SvedbergOpen
DISSEMINATION OF KNOWLEDGE
International Journal of Artificial Intelligence and Machine Learning
Publisher's Home Page: https://www.svedbergopen.com/
Research Paper Open Access

Developing AI-Powered Solutions For Traffic Analysis Using Yolov8 For Object Detection And Textual Descriptions Generation

Deepali Suryawanshi [1] , Dr. Dnyaneshwar Ahire[2]

[1]Department of Electronics and Telecommunications Matoshri College of Engineering and research Center Nasik, India
deepalis94@gmail.com
[2]Department of Electronics and Telecommunications Matoshri College of Engineering and research Center Nasik,
Indiadda007132@gmail.com

Abstract: There is a need for creative traffic monitoring due to the rise in traffic jams and accidents. Conventional approaches are insufficient because they lack precision and speed in real-time and typically involve human interaction. YOLOv8, the most sophisticated iteration of the "You Only Look Once" object identification model, will be used in this work to develop a machine-learning model that can identify traffic. Using traffic images, the model can identify and categorize traffic items, produce textual descriptions, and provide real-time insights for improved traffic control. There were 17 distinct object classes in the custom traffic dataset, including vehicles, buses, and people. Additionally, data augmentation methods such as blurring and grayscale transformations were used to increase the model's resilience under various circumstances. To achieve remarkable performance metrics at mAP50 of 0.385 and mAP50-95 of 0.236, YOLOv8 was further optimized for over 30 epochs. While analyzing photos at a real-time speed of 30 to 50 milliseconds, the system demonstrated a high degree of accuracy in identifying and categorizing important traffic aspects. The model effectively detects objects like cars, buses, and trucks with processing times between 59.8 ms and 103.3 ms per image, which makes it suitable for real-time use. It also generates descriptive text from detected classes, enabling automated traffic analysis. By providing useful insights, this skill improves safety management, urban planning, and traffic monitoring. The suggested method demonstrates that deep learning and computer vision can be used to create automated, scalable traffic systems that address contemporary urban issues.

Keywords: YOLOv8, Artificial Intelligence (AI), Traffic Detection, Textual Description, Object Detection.

I. Introduction

In recent years, road accidents in Bangladesh have taken several lives. The Bangladesh Jatri Kalyan Samity and BRTA claim that several individuals are killed in traffic accidents each year. High accident rates, lengthy travel times, and traffic congestion have been increasing due to the growing population, the number of cars, and the lack of infrastructure. Due to their limited coverage, generally inaccurate procedure, and inability to create real-time data, traditional traffic control approaches that depend only on human participation and simple devices have become more impractical[1].

To overcome these issues, smart solutions based on deep learning and computer vision are becoming important to upgrade traffic monitoring systems. The latest YOLOv8 is a version of the YOLO (You Only Look Once) model[2], offering important improvements in terms of speed, accuracy, and robustness, which make it very suitable for real-time traffic analysis[3]. This study aims to develop an algorithm for generating textual information from still traffic images using AI techniques, specifically through the object detection capabilities of YOLOv8. This new approach promises to enhance the management of traffic by offering real-time data automatically, which will improve the efficiency and reliability of monitoring systems[4].

The study employs a systematic approach and begins with the processing of a custom traffic dataset that is annotated with images[5]. Bounding boxes and class labels are the two kinds of visualization strategies that improve the understanding of the detection results. During training, data augmentation techniques such as blurring and grayscale transformations are used to improve the adaptability of the model to different environments. The model is further refined in several epochs to improve performance for the unique features of a traffic environment. The class is extracted, and the id of the variety class is counted for the detected class along with converting the image into a text description[6].

To examine the efficacy of object detection, this study assesses the model's performance using metrics such as precision, recall, and mean Average Precision (mAP). By providing object counts and converting bounding box detections into text descriptions, the metrics improve in understanding the model's capacity to distinguish between classes, such as cars, buses, and people, while reducing false positives and negatives[7]. It displays the image dimensions, the total number of class counts, and the speed per image both before and after the text description was converted. A more thorough examination of these indicators shows areas for development and confirms the model's efficacy.[8][9].

The objective of this study is to develop an AI-based model that generates textual descriptions using traffic images. Utilizing the YOLOv8 model to identify and classify objects and classes, such as cars and humans, it computes the number of classes in total. In addition, it translates images representing visual data into text, thus giving an actual meaningful description for better traffic analysis and surveillance. This study is important to the development of automated traffic systems as it covers the gap between visual and textual data. The ability of this model to give accurate real-time descriptions supports traffic management, enhances safety monitoring, and supports urban planning to offer valuable understandings of the decision-making process.

II. Related Work

Many studies have been conducted on assistive technologies for the visually impaired, focusing on object detection, navigation, and contextual information generation. Parwateeswar Gollapalli et al. [10] proposed real-time vehicle speed estimation and counting of vehicles using YOLOv8 with Deep SORT multi-object tracking. The Deep SORT model applies accurate tracking by correlating detected objects across video frames based on features of the Dense ReID. Using real-world traffic data from varying complexity, the developed system was evaluated for results above 80% in classifying vehicles and estimating the speed of the vehicle from the dataset. This YOLOv8 with Deep SORT, therefore, gives a robust yet cost-effective solution for monitoring and management in high-speed traffic conditions.

Table 1: Various studies of different authors on this research.

Ref.	Method	Description	Limitations
Saif B. Neamah and Abdulamir A. Karim[11]	YOLOv8 algorithm for object detection	This paper discusses real-time traffic monitoring using the YOLOv8 algorithm and deep learning for vehicle detection, classification, segmentation, tracking, speed and size estimation, and counting.	The illumination and occlusions used in this investigation may have affected the system's accuracy, particularly when it came to size estimates.
Gupta et al.[12]	Vehicle identification and a correlation tracker are two applications of the Haar cascade technique.	It utilized a video-based method by Haar cascade for detection and correlation tracker to measure speed by counting pixel divisions in a video feed per recorded time to measure PPM.	The system suffers from heavy traffic, lacks the estimation of vehicle size, and lacks a counting feature.
Shihab, Ghani, and Mohammed. [13]	Two methods based onthe Haar cascade and YOLOv3	A vehicle detection and classification system was implemented using the techniques of the Haar cascade and YOLOv3 where YOLOv3 performs way better than the Haar cascade for	The system supports multi-object detection and classification but lacks tracking, counting, speed,

		high accuracy and robustness, especially under different lighting conditions.	and size estimation features.
J Lee, KA Cha and M Lee. [14]	YOLOv5, cautionary sentence generation with KoAlpaca	The system for the visually impaired, which incorporates YOLOv5 for object recognition and KoAlpaca for cautionary sentence generation, is enhanced by image augmentation and GPT-based language training.	It needs additional validation and the addition of dynamic objects to better generalize because of its small sample size, controlled environment, and emphasis on static classes.
Ruixin Zhao et al. [15]	Z-YOLOv8	It presents Z-YOLOv8s, enhancing accuracy and efficiency under uncertainties by integrating RepViT, LSKNet, SPD-Conv, and SoftPool-SPPF. It improves mAP@0.5 on BDD100K and KITTI datasets.	It restricts deployment to devices with limited resources and requires additional optimization for actual traffic situations.

The study's gap is in addressing the limitations of current systems, which do not include size estimations, counting, or appropriate handling of dynamic objects in the real world. It is necessary to further optimize models like YOLOv8 with Deep SORT to create a resilient system in complicated traffic circumstances and to deploy it efficiently on devices with limited resources. To expand the generalization of systems for dynamic conditions and validate their performance in diverse real-world settings, more investigation is required.

III. Methodology

1. The Proposed Model

This research methodology provides a systematic approach to the development and evaluation of amodel for the use of AI techniques to generate textual information from still traffic photos. The procedure consists of six essential steps that are intended to guarantee that every step is executed precisely to achieve reliable object detection, strong training, and efficient result visualization. The following diagram shows the flow of the proposed model.

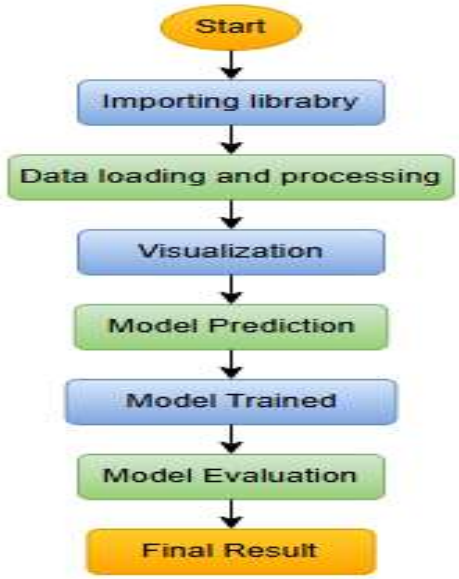

Figure 1: Flow diagram of the proposed model.

The following are the steps of proposed model are as follows:

1. **Import library:** The first step in the implementation is importing the necessary Python modules that allow for object identification, data management, and visualization. To perform object identification, the Ultralytics library offers the YOLOv8 model and related tools. The libraries used include matplotlib and pandas for data processing and visualization, and cv2 and PIL for handling images and creating bounding boxes. These provide a basis for the model functionality.
2. **Data loading and processing:**Images and label files from the directories are used in the study as part of a structured dataset.The model uses custom traffic dataset images forthe loading and processing of the images.To determine the actual reality of items that are detected, images and their labels are taken into a system. To demonstrate the detection result, a random selection of 16 images is used for illustration purposes. Providing the dataset for both training and inference tasks is ensured by this process.
3. **Visualization:** The approach employs OpenCV to create bounding boxes around the identified items in the pictures to enhance interpretability. The detected items, including vehicles, buses, and people, are identified by labels attached to these bounding boundaries. A clear visual depiction of the detections is provided by Matplotlib, which is used to illustrate the results. Sample images from the dataset are displayed to show that the model works in practical situations.
4. **Model Prediction:** The traffic image initial detection is performed using a pre-trained YOLOv8 variant. The previously mentioned approach works effectively with objects in dynamic and complex contexts, such as traffic situations. After putting the images through the model, the method produces bounding boxes, class labels, and discovered objects as a reliable starting point for additional analysis.
5. **Model Trained:**Fine-tuning for the custom traffic dataset was achieved by training the YOLOv8 framework over 30 epochs. By transforming the data using blurring and grayscale, the generalization across various training conditions was improved. The key performance indicators during training were precision, recall, and mAP, which enabled to monitoringof the model's development and improved performance.
6. **Model Evaluation and Result:**The detection metrics for the trained model are given in the last phase. Metrics are calculated on class, like the detection accuracy of specific items, like vehicles, buses, and people. A complete analysis of the performance of the model is also given by computing average precision and recall values. These measurements not only confirm the model effectiveness but also point out areas that still require work.

This structured methodology ensures the difficult testing and optimization of the model in generating reliable textual descriptions of traffic images toward furthering the goals of automated traffic analysis and monitoring.

2. Feature extraction and text generation of the trained model

The methodology optimizes the proposed model through the development of an efficient system to detect and analyze objects in still images using YOLOv8. It starts by importing necessary libraries for loading and processing images, the Python function is used for displaying inputs, YOLO from the ultralytics library for object detection, and Counter from collections for counting the occurrence of objects.The primary purpose of image analysis is to improve the process. It takes an image file path as input loads the image using the function, and displays it for visual confirmation. The YOLOv8 model is loaded with pre-trained weights optimized for lightweight operations. Object detection is performed on the input image, resulting in the output that consists of bounding boxes, class IDs, and confidence scores.

The detected class IDs are mapped to human-readable class names using a dictionary called CLASS_NAMES_DICT, and the occurrences are counted using Counter. The information is formatted into a text summary for easy interpretation, like "car: 3, person: 1". The function returns both a dictionary of detected objects with their counts and the textual summary. This simple script enhances the proposed model by enabling automatic image analysis, making it ideal for traffic monitoring, object counting, and tracking, and it produces results that are easy to read.

Performance Evaluation metrics:

In this research study, several equations can be used to describe important aspects of the YOLOv8 model, which are detection, loss functions, and performance metrics. Some of the most relevant key equations in this regard are given below:

1. Mean Average Precision (mAP) Calculation:

It is a common metric used in object detection models to estimate its performance. The mAP is calculated as the average of the AP over all classes.

$$\text{mAP} = \frac{1}{N}\sum_{i=1}^{N} \text{AP}_i \quad \ldots\ldots\ldots\ldots (1)$$

Where:
N is the number of classes.
AP_i is the Average Precision for class *i*, which is calculated by:

$$\text{AP}_i = \int_0^1 \text{Precision}(r)\text{dRecall}(r) \quad \ldots\ldots\ldots\ldots (2)$$

2. Precision and Recall:

Precision and recall are the measures to assess the performance of the model in classification. Precision is the percentage of true positives compared to total predicted positives. Recall is the ratio of true positive detections against the total actual positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \quad \ldots\ldots\ldots\ldots (3)$$

Where:
TP = True Positives (correctly predicted objects)
FP = False Positives (incorrectly predicted objects)

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \quad \ldots\ldots\ldots\ldots (4)$$

Where:
FN = False Negatives (missed objects)

3. Loss Function

YOLO models usually adopt a combined loss function that includes classification, localization or bounding box prediction, and confidence score.

$$\text{Loss} = \lambda_{\text{obj}} \cdot \text{Loss}_{\text{obj}} + \lambda_{\text{cls}} \cdot \text{Loss}_{\text{cls}} + \lambda_{\text{bbox}} \cdot \text{Loss}_{\text{bbox}} \quad \ldots\ldots\ldots\ldots (5)$$

These equations help to validate the fundamental procedures and metrics evaluated in the research, based on the model's detection and classification capabilities.

IV. Results

It illustrates the dataset, the outcome of the proposed model, which uses YOLOv8 to display the class bounding boxes, and the feature extraction and text generation of the trained model results, which displays class detection, counts the number of distinct classes, and converts the image into a text description.

Dataset (Custom traffic Dataset):

The dataset comprises labeled images of vehicles and number plates, prepared using LabelImg in YOLOv5 format. Around 1,000 training images were collected from traffic scenes, vehicles, number plates, and CCTV footage. Frames were extracted from videos using OpenCV, and bounding boxes were drawn around objects of interest, labeled with their respective classes. LabelImg, an open-source Python-based graphical labeling tool, was used for annotation.The images were labeled under different classes. The dataset ensures an extensive range for object detection by including 17 different classes of traffic-related objects. These classes include Three Wheelers, Bicycle, Bus, Car, Minibus, Minivan, Motorbike, Pickup, Police Car, Rickshaw, Scooter, SUVs, Taxi, CNG, Truck, Van, and Wheelbarrow. This diversified categorization ensures a comprehensive dataset capable of capturing various elements of a traffic scene, enhancing the model's ability to detect and classify objects effectively.

The proposed model result:

The study produced significant results in different performance metrics and stages of analysis. Input image dimensions were standardized at 640x229 with three color channels to maintain consistency in processing. Results of object detection showed the model's capability to correctly identify objects such as cars, buses, persons, and trucks with great speed and accuracy.

The yolov8x model with 68,172,831 parameters is used, updated for 21 classes from the dataset. The training data consists of 2,704 images, with 300 for validation, where duplicate labels were removed during preprocessing. The initial performance was low mAP50 (0.187 in epoch 1), improving to 0.234 by epoch 5. The input image size was adjusted from 229 to 256 to match the model's stride, and labeling consistency was emphasized to prevent issues. The AdamW optimizer, with appropriate learning rates, supports the large-scale model. Data augmentations like blurring, grayscale, CLAHE and flipping enhance the model's robustness.

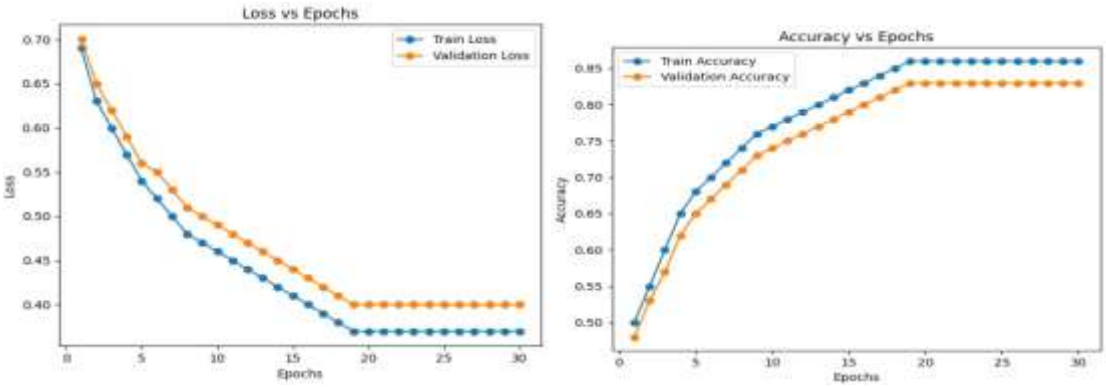

Figure 2: Comparison of loss vs epochs and accuracy vs epochs.

The comparison of training loss and validation loss vs. epochs in Figure 2 illustrates that validation loss is greater than training loss. Training accuracy is higher than validation accuracy, according to the comparison of accuracy vs. epochs in training accuracy and validation accuracy.

At a training epoch of 30, the YOLOv8 framework attained mAP at 0.385 in mAP50 and at 0.236 in mAP50-95. Therefore, it indicates effective multiple object class detection ability. The model's best accuracy was seen for important key classes such as buses, cars, and three-wheelers, which reflects robust performance in different scenarios.

Progressive model optimization during training is shown by comprehensive performance metrics that demonstrate epoch-wise loss reduction in classification, bounding box regression, and other relevant loss categories. Additionally, the model maintained an inference speed of roughly 30 to 50 milliseconds per image, guaranteeing real-time applicability for applications involving traffic monitoring.

Figure 3: Detected images using Yolov8 model.

In Figure3, the images represent the object detection capability of the YOLOv8 model. In the first image, there are bounding boxes and confidence scores over the detected traffic entities like buses, cars, and pedestrians. In the second image, trucks and cars have been identified with high confidence levels. These results indicate the accuracy of the model and its ability to detect and classify multiple objects in real time from diverse traffic scenes.

Feature Extraction and text generation results:

The table shows the feature extraction result of the input image using the proposed trained model.

Table 2: Various input images of consumed time and detected classes.

Sr. no.	Image size (pixel)	Detected classes	Time (ms)
1	448x640	43 cars, 2 trucks, and 1 bus	65.8
2	384x640	1 umbrella and 2 people	69.3
3	448x640	14 cars, 3 motorcycles, 1 bus, and 1 truck	103.3
4	384x640	2 cars	59.8

Figure 4: Detection of classes and text description result

Figure 4 shows the model being applied to an input image with 448x640 pixels in size. It correctly identified 43 cars, 2 trucks, and 1 bus. The processing times are 2.1 ms for preprocessing, 65.8 ms for inference, and 0.8 ms for postprocessing, demonstrating efficient and rapid object detection.

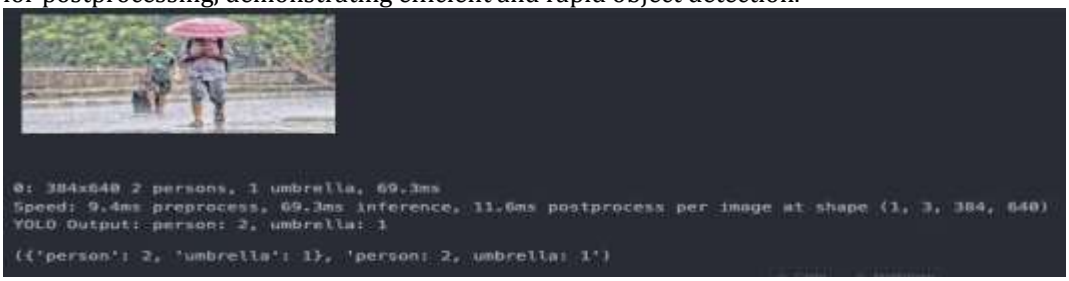

Figure 5: Detection of classes with other input images.

Figure 5 illustrates how this 384x640 pixel input image is being used by the model. It recognized 1 umbrella and 2 people. The object or class identification takes 69.3 ms. After the image is converted to a text description, the preprocessing time is 9.4 ms and the postprocessing time is 11.6 ms.

Figure 6: Detected images of feature extractions.

When the model processes a 448x640 pixel input image, it can identify 14 cars, 3 motorcycles, 1 bus, and 1 truck, as seen in Figure 6. The object and class identification process takes 103.3 ms in total. Subsequent identification, these items are described in the text.

Figure 7: Detected images of feature extractions.

The model's use of this 384x640 pixel input image is seen in Figure 7. It identified 2cars. It takes 59.8ms to identify the object or class. The preprocessing time is 1.3 ms and the postprocessing time is 0.8 ms once the image has been transformed into a written description.The detailed output confirms that this is a robust model with adequate capacity to distinguish and count different classes of vehicles, making it very viable for real-time traffic analysis and monitoring applications.

V. Discussion

The results of the research validate the effective application of the YOLOv8 model to detect and classify traffic objects. This custom traffic dataset is designed with labeled images of both vehicles and number plates curated with careful attention using LabelImg in YOLOv5 format. The dataset contains 17 varied classes, such as cars, buses, motorcycles, and trucks, for comprehensive coverage of traffic objects. This diversity makes the model perform better in generalizing and provides a robust foundation for object detection across different types of traffic scenarios.

The YOLOv8 model with 68,172,831 parameters was trained using a dataset of 2,704 images and validated using 300 images. Initially, its performance is comparatively poor with mAP50 being at 0.187 by epoch 1, improving up to 0.234 by epoch 5. The above indicates that the model has been able to learn and make improvements over time to improve detection accuracy. The adjustments in image size along with the use of the AdamW optimizer helped improve the efficiency and performance of the model. Data augmentation techniques used included blurring, grayscale, and reversing, which helped build robustness in the model with diverse input differences.

Figure 2's loss and accuracy curves show a reduction in loss over epochs that indicates successful training. The interesting thing is that the training accuracy was always higher than the validation accuracy, which would be expected because the model had a better fit on the training data. At epoch 30, the model scored mAP50 of 0.385, which meant it was able to recognize multiple object classes in challenging traffic scenarios. The inference speed of 30-50 milliseconds per image affirms the applicability of the model in real time, which is very significant in traffic monitoring and surveillance. The ability of the model to detect and classify objects like buses, cars, and motorcycles as demonstrated in Figure 3 also supports its efficiency and accuracy.

Feature extraction and text generation results of Table 2 and Figures 4-7 emphasize the model's capability of real-time detection and text description. The model detected and classified various objects with varied image sizes with efficient processing time ranging from 59.8 ms to 103.3 ms for different scenarios. This rapid improvement makes the model viable for practical applications such as automated traffic monitoring systems. the YOLOv8-based model detects and classifies traffic-related objects accurately and efficiently. It has very valuable applications in real-time traffic analysis and can be further extended to more smart city applications. The future work would include optimization of the model in even more complex traffic environments, extending the dataset to more diversified scenarios, and incorporating more features like vehicle tracking or license plate recognition.

VI. Conclusion

This research has successfully shown the potential of AI-based solutions for traffic analysis by using YOLOv8 for object detection and text generation from still images of traffic. The proposed model addresses the most significant challenges in traditional traffic monitoring methods, providing greater accuracy, efficiency, and real-time processing capabilities. Through the detection and classification of different traffic objects, the system provides meaningful textual summaries to aid decision-making in traffic management, safety monitoring, and urban planning. Promising results from the model include its significant performance metrics such as mAP and real-time processing speeds, showing that the model is robust and can be applied to various traffic scenarios.

The study has a few limitations despite its success. Despite its diversity, the exclusive dataset that the model utilizes is unable to capture all real-world traffic circumstances, particularly those associated with extreme weather or varying lighting conditions. Furthermore, further work needs to be done to enhance its ability to detect smaller or partially obscured objects. Its capacity to record temporal dynamics, including traffic flow patterns or vehicle speeds, is limited by its reliance on still photos. Furthermore, processing speeds are likely to become a problem when extended to big datasets or even continuous video streams, even though they are certainly real-time for individual images.

This study's future focus will be on reducing these limitations and enhancing the model's functionality. For improved model generalization, expanding the dataset by a variety of real-world circumstances, such as bad weather or traffic at night, may be beneficial. The addition of video processing power will enable the analysis of dynamic traffic behaviors, such as congestion patterns and flow rates. Additionally, combining multi-modal data streams like GPS and IoT sensors, might offer deeper contextual insights. Additionally, the model's edge device optimization may help with its implementation in resource-constrained settings and broader uptake in smart cities. These advancements will strengthen that AI and computer vision are transforming traffic monitoring systems and tackling modern urban challenges.

References

[1] D. Y.-A. J. of C. & Information and U. 2024, "Road Safety Monitoring Model Based on YOLOV8," Acad. J. Comput. Inf. Sci., vol. 7, no. 3, 2024, doi: 10.25236/ajcis.2024.070313.

[2] M. Sohan, T. Sai Ram, and C. V. Rami Reddy, "A Review on YOLOv8 and Its Advancements," Int. Conf. Data Intell. Cogn. Informatics, pp. 529–545, 2024, doi: 10.1007/978-981-99-7962-2_39.

[3] M. Safaldin, N. Zaghden, and M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects," IEEE Access, vol. 12, pp. 59782–59806, 2024, doi: 10.1109/ACCESS.2024.3393835.

[4] M. Shoman, T. Ghoul, G. Lanzaro, T. Alsharif, S. Gargoum, and T. Sayed, "Enforcing Traffic Safety: A Deep Learning Approach for Detecting Motorcyclists' Helmet Violations Using YOLOv8 and Deep Convolutional Generative Adversarial Network-Generated Images," Algorithms, vol. 17, no. 5, 2024, doi: 10.3390/a17050202.

[5] "Traffic vehicles Object Detection | Kaggle.", 2024. [Online]. Available: https://www.kaggle.com/datasets/saumyapatel/traffic-vehicles-object-detection

[6] A. Patil and F. Gonsalves, "Traffic Sense: An Integrated Approach to Traffic Management through Object Detection and CNN," Int. J. Multidiscip. Res. Sci. Eng. Technol., vol. 7, no. 6, 2024, doi: 10.15680/IJMRSET.2024.0706053.

[7] R. Gerlich, I. Schoolmann, … J. B.-D. S. in, and U. 2024, "AI-based Formalization of Textual Requirements," Researchgate, 2024, Accessed: Dec. 07, 2024.

[8] A. A. Verma, B. Chakravarthi, A. Vaghela, H. Wei, and Y. Yang, "eTraM: Event-based Traffic Monitoring Dataset," Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2024, doi: 10.1109/CVPR52733.2024.02136.

[9] B. Khalili, A. S.- Sensors, and U. 2024, "SOD-YOLOv8—Enhancing YOLOv8 for Small Object Detection in Aerial Imagery and Traffic Scenes," Sensors, 2024, doi: https://doi.org/10.3390/s24196209.

[10] P. K. Gollapalli, P., Muthyala, N., Godugu, P., Didikadi, N., & Ankem, "Speed sense: Smart traffic analysis with deep learning and machine learning," World J. Adv. Res. Rev., pp. 2240–2247, 2024.

[11] S. B. Neamah and A. A. Karim, "Real-time Traffic Monitoring System Based on Deep Learning and YOLOv8," ARO-THE Sci. J. KOYA Univ., vol. 11, no. 2, pp. 137–150, 2023, doi: 10.14500/aro.11327.

[12] R. Gupta, U., Kumar, U., Kumar, S., Shariq, M., and Kumar, "Vehicle speed detection system in highway," Int. Res. J. Mod. Eng. Technol. Sci., vol. 04, no. 05, pp. 406–411, 2022.

[13] M. Shihab, R. Ghani, A. M.-I. J. O. COMPUTERS, and U. 2022, "Machine Learning Techniques for Vehicle Detection," Iraqi J. Comput. Commun. Control Syst. Eng., pp. 1–12, 2022, doi: 10.33103/uot.ijccce.22.4.1.

[14] J. Lee, K. A. Cha, and M. Lee, "Multi-Modal System for Walking Safety for the Visually Impaired: Multi-Object Detection and Natural Language Generation," Appl. Sci., vol. 14, no. 17, 2024, doi: 10.3390/app14177643.

[15] R. Zhao, S. H. Tang, E. E. Bin Supeni, S. A. Rahim, and L. Fan, "Z-YOLOv8s-based approach for road object recognition in complex traffic scenarios," Alexandria Eng. J., vol. 106, pp. 298–311, 2024, doi: 10.1016/j.aej.2024.07.011.