Vehicle Detection and COUNTING System Using YOLOv7

[1]*Karthik G*
Assistant Professor: Mr. S. Vignesh
M.Sc., Data Science and Business Analysis
Rathinam College of Arts and Science
Coimbatore, India
karthikgopal1405@gmail.com, 6374547453

Abstract—The lack of proper road maintenance in highway lanes has led to an increased number of accidents and injuries to the public. In order to address this issue, a solution was developed using the YOLO v7 (You Look Only Once) machine learning model for object detection in videos. The model detects objects and creates bounding boxes for three classes of vehicles (Car, Bike, Heavy Truck) and generates a txt file with the object count. By analyzing this count, we can determine the number of vehicles that have traversed a specific road and alert the government or public safety officials to prioritize road maintenance during specific periods to prevent accidents and injuries. The proposed solution aims to reduce the number of accidents and injuries caused by inadequate road maintenance in highway lanes.

Keywords—YOLOv7,Object detection

I. INTRODUCTION

The world is constantly changing, and the pace of change has accelerated more than ever before. As a result, the technologies we have developed, as well as those current in development, must keep up with the challenges that arise from these changes. This is especially true since our natural resources are depleting rapidly, and we must find solutions that are both efficient and economically beneficial. Artificial Intelligence (AI) is one of the most advanced technologies available today, and it can be used to solve a wide range of problems with minimal impact on our planet.AI systems are created by teaching them about specific tasks or objectives that need to be achieved. This process is known as knowledge transfer, and there are several methods for doing so, including machine learning, deep learning, and creating neural networks. With these methods, an AI system can learn from vast amounts of data and adapt to new situations, making it a highly effective problem-solving tool.One of the many challenges we face today is the increasing number of vehicles on the road, which is causing damage to our infrastructure at an unprecedented rate. Fortunately, the proposed model can help address this issue. By using a trained YOLO v7 model with a custom dataset, the system can accurately detect vehicles and generate a txt file with information about each one. The system can then count the vehicles and calculate how many have crossed from point A to point B. This information can be used to inform the road safety department about necessary maintenance, which can help reduce the impact of high volume vehicle usage on our roads.

II. SCOPE OF THE PROJECT

The proposed vehicle detection and counting system, which utilizes YOLO v7 and imagehash technologies, and is developed using Streamlit framework, is expected to benefit a wide range of stakeholders. Firstly, the road safety department can utilize the system to identify and monitor traffic flow, track congestion and road usage patterns, and identify potential safety hazards. This information can then be used to optimize road maintenance and infrastructure development, leading to improved road safety and reduced traffic accidents.

In addition, the system can be useful for transportation and logistics companies in monitoring and optimizing their fleet operations. They can use the data generated by the system to manage their fleets more effectively, monitor driver behavior, and optimize routes to reduce fuel consumption and emissions.

Furthermore, urban planners and policymakers can utilize the data generated by the system to analyze and optimize urban mobility and transportation planning. The data can be used to identify high-traffic areas, optimize public transportation routes, and plan infrastructure development and road maintenance.

Overall, the proposed vehicle detection and counting system has the potential to benefit a wide range of stakeholders in various industries, including transportation, logistics, urban planning, and government. It can lead to improved road safety, reduced traffic congestion, and more efficient transportation and logistics operations, ultimately contributing to a more sustainable and livable urban environment.

III. METHODOLOGY

YOLO Architecture :

”You Only Look Once: Unified, Real-Time Object Detection,” Joseph Redmon ,Santosh Divvala, and Ross presented the YOLO Deep Learning architecture, which takes a completely different approach. It uses a smart convolutional neural network (CNN) for real-time object detection. YOLO is a Deep Learning algorithm used for object detection from images and videos .YOLO stands for “You Only Look Once”, it is a popular family of real-time object detection algorithms. It looks the image at once, in a clever way and it will detect all the classes in the image. The YOLO algorithm performance is better compared to other object detection algorithms in Deep Learning in addition to improved bounding box Intersection over Union and improved prediction accuracy, YOLO offers the intrinsic benefit of speed. With a maximum frame rate of 45 FPS, YOLO is a lot faster algorithm than its competitors.

YOLO executes all of its iteration with the aid of a single layer which is fully connected, in contrast to techniques like Faster RCNN that functions by first employing the Region Proposal Network to identify potential regions of interest before doing recognition on those regions independently. Thus, methods that employ region proposal networks end up executing numerous iterations on the same image, but YOLO per-forms this action in a single flow. Average Precision (AP) is the area under a precision vs. recall curve for a collection of predictions is used to compute average precision. Recall is calculated as the ratio of all predictions made by the model for a class to all labels that have already been assigned to that class. The ratio of true positives to all of the predictions the model made, on the other hand, is known as precision. We can find the average precision for each class in the model by looking at the area under the precision vs recall curve. Mean Average Precision is the word used to describe the average of this number

YOLO v7 :

The present YOLO algorithm which is the YOLO v7 outperforms all earlier object detection algorithms and YOLO versions in terms of speed and precision. As a baseline, the performance of YOLOv7 was compared to that of YOLOv4 and YOLOv5, as well as YOLOR. The models followed the identical training conditions. When compared to cutting-edge object detectors, the new YOLOv7 has the best speed-to-accuracy balance. With speeds ranging from 5 to 160 frames per second, YOLOv7 outperforms all prior object detectors in terms of both speed and accuracy. While obtaining 30 FPS or more utilizing a GPU V100, the YOLO v7 algorithm achieves the highest accuracy among all other real-time object detection models. The modifications that were made to Yolo v7 are listed below.

1. E-ELAN.
2. Concatenation-based Models Scaling with Models.

From YOLOv4, Scaled YOLOv4, and YOLO-R, the architecture is derived. In order to create a new and superior YOLOv7, additional tests were conducted using these models as a foundation.

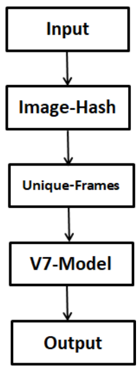
The computational building element of the YOLOv7 backbone is the E-ELAN. It draws influence from earlier studies on network effectiveness. It was created by looking at the following factors that influence speed and accuracy.

Image Hash:

Image hash is a Python library that generates a unique hash value for an image based on its content. It is commonly used for identifying duplicate or similar images in large datasets, as images with similar content will have the same or similar hash values. This makes it easy to compare and find similar images in a dataset without the need for complex image processing techniques.

The Python code utilizes image hashing technique to identify unique frames from a video file. This technique generates a unique fixed-length hash string for each image, which can be used for comparing two images and determining whether they are the same or not.

The video frames are saved as image files in the 'frames' folder, and the average hash is calculated for each image using the 'image hash' library. This generated hash is stored as a string in the 'hash' variable.



To store only unique frames, a set named 'hashes' is created to store all the unique hashes for each frame. When a new hash is generated, it is compared with the set of unique hashes. If the hash is not present in the set, it means that the frame is unique, and the hash is added to the set of unique hashes. The frame is saved to a separate folder named "unique-frames" using the 'cv2.im write' function.

By comparing the generated has with the set of unique hashes, only unique frames are saved in the 'unique-frames' folder, reducing the redundancy and saving storage space, especially when dealing with a large number of frames. Using the saved unique frames, further processing, such as object detection, can be carried out without processing redundant frames. This improves the performance of the object detection algorithm and reduces processing time. Thus, image hashing is an efficient technique used to identify and store unique frames from a video file, which can be utilized in various computer vision tasks.

Object Detection and StreamLit Web:

Object detection using YOLOv7 on the saved unique frames. The unique frames are stored in a folder named 'unique-frames'.

For object detection, the YOLOv7 algorithm is used, which is a state-of-the-art deep learning model for object detection. The 'detect.py' script from the YOLOv7 repository is used for this purpose.

To run the 'detect.py' script, a subprocess is created using the 'subprocess' library in Python. The subprocess runs the 'detect.py' command with the required arguments, such as the path to the saved unique frames folder, the path to the YOLOv5 weights file, and the confidence threshold.

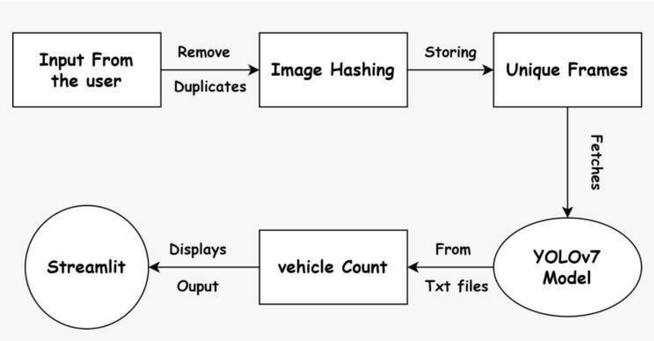
Once the object detection is completed using YOLOv7, the number of vehicle detections is counted for each class. The function 'count-vehicle-detections' is used for this purpose, which takes as input the path to the video file and the path to the directory containing the label files generated by YOLOv7 during object detection.

The function reads the label files, extracts the class name and bounding box coordinates of each detection, and counts the number of detections for each class. The function returns a dictionary containing the count of vehicle detections for each class.

The results of the object detection are then rendered in a web page using the 'render-template' function from the Stream lit. The web page displays the number of vehicle detections for each class.

System Design:

The system design includes creating a YOLOv7 model trained on a custom dataset to classify three classes. Streamlit web application is utilized to enable the end-users to input videos for object detection. Image hashing is utilized to filter out duplicate images and store unique images in a separate folder. Object detection is then performed on the unique folder, and the system displays the count of detected objects alongside the corresponding files. This approach offers a simple and intuitive user interface and facilitates efficient object detection and counting.



IV. RESULT

The proposed solution using YOLO v7 machine learning model for object detection in highway lanes has the potential to significantly reduce the number of accidents and injuries caused by inadequate road maintenance. The model detects and creates bounding boxes for three classes of vehicles (Car, Bike, Heavy Truck) and generates a count of the number of objects that have traversed a specific road. By analyzing this count, government officials and public safety officials can prioritize road maintenance during specific periods and prevent accidents and injuries. This solution is expected to improve road safety and reduce the number of accidents and injuries, ultimately leading to a safer and more efficient transportation system. However, it is important to note that the implementation of this solution may require significant investment and infrastructure, including the installation of cameras, and the development of an automated system for analyzing the object count. Further, it is essential to ensure the privacy and security of the data collected by the system. In conclusion, the proposed solution using YOLO v7 machine learning model for object detection has the potential to make a significant contribution to improving road safety and reducing the number of accidents and injuries caused by inadequate road maintenance in highway lanes.

Figure 1

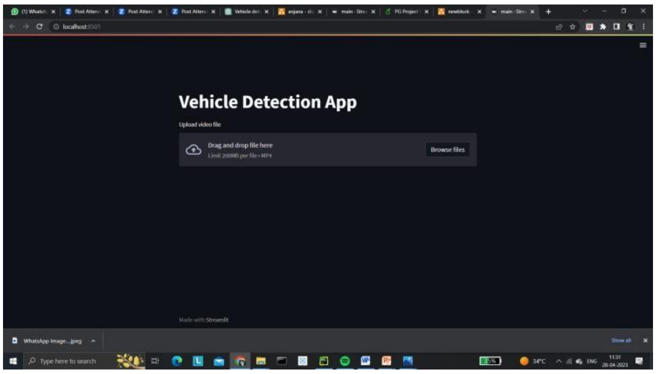


Figure 1. Stream lit page

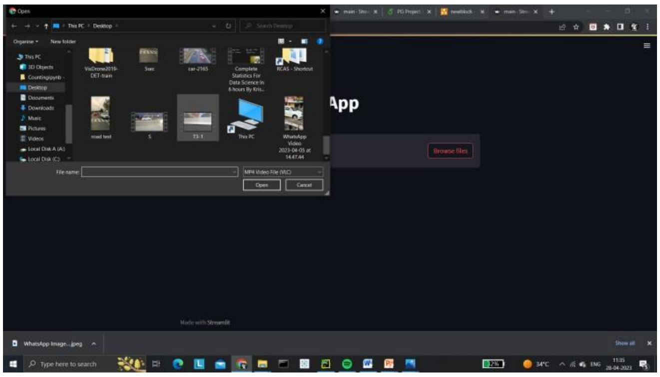


Figure 2 . Inserting Input

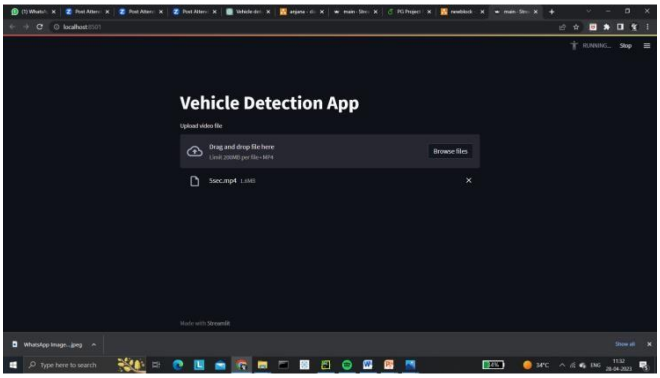


Figure 3 Model Prediction

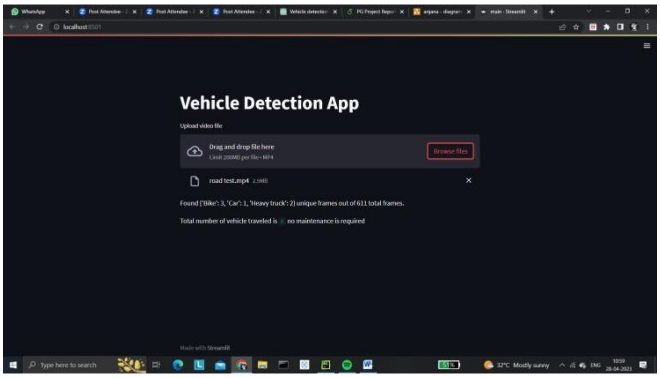


Figure 4 Output

V. CONCLUSION

The proposed solution that employs the YOLO v7 machine learning model for object detection shows promise in addressing the issue of inadequate road maintenance in highway lanes, which has led to an increased number of accidents and injuries to the public. By accurately detecting and counting the number of vehicles that have traversed a specific road and generating data for government or public safety officials to prioritize road maintenance during specific periods, the solution can reduce the number of accidents and injuries caused by inadequate road maintenance. Nonetheless, it is important to note that the solution may require further testing and refinement to ensure its effectiveness and accuracy in various environments and situations. If implemented effectively, the solution has the potential to improve public safety, prevent accidents, and save lives by ensuring proper road maintenance.

REFERENCES :

[1]S. Choudhury, S.P. Chattopadhyay, T.K. Hazra, "Vehicle detection and counting using Haar feature-based classifier", Industrial Automation and Electromechanical Engineering Conference, pp. 106-109, 2017

[2] Swathy S Pillai, Radhakrishnan B, “Night Time Vehicle Detection Using Tail Lights: A Survey”, International Journal of Engineering Research and General Science Volume 4, Issue 2, March-April, 2016.

[3] Orcan Alpar, “Corona segmentation for nighttime brake light detection”, IET Intell. Transp. Syst., 2016, Vol. 10, Iss. 2, pp. 97–105

[4] R. O’Malley, E. Jones, and M. Glavin, “Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions,” IEEE Trans. Intell. Transp. Syst., vol. 11, no. 2, pp. 453–462, Jun. 2010.

[5] S. Eum and H. G. Jung, “Enhancing light blob detection for intelligent headlight control using lane detection,” IEEE Trans. Intell. Transp. Syst., vol. 14, no. 2, pp. 1003–1011, Jun. 2013.

[6] P. Menezes, J. C. Barreto, J. Dias, "Face Tracking based on Haar-like Features and Eigen faces." IFAC/EURON Syfmposium on Intelligent Autonomous Vehicles. Vol. 500, 2004.

[7] C. Emmy Premal, S. S. Vinsley, Image Processing Based Forest Fire Detection using YCbCr Colour Model, 2014 International Conference on Circuits, Power and Computing Technologies.

[8] Sun Yat-sen University Night-time Vehicle Dataset (http://www.carlib.net/?page_id=35)

[9] L. Cuimei, Q. Zhiliang, J. Nan, W. Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers", 2017 13th IEEE Int. Conf. Electron. Meas. Instruments, pp. 483-487, 2017.

[10] Z. Jiao, G. Jia, and Y. Cai, “A new approach to oil spill detection that combines deep learning with unmanned aerial vehicles,” Comput. Ind. Eng., vol. 135, pp. 1300–1311, Sep. 2019, doi: 10.1016/j.cie.2018.11.008.

[12] Y. Wu, Y. Qin, Z. Wang, and L. Jia, “A UAV-Based Visual Inspection Method for Rail Surface Defects,” Appl. Sci., vol. 8, no. 7, p. 1028, Jun. 2018, doi: 10.3390/app8071028.

[13] B. Mishra, D. Garg, P. Narang, and V. Mishra, “Drone-surveillance for search and rescue in natural disaster,” Comput. Commun., vol. 156, pp. 1–10, Apr. 2020, doi: 10.1016/j.comcom.2020.03.012.

[14] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards RealTime Object Detection with Region Proposal Networks,” ArXiv150601497 Cs, Jan. 2016, Accessed: May 24, 2020. [Online]. Available: http://arxiv.org/abs/1506.01497.

[15] A. Ammar, A. Koubaa, M. Ahmed, and A. Saad, “Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3,” ArXiv191007234 Cs Eess, Apr. 2020, Accessed: May 25, 2020. [Online]. Available: http://arxiv.org/abs/1910.07234.

[16] “Computational Vision and Geometry Lab.” https://cvgl.stanford.edu/projects/uav_data/ (accessed May 25, 2020).

[17] ÜY. Kharuzhy, jekhor/aerial-cars-dataset. 2020.

[18] I. Bozcan and E. Kayacan, “AU-AIR: A Multi-modal Unmanned Aerial Vehicle Dataset for Low Altitude Traffic Surveillance,” ArXiv200111737 Cs, Feb. 2020, Accessed: May 25, 2020. [Online]. Available: http://arxiv.org/abs/2001.11737.

[19] Z. Xu, H. Shi, N. Li, C. Xiang, and H. Zhou, “Vehicle Detection Under UAV Based on Optimal Dense YOLO Method,” in 2018 5th International Conference on Systems and Informatics (ICSAI), Nanjing, Nov. 2018, pp. 407– 411, doi: 10.1109/ICSAI.2018.8599403.

[20] G. Oltean, C. Florea, R. Orghidan, and V. Oltean, “Towards Real Time Vehicle Counting using YOLO-Tiny and Fast Motion Estimation,” in 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging
(SIITME), Cluj-Napoca, Romania, Oct. 2019, pp. 240–243, doi: 10.1109/SIITME47687.2019.8990708.

[21] J. Tao, H. Wang, X. Zhang, X. Li, and H. Yang, “An object detection system based on YOLO in traffic scene,” in 2017 6th International Conference on Computer Science and Network Technology (ICCSNT), Dalian,

China, Oct. 2017, pp. 315–319, doi: 10.1109/ICCSNT.2017.8343709.

[22] B. Xu, B. Wang, and Y. Gu, “Vehicle Detection in Aerial Images Using Modified YOLO,” in 2019 IEEE 19th International Conference on Communication Technology (ICCT), Xi’an, China, Oct. 2019, pp. 1669–1672, doi: 10.1109/ICCT46805.2019.8947049.

[23] J. O. du Terrail and F. Jurie, “Faster RER-CNN: application to the detection of vehicles in aerial images,” ArXiv180907628 Cs, Sep. 2018, Accessed: Jun. 13, 2020. [Online]. Available: http://arxiv.org/abs/1809.07628.

[24] J. Lee, J. Wang, D. Crandall, S. Sabanovic, and G. Fox, “Real-Time, Cloud-Based Object Detection for Unmanned Aerial Vehicles,” in 2017 First IEEE International Conference on Robotic Computing (IRC), Taichung, Taiwan, Apr. 2017, pp. 36–43, doi: 10.1109/IRC.2017.77.

[25] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” ArXiv180402767 Cs, Apr. 2018, Accessed: Jun. 08, 2020. [Online]. Available: http://arxiv.org/abs/1804.02767.

[26] W. Liu et al., “SSD: Single Shot MultiBox Detector,” in Computer Vision – ECCV 2016, Cham, 2016, pp. 21–37, doi: 10.1007/978-3-319-46448-0_2.

[27] “tensorflow/models,” GitHub. https://github.com/tensorflow/models (accessed Jun. 08, 2020).

[28] “YOLO: Real-Time Object Detection.” https://pjreddie.com/darknet/yolo/ (accessed Jun. 08, 2020).