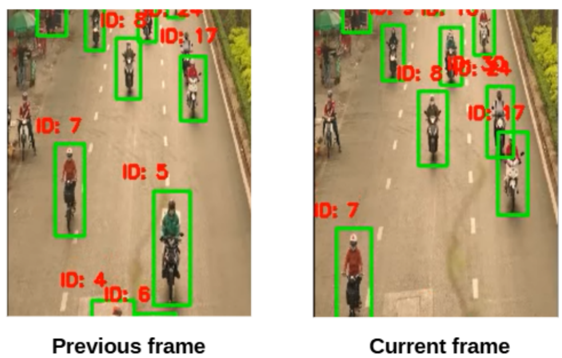

Figure 10: Results of DeepSORT algorithm after assigning IDs to objects in the video.

After using YOLOv8 to detect violating vehicles, the system filters for two-wheelers and applies the DeepSORT algorithm to assign unique IDs to each motorcycle. The pre-trained ConvNeXt model is then used to classify violations for these two-

Figure 11: Final results using the DeepSORT algorithm.

wheelers. If a vehicle is classified as violating (e.g., not wearing a helmet or overloading), the violation is highlighted in the processed video, as shown in Figure 11. For high-quality videos with fixed overhead angles, the algorithm performs well with high accuracy. However, the performance deteriorates for videos with unstable angles, blurry objects, or rapid movement, resulting in less reliable predictions. The results are also influenced by the ConvNeXt model’s classification performance, which struggles with blurry images, further affecting the overall accuracy.

6 Conclusion

We have successfully implemented a framework for the problem of traffic rules violation detection, achieving key objectives such as developing models for vehicle and license plate detection. Additionally, we classified traffic violations for two-wheelers and counted the number of violating vehicles in video data. However, limitations in the dataset and video quality led to some cases where the results were less accurate.

Our team plans to further enhance the system by integrating additional modules to handle a broader range of tasks and improve the overall performance of the framework.

References

Shiva Asadianfam, Farzan Ashrafi, and Reza Faraji. 2020. Tvd-mrdl: Traffic violation detection system using mapreduce-based deep learning for large-scale data. *Multimedia Tools and Applications*, 80(14):21651–21668.

Ilyas El Jaafari Ayoub Ellahyani, Mohamed El Ansari and Said Charfi. 2016. Traffic sign detection and recognition using features combination and random forests. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 7(1):686–693.

Rahul Bhakad, Kunal Gaikwad, and Shantanu Bawankule. 2022. Traffic sign detection and recognition using deep learning. *International Journal of Engineering Research & Technology (IJERT)*, 11(3):295–298.

Karthikeyan D, Enitha C, Bharathi S, and Durka Devi K. 2020. Traffic sign detection and recognition using image processing. *International Journal of Engineering Research & Technology (IJERT)*, Special Issue - 2020.

Divya A. Dhokchoukle Dipti D. Dokhe, Kaveri S. More and Gayatri D. Karanjkar. 2023. Traffic sign detection and recognition using yolov5 and gis datasets. *International Journal of Research Publication and Reviews*, 4(5):559–592.

Dipti D. Dokhe, Kaveri S. More, Divya A. Dhokchaule, and Gayatri D. Karanjkar. 2023. Traffic sign detection and recognition using yolov5 and gtsrb dataset. *International Journal of Research Publication and Reviews*, 4(5):5589–5592.

Ayoub Ellahyani, Mohamed El Ansari, Ilyas El Jaafari, and Said Charfi. 2016. Traffic sign detection and recognition using features combination and random forests. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 7(1):686–693.

Hasan Fleyeh and Mark Dougherty. 2005. Road and traffic sign detection and recognition. In *Advanced OR and AI Methods in Transportation*, pages 644–653.

Priyanka Goyal, Aman Gupta, and Arjun Singh. 2023. Detection of traffic rule violation in university campus using deep learning. *International Journal of System Assurance Engineering and Management*, 14(1):45–53.

Bharathi S Karthikeyan D, Enitha C and Durka Devi K. 2020. Traffic sign detection and recognition using image processing. *International Journal of Engineering Research Technology (IJERT), Special Issue*.

Koteswara Rao Kodepogu, Vijaya Bharathi Manjeti, Mareedu Divya, Madireddy Anusha, Mannem Jaswanth, and Kancharla Pawan Kumar. 2023. Traffic sign detection and recognition by using auto encoder deep learning classification models. *Scope*, 13(4):1388–1400.

Mohana, A. Bhavani, and P. Gokul. 2020. Traffic signal violation detection using artificial intelligence and deep learning. *2020 International Conference on Communication and Signal Processing (ICCSP)*, pages 361–365.

Zhongbing Qin. 2020. Traffic-sign recognition using deep learning. Master's thesis, Auckland University of Technology, School of Engineering, Computer & Mathematical Sciences.

Kunal Gaikwad Rahul Bhakdad and Shantanu Bawanule. 2022. Traffic sign detection and recognition using deep learning. *International Journal of Engineering Research Technology (IJERT)*, 11(3):295–298.

Omkar Sargar, Siddhesh Shinde, Neel Gokhale, and Pooja Patil. 2023. Traffic monitoring and violation detection using deep learning. In *Proceedings of the 4th International Conference on Computing Communication and Signal Processing*, pages 201–214. Springer.

Harikesh Kumar Sharma and Anupama Jamwal. 2023. Traffic sign recognition system using cnn. *International Journal of Innovative Science and Research Technology*, 8(11):339–345.

Aniruddha Tonge, Shashank Chandak, Renuka Khiste, Usman Khan, and L. A. Bewoor. 2020. Traffic rules violation detection using deep learning. *Fourth International Conference on Electronics, Communication and Aerospace Technology (ICECA-2020)*, pages 1250–1257.

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. 2020. Resnest: Split-attention networks. *arXiv preprint arXiv:2003.03256*.

Barret Zoph, Vijay Vasudevan, Jonathan Shlens, and Quoc V. Le. 2018. Learning transferable architectures for scalable image recognition. *arXiv preprint arXiv:1808.10524*.