Reducing Traffic Congestion Using Real-Time Traffic Monitoring with YOLOv8

Sameerchand Pudaruth[1], Irfaan Mohammad Boodhun[2], Choo Wou Onn[3]
ICT Department-Faculty of Information-Communication & Digital Technology,
University of Mauritius, Moka 80837, Mauritius[1,2]
Faculty of Data Science and Information Technology, INTI International University,
Persiaran Perdana BBN, Putra Nilai, Nilai 71800, Malaysia[3]

***Abstract*—The voluminous number of vehicles present on principal roads together with ongoing road expansion projects are triggering serious roadblocks during peak hours in many places in Mauritius. Consequently, an innovative solution has been proposed using the strength of deep learning neural networks and cutting-edge computer vision methodologies to help reduce this problem. The idea is to create a reliable system that is adequate to measure traffic density and traffic flow on important roads of Mauritius in real-time. A dataset of 2800 frames was collected and used to train and test the YOLO models. A setup was designed for detecting, tracking and counting vehicles such as buses, cars, motorbikes, trucks and vans. Relevant traffic information from videos can also be retrieved to generate statistics for traffic density. Moreover, the system can estimate individual speed of vehicles as well as determining traffic flow on bidirectional roads. The overall mean counting accuracy was 96.1% and the overall mean classification accuracy was 94.4%. For traffic flow, the overall mean accuracy was 93.9%, while traffic density was estimated with an overall mean accuracy of 95.3%. In comparison with manual approaches used in Mauritius to understand the state of traffic, the proposed system is a modern, low-cost and effective solution that can adopted to potentially reduce traffic congestions and traffic accidents.**

Keywords*—*Computer vision; deep learning; vehicle detection and tracking; traffic accidents; traffic congestion

I. Introduction

Traffic congestion is a persistent threat for the long-term sustainability of transport networks. Whenever travel volume surpasses the capacity of roads, reduction in normal flow of vehicles happens, causing major time delays, rise in financial losses and significant stress for drivers [1]. Unfortunately, roadblocks are a recurring issue worldwide capable of even causing motionlessness for several days as occurred in Beijing, China on August 2010 [2]. Traffic congestion is a term that can be quantified by measuring its severity, duration and extent. Severity shows the intensity of congestion, duration represents the interval that the traffic network was affected and extent being the length of road being compromised [3]. Furthermore, studies show that after drivers are stuck in traffic, they tend to be more reckless as well as being easily distracted, potentially leading to accidents [4]. Although Mauritius is known for being a paradise island, its habitants are far from being exempted from the daily struggles of traffic jam. According to reports by the Road Development Authority (RDA) in Mauritius, a significant portion of the budget allocated for road projects has been spent to decongest roads in several parts of the country [5]. In Mauritius, it is currently impossible to obtain reliable information about the state of traffic on major roads at any time. Some genuine attempts have been made to understand traffic density by placing individuals at different roads and manually counting moving vehicles. However, the growth of the transportation network has made the method extremely expensive and unsustainable. Moreover, these manual methods do come with a degree of inaccuracy due to human errors.

Frequent travel delays is costly in the long run. Thus, an automated traffic density estimation system is needed to relieve pressure on road users. As the increase in number of vehicles being purchased yearly is not coming to a halt, the concentration of vehicles on major streets during peak hours does raise a safety concern that needs to be tackled immediately. Having a reasonable but limited budget, Mauritius requires relevant and timely statistics to make optimal choices in the ongoing road decongestion program. The goals of this project are to take advantage of computer vision techniques and neural networks to understand the state of traffic from video footages. Firstly, a large number of video recordings of moving vehicles on important roads of Mauritius will be taken. The vehicles present in the frames will be meticulously labelled to form a robust dataset for training, validation and testing. After the training phase, the system shall be able to detect, track and count moving vehicles from video frames with reasonable accuracies. Then, with the help of appropriate algorithms, traffic flow, traffic density and driving speed will be estimated. Finally, an output video containing all the essential information about road traffic analysis will be generated. The system will be able to provide real-time traffic data, allowing authorities and drivers to make informed decisions on routes, reducing congestion. With the availability of real-time data, authorities can predict future traffic patterns and manage resources effectively, such as deploying traffic police or adjusting traffic signals.The system will be cheaper and more sustainable in the long run compared to manual methods.

This paper proceeds as follows. In the next Section II, we present the related works, the object detection and object tracking algorithms. The methodology is described in Section III. The implementation, testing, results and evaluation are presented in Section IV. Section V concludes the paper.

II. Literature Review

This section presents a comprehensive overview of the existing research on vehicle counting, road traffic density evaluation and vehicle speed estimation. Mandal and Adu-Gyamfi [6] built algorithms to count and track vehicles. They developed a reliable approach for vehicle counting on highways and addresses the occlusion issue of small vehicles behind trucks. CenterNet, Detectron2, YOLOv4 and EfficientDet were the four object detectors that were used. Intersection over union (IOU) tracker, Simple Online and Real-time Tracking (SORT), Feature Based Object Tracker were among the object trackers that were tested but Deep SORT was found to be the most reliable one. The combination of CenterNet and Deep SORT outperformed the rest by achieving an average counting accuracy of 95%. Zhu et al. [7] developed an enhanced Single-Shot Detector (SSD) capable of outperforming popular object detectors. ResNet achieved the best results with a classification accuracy of 96.11%, while the accuracy for Speeded-up Robust Features (SURF) was 45.74%. Consequently, a ResNet with reduced layers was picked to be the base network of the improved SSD. An overall accuracy of 90.3% was obtained for the counting and classification of vehicles. This was 3% higher than the one obtained by the traditional SSD.

Song et al. [8] developed a system to conduct vehicle detection and counting on different highways. Firstly, video frames go through the meanshift algorithm and Gaussian filters for smoothing and enhancement. Then, the results are fed to a flooding filling algorithm to identify and extract only the road surfaces. The YOLOv3 algorithm was used to locate the positions of vehicles and the Oriented FAST and Rotated BRIEF (ORB) algorithm was employed to predict driving directions. The experiments revealed that the accuracy for the counting of vehicles and driving directions were 93.% and 92.3%, respectively. Lira et al. [9] constructed a system where video footages obtained from drones were used to perform vehicle detection by using the Mixture of Gaussian (MOG2) algorithm. However, tracking of motorbikes and distant vehicles proved to be difficult. An overall accuracy of 64% was achieved for vehicle detection. Satyanarayana et al. [10] proposed a distinct approach to detect and classify vehicles from Indian roads. The CNN used was trained with 8000 images. The shapes formed by the ROIs (Regions of Interest) were used to approximate the length, width and type of vehicle. However, occurrences of closely moving two-wheelers being classified as a single vehicle of another class were observed. Also, vehicles having colours similar to roads created some difficulties for the system.

Hasan et al. [11] developed a model based on a Convolutional Neural Network (CNN) to analyze traffic density. The CNN was trained to identify five different classes ranging from "empty lanes" to "traffic not moving". One of the worrying issues of the system was overfitting. Data augmentation and batch normalization were used to reduce overfitting. The model was found to be working at an accuracy of 84.06%, with a further increase of 2.5% after batch normalization. Biswas et al. [12] constructed two automated Python algorithms for counting vehicles and for estimating traffic density. The first method was based on Single Shot Detector (SSD) while the other one made use of MobileNet SSD. SSD was able to achieve a detection accuracy of 92.9% while MobileNet only reached 79.3%. Huy and Duc [13] came up with a real-time traffic density evaluation system which locates, counts and classify vehicles using faster R-CNN (Region-Based Convolutional Neural Network) and CSRT (Channel and Spatial Reliability Tracker) algorithm. The tracker can tolerate certain unforeseeable movements and can work with intermittent frame drops. To estimate traffic density, factors such as speed of moving traffic and number of vehicles present were considered. An accuracy of about 95% on highway streets and a 5% drop on crossroads due to frequent occurrences of unpredictable movements was obtained. Bidwe et al. [14] worked on a convolutional neural network (CNN) to classify traffic images into three traffic density categories, namely, low, medium and high. The CNN was able to achieve classification accuracy of up to 99.6%. However, the system can only find density of a single image and has not been tested to calculate the overall density of a sequence of frames.

Ijeri et al. [15] came up with a traffic control system using image processing. Vehicles came from four different directions, the system determines the traffic density and picks the lane that gets green signal for an allocated time. Canny edge detection algorithm is part of the method employed to estimate traffic density. The system is found to be demanding for real-time traffic control. The overall accuracy of the system was only 45%. Mittal et al. [16] put together a traffic density estimation model intended to be used for improving green lights timings at intersections. A combination of Faster R-CNN and YOLOv5, named as EnsembleNet, was used as the object detector. The model was able to operate in low light scenarios and achieve a detection accuracy of up to 98% compared to 95.8% for YOLOv5 and 97.5% for Faster R-CNN for the same test set. Hasanah et al. [17] implemented a Smart Traffic light management system at road intersections in Indonesia. The Haar Cascade Classifier and an advanced CNN were used together to form an object detector. The algorithm can process footages of one second from CCTV cameras in 0.52s for light traffic and 1.05s for heavy traffic. The detection accuracy was 82% during calm periods but drops to 60% during rush hours.

Fedorov et al. [18] attempted to determine the flow of traffic at the most hectic intersection in the Russian city of Chelyabinsk. The faster R-CNN detector was used for vehicle identification and the SORT (Simple Online and Real Time Tracker) algorithm would associate objects throughout the video. Testing revealed an error rate of 7.25% as the tracker struggles when cars overlapped while waiting at the centre of the intersection. Grents et al. [19] also used the architectural model of the faster R-CNN and SORT tracker to detect and track moving vehicles. The authors also estimated vehicles' speed to understand traffic flow using equations from Makwana and Goel [20]. However, the proposed method could only estimate speed within an error rate of 22%. Using YOLOv4, Khalaf et al. [21] achieved an impressive accuracy of 98.93% for the detection of pedestrians from the KITTI dataset

Despite all the breakthroughs concerning vehicle counting and traffic density estimation, a single setup that can compute all the key aspects of road traffic analysis is still missing. Our plan is to implement a system that is able to detect, track, count moving vehicles, estimate traffic flow, traffic density and driving speed. The proposed model should be able to obtain excellent classification accuracy while operating in real-time.

A. *Object Detection Algorithms*

In the realm of real-time traffic detection systems, object detection stands as a pivotal component, given its fundamental role in identifying every moving vehicle traversing the roadways. While humans effortlessly perform object detection, machines encounter significant complexity in executing this task. This section delves into recent advancements in object detection methodologies, shedding light on notable object detectors.

1) The YOLO algorithm: At the University of Washington, Joseph Redmon and Ali Farhadi built the start of the well-known YOLO (You Only Look Once) object recognition and picture segmentation model [22]. After its initial introduction in 2015, YOLO has grown in popularity due to its fast speed and correctness [23]. The first model receives frames at a resolution of 448 × 448 pixels and applies a convolutional neural network on the whole image in one go. The network divides the input in a K × K grid, detects objects when their centres are found inside a cell and predicts the positions of bounding boxes and the likelihood of their respective classes. A threshold value is used to reject bounding boxes with low confidence score. The primary YOLO model recorded real-time processing speed of 45 frames per second and a faster version achieved a speed of 155 frames per second [22].

A year later, Redmon and Farhadi released an improved version known as YOLOv2 which implemented a technique to perform training on both object classification and identification [24]. Batch normalization was another addition to improve the precision. YOLOv2 was able to operate at 67 fps and was more robust to different sizes of input images [24]. YOLOv3 consisted of subtle upgrades which made the network layers larger and resulted into better accuracy while not impacting the speed significantly [25]. The following release, YOLOv4, brought the CSPDarknet53 as its classifier backbone along with a 10% increase in speed and launched an innovative data augmentation technique [26]. YOLO's capacity was further enhanced in 2021 with the introduction of YOLOv5 by Ultralytics, which incorporated support for panoptic segmentation and object tracking.

YOLOv8 was published by Ultralytics in 2023 [23]. This model includes a new backbone network as well as a new loss function. One of its attributes is compatibility with all the existing YOLO versions. Users may swiftly switch between and assess the performance of various versions [23]. YOLOv8 can now also perform instance segmentation and image classification in addition to object recognition. Anchor free detection is an architectural change brought into the algorithm to reduce the unnecessary large amount of predicted bounding boxes. Terminating mosaic data augmentation after a certain number of epochs is another update to the system. During testing on the COCO dataset, YOLOv8 had a 20% increase in mAP for its small version compared to YOLOv5 [27].

2) Single Shot Multibox Detector (SSD): SSD (Single Shot Multibox Detector) is a state-of-the-art object detection model. Its architecture consists of only one deep neural network. During training, SSD requires input frames and labelled boxes surrounding each object. When analyzing images, the algorithm's convolutional layers produce several default boxes at the location of each object on feature maps of different proportions. The system then compares both input's boxes and generated ones to look for matches. Those which are compatible are considered positives and the rest are considered as negatives. This technique has proven to enhance SSD accuracy for low resolution images. During detection mode, the convolutional layers generate a feature map and a 3 × 3 convolution kernel is applied on the map to predict a set of bounding boxes where objects are detected along with the likelihood of their respective classes [28].

When tested on the VOC2007 dataset, SSD is able to attain a mean average precision (mAP) of 74.3% and runs at 59 fps while using 300 × 300 pixels images. A slight increase of 2.6% in mAP was reached for 512 × 512 input frames. Nevertheless, as SSD ignores the data beyond the proposed boxes, it is commonly acknowledged that it is less reliable in recognizing smaller objects [29]. Fig. 1 shows the SSD model.

3) Mask R-CNN algorithm: Mask R-CNN is a framework introduced in 2017 by Facebook AI researchers that extended the Faster R-CNN model [30]. The latter operates in two stages. The initial phase, known as a Region Proposal Network (RPN), places bounding boxes on potential objects according to its concurrent prediction of objects limits and objectness values at each point. The following step is derived from Fast R-CNN, uses RoIPool to retrieve traits from each bounding box and classify them into their respective classes. Bounding-box regression is also done to refine localization and the size of boxes. The features identified from these two steps can be communicated between them for quicker inference. Mask R-CNN adds to the second stage by also performing a high precision segmentation of each detected object in the images. When compared to existing models, Mask R-CNN exhibits a greater average precision and does not significantly increase the overhead of faster R-CNN, but it only runs at 5fps, which makes it unsuitable for real-time applications [30].

4) EfficientDet: Tan et al. [31] noticed that system optimization has grown in significance over the last few years for the field of computer vision. They did in-depth research of neural network architectures in an effort to identify the ideal combination to build an efficient object detector. After evaluation, EfficientNet was chosen as the network backbone. Innovative features such as BiFPN (bi-directional feature pyramid network) and a new compound scaling method were introduced. BiFPN acts as the feature network and makes multiscale feature fusion simple and quick. The compound scaling solution simultaneously adjust the pixel density, depth and breadth of network for better optimization. With only 77 million parameters and 410 billion FLOPs, EfficientDet-D7 is able to obtain an average precision of 55.1% on the COCO dataset. Fig. 2 shows the architecture of EfficientDet.

B. Object Tracking Algorithms

Object tracking algorithms constitute an essential element within intelligent traffic monitoring systems. Their primary objective is to establish correspondences between identical vehicles observed across distinct frames of a recording. A proficient algorithm achieves this objective with minimal computational expenditure while ensuring a consistent and reliable tracking of a vehicle throughout its presence in the monitored scene.

1) DeepSORT: DeepSORT is an extension of the computer vision tracking model SORT (Simple Online Realtime Tracking). The SORT algorithm is a method for numerous objects tracking that emphasizes on efficient and straightforward methods to achieve real-time processing. It employs a blend of well-known methods such as the Kalman Filter and

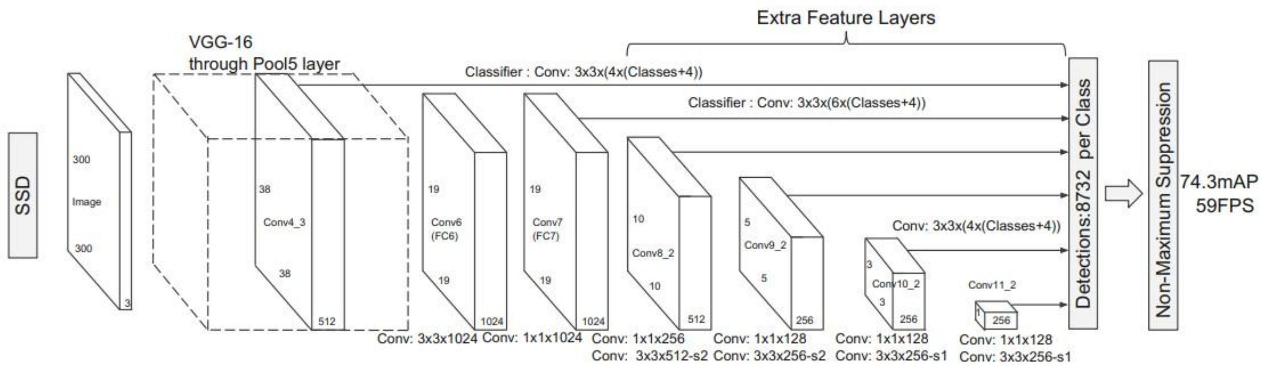

Fig. 1. SSD Model [28].

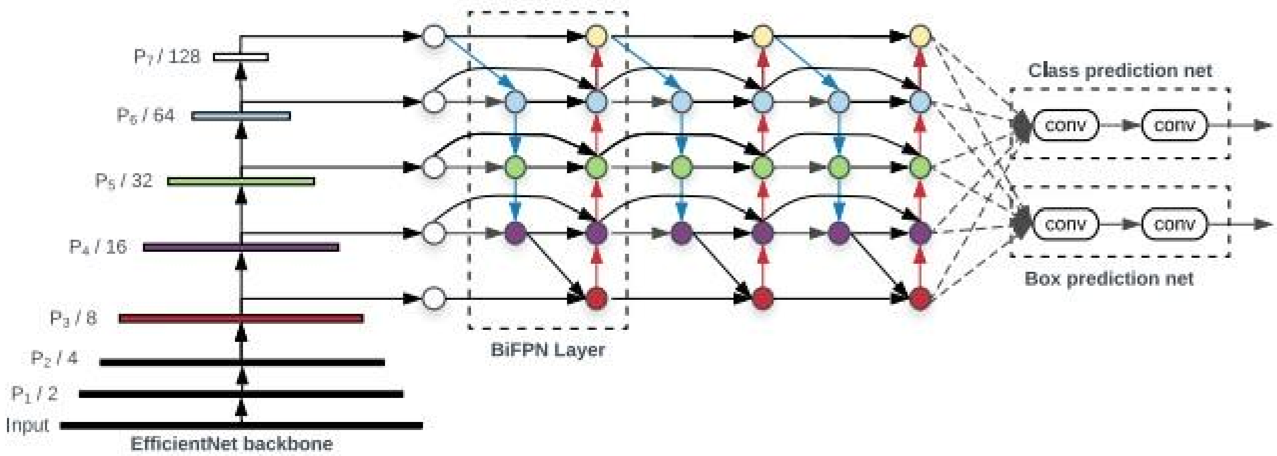

Fig. 2. EfficientDet architecture [31].

Hungarian algorithm to accomplish its goals. When an object appears or exits the frame, a distinct identity is either given to or taken from it. An identified object is labelled as untracked when its detection has an overlap smaller than the defined IOUmin (Intersection over Union) value. An IOU matrix which represents the intersection-over-union distance between each detected object and all expected bounding boxes of pre-existent entities is kept. The matrix is used by the Hungarian algorithm to associate current objects with their next position in the following frame. The Kalman Filter is used to predict future locations of every tracked target which is used for object association. Tracking is terminated if the entity is not found for a specified number of frames [32].

While using the SORT method, some objects' tracking are often lost midway through and afterwards retraced using a new identification. If counting is necessary, it could pose a significant hurdle. DeepSORT tries to include visual appearance of objects as a factor when looking for objects in the upcoming frames. A trained convolutional neural network with two convolutional layers and six residual blocks is incorporated with predicted motion information to form a better object association metric. The technique has shown to eliminate entity switches issue by around 45%. Tracking is proved to be smoother in scenarios with occlusions and unexpected motions [33].

2) ByteTrack: ByteTrack (BYTE) is a simple and robust tracker which main goal is to consider all detected bounding boxes in order to minimize tracking errors due to occluded targets, motion blur or shift in sizes. ByteTrack does not use any convolutional neural network to achieve its aim but rather uses both high and low confidence detection boxes while performing association. Boxes are categorized as low and high confidence according to a selected threshold. The Kalman filter is then applied to predict the expected path of objects in the following frame. The initial association is done with the large confidence value boxes and all the predicted tracks. At this phase, the Hungarian method is utilized to resolve the matching. During the next stage, the resulting unmatched tracks are associated with the low confidence boxes. Any unpaired path is saved so that it may be handled with the following frame. BYTE is a highly customizable model and can easily be merged with other similar tracking algorithms. It was able to obtain an accuracy of 80.3% for multi object tracking while performing at 30 fps [34].

III. Methodology

This section details the architecture and structure of the proposed system. The contents of various elements and how they interact with one another to allow the model to run successfully are listed and thoroughly explained. Before building the components that would be used to extract road traffic information from video recordings, compilation of a robust dataset is essential for training of an object detector.

A. Dataset

The dataset consists of five different classes namely buses, cars, motorbikes, trucks and vans with class ids 0 to 4, respectively. Out of 2800 frames, 22,293 cars, 4852 trucks, 3527 motorbikes, 2308 vans and 1937 buses were manually annotated to form the dataset. Data augmentation techniques such horizontal flips and 10-degree rotations have been used to take the number of frames for the training dataset to a total of 5600 frames. 280 labelled frames were set aside to be used for the validation stage. The images were captured during the day with light, medium and high traffic. Once the videos were captured, any visible faces plate numbers were blurred out as agreed with the Data Protection Office of Mauritius

B. Training of the Model

After intense research and analysis, YOLOv8m has been chosen as the detector to perform vehicle recognition. This choice was made due to the balance of speed and precision it is able to offer. The detector was trained on the dataset for 120 epochs. The process took four hours to be completed.

C. Architectural Framework and Structural Composition of the System

The architecture composes of a detection and classifier module, a tracking unit, a traffic flow computation component, a speed estimation module and a traffic density estimation unit. The detection and classifier module mainly consist of the trained YOLOv8m detector and each video frame, upon being read, is initially transmitted to this detector for processing. It firstly resizes the input image and pass it to its convolutional neural network. Positions of bounding boxes with their corresponding vehicle classes and confidence scores are then determined. Finally, a thresholding of 0.25 is applied to discard low confidence boxes. The output is then transformed into a format compatible with the ByteTrack tracking algorithm and sent to the tracking unit. A reduced ByteTrack algorithm without its association metric serves as the central component within the tracking module. Upon receipt, the detection results are processed by ByteTrack and with the use of the Kalman filter, a list of predicted tracks are created. This list will then be used together with the detection results of the upcoming image for the association of the same vehicle across the sequential video frames. Each track is assigned a tracker ID and paired with the detection exhibiting the highest intersection over union (IOU) score. In case a detection is paired with multiple tracks, only the combination with the highest IOU is considered.

These combinations are then transmitted to the other modules for traffic analysis. The traffic flow computation component is where counting of vehicles is performed. The module requires a reference line perpendicular to the driving direction with end points being pixel coordinates of the recording in order to count vehicles. The end points of the line must be placed in positions such that the visibility of moving targets is optimal while traversing the line. As a vehicle moves from one segment of the line to the other, the counter for its category is incremented. At the end of each 60 s intervals, the number vehicles counted for that period is divided by 60 to calculate traffic flow in vehicles per minute.

The speed estimation module needs two reference lines perpendicular to the driving direction to operate. The calculation of speed for individual vehicles involves determining the time taken to traverse the distance from passing the initial reference line and crossing the subsequent line. The same concept used for vehicle counting is applied to understand when a target is traversing a reference line. Distance between lines can be estimated by adding the length of individual road markings. The traffic density estimation unit makes use of a rectangular box such that two of its edges are perpendicular to the driving direction. The region of interest formed must be located where the detection of vehicle is reliable.

For each 60 s, an average count of vehicles within the specified area in each frame is calculated. This count is divided by the number of lanes and the distance between the edges perpendicular to the driving direction, and then multiplied by 10 to obtain the traffic density per lane in vehicle per 10 m. Fig. 3 shows the activities of the speed estimation module, starting by initializing the speed list to null and positioning the two reference lines needed to calculate the speed of vehicles. Through an examination of tracker IDs associated with recognized vehicles, a distinction is made between new and previously identified vehicles. Newly identified vehicles have their coordinates stored for future reference, while those previously recognized undergo a comparison between their current positions and their last known positions. This assessment aims to ascertain whether these vehicles have crossed either the initial or subsequent reference lines. When a tracker ID is observed to have traversed both lines sequentially, the average speed for that specific duration is computed and then recorded in the speed list.

Fig. 4 describes the workings of the traffic density estimation unit, beginning by defining the coordinates of the rectangular area within which the road traffic density will be estimated. Within each frame, the centroids of all identified vehicles are computed, and their presence within the designated region of interest is determined. Consequently, the traffic density for that particular frame is evaluated based on the count of vehicles within this region. Over a span of 60 s, an average traffic density per lane is calculated.

D. ByteTrack

The instance takes a list of important arguments that have huge impact on its precision and computation speed. track_thresh which is set to 0.25, defines the minimum threshold needed so that an object and a track can be considered a match. track_buffer represent the number of frames a track is kept alive without being updated. The assignment of this parameter, chosen to be 30, necessitates meticulous consideration, particularly in situations characterized by recurrent occlusions. match_thresh is the value used to know if two

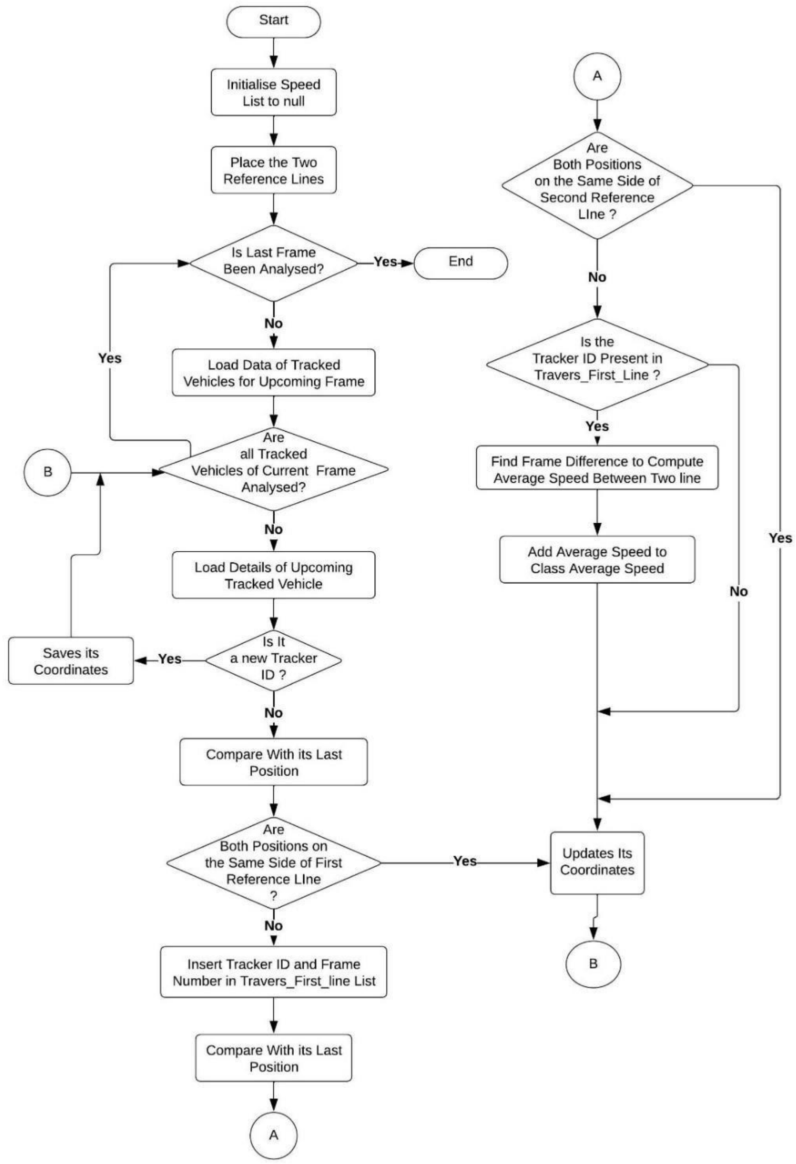

Fig. 3. Speed estimation module.

detections on a frame are the same object and it is configured to a value of 0.8. aspect_ratio_thresh define maximum aspect ratio difference two objects can have to be considered same and it is set to be 3.0. min_box_area, chosen to be 1.0, refers to the minimum area bounding boxes should have to be considered for tracking.

Each detected and tracked vehicle is defined by the coordinates of the four corners of its bounding box together with a tracker ID. Whenever a tracker ID is identified for the first time, the x-y coordinates of the corners are tested in a mathematical calculation to find out if they are below or above the line. The Boolean responses are stored in a dictionary. Upon coming across the tracker ID in an upcoming frame, the mathematical calculation is performed again to look for any changes. If all Boolean responses are identical, we wait until the tracker ID may be encountered again. In a situation where only two responses have changed, it would imply that only part of the vehicle has crossed the line. If all four Boolean values have changed, this indicates that the vehicle has crossed the line and is on the opposite side. Consequently, the new values are stored in the dictionary so that the vehicle is not re-identified as crossing the line whenever the tracker ID pops up again.

For each tracked bounding boxes in each frame, its centre point is found by averaging the positions of its four corners. Then, the method cv2.pointPolygonTest from OpenCV is used with the rectangular region and centre point as parameters to find if the point is inside the area. In case of a positive outcome, the vehicle represented by the bounding box is considered to be inside the rectangular box.

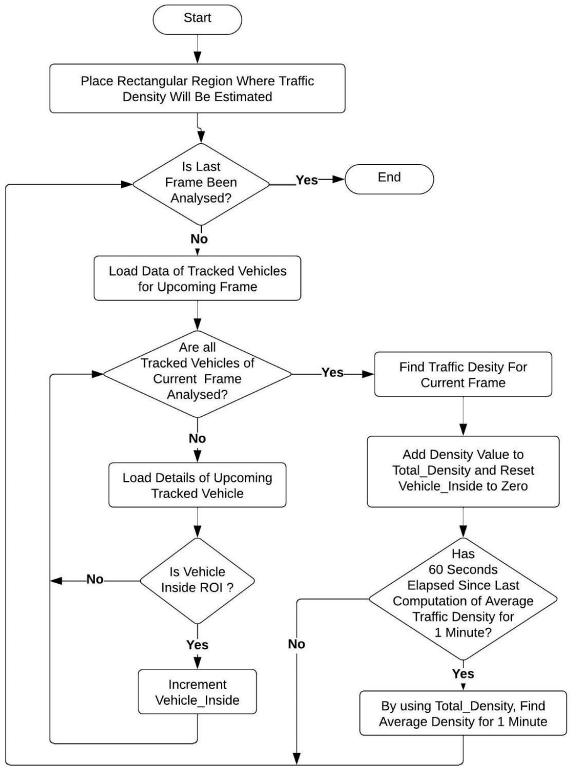

Fig. 4. Traffic density estimation unit.

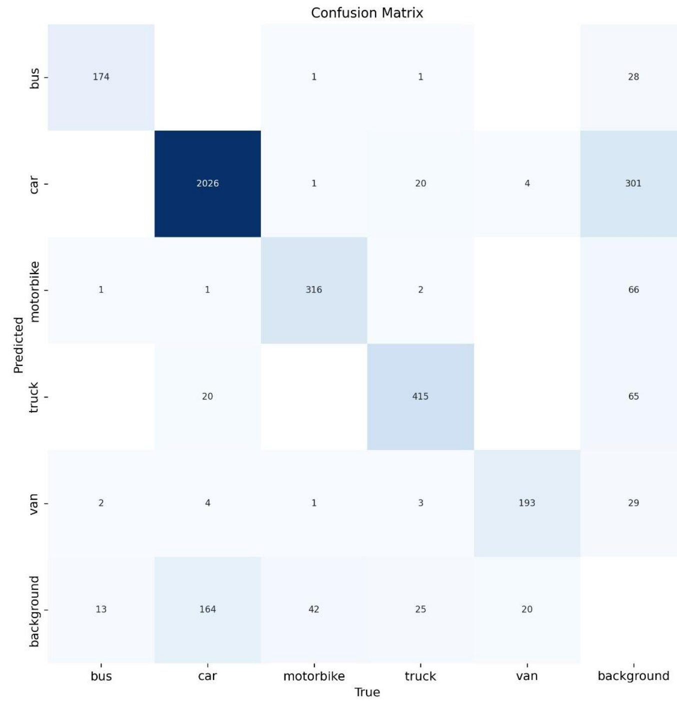

Fig. 5. Confusion matrix.

E. Validation Stage

During the dataset annotation for training purposes, 280 labelled images were set aside to be used for validation. These images were captured under identical conditions and locations as the images used for training. Fig. 5 shows the confusion matrix derived for the validation process. These matrices portray the model's capability to accurately detect and classify 90% of the vehicle instances within the frames. Notably, out of 3429 instances, 486 false positives and 264 false negatives were recorded. The model demonstrates a comparatively higher reliability in identifying buses, while being slightly less accurate for vans.

F. Testing

Test Cases used in this section include scenarios with different road traffic intensities at various locations in Mauritius. The car category includes Support Utility Vehicles (SUV) and the truck category includes lorries, and pickup trucks. The remaining categories are buses, motorbikes and vans.

G. Vehicle Counting and System Speed Comparison

The tracking and counting module have also been built on YOLOv8n and YOLOv8x so that fair accuracy and inference speed comparisons can be made among the three versions. The nano and extra-large versions have also been trained for 120 epochs on the same labelled dataset. From all the videos in test cases, a manual count of vehicles for each category has been done. Classification accuracy has been computed by dividing the count for each vehicle class by their corresponding manual count, find the average and then multiply by 100. The speed of each model has been estimated by the rate at which it can go through a video frame. The reference line for counting is also placed at an optimal position to obtain realistic feedback on correctness.

IV. Experiments, Results and Evaluation

A. Test Case 1: M1 Motorway Near Bagatelle Mall

A two minute video sequence was captured on the bi-directional M1 motorway from the bridge near Bagatelle Mall. The traffic driving to the north had a high traffic density while the traffic moving to the south was sparser. The footage was captured at around 8:30 in the morning. Tables I and II show these results

The results from both directions show that all the three YOLO versions are able to achieve excellent counting and classification accuracies. The traffic flow in both directions for the whole video and for each interval of 60s has also been determined by the system. These values are compared with the actual traffic flow as shown in Tables III and IV.

Observations from Tables III and IV suggest that all the three versions of YOLO are able to estimate traffic flow with very high accuracies. Fig. 6 shows a screenshot from the traffic video used for test case 1.

B. Test Case 2: Brabant Street at Port-Louis

A three minutes video sequence was captured on the bi-directional Brabant Street from the flyover. The road to the north consist of only one lane and had a slow moving traffic while the one to the south consist of two lanes and also had

TABLE I. RESULTS FOR DRIVING DIRECTION: NORTH (TEST CASE 1)

	Manual Count	**YOLOv8m**	**YOLOv8n**	**YOLOv8x**
Car	158	157	158	156
Motorbike	10	10	10	10
Bus	5	5	5	5
Truck	24	22	22	23
Van	10	10	10	10
Total	207	204	205	204
Car Classification Accuracy (%)		99.4	100	98.7
Motorbike Classification Accuracy (%)		100	100	100
Bus Classification Accuracy (%)		100	100	100
Truck Classification Accuracy (%)		91.7	91.7	95.8
Van Classification Accuracy (%)		100	100	100
Counting Accuracy (%)		98.6	99.0	98.6
Classification Accuracy (%)		98.2	98.3	98.9
Speed (FPS)		7	18	3

TABLE II. RESULTS FOR DRIVING DIRECTION: SOUTH (TEST CASE 1)

	Manual Count	**YOLOv8m**	**YOLOv8n**	**YOLOv8x**
Car	46	48	48	45
Motorbike	8	8	8	8
Bus	1	1	1	1
Truck	11	9	10	13
Van	7	7	6	6
Total	73	73	73	73
Car Classification Accuracy (%)		95.7	95.7	97.8
Motorbike Classification Accuracy (%)		100	100	100
Bus Classification Accuracy (%)		100	100	100
Truck Classification Accuracy (%)		81.8	90.9	81.8
Van Classification Accuracy (%)		100	85.7	85.7
Counting Accuracy (%)		100	100	100
Classification Accuracy (%)		95.5	94.5	93.1
Speed (FPS)		7	18	3

TABLE III. TRAFFIC FLOW RESULTS FOR DRIVING DIRECTION: NORTH (TEST CASE 1)

Traffic Flow (Vehicle/min)	Manual Count	YOLOv8m	YOLOv8n	YOLOv8x
0 s–60 s	71	70	71	71
60 s–120 s	68	65	65	65
60 s Intervals Accuracy (%)		97.1	97.8	97.8

TABLE IV. TRAFFIC FLOW RESULTS FOR DRIVING DIRECTION: SOUTH (TEST CASE 1)

Traffic Flow (Vehicle/min)	Manual Count	YOLOv8m	YOLOv8n	YOLOv8x
0 s–60 s	25	24	24	24
60 s–120 s	35	35	35	35
60 s Intervals Accuracy (%)		98.0	98.0	98.0

vehicles moving slowly. The scenario is trickier compared to test case one as a bus stop is found on the road and buses tend to cause vehicle occlusions when they are stationary. This road is also frequently taken by many motorcycles. The footage was captured at around 11:30 in the morning. Tables V and VI show

Fig. 6. YOLOv8m output video.

the results.

TABLE V. RESULTS FOR DRIVING DIRECTION: NORTH (TEST CASE 2).

	Manual Count	YOLOv8m	YOLOv8n	YOLOv8x
Car	14	13	15	14
Motorbike	20	19	19	18
Bus	12	12	9	11
Truck	4	1	1	3
Van	1	1	1	0
Total	51	46	45	46
Car Classification Accuracy (%)		92.9	92.9	100
Motorbike Classification Accuracy (%)		95.0	95.0	90.0
Bus Classification Accuracy (%)		100	75.0	91.7
Truck Classification Accuracy (%)		25.0	25.0	75.0
Van Classification Accuracy (%)		100	100	0
Counting Accuracy (%)		90.2	88.2	90.2
Classification Accuracy (%)		82.6	77.6	71.3
Speed (FPS)		7	18	3

TABLE VI. RESULTS FOR DRIVING DIRECTION: SOUTH (TEST CASE 2)

	Manual Count	**YOLOv8m**	**YOLOv8n**	**YOLOv8x**
Car	10	10	10	10
Motorbike	17	17	17	17
Bus	7	6	9	8
Truck	7	5	1	5
Van	1	1	1	0
Total	42	39	38	40
Car Classification Accuracy (%)		100	100	100
Motorbike Classification Accuracy (%)		100	100	100
Bus Classification Accuracy (%)		85.7	71.4	85.7
Truck Classification Accuracy (%)		71.4	14.3	71.4
Van Classification Accuracy (%)		100	100	0
Counting Accuracy (%)		92.9	90.5	95.2
Classification Accuracy (%)		91.4	77.1	71.4
Speed (FPS)		7	18	3

As test case 2 is a more challenging environment, more variations in the results are seen. YOLOv8m outperforms the two other YOLO versions in classifying the vehicles in their respective categories. Categorization of trucks seems to be more difficult for all the three models. The results from Tables VII and VIII show that all the three versions of YOLO are able to estimate the traffic with accuracies above 80%. Table VII shows a screenshot from the traffic video used for test case 2. Fig. 7 shows the YOLOv8m output video.

TABLE VII. TRAFFIC FLOW RESULTS FOR DRIVING DIRECTION: NORTH (TEST CASE 2)

Traffic Flow (Vehicle/min)	Manual Count	YOLOv8m	YOLOv8n	YOLOv8x
0 s–60 s	13	13	12	13
60 s–120 s	15	10	11	11
120 s–180 s	14	13	11	12
60 s Intervals Accuracy (%)		86.5	81.4	86.3

TABLE VIII. TRAFFIC FLOW RESULTS FOR DRIVING DIRECTION: SOUTH (TEST CASE 2)

Traffic Flow (Vehicle/min)	Manual Count	YOLOv8m	YOLOv8n	YOLOv8x
0 s–60 s	5	4	5	5
60 s–120 s	10	8	7	9
120 s–180 s	21	21	20	20
60 s Intervals Accuracy (%)		86.7	88.4	95.1

Fig. 7. YOLOv8m output video.

C. Test Case 3: Motorway M1 Near Bagatelle Mall

A 4 min video sequence was captured on the bi-directional M1 motorway from the bridge near Bagatelle Mall at around 6 o'clock in the afternoon. Traffic density was very light during this time. Tables IX and X show that all three versions have performed exactly the same while achieving about 99% classification accuracy for both driving directions. The drop in illumination does not seem to have any effect on the detectors' ability to count and classify vehicles.

Tables XI and XII illustrate that all three models had the same performance for traffic flow estimation for each direction. They show that they are able to obtain traffic flow values with accuracies higher than 96%. Fig. 8 shows a screenshot from the traffic video used for test case 3.

In terms of counting accuracy, all the three YOLO versions perform close to each other with YOLOv8x having the slight edge. As predicted, counting results shows that test case 2 was the most challenging as it often had motorbikes and small

TABLE IX. RESULTS FOR DRIVING DIRECTION: NORTH (TEST CASE 3)

	Manual Count	YOLOv8m	YOLOv8n	YOLOv8x
Car	40	38	38	38
Motorbike	5	5	5	5
Bus	2	2	2	2
Truck	6	6	6	6
Van	4	4	4	4
Total	57	55	55	55
Car Classification Accuracy (%)		95.0	95.0	95.0
Motorbike Classification Accuracy (%)		100	100	100
Bus Classification Accuracy (%)		100	100	100
Truck Classification Accuracy (%)		100	100	100
Van Classification Accuracy (%)		100	100	100
Counting Accuracy (%)		96.5	96.5	96.5
Classification Accuracy (%)		99.0	99.0	99.0
Speed (FPS)		7	18	3

TABLE X. RESULTS FOR DRIVING DIRECTION: SOUTH (TEST CASE 3)

	Manual Count	YOLOv8m	YOLOv8n	YOLOv8x
Car	56	55	55	55
Motorbike	1	1	1	1
Bus	0	0	0	0
Truck	3	3	3	3
Van	6	6	6	6
Total	66	65	65	65
Car Classification Accuracy (%)		98.2	98.2	98.2
Motorbike Classification Accuracy (%)		100	100	100
Bus Classification Accuracy (%)		-	-	-
Truck Classification Accuracy (%)		100	100	100
Van Classification Accuracy (%)		100	100	100
Counting Accuracy (%)		98.5	98.5	98.5
Classification Accuracy (%)		99.6	99.6	99.6
Speed (FPS)		7	18	3

TABLE XI. TRAFFIC FLOW RESULTS FOR DRIVING DIRECTION: NORTH (TEST CASE 3)

Traffic Flow (Vehicle/min)	Manual Count	YOLOv8m	YOLOv8n	YOLOv8x
0 s–60 s	17	17	17	17
60 s–120 s	11	10	10	10
120 s–180 s	12	12	12	12
180 s–240 s	15	16	16	16
60 s Intervals Accuracy (%)		96.1	96.1	96.1

TABLE XII. TRAFFIC FLOW RESULTS FOR DRIVING DIRECTION: SOUTH (TEST CASE 3)

Traffic Flow (Vehicle/min)	Manual Count	YOLOv8m	YOLOv8n	YOLOv8x
0 s–60 s	20	19	19	19
60 s–120 s	18	18	18	18
120 s–180 s	16	16	16	16
180 s–240 s	9	9	9	9
60 s Intervals Accuracy (%)		98.8	98.8	98.8

Fig. 8. YOLOv8m output video.

cars being hidden behind large buses waiting at a bus stop. This poses major difficulties for detection and tracking. As for classification accuracy, clear differences can be easily seen among the different models. YOLOv8m achieved a classification accuracy of 94.4% while the accuracy for YOLOv8x was only 88.9%. Test case 2 proved to be challenging for YOLOv8m and YOLOv8x as they misclassified or missed buses and trucks.

As for the operating speed, YOLOv8n outperforms the two others with an average fps of 18 as it is a much lighter detector with fewer parameters. YOLOv8n is more than twice faster than YOLOv8m and almost four times faster than YOLOv8x. The slight variation in speed between different test cases is due to dependency on the number of objects detected. This testing shows YOLOv8m can operate in real-time if a powerful GPU is used. All the three YOLO models are capable of evaluating traffic flow with good accuracies. YOLOv8x is slightly more suited for this task with an overall accuracy of 95.4%.

D. Traffic Density

With a known distance provided for the length of the region of interest (ROI), the system is able to determine traffic density for each lane. Values for traffic density per lane in the ROIs has been manually estimated by counting the number of vehicles present inside the selected areas, divides this figure by the length of the region and the number of lanes then multiplied by ten.

1) Test Case 4: Motorway M1 near bagatelle mall: A three minutes video was recorded on the bi-directional M1 motorway from the bridge near Bagatelle Mall at 9 o'clock in the morning. The video had about 106 vehicles moving north along with about 55 vehicles going south. Both directions consisted of three lanes. A region with a length of sixty was chosen as the length to be inspected for both directions. Tables XIII and XIV show the results.

TABLE XIII. RESULTS OF DRIVING DIRECTION TO THE NORTH (TEST CASE 4)

Interval	Traffic Density per Lane Manually Evaluated (Vehicles/per Lane/10 m)	YOLOv8m (Vehicles/per Lane/10 m)	Accuracy (%)
0 s–60 s	1.80	1.77	98.3
60 s–120 s	1.99	1.94	97.5
120 s–180 s	2.11	2.30	91.0

TABLE XIV. RESULTS OF DRIVING DIRECTION TO THE SOUTH (TEST CASE 4)

Interval	Traffic Density Per Lane Manually Evaluated (Vehicles/per Lane/10 m)	YOLOv8m (Vehicles/per Lane/10 m)	Accuracy (%)
0 s–60 s	1.14	1.17	97.4
60 s–120 s	1.00	1.06	94.0
120 s–180 s	0.90	0.89	98.9

Values obtained from test case 4 show that the system is able to estimate the traffic density with high accuracies. Fig. 9 shows the outputs from test case 4.

Fig. 9. Output video frame (Test Case 4).

2) Test Case 5: Brabant street at port-louis: A three minutes video was captured on the bi-directional Brabant Street from the flyover at 11 o'clock in the morning. The road to the north consist of only one lane and while the one to the south consist of two lanes. The video had about 18 vehicles moving north along with about 31 vehicles going south. A region with a length of thirty meters was chosen as the area to be inspected for both directions. Tables XV and XVI show the results.

TABLE XV. RESULTS OF DRIVING DIRECTION TO THE NORTH (TEST CASE 5)

Interval	Traffic Density per Lane Manually Evaluated (Vehicles/per Lane/10 m)	YOLOv8m (Vehicles/per Lane/10 m)	Accuracy (%)
0 s–60 s	1.86	2.00	92.5
60 s–120 s	1.61	2.02	74.5
120 s–180 s	2.46	2.55	96.3

For the north direction, during the interval 60 s–120 s, a road marking was wrongly identified as a car for a brief period of time which caused a significant decrease in the accuracy. Fig. 10 shows the outputs from test case 5.

3) Test Case 6: M1 Motorway near bagatelle mall: A three minutes recording was also captured on the bi-directional M1 motorway from the bridge near Bagatelle Mall at around 6 o'clock in the afternoon. The video had around 22 vehicles moving north along with about 30 vehicles going south. A

TABLE XVI. Results of Driving Direction to the South (Test Case 5)

Interval	Traffic Density Per Lane Manually Evaluated (Vehicles/per Lane/10 m)	YOLOv8m (Vehicles/per Lane/10 m)	Accuracy (%)
0 s–60 s	1.22	1.26	96.7
60 s–120 s	1.20	1.24	96.7
120 s–180 s	2.71	2.71	100

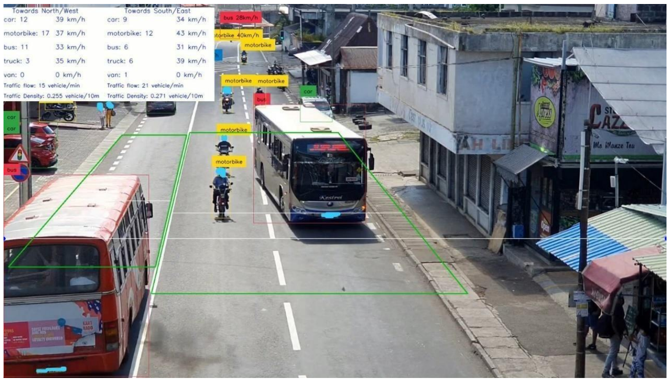

Fig. 10. Output video frame (Test Case 5).

region with a length of forty meters was chosen as the area to be inspected for both directions. There are 3 lanes in both directions. Tables XVII and XVIII show the results. Fig. 11 shows the outputs from test case 6.

TABLE XVII. Results of Driving Direction to the North (Test Case 6)

Interval	Traffic Density per Lane Manually Evaluated (Vehicle/per Lane/10 m)	YOLOv8m (Vehicle/per Lane/10 m)	Accuracy (%)
0 s–60 s	0.80	0.80	100
60 s–120 s	0.40	0.42	95.0
120 s–180 s	0.62	0.64	96.8

TABLE XVIII. Results of Driving Direction to the South (Test Case 6)

Interval	Traffic Density Per Lane Manually Evaluated (Vehicle/per Lane/10 m)	YOLOv8m (Vehicle/per Lane/10 m)	Accuracy (%)
0 s–60 s	0.98	1.00	98.0
60 s–120 s	0.90	0.88	97.8
120 s–180 s	0.64	0.68	93.8

According to the figures obtained from the three test cases, the implementation with YOLOv8m is able to produce an overall mean accuracy of 95.3% for traffic density. An automated real-time traffic monitoring system using computer vision and deep learning offers several advantages. It provides immediate traffic data for better decision-making and congestion management. With automated vehicle counting and tracking, the system ensures higher accuracy and reduces human error. Data obtained from this system can be used by the relevant authorities for decongestion strategies. The system can also be used to enhance road safety by identifying dangerous driving patterns and enable faster emergency responses. Since less time is spent in traffic, it contributes to environmental sustainability by reducing vehicle emissions. For drivers, the system improves their overall driving experience by minimising travel time and stress.

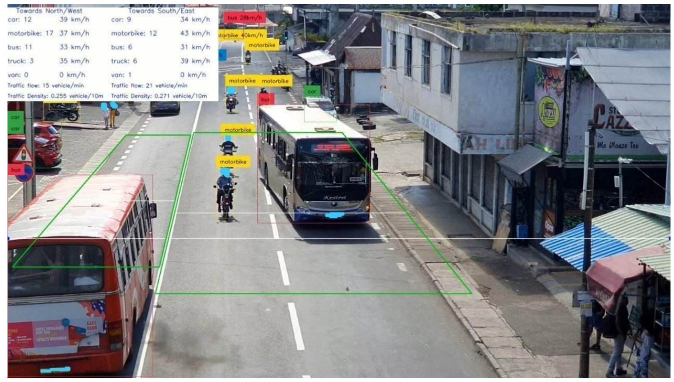

Fig. 11. Output video frame (Test Case 6).

V. Conclusion

The recent changes in road infrastructure to incorporate the metro pathway, combined with the constant increase in the number of vehicles in Mauritius, are causing severe traffic congestions on the country's highways and in major towns. The system designed and implemented can undoubtedly serve as the foundation for a solution to this problem. The proposed system is able to detect, track, classify and count vehicles with reasonable accuracy under different traffic conditions. Traffic flow and traffic density can also be estimated in real-time with good accuracy. The system can also estimate the speed of moving targets and give a representation of average driving speed for each category of vehicles. Once deployed, the proposed system can perform these tasks without any human intervention. It is also designed to be scalable and can be integrated with existing monitoring systems to provide detailed information about the traffic state of any roads in the Republic of Mauritius. This will enable the relevant authorities to gain a comprehensive understanding of traffic conditions on any road in Mauritius, facilitating more informed decision-making for road management and future infrastructure projects. As the system evolves, we aim to enhance its robustness by collecting and analyzing additional data from various locations and times, including night-time monitoring, to account for variations in road structures, traffic patterns, and lighting conditions. Future enhancements will focus on refining detection accuracy under challenging conditions, such as poor weather or very heavy traffic, to ensure that the system remains reliable and adaptable for the long-term needs of the transport infrastructure in Mauritius.

References

[1] T. Afrin and N. Yodo, "A survey of road traffic congestion measures towards a sustainable and resilient transportation system," *Sustainability*, vol. 12, p. 4660, 2020. [Online]. Available: https://doi.org/10.3390/su12114660

[2] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ, USA, 2016, pp. 3464–3468. [Online]. Available: https://doi.org/10.1109/ICIP.2016.7533003

[3] S. Bidwe, G. Kale, and R. Bidwe, "Traffic monitoring system for smart city based on traffic density estimation," *Indian Journal of Computer Science and Engineering*, vol. 13, pp. 1388–1400, 2022. [Online]. Available: https://doi.org/10.21817/indjcse/2022/v13i5/221305006

[4] D. Biswas, H. Su, C. Wang, A. Stevanovic, and W. Wang, "An automatic traffic density estimation using single shot detection (ssd) and mobilenet-ssd," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 110, pp. 176–184, 2019. [Online]. Available: https://doi.org/10.1016/j.pce.2018.12.001

[5] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," 2020, available online: https://arxiv.org/abs/2004.10934v1 (accessed on 23 March 2024).

[6] A. Fedorov, K. Nikolskaia, S. Ivanov, V. Shepelev, and A. Minbaleev, "Traffic flow estimation with data from a video surveillance camera," *Journal of Big Data*, vol. 6, p. 73, 2019. [Online]. Available: https://doi.org/10.1186/s40537-019-0234-z

[7] J. Gorzelany, "The worst traffic jams in history," 2021, available online: https://www.forbes.com/sites/jimgorzelany/2013/05/21/the-worst-traffic-jams-inhistory/?sh=6b5ee2783e1a (accessed on 12 March 2024).

[8] A. Grents, V. Varkentin, and N. Goryaev, "Determining vehicle speed based on video using convolutional neural network," *Transportation Research Procedia*, vol. 50, pp. 192–200, 2020. [Online]. Available: https://doi.org/10.1016/j.trpro.2020.10.024

[9] M. Hasan, S. Das, and M. Akhand, "Estimating traffic density on roads using convolutional neural network with batch normalization," in *Proceedings of the 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, Dhaka, Bangladesh, 2021, pp. 1–6. [Online]. Available: https://doi.org/10.1109/iceeict53905.2021.9667860

[10] M. Hasanah, G. Qorik, O. Pratamasunu, and R. Pawening, "Automatic car detection using haar cascade classifier and convolutional neural network for traffic density estimation," *Indonesian Journal of Artificial Intelligence and Data Mining (IJAIDM)*, vol. 4, pp. 11–18, 2021. [Online]. Available: https://doi.org/10.24014/ijaidm.v4i1.10785

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 386–397, 2017. [Online]. Available: https://doi.org/10.1109/TPAMI.2018.2844175

[12] T. Huy and B. Duc, "Traffic flow estimation using deep learning," in *Proceedings of the 5th International Conference on Green Technology and Sustainable Development (GTSD)*, Ho Chi Minh City, Vietnam, 2020, pp. 180–184. [Online]. Available: https://doi.org/10.1109/GTSD50082.2020.9303163

[13] D. Ijeri, P. Maidargi, and R. Sunagar, "Traffic control system using image processing," in *Proceedings of the 1st IEEE Bangalore Humanitarian Technology Conference (B-HTC)*, Vijiyapur, India, 2020, pp. 1–6. [Online]. Available: https://doi.org/10.1109/B-HTC50970.2020.9298014

[14] A. Khalaf, M. Abdulrahman, I. Al-Barazanchi, J. Tawfeq, P. JosephMg, and A. Radhi, "Real time pedestrian and objects detection using enhanced yolo integrated with learning complexity-aware cascades," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 22, pp. 362–371, 2024. [Online]. Available: https://doi.org/10.12928/telkomnika.v22i2.24854

[15] G. Li, W. Lai, X. Sui, X. Li, X. Qu, T. Zhang, and Y. Li, "Influence of traffic congestion on driver behavior in post-congestion driving," *Accident Analysis and Prevention*, vol. 141, p. 105508, 2020. [Online]. Available: https://doi.org/10.1016/J.AAP.2020.105508

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. Berg, "SSD: Single shot multibox detector," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, The Netherlands, 2016, pp. 21–37. [Online]. Available: https://doi.org/10.1007/978-3-319-46448-0_2

[17] G. Lira, Z. Kokkinogenis, F. Rossetti RJ, D. Moura, and T. Rúbio, "A computer-vision approach to traffic analysis over intersections," in *Proceedings of the 19th IEEE International Conference on Intelligent Transportation Systems (ITSC)*, Rio de Janeiro, Brazil, 2016, pp. 47–53. [Online]. Available: https://doi.org/10.1109/ITSC.2016.7795530

[18] M. Makwana and P. Goel, "Moving vehicle detection and speed measurement in video sequence," *International Journal of Engineering Research and Technology (IJERT)*, vol. 2, pp. 3534–3537, 2013.

[19] V. Mandal and Y. Adu-Gyamfi, "Object detection and tracking algorithms for vehicle counting: A comparative analysis," *Journal of Big Data Analytics in Transportation*, vol. 2, pp. 251–261, 2020. [Online]. Available: https://doi.org/10.1007/s42421-020-00025-w

[20] U. Mittal, P. Chawla, and R. Tiwari, "Ensemblenet: A hybrid approach for vehicle detection and estimation of traffic density based on faster r-cnn and yolo models," *Neural Computing and Applications*, vol. 35, pp. 4755–4774, 2023. [Online]. Available: https://doi.org/10.1007/S00521-022-07940-9

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016, pp. 779–788. [Online]. Available: https://doi.org/10.1109/CVPR.2016.91

[22] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 6517–6525. [Online]. Available: https://doi.org/10.1109/CVPR.2017.690

[23] ——, "Yolov3: An incremental improvement," 2018, available online: https://arxiv.org/abs/1804.02767v1 (accessed on 3 February 2024).

[24] Road Development Authority, "Budget estimates 2022/23," 2022, available online: https://rda.govmu.org/rda/wp-content/uploads/2022/10/4.-Budget-2022-202312.06.22-1.pdf (accessed on 10 February 2024).

[25] R. Satyanarayana GS, P. Deshmukh, and S. Das, "Vehicle detection and classification with spatio-temporal information obtained from cnn," *Displays*, vol. 75, p. 102294, 2022. [Online]. Available: https://doi.org/10.1016/J.DISPLA.2022.102294

[26] J. Seong, Y. Kim, H. Goh, H. Kim, and A. Stanescu, "Measuring traffic congestion with novel metrics: A case study of six u.s. metropolitan areas," *ISPRS International Journal of Geo-Information*, vol. 12, p. 130, 2023. [Online]. Available: https://doi.org/10.3390/IJGI12030130

[27] J. Solawetz and Francesco, "What is yolov8? the ultimate guide," 2023, available online: https://blog.roboflow.com/whats-new-in-yolov8/ (accessed on 3 February 2024).

[28] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes," *European Transport Research Review*, vol. 11, p. 51, 2019. [Online]. Available: https://doi.org/10.1186/s12544-019-0390-4

[29] M. Tan, R. Pang, and Q. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, 2019, pp. 10 778–10 787. [Online]. Available: https://doi.org/10.1109/CVPR42600.2020.01079

[30] Ultralytics, "Yolov8 docs," 2023, available online: https://docs.ultralytics.com/#ultralytics-yolov8 (accessed on 3 February 2024).

[31] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proceedings of the International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649. [Online]. Available: https://doi.org/10.1109/ICIP.2017.8296962

[32] W. Xiang, D. Zhang, H. Yu, and V. Athitsos, "Context-aware single-shot detector," in *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, Lake Tahoe, NV, USA, 2018, pp. 1784–1793. [Online]. Available: https://doi.org/10.1109/WACV.2018.00198

[33] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "Bytetrack: Multi-object tracking by associating every detection box," in *Lecture Notes in Computer Science*. Cham, Switzerland: Springer, 2022, pp. 1–21. [Online]. Available: https://doi.org/10.1007/978-3-031-200472_1

[34] J. Zhu, K. Sun, S. Jia, Q. Li, X. Hou, W. Lin, B. Liu, and G. Qiu, "Urban traffic density estimation based on ultrahigh-resolution uav video and deep neural network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, pp. 4968–4981, 2018. [Online]. Available: https://doi.org/10.1109/JSTARS.2018.2879368