

IMPROVED FAULTY VEHICLE DETECTION AND CLASSIFICATION FOR INTELLIGENT TRAFFIC MANAGEMENT SYSTEM USING MACHINE LEARNING APPROACH

Abubakar Garba Bagudo¹, Abubakar Atiku Muslim², Shamsu Arzika Aliero³, Muhammad Saidu Aliero⁴, Ahmad Ibrahim Bagudo⁵

^{1,3}Department of Computer Science, Abdullahi Fodio University of Science and Tech., Aliero,

²Department of Electrical and Electronics Engineering, Abdullahi Fodio University of Science and Technology, Aliero,

⁴Department of Infor. and Comm. Tech., Abdullahi Fodio University of Science and Technology, Aliero.

⁵Department of Microbiology, Abdullahi Fodio University of Science and Technology, Aliero.

agbyoumen18@gmail.com, alतिकु@gmail.com, easykallamu@gmail.com, msaidua200@gmail.com, ahmadibagudo@ksusta.edu.ng

ABSTRACT

One of the most important aspects of a smart city is an intelligent traffic light control system. The number of people who own cars has dramatically increased, which causes traffic jams in urban areas, especially in Nigeria. One of the most difficult issues individuals deal with on a daily basis is traffic congestion, which reduces productivity at work, increases the risk of accidents, and more. The efficiency of security personnel's traffic control has not been demonstrated. This issue was not resolved by the traffic light's static or constant time under all conditions (low and high vehicle volumes). Additionally, studies that take into account counting, classifying, and detecting the number of vehicles at intersections lack sufficient accuracy in doing so because they are unable to identify and categorize defective vehicles among the vehicles, which could result in delays, waste of time, or even an accident at the intersection. In order to increase efficiency and decrease delays or time wasting, this study suggested creating an enhanced vehicle classifier with an effective timer that could identify and categorize defective vehicles among those on the side of the crossing as well as allocate the proper amount of time. Additionally, the proposed system's scalability was tested on a large dataset using a camera and machine learning approach; vehicles at intersections were detected and classified using a Faster R-CNN based detection model; and a timer was developed using the DevC++ Environment to help with appropriate timing, which in turn reduces fatalities, wastes time, and increases productivity. The suggested system's efficiency was assessed using performance measures, and it was found to have an F1 score of 95.08% and an overall accuracy of 97.52%.

KEYWORDS: Faulty Vehicles, Non-Faulty Vehicles, Faster R-CNN, Vehicle Classifier, Machine learning

I. INTRODUCTION

Allocating right-of-way to opposing traffic movements at junctions is the main function of any traffic light system. This is accomplished by establishing time intervals that permit conflicting traffic flows to safely use the same intersection. By creating a controlled environment where heavy traffic can be stopped to safely allow smaller movements through, traffic signal systems aid in managing the efficient flow of traffic and enhancing safety. These technologies can increase an intersection's traffic capacity when they are timed appropriately. According to Hasanvand et al. (2023) and Arzika et al. (2023), traffic light systems can reduce certain types of accidents, such as right-angle collisions, improve waiting times and work productivity, and increase the safety and efficiency of both vehicles and pedestrians when implemented under the proper conditions. One of the most urgent problems facing today's traffic networks is traffic congestion, which is mostly caused by heavy usage. Experts in traffic control, legislators, and engineers have suggested ways to alleviate traffic congestion, with an emphasis on either increasing the infrastructure by building additional lanes or roads or lowering traffic demand by imposing taxes and fees. However, political and practical issues have prevented these options from working (Bashir et al., 2023).

Numerous strategies have been investigated to deal with traffic control problems. The issue has not been resolved despite the implementation of threshold limitations for density verification by some researchers, such as AbdulHadi et al. (2020). According to Arman & Harco (2020) and Anna et al. (2020), the employment of infrared (IR) sensors has been unsuccessful due to the sensors' inability to accurately detect cars. Similar to this, Bashir et al. (2023) suggested using load cells to detect vehicles by weighing them. However, this approach only yields an approximation that is comparable to manual counting and does not provide accurate data on traffic volume. Given these difficulties, researchers have looked to camera-based techniques as a potentially effective way to identify and categorize moving cars. Nevertheless, there are issues with existing camera-based systems, such as when vehicles are hidden by obstacles or when overlapping vehicles are counted incorrectly as one. Also, when vehicles are faulty, they can no longer move and hence causes road block which may lead to waste of time, loss of properties and even lives because the current systems cannot recognize sides of intersection that are blocked due to faulty vehicles and therefore allocate time even when the vehicles cannot move. By incorporating machine learning approaches, this research seeks to improve vehicle recognition and classification accuracy in camera-based systems in order to address these problems. This method entails gathering data via cameras, which is subsequently utilized for training and assessing the suggested solution (Khac-Hoai & Nam Bui, 2022).

II. RELATED WORKS

One of the biggest problems people deal with on a daily basis is traffic congestion, so a smart transportation system is essential to a smart city. In addition to having a detrimental effect on productivity, traffic bottlenecks also raise the possibility of accidents. Numerous strategies have been put out to address traffic congestion, including the conventional strategy of employing security guards to manage traffic, which has shown itself to be ineffective. In a similar vein, the issue has not been solved by depending solely on set, static traffic light timings, regardless of traffic volumes. Furthermore, faulty and non-faulty vehicle classification is frequently overlooked by systems that use sensors, cameras, and IoT technologies to detect and count vehicles (Hasanvand et al., 2023; Basheer et al., 2023; AbdulHadi et al., 2020; Chilakalapalli et al., 2020; Arman et al., 2020). In a paper published in 2021, Anna et al. proposed a system that makes use of the Internet of Things (IoT) and an Adaptive Neuro-Fuzzy Inference System (ANFIS) to improve traffic conditions. The system prototype was developed utilizing cameras, an

Arduino UNO, the ThingSpeak platform, and a MATLAB SIMULINK environment for density tests. Even those machine learning-based techniques for vehicle recognition and classification, however, do not categorize cars according to their intended use. In computer vision, deep learning, and machine learning, vehicle identification and classification are crucial fields. Extensive pre-processing, such as image pre-processing (such as noise reduction, contrast correction, scaling, and background subtraction), and feature extraction are necessary to achieve the best results.

With the development of technologies like video systems, which aid in the daily supervision and monitoring of city roads and major highways, traffic management has advanced. As a result, Vehicular Detection and Classification systems for Intelligent Transportation Systems (ITS) have been developed. These systems use vehicle detection and classification to monitor smart traffic. Vehicle categorization has been done using machine learning models such as Support Vector Machines (SVM), Random Forests, Decision Trees, and Artificial Neural Networks (ANN). However, there are drawbacks, such as poor vehicle type detection rates and difficulties with camera angles. On the other hand, the Convolutional Neural Network (CNN) model, created by Mayank et al. (2019), can manage spatial invariance and shifting camera views and has demonstrated remarkable vehicle type detection accuracy (around 99.2%). There have been a lot of studies done on vehicle detection and classification. For example, Ali et al. (2023) employed inductive waveform analysis to accurately identify and categorize five vehicle types: automobiles, trucks, buses, motorbikes, and pickups. Using the Faster R-CNN method, Arzika et al. (2023) developed a deep learning-based model which achieves 94.43% accuracy in recognizing and classifying vehicles based on their functions, including VIP, emergency, and others. Nevertheless, the problem of identifying and categorizing defective cars stopped at intersections which might waste time and result in collisions is not addressed by current methods. Furthermore, they fail to take into account the allocation of time-based lanes for fault-free vehicles, which has been noted as a research gap. In order to close that gap, this work presents a deep learning-based model for vehicle recognition and classification as part of a Smart Traffic System that uses a Faster Region-based Convolutional Neural Network (Faster R-CNN) using ResNet50 architecture. This method aims to handle both the allocation of time-based lanes free of defective vehicles and the identification of defective vehicles at intersections.

III. METHODOLOGY

For the purpose of efficiently identifying and categorizing faulty and non-faulty automobiles on intersection sides, we created a model. The suggested model is based on using machine learning and cameras for vehicles detection and categorization. The Figure 1 represents the proposed system.

- (a). Getting Images of vehicles at the side of Intersection: A Camera was proposed to be used to capture images of vehicles at the roadsides or intersection for the model testing and training. But due to the limited resources we used secondary datasets.
- (b). Faulty and Non-Faulty Vehicles Detection and Classification: The captured images were processed using image processing software (labelImg) so as the model can be able to identify vehicles and classify them as faulty (damaged or abnormal vehicles) or non-faulty vehicles (normal).
- (c). Vehicles Estimation to Time Simulator: The density and types (Faulty and Non-Faulty) of the detected vehicles are supposed to be transmitted to a time simulator. The simulator predicts traffic conditions and helps allocate appropriate signal based on timing.
- (d). Allocating Time Based on the Detection and Classification: The Duration of the Traffic signal was assigned according to the Detection and classification results as well as the traffic density. Lanes without vehicles and lanes with faulty vehicles will not be allocated any time while lanes with Non-Faulty vehicles will be allocated time.

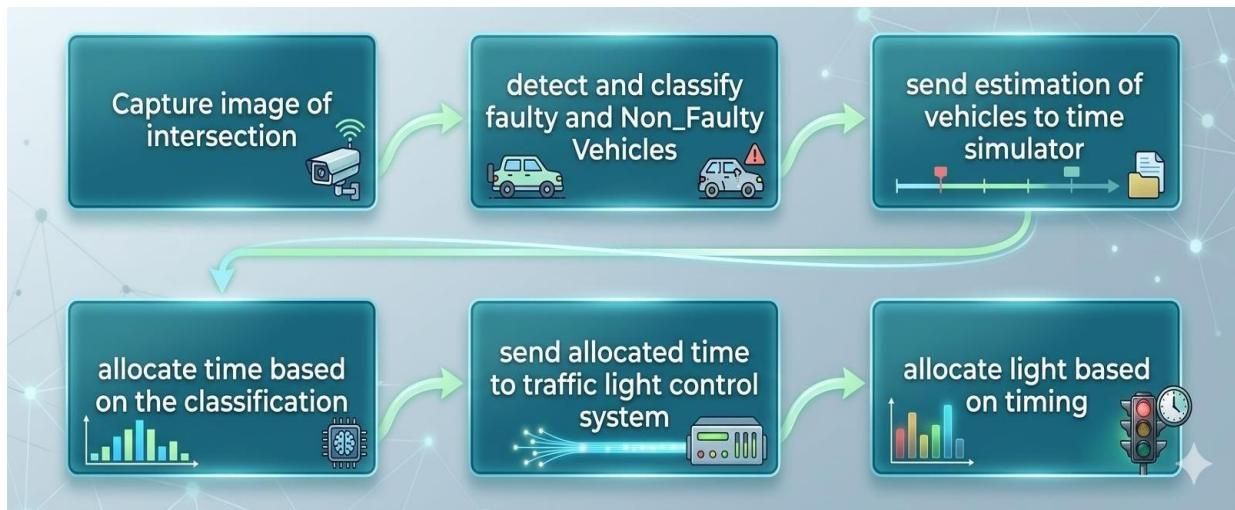


Figure 1: The Overall Block Diagram of the Proposed System

- (e). Send Allocated Time to the Traffic light Simulator: The timing information based on the detection and classification was sent to the traffic light controller, which controls the operations of the traffic signals at the side of intersection.
- (f). Allocate Light Based on Timing: The traffic lights will be on based on the allocated timings, so as to efficiently monitor traffic flow and reduces traffic congestion at the sides of intersection.
- (g). Overall System Operation: The suggested approach starts by taking pictures of each of the intersection's four (4) sides. It then goes on to identify and categorize the automobiles in the pictures as either faulty or non-faulty vehicles. The system's time simulator is required to allot time according to the forecast outcomes. In order to distribute light according to the timing from the time simulator, the traffic light control system receives the time allotted based on the estimation from the developed simulator.

IV. METHODOLOGY

A. Data Collection and Labelling

The datasets utilized in this work were sourced from publicly available and open-source internet sites, including the datasets from COCO, the UCL Machine Learning Repository, and the Kaggle.com. After that, the dataset was appropriately sorted by removing any pictures that had anything other than vehicle objects for classification. The Labeling tool was used to label the dataset for better accuracy and better result. Throughout the investigation, the dataset was used for testing and training. Eighty percent were utilized for instruction, while twenty percent were used for testing. The characteristics of the dataset used in this research work are shown in Table 1.

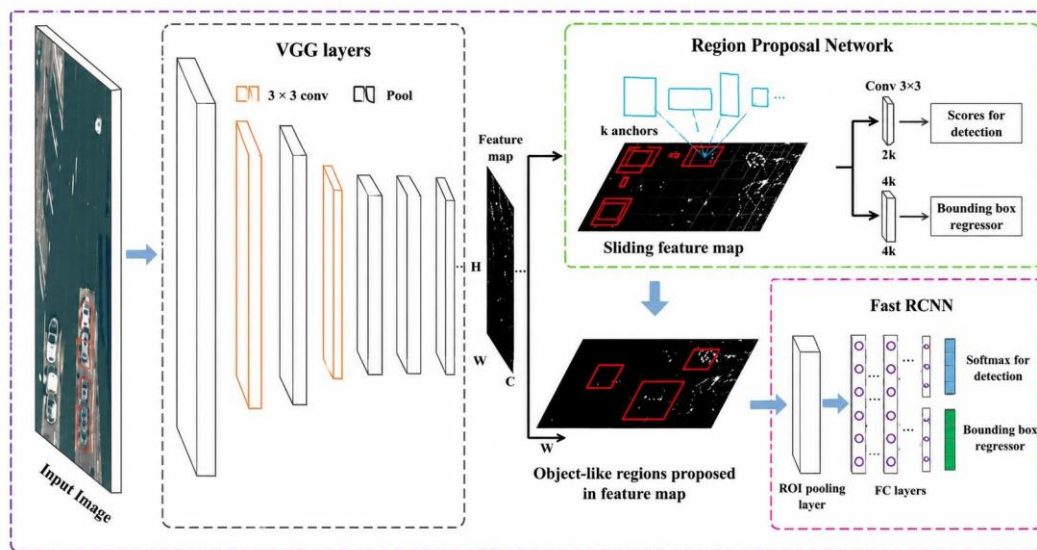
We concur that the model's capacity to generalize to new data may be impacted by the comparatively short dataset used to train deep learning models. In order to overcome this constraint, we used transfer learning using a Faster R-CNN model that had already been trained on a sizable benchmark dataset. This allowed the network to make use of previously acquired feature representations rather than starting from blank.

Table 1: Images Datasets Attributes

S/N	Name of Vehicle Class	Number of Labelled Objects	Type of Images
1	Faulty Vehicle	181	JPG
2	Non Faulty Vehicle	265	JPG

B. The CNN Model

Faster R-CNN (faster Region based Convolutional neuron network), a deep convolutional network for object detection, appears to the observer to be a single, coherent network from start to finish. The network can anticipate different things' placements quickly and accurately. R-CNN, Fast R-CNN, and the Faster R-CNN method replaced CNN. A Fast R-CNN network receives two inputs: a collection of item ideas and a whole image. The network first uses many convolutional (conv.) and max pooling layers to process the full image in order to produce a convolution feature map. For each object proposal, a region of interest (RoI) pooling layer then extracts a fixed-length feature vector from the feature map. Each feature vector is fed into a sequence of fully connected (fc) layers before splitting into two sibling output layers. One of these layers produces four real-valued integers for each of the K object classes, while the other produces softmax probability estimates over K object classes plus a catch-all "background" class. Each set of four numbers encodes refined bounding-box positions for one of the K classes. The Fast R-CNN design is shown in Figure 2.

**Figure 2:** The Faster R-CNN Architecture (Arzika et al, 2023)

C. The Model Set-Up

The implementation of the system consists of the following set of steps:

(a). The use of Labeling Tool in Labelling the Dataset

The data was sorted, examined, and labelled using "Labelimg software" after it was collected. Annotations were applied to the automobile objects in each dataset image during the labelling process, creating xml files that were then converted into csv files using Python software. Following the conversion, the train and test datasets were organized into folders, and the dataset was annotated to make it ready for testing and training. A graphical application for image annotation called LabelImg is used to label photo sets in machine learning research projects. It uses a graphical user interface called Qt and is developed in Python. ImageNet typically stores annotations as XML files in the PASCAL VOC format. The dataset was labelled

using the program. A sample of tagged car objects in an image created using the LabelImg program is displayed in Figure 3.

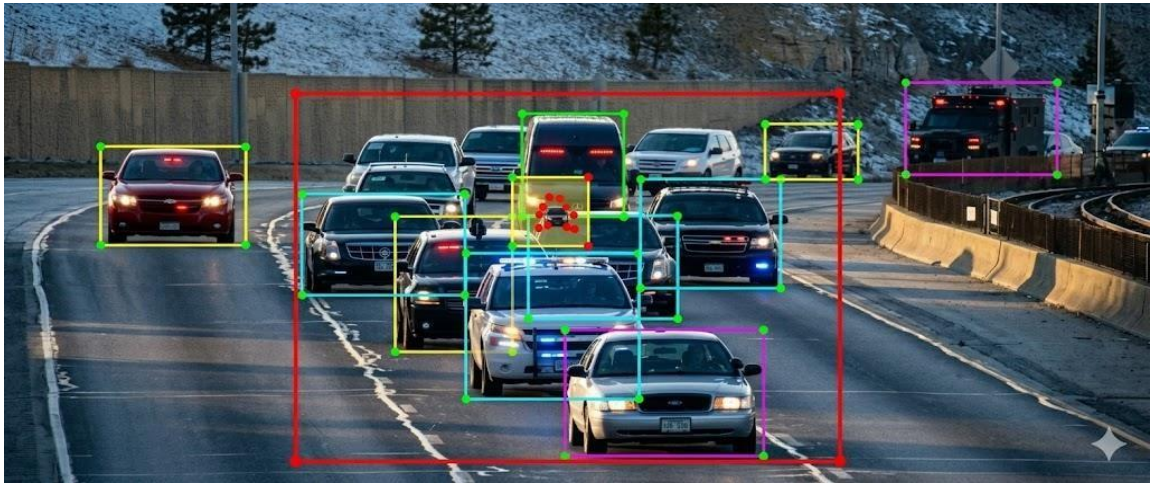


Figure 3: Labeled Image using LabelImg Software (Arzika et al, 2023)

(b). XML Files to CSV Files Conversion

The XML files were created by the labelling processes or operations using the Labelimg software. The XML files were generated automatically once the bounding box of a car object was drawn in an image. A bounding box with its own xml is drawn on each vehicle object. The bounding box coordinates of the tagged vehicle objects in each image file are contained in the xml files. A Python script was used to convert these XML files to CSV files. There were three different files created: (i) the train_labels.csv (ii) the test_labels.csv and (iii) the label_map.pbtxt are shown in Figures 4 and 5.

filename	width	height	class	xmin	ymin	xmax	ymax
images (3).jpg	275	183	Faulty	4	66	103	127
images (3).jpg	275	183	Non_Faulty	197	78	266	127
images (3).jpg	275	183	Non_Faulty	70	70	98	87
images (3).jpg	275	183	Non_Faulty	104	77	155	110
images (3).jpg	275	183	Non_Faulty	140	73	170	97
images (3).jpg	275	183	Non_Faulty	170	73	193	90
images (3).jpg	275	183	Non_Faulty	246	59	272	87
images (3).jpg	275	183	Non_Faulty	219	52	243	76
images (3).jpg	275	183	Non_Faulty	196	65	214	76
images (3).jpg	275	183	Non_Faulty	182	63	196	73
images (3).jpg	275	183	Non_Faulty	176	53	192	67
images (3).jpg	275	183	Non_Faulty	160	65	176	73
images (3).jpg	275	183	Non_Faulty	149	62	168	71
images (3).jpg	275	183	Non_Faulty	144	59	162	64
images (3).jpg	275	183	Non_Faulty	125	70	146	79

Figure 4: A Sample of XML to CSV Files Conversions

```

item {
  id: 1
  name: 'Faulty'
}

item {
  id: 2
  name: 'Non_Faulty'
}

```

Figure 5: The Classes and their Respective Ids of the Labels

(c). Setting Up and Conducting the Training

A few setups and configurations were created prior to the training procedure. Among these settings are: The train and test data used to train the vehicle detector and classifier are contained in the generated train and test record files. In the Google Collaborator environment, the GPU runtime was selected as the runtime type for increased speed. We empirically discovered that a training procedure of 15,000 steps was sufficient for training the Faster R-CNN model using 80% of our datasets. For training, a Google Colab environment with free GPU Runtime access was utilized. After training, the inference graph was generated. It has a recently trained car detector and classifier that can identify and categorize both defective and non-defective vehicles.

(d). Evaluation of the Model

There are two potential positive and two potential negative in vehicle detection and classification: A False Positive (FP) is a class that is mistakenly identified, while a True Positive (TP) is a prediction that is accurately detected. In a similar vein, a False Negative (FN), which is a mistakenly rejected class, and a True Negative (TN), which is a detection that is appropriately rejected. According to Bie et al., (2023), the following metrics were used to assess the newly-trained detector and classifier's performance:

- i. Accuracy is calculated by dividing the total number of true detections by the total number of detections.

It was computed using equation 1:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Where TP is True Positive, TN is True Negative, FP is False Positive and FN is False Negative

- ii. Recall measure provides an answer to the question of what percentage of real positives were correctly identified given a class. It was computed using equation 2:

$$Recall = \frac{TP}{TP+FN} \quad (2)$$

- iii. Precision indicates the likelihood that the classification is accurate. The performance metrics for Recall, Precision, and F1 Score are expressed mathematically using equation 3:

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

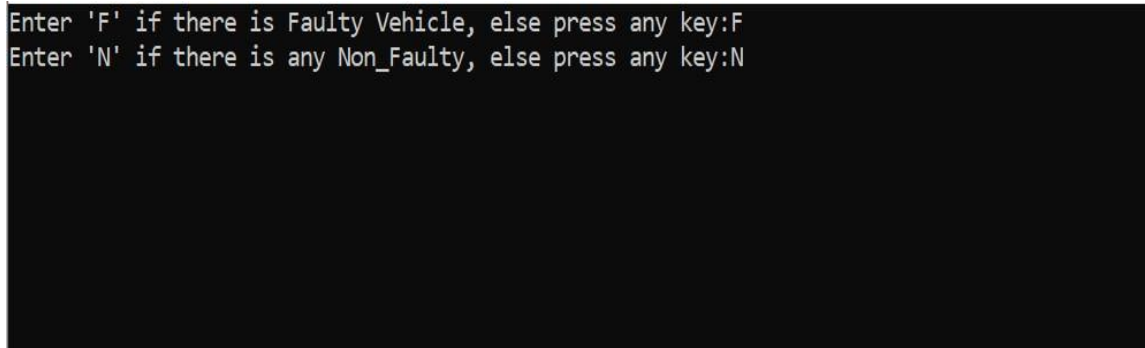
- iv. F1 score is the harmonic mean of precision and recall. It was computed using equation 4:

$$F1\ Score = 2 \times \left(\frac{Precision \times Recall}{Precision + Recall} \right) \quad (4)$$

For class-imbalanced data sets with an unequal distribution of positive and negative predictions, the F1 score is a better metric. A class-imbalanced data set has varying numbers of classifier predictions for each class.

(e). The Time Simulator

The most important component of a smart traffic system is the timing component. This study offered a time simulator that allocates time to the intersection's sides for traffic control based on the availability of the selected vehicle classes (Faulty class and Non_Faulty class) from the model generated results. The simulator was developed using the C++ programming language due to its significance and user-friendliness. The highways under examination include 3 lanes, and the timer was built with a capacity of ten automobiles per each lane. Although this will be done in later work, the simulator is intended to ingest a file containing the estimation of detected classes and vehicle densities and operate based on the vehicle estimation from the model. The Figures 6 and 7 show DevC++ window showing simulator's interface and flowchart representation of the simulator.



```
Enter 'F' if there is Faulty Vehicle, else press any key:F
Enter 'N' if there is any Non_Faulty, else press any key:N
```

Figure 6: DevC++Window Showing Simulator in Getting Detected Classes

The simulator checks for the presence of a faulty vehicle first in order to know whether to allocate time to that side or not. If the Detector & Classifier detects a Faulty vehicle the simulator will jump that side of intersection and move to the next side and allocate time if there is no any Faulty vehicle in that side. This is to ensure optimization of time usage. When there is a faulty vehicle in a particular side of intersection, allocating time to that side may cause accident, traffic jams and indeed causes waste of time and resources because the faulty vehicle may block the side of the road and the side is given green light to pass while the side is blocked and other sides without faulty vehicles will have to wait for their turns, so jumping the side with faulty vehicles in time allocation will help in eradicating such problems. In order to allow any vehicle within the lane's threshold to pass within the allotted time, the timer allocates 50 seconds to that lane.

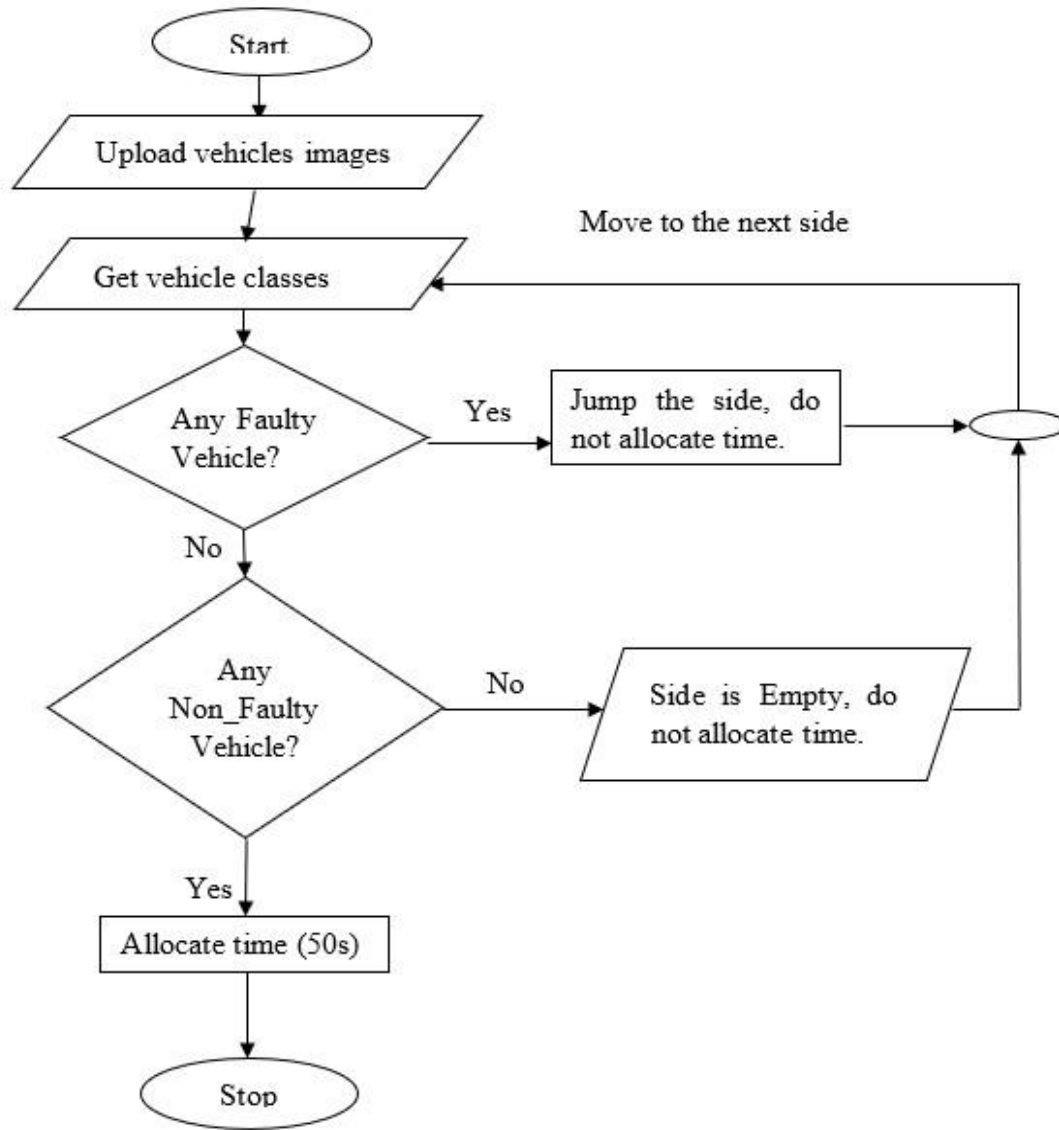


Figure 7: Flow chart of Time Simulator

V. RESULTS AND DISCUSSION

Twenty-three (23) photos, each including several vehicle objects of various classes, totaling one thousand and seventeen (1,017) vehicle objects, were used to test the trained vehicle detection and classification model. One of the shortcomings of earlier research projects, overlapping (Arzika *et al.*, 2023), is no longer an issue. The cars shown in Figures 8, 9, and 10 were detected and classified even though some of them overlapped. Nonetheless, the figures display the 100%, 99%, 98%, and 97% accurately identified and classified automobiles enclosed by various bounding boxes.



Figure 8: Identified Vehicles Objects during Testing Period from Google Collab Simulation Environment

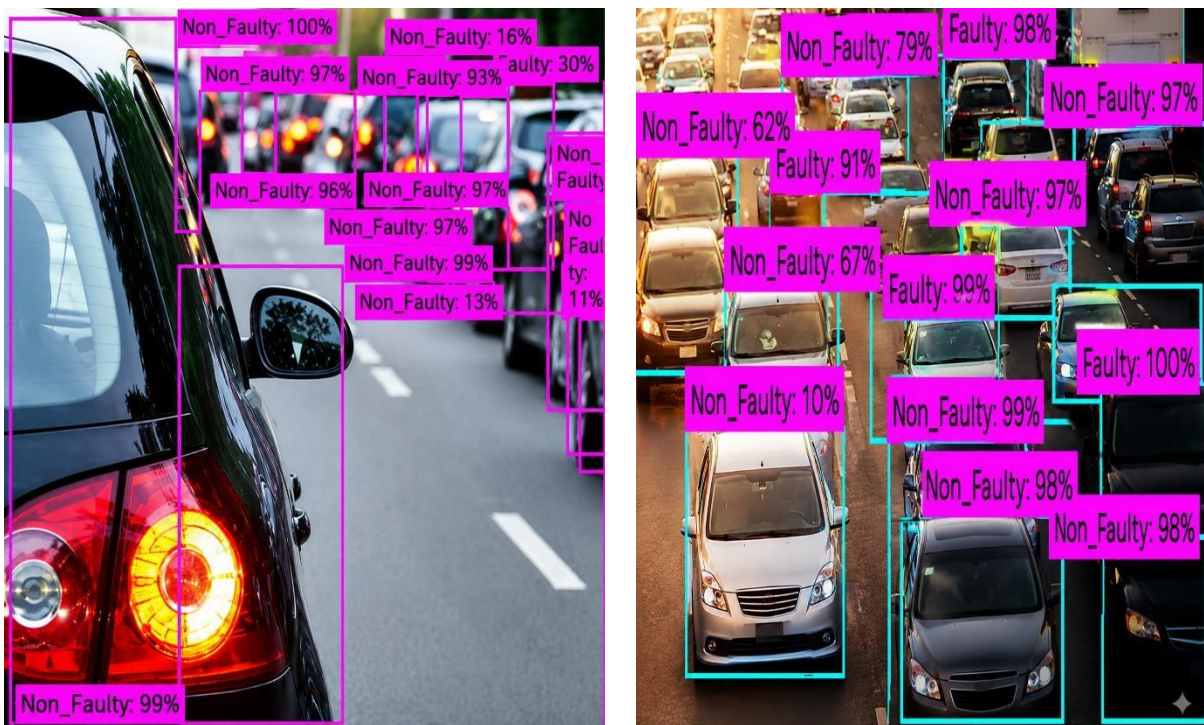


Figure 9: Identified Vehicles Objects during Testing Period from Google Collab Simulation Environment



Figure 10: Identified Vehicles Objects during Testing Period in Overlapping from Google Colab Simulation Environment.

Additionally, the newly trained model was evaluated using performance metrics. Twenty-three (23) photos, each carrying several vehicle objects of various classes, totaling one thousand and seventeen (1,017) vehicle objects, were used to test the model. The tables 2 and 3 show the Confusion Matrix and Accuracy of the developed model.

Table 2: Vehicle Classes in a Confusion Matrix

		Machine Predictions		
Actual Predictions	Classes	Faulty	Non_Faulty	Total Predictions
	Faulty	60	8	68
	Non_Faulty	3	372	375

Table 3: Accuracy of Vehicle Classification

CLASSES	TP	TN	FP	FN	DETECTION ACCURACY
FAULTY	60	372	8	3	97.52%
NON_FAULTY	372	60	3	8	97.52%
TOTAL	432	432	11	11	97.52%

Since the vehicle detector and classifier performed well in every class with an overall accuracy rate of 97.52%, We can say that its performance is significantly optimal in terms of accuracy. The accuracy of

97.52% was the same for the defective and non-faulty classes. We chose to use the F1 score as an additional performance indicator to gauge accuracy even if the vehicle classification did not turn out to be a class imbalanced dataset. According to Arzika et al., (2023), the F1 score is the harmonic mean of recall and precision. The harmonic mean shows the ratio of the number of real-world vehicle samples that are correctly classified to the likelihood that the classification is correct. The detector and classifier's performance results in terms of recall, precision, and F1 score are displayed in Table 4.

Table 4: The Model F1 Score Performance

CLASSES	RECALL	PRECISION	F1 SCORE
FAULTY	95.24%	88.24%	91.61%
NON_FAULTY	97.89%	99.20%	98.54%
AVERAGE	96.57%	93.72%	95.08%

According to the F1 score performance metric, the Non_Faulty vehicles class has the highest F1 score of 98.54%. This indicates that every vehicle that is detected and classified as Non_Faulty with 99.20% precision accuracy is an actual Non_Faulty vehicle, and every actual Non_Faulty with 97.89% recall accuracy is classified as Non_Faulty. When compared to the Non_Faulty class, the Faulty class has the lowest F1 score of 91.61%. This indicates that for every 100 vehicles categorized as Faulty, 88.24% of them are actually Faulty, and for every 100 true Faulty in the waiting lane, 88% are appropriately identified. In summary, the detector and classifier performed well, as indicated by the overall F1 score of 95.08%. Nonetheless, the following are of the vehicle detection and categorization errors:

- i. Due to issues with labelling, several vehicles were not found.
- ii. Bounding boxes and vehicle size issues cause several vehicle objects to combine into a single box.
- iii. The detected vehicle was misclassified because of the similarities between the vehicles; this is because the classes of Faulty Vehicles and Non_Faulty vehicles have some characteristics in common. Some of the mistakes are depicted in Figure 11.

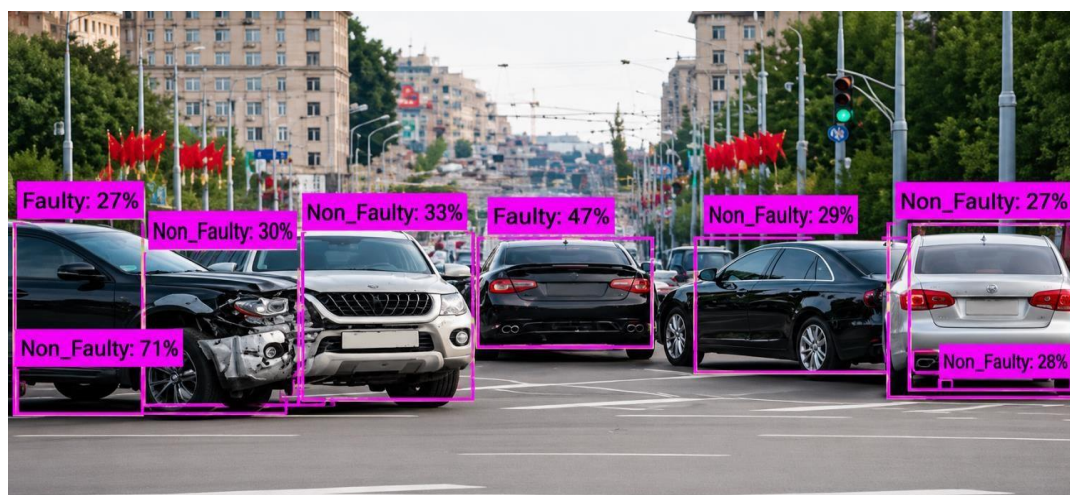


Figure 11: Errors in Vehicle Class Detection from Google Colab Simulation Environment.

This study shows that when it comes to vehicle recognition and classification, the machine learning technique performs better since it provides high accuracy utilizing performance metrics like recall, precision, F1-score, and so forth. Furthermore, in contrast to other studies when these factors were heavily taken into account, the vehicle detection accuracy was not significantly impacted by vehicle overlap, night-time photo capture, or noisy background. Regarding classification, every class had been accurately identified, and there was a good chance that the forecast would be accurate. In the meanwhile, there were still misdetections that might have been brought on by threshold setting or the labelling procedure. The classifier accurately predicted examples of all the classes (Faulty and Non_Faulty) based on the F1-score result. The time simulator was verified, and no problems were noted. Future work will take into account the estimation or counting of vehicles in each class, which is another crucial component as it can aid in producing inputs for the time simulator. Therefore, research studies using Faster Region Based Convolutional Neural Networks (Faster R-CNN) and machine learning approaches outperform the current implementations of ANN and SVM in terms of vehicle detection and classification, as demonstrated by the overall accuracy of 97.52% and the F1 score result of 95.08%.

LIMITATIONS

We concur that a more thorough assessment of the suggested Faster R-CNN model's performance would come from contrasting it with other cutting-edge object identification techniques. However, this study's main goal was to use the provided dataset to examine the viability and efficiency of Faster RCNN for faulty vehicle detection. Comprehensive benchmarking against several other models was not carried out because of the current study's narrow scope and computing limitations. Additionally, we have recognized thorough comparative trials as a crucial area for future research and acknowledged this as a weakness of the current work. We also concur that a quantitative assessment of the suggested time allocation simulator would strengthen the evidence supporting its efficacy. Developing and proving the viability of an adaptive time allocation framework coupled with the Faster R-CNN-based problematic vehicle detection system was the main goal of this project. As a result, rather than conducting a thorough assessment of traffic performance, the current study concentrated on the operational design of the simulator. We acknowledge this limitation and that comprehensive simulation experiments utilizing common traffic performance metrics, such as average vehicle waiting time, queue length, traffic throughput, delay, and congestion level under various traffic conditions, will be part of future work.

VI. CONCLUSION

This study shows that when it comes to vehicle recognition and classification, the machine learning technique performs better since it provides high accuracy utilizing performance metrics like recall, precision, F1-score, and so forth. Furthermore, in contrast to other studies when these factors were heavily taken into account, the vehicle detection accuracy was not significantly impacted by vehicle overlap, night-time photo capture, or noisy background. Regarding classification, every class had been accurately identified, and there was a good chance that the prediction would be accurate. In the meanwhile, there were still misdetections that might have been brought on by the threshold setting or labelling procedure. The classifier accurately predicted examples of all the classes (Faulty and Non_Faulty) based on the F1-score result. The time simulator was verified, and no problems were noted. Future work will take into account the estimation or counting of vehicles in each class, which is another crucial component as it can aid in producing inputs for the time simulator. Therefore, research studies using Faster Region Based Convolutional Neural Networks (Faster R-CNN) and machine learning techniques outperform the current implementations of ANN and SVM in terms of vehicle detection and classification, as demonstrated by the overall accuracy of 97.52% and the F1 score result of 95.08%. Considering the conclusions mentioned above, the recommendations below are proposed:

- i. In machine learning, a crucial part of the process is preparing the training and testing datasets and labelling vehicle objects in images, as these serve as the input from which the machine learns. It is advisable to label the vehicle objects, particularly the “Others class” due to its low F1 score, in a manner that allows the classifier to differentiate between their features.
- ii. Aside from the Faster RCNN, the Tensor-flow object detection API includes various other pretrained models. Future research can choose from other pre-trained models like Mask CNN, ANN, SSD, etc.
- iii. Therefore, it is advisable to incorporate this method into the traffic control systems of future smart cities for vehicle detection and classification.
- iv. Upon implementation, the research work will contribute to reducing traffic jams, thereby decreasing accidents and delays while enhancing work productivity.
- v. Once put into practice, it can assist the government in evaluating roads to determine when and where to construct underpasses, flyovers, and other road improvements.
- vi. It aids traffic control systems and the broader intelligent transportation system in managing government officials' convoys with special attention, thereby contributing to enhanced work productivity.

REFERENCES

- [1] Hasanvand, M., Nooshyar, M., Moharamkhani, E., & Selyari, A. (2023, April). Machine learning methodology for identifying vehicles using image processing. In *Artificial Intelligence and Applications* (Vol. 1, No. 3, pp. 170-178).
- [2] Arzika, S. A., Adeshina, S. A., & Suleiman, M. A. (2023). Improved traffic control system through detection and classification of vehicles using camera-based crowd sensing with deep learning approach. *Nile Journal of Engineering and Applied Science*, 1(1), 0-0.
- [3] Basheer Ahmed, M. I., Zaghdoud, R., Ahmed, M. S., Sendi, R., Alsharif, S., Alabdulkarim, J., ... & Krishnasamy, G. (2023). A real-time computer vision-based approach to detection and classification of traffic incidents. *Big data and cognitive computing*, 7(1), 22.
- [4] Ali, S., Jalal, A., Alatiyyah, M. H., Alnowaiser, K., & Park, J. (2023, June). Vehicle detection and tracking in UAV imagery via YOLOv3 and Kalman filter. In *CMC*.
- [5] AbdulHadi M. Alaidi, Ibtisam A. Aljazeera, Haider Th. Salim AlRikabi, Ibrahim Nasir Mahmood and Faisal Theyab Abed (2020). “Design and Implementation of a Smart Traffic Light Management System Controlled Wirelessly by Arduino” in international journal of interactive mobile technology iJIM – Vol. 14, No. 7,.
- [6] Chilakalapalli Yaswanthasai , Chaduvula Sushma Rani, Nagam Rangasai, Sathujoda Nikhil, Macharla Umamahesh, Tetla Srilatha (2020). “Smart Traffic Monitoring” in International Journal Of Innovative Research In Technology IJIRT, Volume 6 Issue 12.
- [7] Arman Syah Putra and Harco Leslie Hendric Spits Warnars (2020). “Intelligent Traffic Monitoring System (ITMS) for Smart City Based on IoT Monitoring”.
- [8] Anna Merine George, V.I.George and Mary Ann George (2021). “IOT based Smart Traffic Light Control System” in International Conference on Control, Power, Communication and Computing Technologies (ICCPCT).
- [9] Khac-Hoai Nam Bui, "Real-Time Traffic Flow Management Based on Inter-Object Communication: A Case Study at Intersection," in 7th International Conference on Ambient Intelligence (ISAmI'2020).
- [10] Aneesa Saleh, Steve A. Adeshina, Ahmad G aladima, & Okechukwu Ugweje (2022), “An Intelligent Traffic Control System” in International Conference on Electronics Computer and Computation (ICECCO), 978-1-5386-2501-9/22/ c 2022 IEEE.
- [11] Mayank singh Chauhan, Arshdeep Singh & Mansi Khemka (2019) “Embedded CNN based vehicle classification and counting in non-laned traffic” ICTD '19, January 4-7, Ahmedabad, India.
- [12] Bie, M., Liu, Y., Li, G., Hong, J., & Li, J. (2023). Real-time vehicle detection algorithm based on a lightweight You-Only-Look-Once (YOLOv5n-L) approach. *Expert Systems with Applications*, 213, 119108.