Bangladeshi Native Vehicle Classification Employing YOLOv8

Siraj Us Salekin[1], Md. Hasib Ullah[1], Abdullah Al Ahad Khan[1], Md. Shah Jalal[1], Huu-Hoa Nguyen[2], and Dewan Md. Farid[1(✉)]

[1] Department of Computer Science and Engineering, United International University, United City, Madani Avenue, Badda, Dhaka 1212, Bangladesh
ssalekin213059@mscse.uiu.ac.bd, dewanfarid@cse.uiu.ac.bd

[2] College of Information and Communication Technology, Can Tho University, 3/2 Street, Ninh Kieu District, Can Tho City, Vietnam
nhhoa@ctu.edu.vn
https://cse.uiu.ac.bd/profiles/dewanfarid/

Abstract. Traffic congestion poses a significant challenge in Bangladesh due to the growing number of vehicles. To tackle the obstacle needs an effective intelligent system that can reduce the traffic congestion. With a vision of building such an intelligent system, this paper presents a study on vehicle classification using the YOLO (You Only Look Once) v8 transfer learning model, customized for Bangladeshi native vehicles. Besides, we propose a transfer learning model-based system that helps to analyse the video footage of vehicle movements from the elevated viewpoints of foot over-bridges. Initially, the Bangladeshi Native Vehicle Image dataset is gathered, processed, and used to train the model. Once the model is trained and evaluated, the model is integrated into the vehicle detection system. The system detects and tracks the vehicles, providing practical traffic volume and movement insights. After the result analysis, We have found a high mean average precision (mAP) of 91.3 % using intersection over union (IoU). The model's performance enables proactive measures to reduce congestion and optimise traffic flow. To build an efficient transportation network, this system can assist the Bangladesh Road Transport Authority (BRTA) and Bangladesh Police Traffic Division to address the challenges of increasing traffic and enhance traffic management.

Keywords: Deep Transfer Learning · YOLOv8 · Vehicle Classification

1 Introduction

Dhaka, the capital of Bangladesh, is grappling with severe traffic congestion resulting from haphazard growth and insufficient planning. Over the past 10 years, average driving speeds have dramatically declined from 21 kilometers per hour to a mere 6 kilometers per hour. If the current trend persists, it could further plummet to just 4 kilometers per hour by 2035, slower than the average walking pace. This congestion wastes 3.2 million working hours per day

N. Thai-Nghe et al. (Eds.): ISDS 2023, CCIS 1949, pp. 185–199, 2024.
https://doi.org/10.1007/978-981-99-7649-2_14

and imposes billions of dollars in economic costs annually [3]. The poor traffic management is causing delays and leading to a staggering 40% fuel wastage, amounting to daily losses of Bangladeshi BDT 41.5 million ($483,872). This fuel consumption and traffic congestion issue is adversely affecting the economy. Dhaka commuters face long travel times, with just 3.8 kilometers taking up to one and a half hours. The country is estimated to lose 1.38 billion BDT ($160 million) daily due to traffic jams. Insufficient road capacity, with only 7% of the required road capacity available, and an overload of vehicles contribute to traffic congestion. Road accidents, mental discomfort, and loss of valuable time are additional consequences of heavy traffic [9]. Experts recommend government intervention to establish discipline in the public road transport system. So we propose a method that can facilitate reducing traffic congestion in Bangladesh. To develop the system, we need Object Detection which is related to Computer Vision and is also a part of Deep Learning.

Object detection is an advancing field in computer vision, focusing on identifying and locating objects in images or videos. It's complex due to variations in appearance, lighting, occlusion, and cluttered backgrounds. Two main approaches exist: two-stage detectors proposing object locations and then classifying them, and one-stage detectors predicting object categories and bounding them in a single pass. Deep learning models like Convolutional Neural Networks (CNNs) are commonly used to extract features and classify objects. Challenges include detecting small objects, handling clutter, and enhancing robustness to changes in object appearance. Deep learning, a subset of machine learning, employs artificial neural networks to analyse large datasets and make predictions. It has gained popularity for solving complex problems in computer vision, natural language processing, and speech recognition. Deep learning algorithms recognise patterns by adjusting neuron weights through back-propagation. However, training and deploying deep learning models require substantial data and computational resources, and their inner workings may be hard to comprehend. Ongoing research addresses these challenges and enhances deep learning capabilities [5]. Also, to make it more accessible, researchers developed the Transfer Learning technique.

Transfer learning, a technique in deep learning, utilities knowledge gained from one task to improve performance in another. It reduces data and computation needs for training deep learning models. Pre-trained models like VGG, ResNet, YOLOv8, and BERT serve as common benchmarks. Transfer learning offers advantages such as faster training, higher accuracy, and learning from limited data. Challenges include selecting appropriate pre-trained models and managing domain discrepancies. Research focuses on refining pre-training methods, enhancing adaptation and fine-tuning, and deepening the understanding of transfer learning principles [1]. YOLOv8, developed by Ultralytics, is a cutting-edge computer vision pre-trained deep-learning model succeeding YOLOv5. This model offers built-in capabilities for object detection, classification, and segmentation tasks. It provides an accessible Python package and command line inter-

face for easy usage. YOLOv8 enables the way of training its model on the custom data set to maximise the result for greater context [13].

Due to the efficiency of transfer learning, we have used the YOLOv8 deep-learning model for vehicle classification on the Bangladeshi Vehicle Annotated image data set named "Poribohon-BD" [12]. It requires data preparation, model training, evaluation, and deployment to achieve accurate and efficient vehicle detection results using our system. The annotated images are fed into the model and adjusted its weights and biases to optimise the detection performance. Once the model has been trained and evaluated, it is used to detect vehicles in new images or videos. And we have used our custom deep-learning model to develop a system that can detect traffic volume for both sides of the road from a given recorded video of the road from elevated viewpoints. We are hopeful that our system can help the authority of Bangladesh to give a proper insight into traffic congestion to take necessary steps on a particular road. Also, they can use our model in surveillance cameras for real-time vehicle detection in Bangladesh.

2 Literature Review

Transfer learning is used in Image Processing a lot these days. Deep Neural Networks are effective in recognizing intricate image features. The dense layers are responsible for image detection, and adjusting the higher layers does not impact the fundamental logic. Some related articles about machine learning, deep learning, and transfer learning in computer vision are discussed in this section.

Tabassum et al. developed a transfer learning method using YOLOv5 for detecting local vehicles on Bangladeshi roads using 9000 annotated images. The method achieved a 73% IoU score and a 55 frame per second speed after 56600 iterations. This approach holds promise for traffic management applications in Bangladesh [11].

Yiren et al. used deep neural networks to tackle vehicle detection and classification challenges in road images. Their YOLO detection model achieved 93.3% precision and 83.3% recall, comparable to the state-of-the-art DPM(Deformable Part Models) method had 94.4%. They explored fine-tuning and feature-extraction ways, and proposed techniques for addressing poor lighting conditions. This approach holds the potential for limited dataset training and can be extended for traffic development and planning purposes [18].

Shaoyong Yu et al. developed a deep-learning approach for classifying vehicles in complex transportation scenes. Their model uses a Faster R-CNN method for vehicle detection and a joint Bayesian network for classification. The classification model achieved 89% accuracy, but misclassifying non-vehicle regions may affect overall accuracy. Future work aims to improve detection accuracy, and speed, and incorporate feature classifiers for similar-looking vehicles [15].

Chen et al. developed a real-time vehicle classification model using AdaBoost and deep convolutional neural networks, achieving 99.50% accuracy in five vehicle groups. The model efficiently identifies vehicle images in 28ms and has low

storage requirements, with training taking only 8 min. This model holds potential for intelligent transport systems and real-time traffic supervision [2].

Mahibul Hasan et al. developed a Bangladeshi model using transfer learning into ResNet50 and data augmentation to classify native vehicle types, achieving 98% accuracy in 13 standard classes. The Deshi-BD dataset used 10,440 images for training. This approach has strong generalization capabilities and the potential to address road traffic accidents in Bangladesh [4].

Maungmai et al. developed a vehicle classification system using Convolutional Neural Networks to classify vehicle characteristics, including type and color, from cropped images. The method outperformed Saripan et al.'s method and a deep neural network, with over 80% accuracy in type classification and 1.8% in color classification. Future research should explore different input image sizes, deeper CNN structures, and hyper-parameters [6].

Zhou et al. used deep neural networks to detect and classify vehicles in a public dataset. They achieved a 95.6% vehicle detection precision rate using state-of-the-art methods, including Alexnet-based methods. The method successfully detected and classified 714 passenger vehicles and 226 other vehicles, proving useful in multi-lane highway scenarios [17].

Yiren et al. also developed a visual attention-based CNN model for image classification, using a processing module to highlight specific parts of an image and weaken others. They improved image classification by computing information entropy and guiding reinforcement learning agents to select critical parts. The model was tested on a surveillance-nature dataset and showed better performance than large-scale CNNs in vehicle classification tasks. The VGG58 model (ImageNet-Based) showed a larger performance boost on Vehicle-58 (4%) than Vehicle-5 (3%) custom models. The accuracy rate was about 3% higher than large-scale CNNs, demonstrating the effectiveness of visual attention in improving image classification performance [16].

Vijayaraghavan et al. developed a convolution neural network for detecting and classifying vehicles using an entire image as input and a bounding box with feature class probabilities. The model outperformed state-of-the-art Fast R-CNN and cars, with an accuracy of 87% and 76%, respectively [14].

Neupane et al. developed a large training dataset, domain-shift problem, and real-time multi-vehicle tracking algorithm using deep learning. They created a 30,000-sample dataset and fine-tuned YOLO networks. The YOLOv5 model achieved 95% accuracy and performed well under various conditions. Future work should validate vehicle speed and compare internet network speed effects on deep learning models for road safety [7].

Overall, all the authors discussed and proposed the various uses of deep learning, and transfer learning for vehicle detection, and classification. Therefore an updated deep learning and transfer learning model will be beneficiary in the context of vehicle detection in the current world. Table 1 showcases some of the relevant research on vehicle detection and classification. After going through above mentioned research, we have found that there is a lack of implementing the latest YOLOv8 model for vehicle detection and tracking in the context of

Bangladesh. YOLOv8 is the updated deep neural network model for object detection which is showing promising results in custom image datasets for Transfer Learning. So, what we have done in our research is listed below:

Native vehicle detection in Bangladesh.
Native vehicle tracking in Bangladesh.
No. of Traffic volume & movement insights.
A highly accurate Transfer Learning Model which can be used in real-time native vehicle tracking in Bangladesh.
Providing a tool that can be used by the Road Transport Authority of Bangladesh (BRTA) to reduce traffic congestion.

3 Methodology

In Fig. 1, we have illustrated how the proposed system is built and the process of detecting Bangladeshi native vehicles on given input videos. Now, a detailed process is discussed in this section.

3.1 Dataset Description

Poribohon-BD [12] is a vehicle dataset consisting of 15 Bangladeshi vehicles, including a bicycle, boat, bus, car, CNG, easy-bike, horse-cart, launch, leguna, motorcycle, rickshaw, tractor, truck, van, and wheelbarrow. The dataset consists of 9058 JPG images depicting a wide variety of poses, angles, illumination, and weather conditions, as well as backgrounds. Each image is accompanied by an annotation file in XML format that indicates the exact positions and labels of each object. Data augmentation techniques were utilized to ensure that image counts for each vehicle classification were comparable. The faces of people were obscured in order to protect their privacy. The dataset is organized into separate folders for each vehicle classification, with an additional folder titled'Multi-class Vehicles' containing images and annotations of vehicles of multiple classes. Poribohon-BD is compatible with widely-used CNN architectures such as YOLO, VGG-16, R-CNN, and DPM. It functions as a valuable resource for Bangladeshi vehicle classification and detection research. But to train the model of YOLOv8 using this dataset is not possible because YOLOv8 expects PyTorch TXT annotated data. So, we have to re-annotate the dataset.

3.2 Dataset Processing

In the Yolov8 transfer learning process, PyTorch TXT annotation format is needed. XML annotation is not the right format. The available dataset is not in the right format to work with the latest Yolov8 transfer learning model.

So, we used Roboflow and that is an online tool that simplifies the conversion of XML annotation image files to PyTorch TXT format. By uploading your dataset and configuring settings, Roboflow exports the dataset with transformed annotations. The converted dataset, including images and TXT annotation files, is ready for PyTorch-based training [8]. This user-friendly process facilitates the integration of annotated data into machine learning workflows. This way we have re-annotated our data so that it can be used in the YOLOv8 object detection model. That is why we say our data is customized. Also in Fig. 1, this dataset is mentioned as custom data. In the future, anyone can access the customised dataset for further research.

Table 1. Research findings based vehicle classification models and its accuracy.

Author	Task	Method	Model	Accuracy
Tabassum et al. (2020) [11]	Classification	Transfer Learning	YOLOv5	73%
Yiren et al. (2016) [18]	Classification, Detection	Transfer Learning	Darknet + Coffee Framework and YOLO with DMP	93.3%, 94.4%
Shaoyong Yu et al. (2017) [15]	Classification	Deep-Learning	R-CNN	89%
Chen et al. (2018) [2]	Classification	Deep-Learning	AdaBoost + CNN	99.50%
Mahibul Hasan et al. (2021) [4]	Classification	Transfer Learning	ResNet50-Based	98%
Maungmai et al. (2019) [6]	Classification	Deep Learning	CNN	80%
Zhou et al. (2016) [17]	Classification, Detection	Transfer Learning	AlexNet-Based	95.6%
Yiren et al. (2016) [16]	Classification	Guided Reinforcement Learning	Imagenet-Based	$> 90\%$
Vijayaraghavan et al. (2019) [14]	Classification, Detection	Deep-Learning	Fast-RNN	87%
Neupane et al. (2022) [7]	Detection, Real-Time Tracking	Transfer-Learning	YOLOv5	95%

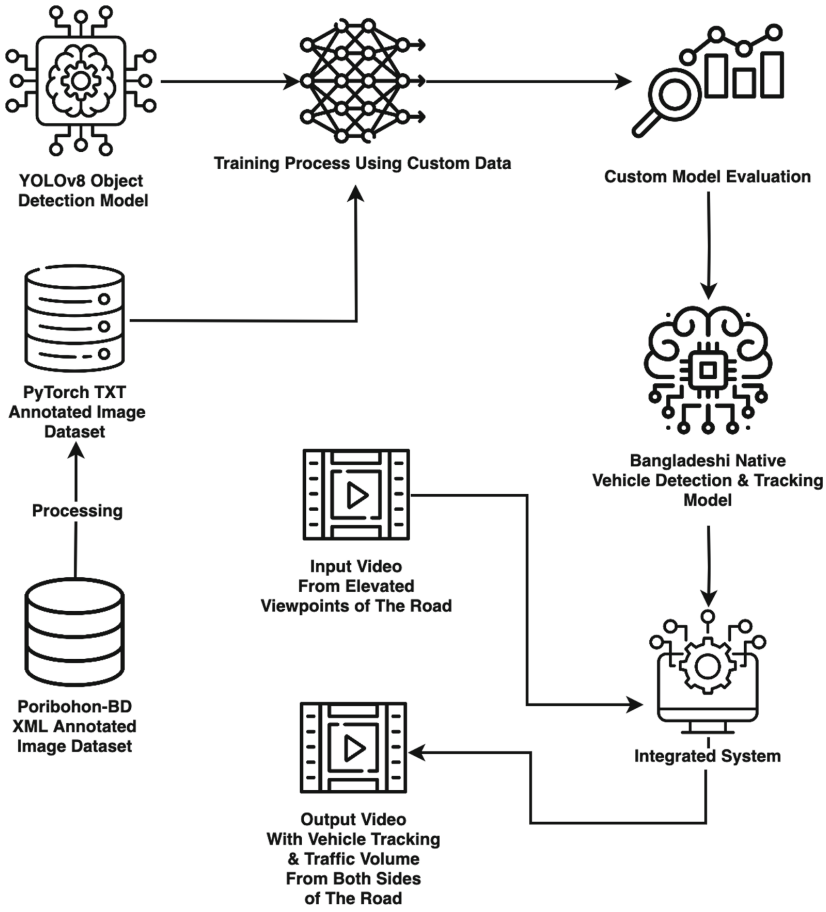

Fig. 1. Overview of the methodology.

3.3 Model Training

Once Dataset is processed, it is used by YOLOv8 object detection model. Basically, a training process is done on the custom dataset. Afterward, the trained model is evaluated. If the evaluation is not satisfactory then the training epochs are increased. In our case, 150 epochs are satisfactory to get our desired output. YOLOv8 algorithm divides an image into a grid. In YOLOv8, we utilized images that were 640 pixels wide by 640 pixels height. YOLOv8 employs a unique approach in which each grid cell spans an area of 32 pixels wide and 32 pixels height, and the number of grid cells can be calculated by dividing 640 by 32. That brings us to 20. It's the same as dividing the image into 20×20 grids (each grid is 32 pixels wide and 32 pixels height). These options may be changed. However, for our study, we left the default settings alone. Each grid cell predicts bounding boxes and confidence scores for objects that may be present in the cell. Non-maximum

suppression is used to remove overlapping bounding boxes, and the output is a list of bounding boxes, confidence scores, and class labels. Using Table 2, we showcase all the parameters, modules, arguments and their corresponding values used for the transfer learning process in different layers. The total breakdown is shown in Table 2. Model Training Summary: 225 layers, 11141405 parameters, 11141389 gradients, and 28.7 GFLOPs(Giga Floating Point Operations per Second). The model transferred 349/355 items from pre-trained weights. The trained model is used to make a system that can take videos as input and give an output video consisting of annotated traffic volume from both sides of the road. The video also showcases the classification, detection, and annotation of native vehicles of Bangladesh.

3.4 YOLOv8 Architecture

As we have trained our model based on YOLOv8, therefore we now want to discuss about YOLOv8's architecture. YOLOv8's architectural framework utilizes key components of deep learning to complete object detection duties. In Fig. 2

Table 2. Model parameters & arguments summary

index	from	n	params	module	arguments
0	−1	1	928	ultralytics. nn. modules . Conv	[3, 32, 3, 2]
1	−1	1	18560	ultralytics. nn. modules . Conv	[32, 64, 3, 2]
2	−1	1	29056	ultralytics. nn. modules . C2f	[64, 64, 1, True]
3	−1	1	73984	ultralytics. nn. modules . Conv	[64, 128, 3, 21]
4	−1	2	197632	ultralytics. nn. modules . C2f	[128, 128, 2, True]
5	−1	1	295424	ultralytics. nn. modules . Conv	[128, 256, 3, 2]
6	−1	2	788480	ultralytics. nn. modules . C2f	[256, 256, 2, True]
7	−1	1	1180672	ultralytics. nn. modules . Conv	[256, 512, 3, 2]
8	−1	1	1838080	ultralytics. nn. modules . C2f	[512, 512, 1, True]
9	−1	1	656896	ultralytics. nn. modules. SPPF	[512, 512, 5]
10	−1	1	0	torch. nn. modules. upsampling. Upsample	[None, 2, Nearest]
11	[−1, 6]	1	0	ultralytics. nn. modules . Concat	[1]
12	−1	1	591360	ultralytics. nn. modules . C2f	[768, 256, 11]
13	−1	1	0	torch. nn. modules. upsampling. Upsample	[None, 2, Nearest]
14	[−1, 4]	1	0	ultralytics. nn. modules . Concat	[1]
15	−1	1	148224	ultralytics. nn. modules . C2f	[384, 128, 11]
16	−1	1	147712	ultralytics. nn. modules . Conv	[128, 128, 3, 21]
17	[−1,12]	1	0	ultralytics. nn. modules . Concat	[11]
18	−1	1	493056	ultralytics. nn. modules . C2f	[384, 256, 11]
19	−1	1	590336	ultralytics. nn. modules . Conv	[256, 256, 3, 2]
20	[−1,9]	1	0	ultralytics. nn. modules . Concat	[1]
21	−1	1	1969152	ultralytics. nn. modules . C2f	[768, 512, 1]
22	[15,18,21]	1	2121853	ultralytics. nn. modules . Detect	[15, [128, 256, 512]

the architecture of YOLOv8 is illustrated in detail [10]. The core, also known as the Backbone, is comprised of convolutional layers that extract relevant image features. The SPPF(Feature (feature selection) level, followed by a sequence of convolutional levels, manages characteristics at multiple dimensions. Simultaneously, the Upsample level boosts the resolution of these characteristic maps. To improve detection precision, the C2f module combines superior-level characteristics with context-based data. The last portion Detection module employs a combination of convolution and linear strata to transform high-dimensional characteristics into the resultant object categories and bounding boxes. The YOLOv8 architectural framework is designed to be quick and resourceful while maintaining superior precision in detection. Regarding the blueprint legend, the rectangles represent the strata, with marks denoting the stratum type (Conv, Upsample, etc.) and any significant parameters (channel count, kernel size, etc.). The directional arrows represent the data transition between strata, with the arrow orientation representing the data movement from one stratum to the following stratum.

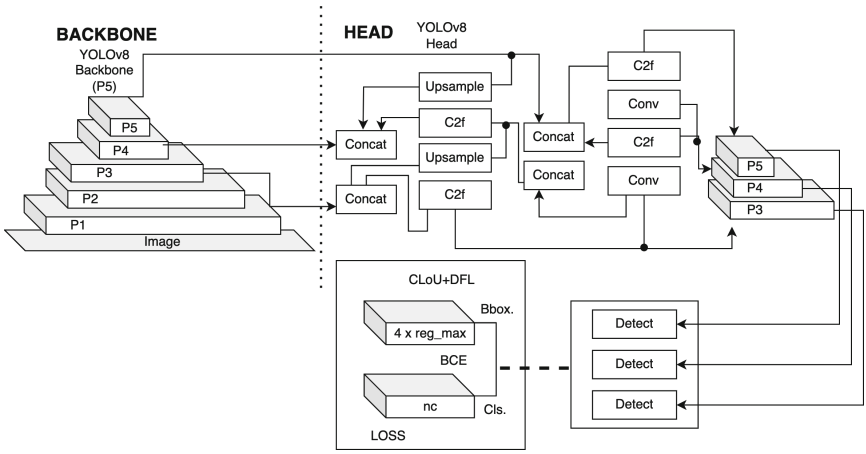

Fig. 2. YOLOv8 Architecture

3.5 Integrated System

After developing the customized YOLOv8 model, we have used the model to create the system that takes any video from the elevated viewpoints of foot over-bridges in Bangladesh and gives native vehicle detection result including the number of vehicles that passes the road. The number of volume indicates the number of vehicles from the both side of the road. Finally, a video consisting of number of vehicles from both side of the road will be the output. The default behavior of our model allows simultaneous detection of up to 300 vehicles. To modify this behavior and enhance detection capabilities, it is imperative

to customize the settings within YOLOv8's detection module before initiating the transfer learning process. It's important to note that achieving improved detection performance through these adjustments is contingent upon employing robust hardware infrastructure. Otherwise, the model will continue to exhibit its default behavior of detecting a maximum of 300 objects. The model can be used to analyze video frames, identifying detected vehicles with timestamps. Tally the vehicle counts for each timestamp to create a dataset. To find the peak vehicle time, analyze the data, potentially using charts or graphs. This reveals the timestamp with the highest vehicle count. In our research, we didn't employ this method, but it's an option for future use.

4 Result Analysis

In this section, the proposed model is evaluated based on multiple factors like confusion matrix, precision-recall curve, and validation set results.

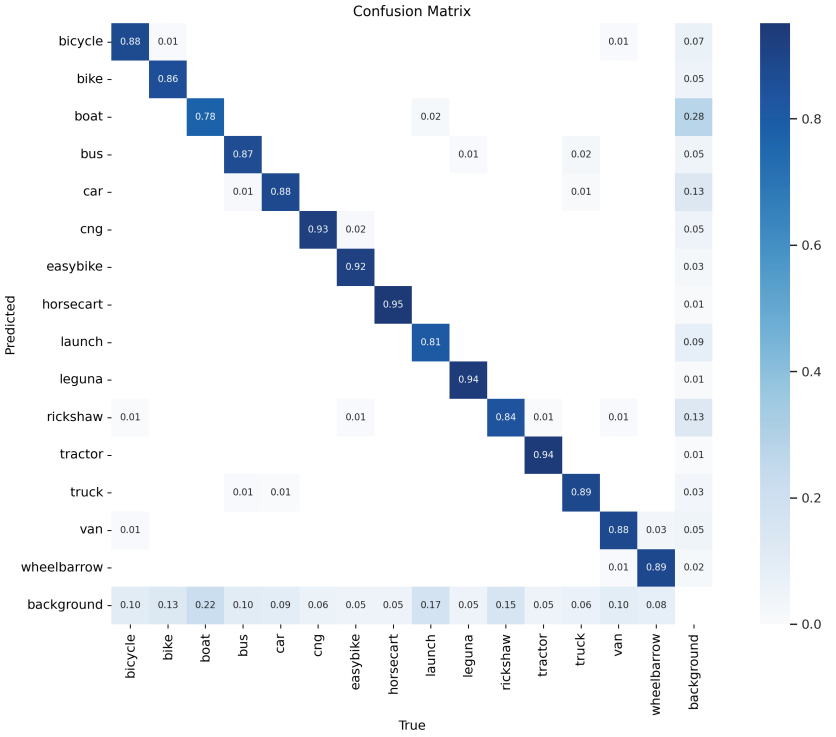

Fig. 3. Confusion matrix of the proposed model.

In the context of object detection algorithms such as YOLO, "mAP @5" refers to the mean average precision at an intersection over union (IoU) threshold of

Table 3. Validation set result.

Class	Instances	Box(P)	R	mAP50	mAP(50–95)
all	5238	0.903	0.854	0.913	0.663
bi-cycle	307	0.885	0.827	0.914	0.62
bike	379	0.913	0.831	0.898	0.634
boat	732	0.79	0.701	0.784	0.446
bus	334	0.902	0.83	0.902	0.667
car	624	0.882	0.848	0.903	0.646
cng	433	0.931	0.91	0.953	0.732
easy-bike	274	0.943	0.912	0.96	0.749
horse-cart	62	0.872	0.935	0.957	0.743
launch	370	0.887	0.759	0.848	0.581
leguna	370	0.931	0.902	0.961	0.807
rickshaw	609	0.872	0.783	0.872	0.584
tractor	102	0.947	0.931	0.961	0.743
truck	341	0.947	0.886	0.93	0.701
van	390	0.919	0.856	0.917	0.692
wheelbarrow	131	0.925	0.893	0.931	0.603

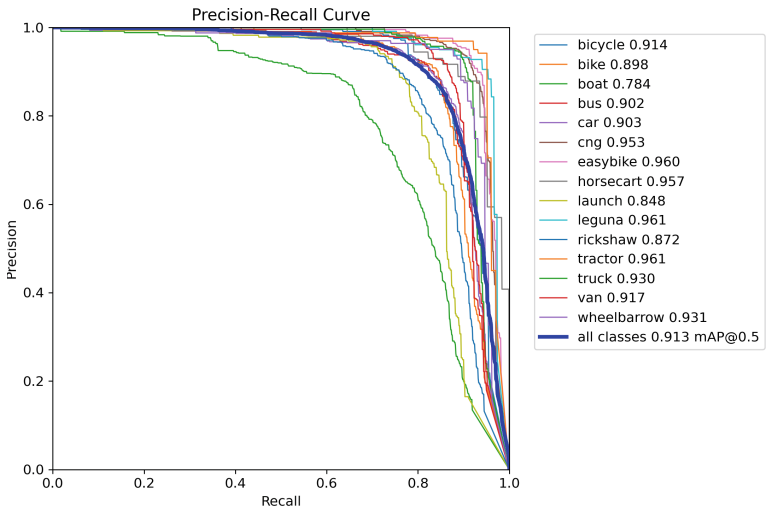

Fig. 4. Precision-recall curve of the proposed model.

Fig. 5. Vehicle Detection by the model

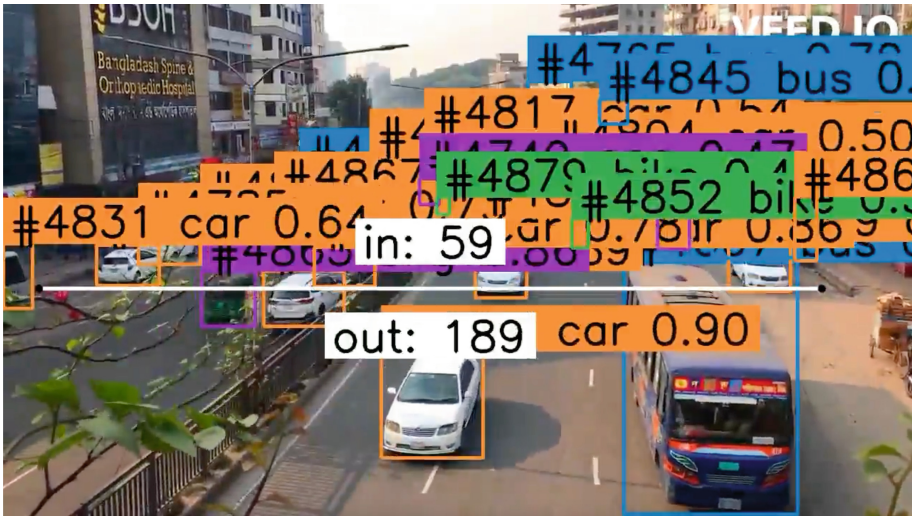

Fig. 6. Vehicle Tracking by the model

0.5. Mean Average Precision (mAP) is a common evaluation metric used to assess the performance of object detection models. It measures both the precision and recall of the model across different classes and IoU thresholds. Table 3 showcases these evaluations of our model on the validation set. In our case, an mAP of 0.913

@5 means that, on average, the object detection model achieved a precision of 0.913 when considering an IoU threshold of 0.5. This indicates that the model's predictions are highly accurate and had a good overlap with the ground truth bounding boxes. It's important to note that the mAP value is typically reported per class, so "all classes" indicates that this performance metric is an average across all the classes in the dataset being evaluated. Figure 5 illustrates the precision-recall curve that showcases the effectiveness of our model in detecting native Bangladeshi vehicles. Also, Fig. 4 showcases the confusion matrix that indicates how much accurate our model is to classify native Bangladeshi vehicles. A classification example using our model is shown in Fig. 6. mAP, IoU, Precision-Recall Curve and Confusion matrix are used to evaluate the model. One thing to consider is that the light change in the morning or afternoon will affect the detection of the vehicles as most of the images used in the training are captured in broad daylight or using flashlight. However, sufficient low light images are also used and the accuracy of the model will decrease slightly which is negligible in real life. Finally, we can say that our model is robust enough to be used in a system that can detect Bangladeshi native vehicles. Eventually, the system takes videos as input and successfully gives an output video consisting of annotated traffic volume from both sides of the road. Finally, a video consisting of number of vehicles from both side of the road is the output of the integrated system. An image is included to showcase the classification, detection, and annotation of native vehicles of Bangladesh in a video. An example of that is showcased in Fig. 6.

5 Conclusion

Our research offers a promising strategy for addressing the expanding problem of traffic congestion in Bangladesh. We have effectively developed a vehicle detection system based on the YOLOv8 model of transfer learning that has been customised for Bangladesh. This system, which is incorporated with elevated viewpoints and video footage analysis, enables the accurate detection and tracking of vehicles, thereby providing valuable information about traffic volume and movement. Through the collection and processing of the Bangladeshi Native Vehicle Image dataset, we trained the model to attain a remarkable mean average precision (mAP) of 91.3%. The YOLOv8 model utilized in this research is a potent instrument for detecting objects in images or videos. It is intended to derive significant image features using convolutional layers. Then, these features are analyzed at various levels in order to capture specific characteristics and improve their resolution. In addition, the model includes a module that integrates various features to enhance detection precision. Then the model is used in the vehicle detection system. The capability of our system to detect local vehicles in video footage and to provide traffic volume data, including vehicle counting, is a significant step toward enhancing traffic management in Bangladesh. This instrument can aid the Bangladeshi Road Transport Authority in regulating traffic and addressing the challenges posed by growing traffic congestion. Our

research proposes a practicable and efficient method for constructing an effective transportation network in Bangladesh. By utilising advanced technology and deep learning algorithms, we can make informed decisions to reduce congestion, optimise traffic flow, and enhance the nation's transportation infrastructure. In the future, we are hopeful to utilise the mentioned model in this paper in real-time vehicle detection system in Bangladesh. We also believe, our study will help researcher to build a proper vehicle classification and detection system to tackle the issue of traffic congestion in their respective countries.

Acknowledgements. We appreciate the support for this research received from the a2i Innovation Fund of Innov-A-Thon 2018 (Ideabank ID No.: 12502) from a2i-Access to Information Program - II, Information & Communication TechnologyDivision, Government of the People's Republic of Bangladesh and Institute for Advanced Research (IAR), United International University (Project Code: UIU/IAR/01/2021/SE/23).

References

1. Baheti, P.: What is transfer learning? [examples & newbie-friendly guide] (2021). https://www.v7labs.com/blog/transfer-learning-guide
2. Chen, W., Sun, Q., Wang, J., Dong, J.J., Xu, C.: A novel model based on AdaBoost and deep CNN for vehicle classification. IEEE Access **6**, 60445–60455 (2018)
3. Fan, Q.: Toward great dhaka. World Bank (2017). https://www.worldbank.org/en/news/speech/2017/07/19/toward-great-dhaka
4. Hasan, M.M., Wang, Z., Hussain, M.A.I., Fatima, K.: Bangladeshi native vehicle classification based on transfer learning with deep convolutional neural network. Sensors **21**(22), 7545 (2021)
5. IBM: What is deep learning? (nd). https://www.ibm.com/topics/deep-learning
6. Maungmai, W., Nuthong, C.: Vehicle classification with deep learning. In: 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), pp. 294–298. IEEE (2019)
7. Neupane, B., Horanont, T., Aryal, J.: Real-time vehicle classification and tracking using a transfer learning-improved deep learning network. Sensors **22**(10), 3813 (2022)
8. Roboflow: Give your software the power to see objects in images and video (2023). https://roboflow.com/
9. Sakib, S.N.: Bangladesh loses (2021). https://www.aa.com.tr/en/asia-pacific/bangladesh-loses-40-of-fuel-due-to-poor-traffic-management/2449934
10. Solawetz, J.: What is YOLOv8? the ultimate guide (2023). https://blog.roboflow.com/whats-new-in-yolov8/
11. Tabassum, S., Ullah, M., Al-Nur, N., Shatabda, S.: Native vehicles classification on bangladeshi roads using CNN with transfer learning (2020). https://doi.org/10.1109/TENSYMP50017.2020.9230991
12. Tabassum, S., Ullah, M., Al-Nur, N., Shatabda, S.: Poribohon-BD: bangladeshi local vehicle image dataset with annotation for classification. Data in Brief **33**, 106465 (2020). https://doi.org/10.1016/j.dib.2020.106465
13. Ultralytics: revolutionizing the world of vision AI (2023). https://ultralytics.com/
14. Vijayaraghavan, V., Laavanya, M.: Vehicle classification and detection using deep learning. Int. J. Eng. Adv. Technol. **9**(1S5), 24–28 (2019)

15. Yu, S., Wu, Y., Li, W., Song, Z., Zeng, W.: A model for fine-grained vehicle classification based on deep learning. Neurocomputing **257**, 97–103 (2017)
16. Zhao, D., Chen, Y., Lv, L.: Deep reinforcement learning with visual attention for vehicle classification. IEEE Trans. Cogn. Dev. Syst. **9**(4), 356–367 (2016)
17. Zhou, Y., Cheung, N.: Vehicle classification using transferable deep neural network features. CoRR abs/1601.01145 (2016). http://arxiv.org/abs/1601.01145
18. Zhou, Y., Nejati, H., Do, T.T., Cheung, N.M., Cheah, L.: Image-based vehicle analysis using deep neural network: a systematic study. In: 2016 IEEE International Conference on Digital Signal Processing (DSP), pp. 276–280. IEEE (2016)