INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Check for updates

OPEN ACCESS

Received: 22-12-2023

Accepted: 09-04-2024

Published: 07-08-2024

Citation: Pandey A, Raja R (2024) Smart Skyways: Aerial Traffic Control Using Artificial Intelligence Algorithms. Indian Journal of Science and Technology 17(31): 3272-3280. https://doi.org/10.17485/IJST/v17i31.836

* **Corresponding author**.

agnitu1984@gmai.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Smart Skyways: Aerial Traffic Control Using Artificial Intelligence Algorithms

Agnivesh Pandey[1]*, Rohit Raja[2]

1 Assistant Professor and Research scholar, Department of IT SOSE & T, GGV Bilaspur, India
2 Associate Professor, Department of IT SOSE & T, GGV Bilaspur, India

Abstract

Objectives: This study presents a resilient deep learning method for vehicle detection and classification in traffic aerial images. **Methods:** The Roboflow dataset used for model training includes 220 aerial images, annotated in YOLOv8 format in four classes of moving objects. Our suggested YOLO version eight-based approach uses real-time photos from Aerial cameras for the training and testing and validation for the classification and detection of moving objects. **Findings:** YOLOv8, a state-of-the-art object detection algorithm, enhances the accuracy and efficiency of traffic monitoring on the Roboflow dataset. This approach achieves a mean average precision (mAP) of 0.86, demonstrating the robustness of our model in detecting traffic-related objects. The precision and recall values of 0.89 and 0.96, respectively, further affirm the model's effectiveness in accurately identifying and classifying objects. **Novelty:** The research introduces a novel application of YOLOv8 for traffic analysis, demonstrating its advanced object detection and classification capabilities and achieving an impressive mean average precision of 0.86, enhancing computer vision for real-time traffic monitoring, which offers high precision and recall, enabling traffic flow optimization, anomaly detection, and intelligent transportation systems, paving the way for smart city solutions and data-driven urban planning.

Keywords: Object Detection; YOLOv8; RealTime Monitoring; Intelligent Transportation Systems; Aerial Images

1 Introduction

Road traffic management is essential for urban planning and transportation networks, traffic patterns and congestion management which generally driven by aerial photos and machine intelligence. Aerial imagery and machine learning algorithms can help city officials and transportation planners to understand traffic flow, identify issues, and find solutions to reduce congestion. Advanced technologies can also predict traffic patterns and optimize traffic signal timing for efficiency. Data privacy problems, algorithm biases, and the need for specific training make these technologies difficult to use. Although these challenges remain, aerial pictures and machine learning may improve road traffic management and justify additional study. Real-time traffic flow insights are one benefit of aerial pictures and machine learning in road traffic management.

Transportation officials can immediately spot bottlenecks, accidents, and construction zones by examining high-resolution aerial photographs. They can target solutions like signal timing or traffic rerouting to reduce congestion and improve traffic flow. Machine learning systems can also forecast traffic trends and optimize traffic signal timing using past data. Cities may increase traffic efficiency and plan for long-term infrastructure improvements to handle expanding populations and changing transportation needs by using this innovative technology.

Intelligent traffic management systems optimize traffic flow in smart cities using modern technologies and data analytics to reduce congestion, improve transportation efficiency, and improve urban livability by providing commuters with correct information[(1)]. Traffic incidents are detected and addressed by intelligent traffic management systems using sensors, cameras, and complex algorithms to ensure safety and well-being. They promote sustainable commuting and environmental sustainability by integrating transport modes. This integrative strategy minimizes congestion and improves urban living[(2)]. Fusion-based intelligent traffic congestion control system employing machine learning to collect and route traffic data. The remote visibility of traffic flow and vehicle availability reduces congestion and improves traffic flow[(3)]. Urban planning and smart cities require ITS to improve safety, efficiency, and the environment. Scalability, different service needs, and large data creation make them difficult. Investigates the use of machine learning (ML) in ITS applications including cooperative driving and road hazard warning and suggests ways to improve ITS using ML[(4)]. Fuzzy logic traffic signal system includes two fuzzy controllers for the main and secondary driveways to adjust to traffic conditions. VISSIM and MATLAB simulations showed this strategy improves low-volume traffic conditions. ANN and fuzzy controllers are employed in this smart traffic signal system. The system uses sliding window algorithms to segment, normalize, and convert traffic camera images to grey scale. Segmented images are transmitted through the ANN and used in a fuzzy controller to set green and red-light timings. System execution takes 1.5 seconds and averages two errors[(5)]. Yolo machine learning algorithm is utilized for various type of object detection efficiently[(6,7)]. Deep learning-based fire detection requires more diversified and larger datasets for training and testing due to low-quality cameras or bad illumination conditions influencing algorithm accuracy. It needs better YOLO algorithms[(8)]

Recent advances in yolo shows the success in forest fire detection and tracking[(9)], detection of tiny targets and multiple scales better procedure Remote sensing photos on SIMD dataset use Yolo-SE[(10)]. The rise in cars in metropolitan areas leads to road capacity decline and service issues, with fixed signal timings causing traffic problems. New Intelligent Transport System technologies are needed to address this issue. Current traffic control techniques include manual control, static timer traffic lights, and vehicle classification. An innovative solution incorporating advanced sensors and machine learning algorithms is proposed.

The main contribution of this work is as follows

- Create AI Enable framework that can classify and detect vehicle classes and track vehicle in Aerial Images.
- Propose a method for counting automobiles In Aerial View.

The use of smart grid technology goes beyond simple vehicle categorization to include anomaly and possible danger identification on the road. The suggested system can quickly intervene and mitigate abnormalities like accidents, stuck cars, or unpredictable driving behavior by integrating anomaly detection algorithms. This dual purpose helps with road safety, which is an important aspect of urban planning and administration, in addition to streamlining traffic.

2 Methodology

Smart traffic management system requires a good annotated date set for the training the model. We employed a systematic approach to implement land vehicle detection using machine learning algorithms on aerial images. The Roboflow dataset utilized for training and evaluation was carefully curated, consisting of high-resolution aerial imagery captured under diverse environmental conditions.

2.1 Overview of the proposed system

The suggested system is a smart traffic control system that uses YOLOV8 to recognize traffic for each lane and track and count the different vehicles. The suggested solution is flexible and dynamic in response to varying incoming traffic. Compared to the current static system, the suggested one will be able to handle traffic in a more dynamic manner. The goal of the suggested system is to create a desktop stand- alone program for YOLOV8-based smart traffic control.

The typical YOLO-based object identification approach detects the input image's car, bus, bike, and truck. YOLO models' standard object identification technique is shown in Figure 1. The input image is segmented into S×S grids, with 'n' bounding boxes for item detection. Each grid predicts bounding boxes with confidence scores to indicate object presence. The confidence score is zero if there are no objects in the grid and close to one if there are. Bounding box coordinates are predicted by the confidence score versus the ground truth. As indicated in Eq. (1), the confidence score is the product of object probability and intersection over union truth prediction.

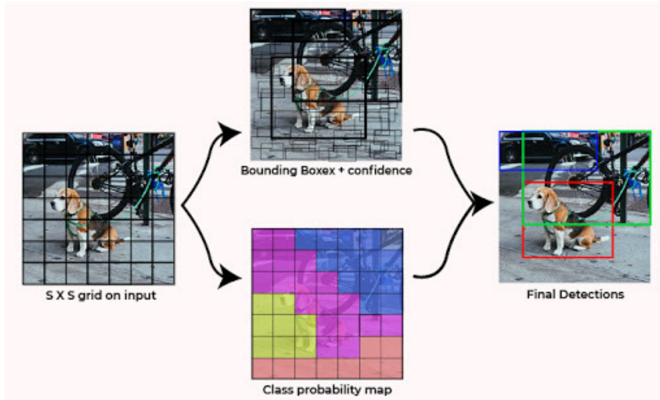

Fig 1. Yolo object detection process

$$CS = P(O) \times T\,(1)\,P(IOU) \tag{1}$$

where O object, IOU intersection over union, P probability, TP true prediction, and CS confidence score

Furthermore, each grid provides an estimation of the conditional class probability for every object, in addition to the confidence score. The conditional class probability in each grid cell is what we refer to as the probability that the detected object belongs to a particular class, and equation (2) represents it.

$$CP = P(C|O) \tag{2}$$

Each grid cell specifies a single set of class probabilities, were CP conditional class probability, C class that is regardless the quantity of boxes. As stated in Equation (3), the conditional probability is multiplied by the individual box confidence predictions in order to provide the class-specific confidence ratings for each box.

$$P(C|O) \times P(O) \times T\,(3)\,P(IOU) = P(C) \times TP(IOU) \tag{3}$$

Not only do these ratings indicate the likelihood of the class being included in the box, but they also indicate the degree to which the predicted box corresponds to the actual item. After everything is said and done, a bounding box that has a class probability that is greater than a threshold value is chosen and utilized in order to locate the objects in the image.

Algorithm for car and truck detection and counting is given below, which gives the steps from detection and counting.

For all (I) image do:
divide I into S x S grids
place 'n' BB within each grid to objects detection
calculate the confidence score (CS) to represent the objects within each box
calculate the coordinates (w-width, h-height) of the BB with the CS
calculate the conditional class probability (Cp) for each detected object
For all detected object do:
initialize the counter for each object
increment the counter for each detected object, respectively
end
end

Figure 2 depicts the layout of the planned architecture for the detection of automobiles, buses, bicycles, and trucks. The architecture of the network is achieved by implementing it through a variety of different paths in order to finalise the configuration with the least amount of error possible. The architecture is comprised of 21 layers. Following the application of the convolution and pooling layers, the input image, which has a resolution of 416 × 416, is processed in order to get the feature maps. To facilitate the generation of bounding boxes through the utilisation of the convolution technique, the image is down-sampled to a resolution of 208 × 208. For the purpose of more correctly identifying automobiles and trucks that are present in congested areas, the image is down-sampled by a factor of 32. The performance of the network is gradually improved as a result of the suggested architecture's utilisation of a lower number of layers in comparison to the YOLO network. A pass-through layer is incorporated towards the conclusion of the process. This layer is responsible for binding the 52 x 52 layers into the final convolution layer, which aids in the detection of smaller objects that are captured by aerial photography.

2.2 Vehicle detection module

YOLOV8 is a real-time object recognition method that uses a single convolutional neural network to forecast bounding boxes and probabilities for each position in an image. This method is well-liked since it operates in real-time and produces excellent accuracy. Predictions only need to be made once during forward propagation since the approach only makes one pass over the image. Bounding boxes and identified objects are produced using the procedure after nonmax suppression.

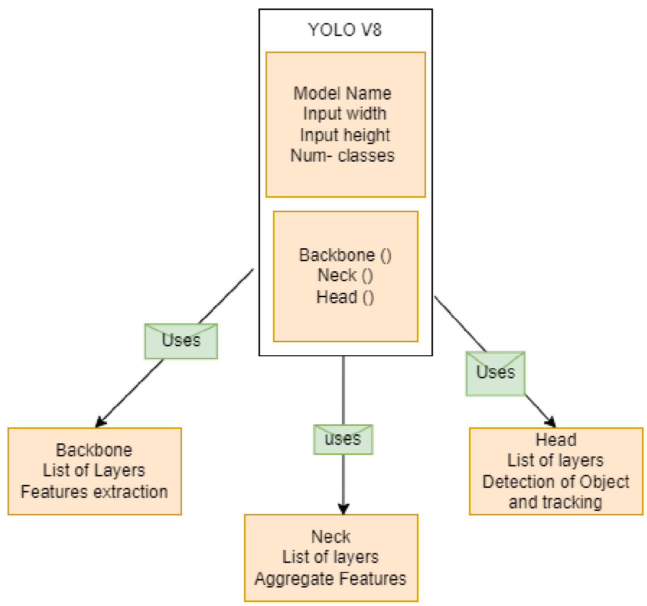

Fig 2. Proposed Yolov8 for object detections

The model was trained on a dataset[11] that was produced by manually identifying images from Kaggle using Roboflow and scraping them. The model was then trained using pertained weights[8]. The parameters of the video file were adjusted to comply with the model's specifications. The number of output neurons in the last layer was changed to match the number of classes the model was predicted to detect. The quantity of filters was also altered. After the model was trained, the weights were changed until the loss was significantly reduced. The weights were imported into the code to enable the detection of automobiles using the OpenCV library. The standard for successful detection was set. . Subsequently, the model generates the results in video file format, with labels serving as the keys and confidence and coordinates as the values. Using OpenCV, bounding boxes may be created on the images depending on the given labels and coordinates. provides a visual description of the steps in the categorization detection process by highlighting the key stages and the information flow in (Figure 3).

The compilation of collected datasets for the purpose of training YOLO model, we have opted to use the Roboflow dataset[11]. This is the dataset that we have taken into consideration. Roboflow is a framework that was developed with the intention of simplifying the process of managing and preparing datasets for applications that are related to computer vision. Annotation is a process comprised of: Annotation is comprised of a number of different components, including the identification of objects in images, the assignment of class labels to those various items, and the providing of bounding box coordinates. The YOLO architecture makes use of a format of annotation that is one of a kind due to its exceptional characteristics. When it comes to: There are many distinct class designations that are assigned to various types of vehicles. Some examples of these classifications include the terms "car," "truck," and "bus," among others. It is imperative that this phase be completed in order for the model to be able to distinguish between the several types of objects. What are the coordinates of the box that is bounded? at the box

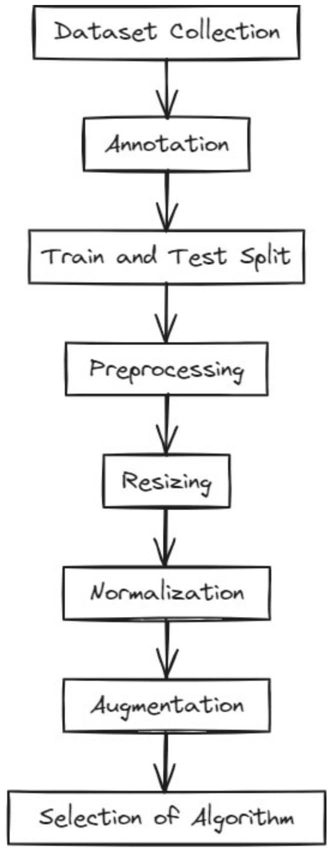

Fig 3. The functioning of the complete classification detection procedure

that encloses an object, the corners that are determined by the bounding box coordinates are the corners that are located at the top-left and bottom-right corners of the box. In most instances, these coordinates are scaled to values that fall between 0 and 1, and they are relative to the dimensions of the illustration. In other words, they are scaled to the ideal range. Establishing subdivisions: The process of image segmentation involves dividing a picture into numerous regions or segments, with each segment representing a major piece of the overall image. The method of semantic image segmentation, which is particularly achieved with YOLO, is responsible for doing the assignment of a class name to each pixel that is contained within an image. Because of this, the image is successfully divided into portions that are more digestible, depending on the semantic content of the graphic.

2.3 YOLO Algorithm

The core idea behind YOLO is to partition the input image into a grid of cells and, for each cell, estimate the probability of the presence of an object and the bounding box coordinates of the object. The process of YOLO can be broken down into multiple parts.

A convolutional neural network (CNN) is used to process the input image in order to extract hierarchical features. This process is referred to as feature extraction. Because of this, the network is able to effectively capture information at a variety of different levels of abstraction, ranging from low-level features such as edges to high-level features such as object forms. Fully Connected Layers: After the features have been retrieved, they are then put through fully connected layers, which are responsible for providing predictions regarding class probabilities and bounding box coordinates. These fully connected layers are responsible for making global predictions by utilizing the features that were extracted by the convolutional layers that came before them. The image is separated into a grid of cells, which is referred to as grid division. Predicting a collection of bounding boxes and the class probabilities that are connected with them is the responsibility of each individual cell. As a result of the predictions being localised to particular cells, the network is able to concentrate on various parts of the image. Bounding Box and Class Prediction: The network makes a prediction about the bounding box coordinates (x, y, width, and height) for each grid cell based on the position of the cell. A further prediction is made on the class probabilities of the objects that are contained within that cell. Each enclosing box is connected to a certain class probability because of its association. The output of the network is a collection of bounding boxes, each of which has associated probabilities for each object. This type of suppression is known as non-max suppression. Non-maximum suppression is utilized in order to improve the accuracy of the forecasts. By taking into consideration the bounding box that has the highest probability for each object, this technique eliminates bounding boxes that are redundant and have a low level of confidence. By doing so, it helps to ensure that only the predictions that are the most accurate and relevant are included. Final Output: Following the non-max suppression step, the final output consists of a collection of predicted bounding boxes together with the class labels that are connected with them. Each bounding box is connected to a particular class, and the combination of these bounding boxes and classes is what represents the objects that have been identified in the image.

2.4 Evaluation Metrics

We used many well-known assessment metrics, including recall, precision, accuracy, F1-score, and mean average precision (mAP), to assess the prediction performance of the algorithms in this work. The most widely used metric for Assessing accuracy is the ratio of instances that were properly classified to all test photos. This can be demonstrated in Eq (4)

$$Accuracy = \left(TP + \frac{TN}{TP} + FN + FP + TN\right) * 100 \tag{4}$$

Precision, also referred to as a positive predictive value, is defined as the proportion of correctly identified labels in patients who are truly positive. It is expressed in Eq. (5)

$$Precisions = \left(\frac{TP}{TP + FP}\right) * 100 \tag{5}$$

The F1-score, also referred to as the F-measure, is a weighted average that combines precision and recall. The notation for the F-measure in Eq. (6)

$$F1 - score = 2 * \left(\frac{precision * Recall}{Precisions + Recall}\right) 100 \tag{6}$$

Recall or sensitivity are used to calculate the percentage of correctly classified objects. And it is shown in Eq (7)

$$Recall = \left(\frac{TP}{TP + FN}\right) * 100 \tag{7}$$

3 Results and Discussion

3.1 Evaluation of Vehicle Detection Module

Good precision rate was attained when testing the vehicle detection module with multiple photos that had varying numbers of vehicles the proposed model shows detection of real word traffic objects with average efficiency of 0.577 and the previous model reported in [9] and [10] are fail to detect the real time traffic or rode objects like car and bus. Real-world video from traffic cameras may be utilized to train the model and increase the accuracy and precision of the system.

3.2 Performance of Object Detection

Accuracy and Precision of the Yolov8 Model: Robust performance was shown by the YOLOv8 model in precisely identifying items in the smart grid environment. The model's performance in recognizing and categorizing things connected to traffic, such cars, bike, truck and bus is demonstrated by the Precision and recall metrics shown in (Figure 4) provides a graphical representation of the model's performance across different confidence thresholds.

Figure 5 presents the Confusion Matrix, a visual representation that elucidates the model's classification performance in granular detail. This matrix encapsulates the true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions for each class, offering a comprehensive snapshot of the model's ability to distinguish between different categories. Each cell in the matrix provides insights into the model's accuracy, highlighting areas of strength and potential improvement. The Confusion Matrix serves as a crucial diagnostic tool, guiding further optimization strategies and refining the model's

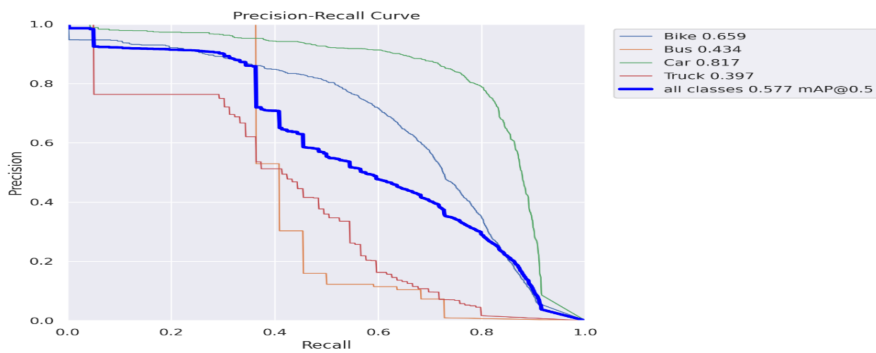

Fig 4. Precision Recall curve for predicted object (bike, Bus, Car and truck) and mAP for all classes at 50% confidence

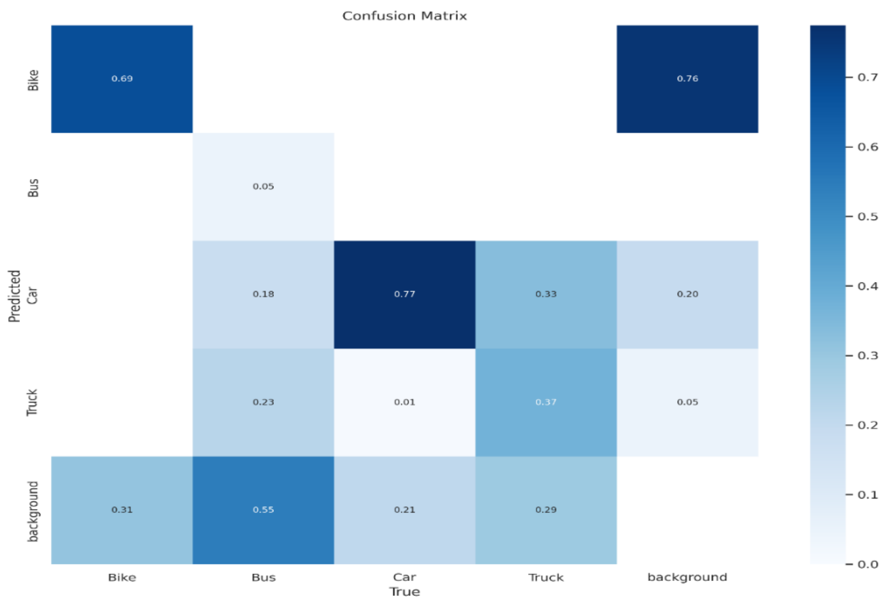

Fig 5. Confusion Matrix for all four classes (bike, Bus, Car and truck) and background for the fine tune model for test dataset

classification capabilities. Effect on the Effectiveness of Traffic Management The smart grid's ability to accurately detect objects improves traffic control effectiveness. Prompt decision-making and response to traffic accidents are made possible by real-time monitoring and analysis, which enhances grid resilience overall.

Figure 6 encapsulates the Predicted Results, offering a visual representation of the model's output in terms of bounding box predictions and class labels. Each bounding box signifies a detected object, with accompanying class labels and confidence scores providing valuable insights into the model's decision-making process.

This visual representation enables a qualitative assessment of the model's performance, allowing for an examination of correct and erroneous predictions. By juxtaposing the predicted results against ground truth annotations, this figure aids in identifying areas of success and potential improvement in the model's object detection capabilities. The visualization of predicted results serves as an intuitive means to assess the model's ability to accurately localize and classify objects in the given dataset, contributing to a holistic understanding of its performance.

The proposed method is summarized in Table 1 which provides a comprehensive overview of the performance of our object detection model across different classes. The evaluation metrics include precision, recall, mean. Average Precision at IoU 0.5 (mAP50), and mean Average Precision across all IoU thresholds from 0.5 to 0.95 (mAP50-95#). These metrics offer insights into the model's ability to accurately detect and classify objects in the given dataset. The model demonstrates a reasonable performance across all classes, with an overall precision of 0.624 and recall of 0.601. The mean Average Precision at IoU 0.5

Fig 6. Predicted Results for all four classes (bike, Bus, Car and truck) and background for the fine tune model for test dataset

(mAP50) is 0.577, indicating that the model performs well in terms of both precision and recall at a relatively relaxed intersection over union threshold.

3.3 Class-Specific Analysis

1. **Proposed method shows the good results for the selected four classes of objects**

- Bike

The class 'Bike' exhibits a precision of 0.491 and a relatively high recall of 0.729, resulting in a mAP50 of 0.659. Despite a lower precision, the model excels in capturing a significant portion of instances for this class.

- Bus

For the 'Bus' class, the model achieves an impressive precision of 0.895 but at the cost of lower recall (0.364). The trade-off between precision and recall is evident, reflected in the mAP50 of 0.434 and mAP50-95# of 0.366.

- Car

The 'Car' class showcases balanced performance, with a precision of 0.751, recall of 0.819, and high mAP50 (0.817) and mAP50-95# (0.47) scores. This indicates the model's proficiency in accurately identifying and localizing cars in the dataset.

- Truck

Performance for the 'Truck' class is comparatively lower, with a precision of 0.358 and recall of 0.495. The model struggles to achieve a satisfactory balance between precision and recall for this class, resulting in lower mAP50 (0.397) and mAP50-95# (0.22) scores.

Table 1. Model Performance for All classes

Class	Images	Instances	Precisions	Recall	mAP50*	mAP50-95#
All	199	3981	0.624	0.601	0.577	0.332
Bike	199	2369	0.491	0.729	0.659	0.27
Bus	199	22	0.895	0.364	0.434	0.366
Car	199	1491	0.751	0.819	0.817	0.47
Truck	199	99	0.358	0.495	0.397	0.22

* Mean Average precisions at confidence 50% # Mean Average precisions at confidence 50-95%

4 Conclusion

The class-specific analysis sheds light on the strengths and weaknesses of the model across different object categories. These findings can guide further improvements in the model architecture, dataset augmentation strategies, and training procedures for specific classes with suboptimal performance.

In conclusion, while our model demonstrates competitive object detection capabilities, ongoing efforts are required to enhance its performance across all classes. This study contributes valuable insights for refining the model's robustness and generalizability in real-world scenarios, fostering advancements in object detection research.

Acknowledgement

The authors thank the Department of Information Technology, Central University, Bilaspur, Chhattisgarh, India for providing infrastructure facilities for this work.

References

1) Hilmani A, Maizate A, Hassouni L. Automated real-time intelligent traffic control system for smart cities using wireless sensor networks. 2020. Available from: https://doi.org/10.1155/2020/8841893.
2) Siddiqi SJ, Naeem F, Khan S, Khan KS, Tariq M. Towards AI-enabled traffic management in multipath TCP: A survey. *Computer Communications*. 2022;181:412–439. Available from: https://doi.org/10.1016/j.comcom.2021.09.030.
3) Saleem M, Abbas S, Ghazal TM, Khan MA, Sahawneh N, Ahmad M. Smart cities: Fusion-based intelligent traffic congestion control system for vehicular networks using machine learning techniques. *Egyptian Informatics Journal*. 2022;23(3):417–443. Available from: https://doi.org/10.1016/j.eij.2022.03.003.
4) Yuan T, Neto WDR, Rothenberg CE, Obraczka K, Barakat C, Turletti T. Machine learning for next-generation intelligent transportation systems: A survey. 2022. Available from: https://doi.org/10.1002/ett.4427.
5) Vogel A, Oremovic I, Simic R, Ivanjko E. Fuzzy traffic light control based on phase urgency. *In2019 International Symposium ELMAR*. 2019;p. 9–14. Available from: https://doi.org/10.1109/ELMAR.2019.8918675.
6) Terven J, Cordova-Esparza D. A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond. 2023. Available from: https://doi.org/10.48550/arXiv.2304.00501.
7) Sohan M, Ram TS, Reddy CVR. A Review on YOLOv8 and Its Advancements. In: Jacob, J I, Piramuthu, S, Falkowski-Gilski, P, editors. Data Intelligence and Cognitive Informatics. ICDICI 2023. Algorithms for Intelligent Systems. Springer. 2024. Available from: https://doi.org/10.1007/978-981-99-7962-2_39.
8) Mudawi A, Qureshi N, Abdelhaq AM, Alshahrani M, Alazeb A, Alonazi A, et al. Vehicle Detection and Classification via YOLOv8 and Deep Belief Network over Aerial Image Sequences. 2023. Available from: https://doi.org/10.3390/su151914597.
9) Talaat FM, Zaineldin H. An improved fire detection approach based on YOLO-v8 for smart cities. *Neural Comput & Applic*. 2023;35:20939–20954. Available from: https://doi.org/10.1007/s00521-023-08809-1.
10) Wu T, Dong Y, Yolo-Se. Improved YOLOv8 for Remote Sensing Object Detection and Recognition. *Appl Sci*;2023:12977–12977. Available from: https://doi.org/10.3390/app132412977.
11) Roboflow. Detecting Vehicles on Aerial Images Dataset. *Roboflow Universe Roboflow*. 2023. Available from: https://universe.roboflow.com/aerial-vehicle-detection/vehicle-detection-in-aerial-images.