REAL-TIME THREE-WHEELER TRAFFIC DETECTION ON EXPRESSWAYS USING YOLO MODEL

Rahat Morshed Nabil*[1], Tarunnyamoye Kundu[2] and Uday Karmoker[2]

[1] *Undergraduate Student, Khulna University, Bangladesh, e-mail: imrmnabil@gmail.com*
[2] *Undergraduate Student, Khulna University, Bangladesh, e-mail: tarunnyamoye1917@gmail.com*
[3] *Undergraduate Student, Rajshahi University of Engineering & Technology, Bangladesh, e-mail: karmok3r@gmail.com*

***Corresponding Author**

ABSTRACT

In recent years, the development of intelligent transportation systems (ITS) has become increasingly critical to ensure safe and efficient traffic management on expressways. One of the key challenges in this domain is the precise and real-time detection of various types of vehicles including truck, car, motor-bike, three-wheelers, rickshaws, pedestrians etc. Among these three-wheelers, those are the core concern of this study which are commonly used for transportation in many regions. This paper presents a novel approach for real-time three-wheeler traffic vehicle detection on expressways using the You Only Look Once (YOLO) algorithm i.e., YOLOv8. Expressways are primarily designed for conventional vehicles such as cars, buses, minibuses, microbuses, and various sizes of trucks. However, three-wheelers like CNG vehicles, electrical vehicles, auto rickshaws etc. often lead to undesirable accidents. To confront the complexities of identifying three-wheelers in real-world traffic scenarios, a multifaceted approach was adopted. The study began with the extensive collection of data from Rajshahi City Corporation area specifically from Zero Point, Rajshahi, followed by meticulous annotation of this dataset. Such as blurring, cropping, rotating, scaling, flipping. Moreover, the data were collected by the authors on different light conditions i.e., sunny, gloomy etc. Subsequently, arigorous model training was conducted, which was then complemented by comprehensive model testing to ensure its efficacy. The culmination of these efforts led to the successful deployment of our custom-trained YOLO model on real-time video footage from roadside CCTV cameras. In practical the police box near entering road or expressways can use this trained model as a module. YOLO, being an open-standard-based algorithm, serves as a foundational pillar upon which sustainable solutions within the realm of smart transportation can be built. Notably, it has been observed that 88.2% accuracy can be gained by implementing YOLOv8m model which was 83.6% in case of YOLOv8n.

Keywords: *YOLO, three-wheeler, traffic management, ITS, expressway*

1. INTRODUCTION

In 2008, Bangladesh introduced a new and popular addition to its urban transportation system: battery-operated Electrical vehicle, fondly known as Easy-bikes, Auto-rickshaw, Three-wheeler etc. Just if we pull an example, with over 15,000 Easy-bikes and 7,000 auto-rickshaws zipping around the Rajshahi Metropolitan area (Basri et al. 2014). These vehicles have gained favour among urban commuters due to their lower travel costs and the reasonable safety and comfort they provide. However, a challenge arises as these electric vehicles (EVs) lack a proper braking system, with only the front wheels equipped with brakes. This issue raises concerns about safety, leading to the prohibition of these EVs on expressways due to the potential for accidents. Recent data reveals that a significant portion of accidents in Bangladesh, around involve auto-rickshaws and three-wheelers.

Table 1: Increasing Rate of Auto for Past 5 years (Basri et al. 2014)

Year	No. of Auto	Population (Million)	No of Auto for Every 1000 People
2011	6000	0.65	0.92
2012	7500	0.68	11.00
2013	9500	0.72	13.19
2014	12000	0.77	15.85
2015	15000	0.81	18.51

Starting with expressways, the speed limit is 80 km/h. So, the EVs being slow moving vehicle are not allowed to use expressways. In Bangladesh a developing country is gaining focus on constructing expressways starting from Dhaka elevated expressway (Asad, 2023). In recent days more will be built to reduce traffic congestion. In Bangladesh, where EVs are so popular, there is a chance of EVs entering the expressways. The entrance of EVs into expressways can occur accidents due to uneven speed. There is a minimum speed on expressways to be maintained that’s why EVs cannot lie in the same line.

Table 2: Speed Limits of Expressways and Urban Roads in Different Countries

	Ethiopia	Bangladesh	Brazil	USA
Urban roads	Not specified	40-50 km/h	60 km/h	40 km/h
Expressways	100-120 km/h	80 km/h	120 km/h	113-137 km/h

The traffic signal with CC cameras can detect the EVs while entering the expressway. The automated traffic control can be achieved through this. As an integral component of the intelligent transportation system (ITS), an effective real-time vehicle counting method is a crucial gamechanger for the expressway management department to implement traffic control measures and enhance safety. The expressway's high traffic flow necessitates advanced techniques for Easy-bike detection, tracking, and counting. There are available datasets fort vehicle detection such as KITTI (Geiger et al., 2012), PASCAL VOC (Everingham et al., 2010), LISA-2010 (Sivaraman et al., 2010), MS-COCO (Lin et al., 2014) etc. However, these datasets do not include the South Asian i.e., Indian or Bangladeshi Three-Wheeler in account. While traditional machine learning and classification methods have been widely used, recent advancements in computer vision have ushered in deep learning as the mainstream approach. This research paper focuses on a novel three-wheeler vehicle counting method for expressway management. Leveraging the You Only Look Once (YOLO) model, trained on a dataset derived from urban road video sequences, this method aims to enhance the accuracy of easy-bike detection. Unlike natural images, the expressway dataset caters specifically to the challenges posed by vehicle counting, addressing the unique characteristics of easy-bikes in the urban transportation landscape.

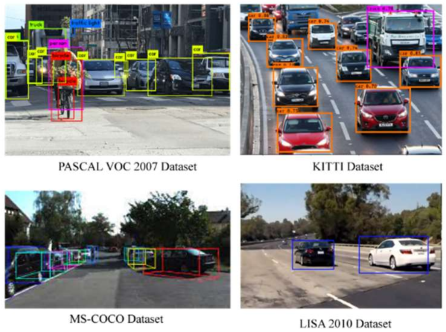

Figure 1: Existing Dataset

Figure 2: Our Target Dataset

YOLOv8 is the version of YOLO by Ultralytics released in 2023. Which is a significent imporvement over its previous versions e.g., YOLOv5. YOLOv8 demonstrates impressive accuracy on the COCO which is a large-scale object detection, segmentation, and captioning dataset. (Lin, T. Y. et al.,2014)

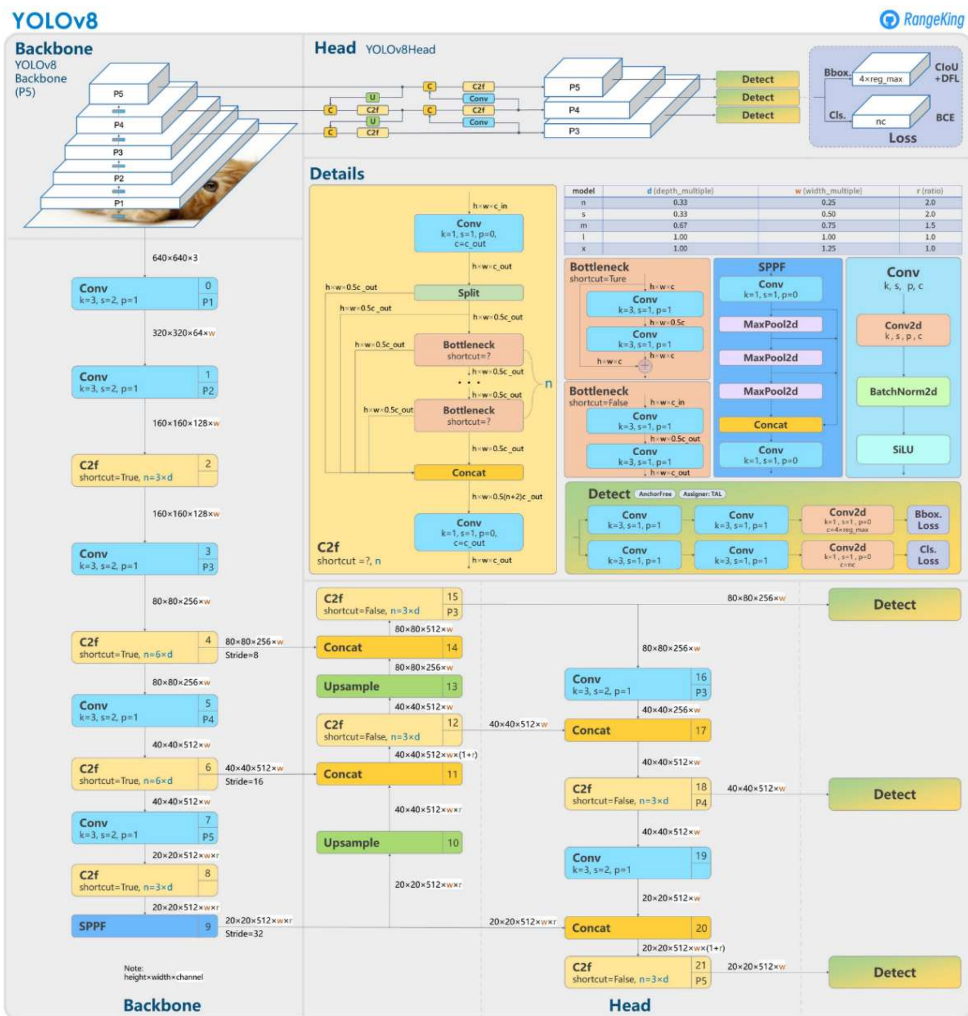

Figure 3: YOLOv8 Architecture (Jacob et al. 2023)

Specifically, the YOLOv8m model, categorized as the medium variant, attains a notable 50.2% mean Average Precision (mAP) on COCO. Notably, when assessed against Roboflow 100, a dataset designed to assess model performance across diverse task-specific domains, YOLOv8 outperforms YOLOv5 by a significant margin. In table 3, a comparison between YOLOv5 & YOLOv8 about the performance of the average precision (AP) is shown. From the table, it is obvious that for each model of YOLOv5 and YOLOv8 the later one is 6.31% to 33.21% faster in performance. This led us to prioritize YOLOv8 to YOLOv5.

Table 3: Object detection performance comparison between YOLOv5 & YOLOv8 (YOLOv8 Vs. YOLOv5: Choosing the Best Object Detection Model, n.d.)

Model Size	YOLOv5	YOLOv8	Difference
Nano	28	37.3	+33.21%
Small	37.4	44.9	+20.05%
Medium	45.4	50.2	+10.57%
Large	49	52.9	+7.96%
Xtra Large	50.7	53.9	+6.31%

Additional details on this performance comparison are elaborated in our subsequent performance analysis within this article (Jacob et al. 2023). While YOLOv8 currently lacks a published paper, withholding direct insight into its research methodology and ablation studies during development, we have undertaken an analysis of the repository and available information about the model to initiate documentation on the novel aspects of YOLOv8. Figure 4 demonstrates the practical applicability of the model trained. Being fed the data from urban busy roads, the model implemented as a detection module can be used in the police check posts at the ends of expressways.

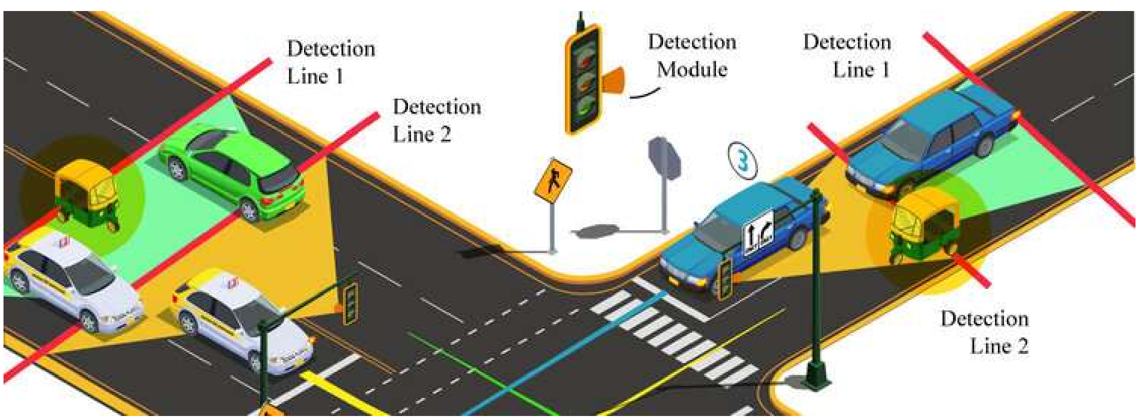

Figure 4: Real-time applicability of three-wheeler detection module

2. OBJECTIVE

The core objective of this study is to,

1. Develop a real-time three-wheeler traffic vehicle detection system for expressways.
2. Focus on the prevention of undesirable accidents caused by three-wheelers on expressways.
3. Utilize the You Only Look Once (YOLO) algorithm, specifically YOLOv8, as the core methodology for detection.

4. Validate the model's performance through deployment on real-world video footage, demonstrating its practical applicability and robustness.
5. Highlight the achieved accuracy.

3. METHODOLOGY

The methodology employed in this study involved a multifaceted approach. Extensive data collection from the Rajshahi City Corporation area, including Zero Point, provided a diverse dataset for training. Meticulous annotation, incorporating techniques like blurring and scaling, enhanced the dataset's representativeness. Rigorous model training and comprehensive testing, complemented by real-time deployment on roadside CCTV footage, validated the effectiveness of the custom-trained YOLOv8 algorithm in real-world three-wheeler traffic detection scenarios. This study was conducted following this flow diagram.

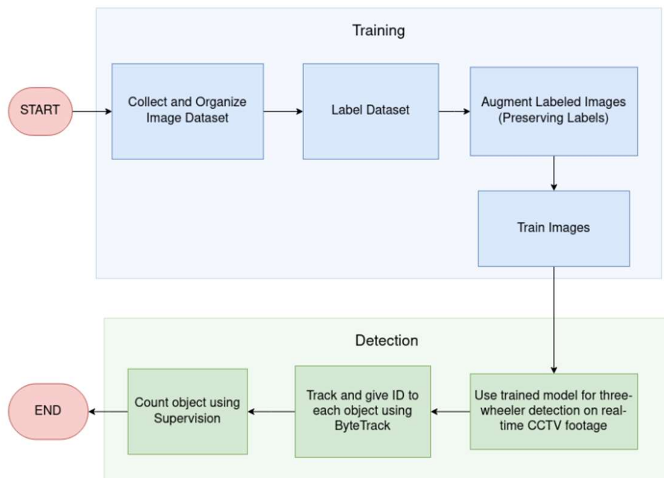

Figure 5: Flow Diagram of Working Procedure; Training and Detection

3.1 Data Collection

Three-wheeler detection, the images were collected from different places in Rajshahi city in different light conditions. The places were preferably busy urban roads. Around 700 number of images were taken. The images contain various types of vehicles like cars, autorickshaws, rickshaws, buses, motorcycles etc. Also, a large number of pedestrians were present in those images. The repeated images were filtered out to remove redundancy, as this removal makes the dataset stronger for the model.

3.2 Data labeling

We utilized the open-source software 'LabelImg' (HumanSignal, 2022) to annotate the images, with a specific focus on detecting three-wheelers. To achieve this, a single class was employed for labeling, facilitated by the YOLO object detection model's support for such labeling tasks.

3.3 Data augmentation

To enhance the dataset's sample size, we employed the data augmentation principles of computer vision, primarily benefiting the training of our model. This approach proves advantageous not only in model training but also holds significant utility in data detection tasks. In the course of our ongoing project, we encountered challenges such as occlusion and variations in viewpoint. To mitigate these issues, we implemented various techniques, elaborated upon in the subsequent discussion. In this study we used an open-source Python library 'Albamentations' to generate various random augmented image data preserving bounding box values from actual labeled image set (Buslaev et al. 2020).

1. Blurring: Different light conditions can also hamper the accuracy of object detection. Thus, we applied blurring so that the model learns to detect objects under poor light conditions. Blur function of the Albumentations library was used with the value of probability parameter p = 0.5. Here the value of the probability parameter ‘p’ means the probability of the function being perform its task during augmented image generation. The higher the value is, the higher the probability the function works on an image. Here, p = 0.5 indicates that there is a 50% chance for each image to undergo blurring.
2. **Flipping:** The images were horizontally flipped. This was done to make the model learn to detect three-wheelers from both sides. To implement this action the HorizontalFlip function was used with the value of probability parameter p = 0.5. (Albumentations-teams, 2023)
3. **Cropping:** Sometimes, the object that we are trying to detect can be partially obscured. To limit the effect of this occlusion cropping was done. The RandomCrop function was used to randomly crop different portions of images. (Albumentations-teams, 2023)
4. **Rotating:** Another issue that was mentioned is the viewpoint variation. Rotating was done to make the model learn about different viewpoints of the three-wheelers. To rotate images randomly in an angle, range from -30 degree to 30 degree the Rotate function was used with parameters limit = [-30,30] and p = 0.5. (Albumentations-teams, 2023)
5. **Scaling**: Scaling in order to help the model learn about three-wheeler vehicles of different sizes, scaling was done to alter the size. The scaling was done in a random manner using the RandomScale function. (Albumentations-teams, 2023)

Though the library that was used here augmented bounding boxes also, some minor issues of bounding boxes that were automatically generated for augmented images were fixed.

3.4 Training Image Data

Initially, the labelled dataset was divided into two segments. Approximately 80% of the dataset's images were allocated for training purposes, while the remaining 20% were designated for the testing dataset. The training process involved the use of a pre-trained model named 'yolov8m.pt.' Training, in this context, refers to imparting knowledge to the deep learning model about the distinctive features of three-wheelers. The YOLOv8 model relies on a deep learning framework, such as PyTorch, which provides essential tools and libraries for constructing and training neural networks. YOLOv8 operates by dividing the input image into a grid of cells, predicting bounding boxes and class probabilities for each cell. Subsequently, a non-maximum suppression (NMS) algorithm is employed by YOLOv8 to filter out overlapping bounding boxes, selecting the most relevant ones for each object in the image. It is pertinent to mention that certain configuration parameters for YOLOv8 were set during the training process.

Table 4: YOLOv8 Configuration Parameters

Parameters	Values
Epoch	300
Learning Rate	0.01
Image Size	640
Batch Size	8
Number of images	1637

3.5 Detection using trained model

Once training was finished, it was applied to new video footage containing scenes with three-wheelers. The video Footage was collected from different busy roads of Rajshahi city. For detection, the YOLOv8 object detection model was used with a confidence value of 0.5. The algorithm is capable of detecting objects in real-time footage.

3.6 Tracking of detected object using ByteTrack

In order to ensure the precise counting of each individually detected object, it is imperative to implement object tracking. This system employs the ByteTrack object tracking algorithm, specifically designed to track three-wheelers that have been detected individually. Through this algorithm, each three-wheeler is assigned a unique identity, facilitating accurate tracking and counting of these objects. (Zhang, 2022)

3.7 Count tracked vehicles using supervision

Within this system, there were two horizontal lines marked on the video footage using supervision (RoboFlow, 2023). The program incorporated two counting variables, UP_COUNT and DOWN_COUNT, designed to increase with each vehicle crossing these lines. The logic was structured such that if a tracked object intersected the upper line before the lower line, the system identified the vehicle as approaching the camera, leading to an increment in UP_COUNT by 1. Conversely, if the vehicle touched the lower line first, followed by the upper line, it was recognized as moving away from the camera, resulting in an increment of 1 in DOWN_COUNT.
By employing these steps, we assessed the overall count of vehicles entering and exiting a road based on the video footage.

4. EXPERIMENTAL RESULTS AND DATA ANALYSIS:

In this section used dataset, performance measurement and result estimation are discussed.

4.1 System Setup

The computation and system specification for our study was as following.

CPU: AMD Ryzen 5 3500X 6-Core Processor (3938 MHz)
Graphics Card: Nvidia GeForce GTX 1650 SUPER (4GB VRAM)
Main Memory: 8GB (3200MHz)
STORAGE: 128GB (SSD)
Operating System: Fedora 39 (Linux based distribution)

4.2 Used dataset

All the experiments in this study performed using the YOLO algorithm with the help of Ultralytics YOLOv8 model and the programs are written in Python 3.8 in anaconda environment. In the training process automatic mixed precision (amp) was used which decreased the memory usage significantly in training process so that larger batch size can be used. To train the model 1637 images with 7320 three-wheeler objects are used. The dataset was split into 80% training and 20% validation sets. All the images were collected by the authors from streets of Rajshahi where three wheelers are very available.

4.3 Performance measurement

In this study, the training of the proposed model was executed using the 'yolov8m.pt' model as the foundation. For traffic detection with decent performance we have prioritize two characteristics, speed and precision.

Table 5: Comparision among different YOLOv8 Models (Ultralytics, 2023)

Model	**Size** (pixels)	**mAP^{val}** 50-95	**Speed** CPU ONNX (ms)	**Speed** A100 TensorRT (ms)	**Parameters** (M)	**FLOPs** (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

From table 5, we can see that for YOLOv8n, the speed is maximum which is 80.4 ms. On the other hand, the precision is best found with the extra-large model, YOLOv8x. It is justified by the mAP(50-95) value 53.9. But the speed is very low for the large dataset. Traffic cameras and detection module is not as fast as it demands. So, for a real time detection of three wheelers, we chose a model in between these. i.e., YOLOv8m. For a performance comparison we have taken the nano model YOLOv8n.

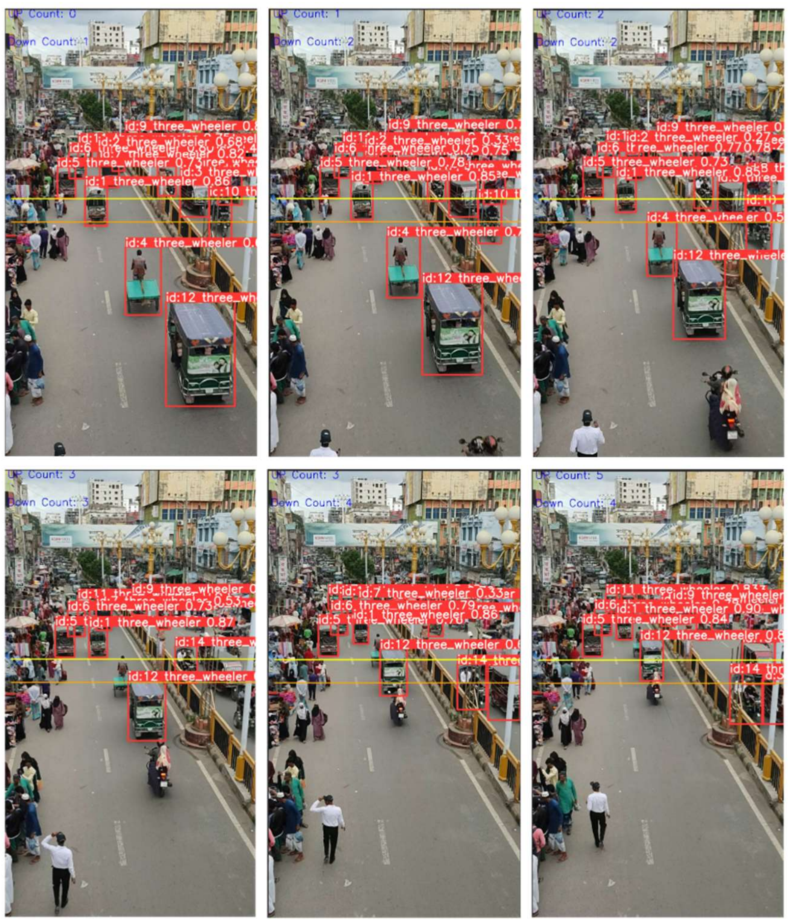

Figure 6: Detection of Three-wheelers on a busy urban road at Saheb Bazar Zero Point, Rajshahi, Bangladesh

To facilitate a thorough comparative analysis, an additional training process was carried out utilizing the 'yolov8n.pt' on the identical dataset. This approach allows us to assess and contrast the performance of our proposed model against the alternative training configuration, providing valuable insights into the impact of model variations on the outcomes. This deliberate comparison serves to enhance the robustness of our findings and contributes to a more comprehensive understanding of the model's behavior under different training conditions.

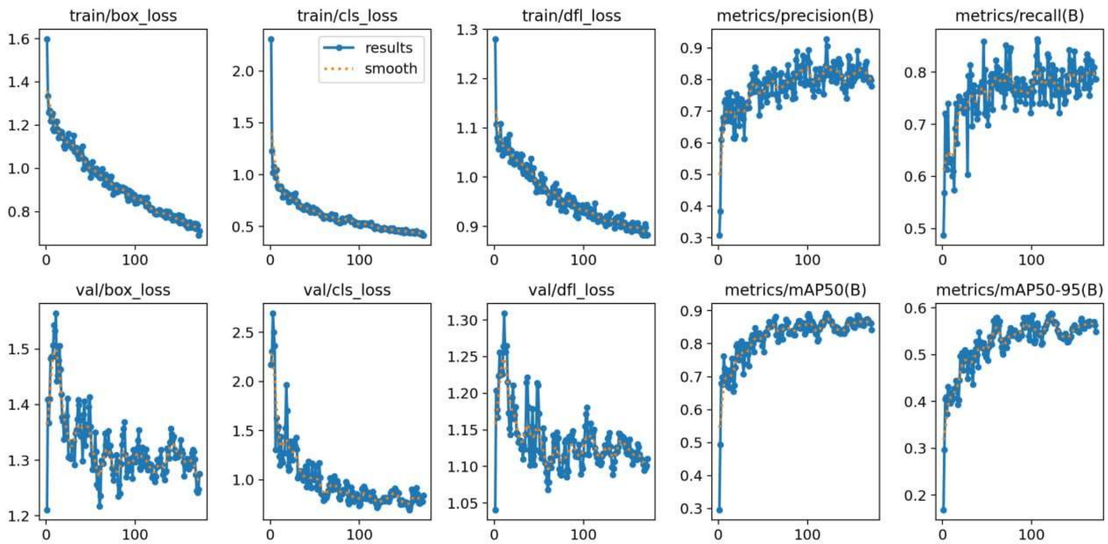

Figure 7: Testing Results Using 'yolov8m.pt Pretrained Weight'

Figure 7. shows the overall result that includes box loss, class loss, recall and precision of the proposed model trained with ‘yolov8m.pt’. The Precision-Recall curve in Figure 9. shows that the mean average precision (mAP) of the trained model is 88.2%.

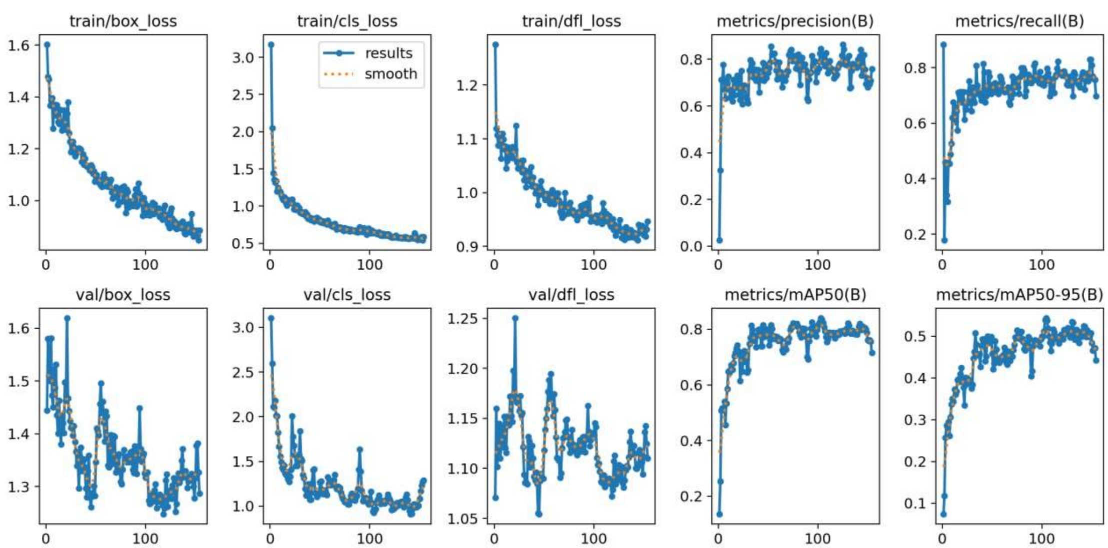

Figure 8: Testing Results Using 'yolov8n.pt Pretrained Weight'

Figure 8. shows the overall result that includes box loss, class loss, recall and precision of the model trained with ‘yolov8n.pt’. The Precision-Recall curve in Figure 10. shows that the mean average precision (mAP) of the trained model is 83.6%.

The true positive rate is determined by the ratio of correct positive predictions to the total number of actual positives, as outlined in Eq. (1). Precision, on the other hand, is computed as the ratio of correct positive predictions to the total number of positive predictions, as described in Eq. (2). (Precision-Recall, n.d)

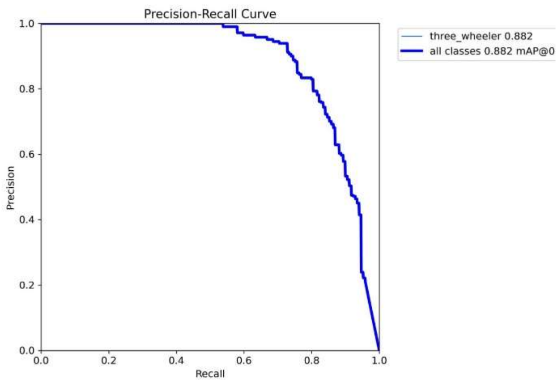

Figure 9: PR curve for 'yolov8m.pt'

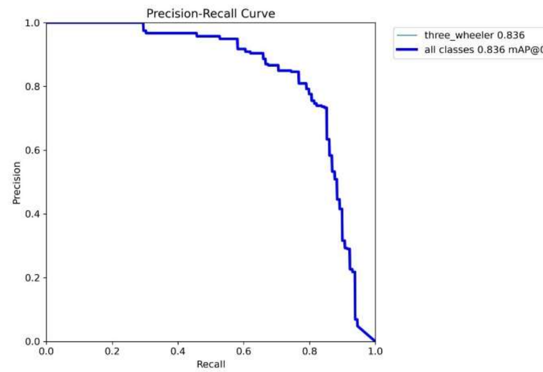

Figure 10: PR curve for 'yolov8n.pt'

$$\mathbf{Recall} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \tag{1}$$

$$\mathbf{Precision} = \frac{\mathbf{TP}}{\mathbf{TP+FN}} \tag{2}$$

$$\mathbf{FM} = \mathbf{2} * \frac{\mathbf{Recall*Precision}}{\mathbf{Recall+Precision}} \tag{3}$$

In the context where TP represents True Positives, FP denotes False Positives, TN signifies True Negatives, and FN stands for False Negatives, F-Measure (abbreviated as FM and calculated as per Equation 3) incorporates both Precision and Recall in its computation.

5 CONCLUSIONS

This paper proposed an improved three-wheeler detection approach on the YOLOv8 algorithm. It can further be expanded to detect the delay due to the passenger onboarding de-boarding. Counting the daily traffic volume on a particular highway or expressway. A real-time footage from traffic CCTV camera can be fed into the algorithm for a real time count. From Figure 8 and Figure 9 it can be observed Precision jumped from 83.6% to 88.2%. In YOLOv8 algorithm there are versions ranging from Nano to X-Large. This study started with YOLOv8n (where ‘n’ stands for nano) that was 83.6% precise. After choosing the YOLOv8m (where ‘m’ stands for medium) the precision was improved 4.6%. We are expecting near about 90% in large models, but this could result in a delay in processing.

In brief we can say that,

1. This study has developed a real-time three-wheeler traffic vehicle detection system for expressways.
2. The trained model can the prevent undesirable accidents caused by three-wheelers on expressways by pre-detecting three wheelers before entering the expressway.
3. A proper utilization of YOLO algorithm, specifically YOLOv8, as the core methodology for detection was done
4. Validation of the model's performance through deployment on real-world video footage, was examined comparing YOLOv8n and YOLOv8m

5. Finally, accuracy of 88.20% was achieved and highlighted

As we are focused in real-time detection, a small delay impacts a lot. This study has some limitations and shortcoming too.

1. This trained model performs slow in case of low light condition
2. Extensive traffic can hamper the detection of this model

The model can be trained with more data fed. This can eventually lead to better precise model. Scopes are open for further studies for improving the dataset model

REFERENCES

Albumentations-teams (07 Dec 2023) Albumentations. GitHub. https://github.com/albumentations-team/albumentations

Asad Abedin Joy (02 Sep 2023) Elevated expressway ushers in new era of connectivity in Dhaka. Dhaka tribune. https://www.dhakatribune.com/bangladesh/324208/elevated-expressway

Basri, R., Khatun, T., Reza, M. S., & Khan, M. M. H. (2014). Changing modes of transportation: A case study of rajshahi city corporation. Bangladesh J. Pol. Econ, 29(3), 1-23.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: fast and flexible image augmentations. Information, 11(2), 125.

Eames, B. K., & Grover, R. (2010). Formalizing functional flow block diagrams using process algebra and metamodels. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 41(1), 34-49.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. International journal of computer vision, 88, 303-338.

Geiger, A., Lenz, P., & Urtasun, R. (2012, June). Are we ready for autonomous driving? the kitti vision benchmark suite. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3354-3361). IEEE.

HumanSignal (23 Sept 2022) labelImg. GitHub. https://github.com/HumanSignal/labelImg

Jacob Solawetz, Francesco (11 Jan 2023) What is YOLOv8? The Ultimate Guide. Robo Flow.
https://blog.roboflow.com/whats-new-in-yolov8

Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13 (pp. 740-755). Springer International Publishing.

Precision-Recall. scikit-learn
https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall

Roboflow (Dec 2023) supervision. GitHub. https://github.com/roboflow/supervision

Sivaraman, S., & Trivedi, M. M. (2010). A general active-learning framework for on-road vehicle recognition and tracking. IEEE Transactions on intelligent transportation systems, 11(2), 267-276.

Ultralytics (28 Dec 2023) Ultralytics. GitHub. https://github.com/ultralytics/ultralytics

YOLOv8 vs. YOLOv5: Choosing the Best Object Detection Model. (n.d.). https://www.augmentedstartups.com/blog/yolov8-vs-yolov5-choosing-the-best-object-detection-model

Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., ... & Wang, X. (2022, October). Bytetrack: Multi-object tracking by associating every detection box. In European Conference on Computer Vision (pp. 1-21). Cham: Springer Nature Switzerland.