ISSN 2277-2685
IJESR/April-June. 2025/ Vol-15/Issue-2s/288-293
Akkinapalli Usha Praneetha *et. al.*, / International Journal of Engineering & Science Research

An Efficient Algorithm for Vehicle Detection and Counting

Akkinapalli Usha Praneetha
Pg Scholar, Department Of Mca, Dnr College, Bhimavaram, Andhra Pradesh.
K.Rambabu
(Assistant Professor), Master Of Computer Applications, Dnr College, Bhimavaram, Andhra Pradesh.

***Abstract:** The technology of detection within the captured video has implementation within the sort of fields. This emerging technology when implemented over the real-time video feeds could even be beneficial. The supreme good thing about vehicle detection within the real-time streaming video feed is to trace vehicles in busy roads or Bridges like Padma or Jamuna Bridge. An accident occurred anywhere which may rather be detected. Vehicle detection also called computer vision beholding, basically the scientific methods and ways of how machines see instead of human eyes. This chapter aims to explore the prevailing challenging issue within the planet of unsupervised surveillance and security, Helps traffic police, Maintaining records and Traffic surveillance control. The detection of vehicles is implemented with enhanced algorithms and machine learning libraries like OpenCV, TensorFlow, and others. The varied approaches are accustomed identify and count the number of vehicles. This approach even helps in getting vehicle count from night time videos.*

***Keywords:** Image classifications, Video tracking, Information analysis, Vehicle detection, Vehicle count.*

I. INTRODUCTION

Since the population and transport system increase day by day, the demand for managing them increase at the identical time. the globe is getting populated so fast. Therefore the quantity of machines from any types including vehicles increased at the identical time. That being said, new topics like traffic, accidents and plenty of more issues are needed to be managed. it's hard to manage them with the old methods, new trends and technologies are found and invented to handle each and each milestone that human kind is trying achieve.

one amongst these challenges is traffic in highways and cities. Many options like light, sign, etc. deployed so as to cope with this phenomena. It seems that these options aren't enough or not so efficient alone. New technologies like object detection and tracking are invented so as to utilize automated camera surveillance to provide data that may give meanings for a choice making process.

This phenomena are used for various quite issues. The new trend Intelligent Transport System (ITS) has many elements which object detection and tracking is one among them. this technique is employed to detect vehicles, lanes, traffic sign, or vehicle make detection. The vehicle detection and classify ability gives us the likelihood to enhance the traffic flows and roads, prevent accidents, and registering traffic crimes and violations. Objective of this chapter briefly presents the methodology for the detection and count of a specific variety of vehicle within the streaming video.

This work is required a particular mechanism within the surveillance and security. Efficient identification of a vehicle in live stream video requires massively enhancement within the surveillance techniques to counter the terrorism. In high speed stream handle, the live video stream is additionally simultaneous copied to the storage server to explore the insights into the chunk of video frames for further process [1– 3]. This object detection and identification within the captured video aims to acknowledge the moving vehicle within the chunk video frames to track the movement of that exact target in whole video stream [4].

Machine learning technologies are successfully employed in the face detection and face recognition. The video object co-segmentation is a few task of computer vision in which it may be widely used. the precise styles of the vehicle count help to reinforce the surveillance technique for the captured live stream. It are often finished the assistance of various machine learning technologies using Python like OpenCV and Tensor- Flow. Computer vision may be employed in object detection, object tracking, object classification, video surveillance, and background modelling [2].

We are able to consider numerous examples, tracking of a football and therefore the cricket equipment within the football and cricket matches,

respectively. On the opposite hand, it's to acknowledge the people within the stream video or to detect the movement of car, bike, and truck within the stream. That is an example to detect a particular target in surveillance video footage.

The vehicle detection within the streaming video also aims to resolve some real-time problem, for example, the difficulty that video feeds can not be processed in real [5, 6]. We concentrate on the vehicle detection within the captured video stream in order that the track of the overall vehicles can be maintained and also the total moving vehicles within the captured video stream can be calculated [7].

Further, we will explore the phases like classification of the vehicles supported light and heavy vehicles or two-wheelers and four-wheelers or based on brands. The classification of the moving vehicles can do using the artificial neural network, AdaBoost algorithm, and support vector machine. The vehicle detection within the captured video stream is finished with the assistance of the OpenCV through which the count of the entire number of moving vehicle is being maintained. The initial approach is to detect and count the entire number of moving vehicles within the captured video stream with the assistance of ML libraries like OpenCV.

The other approach is to coach the model to acknowledge a selected object with the assistance of ML libraries like TensorFlow. Further, both approaches are to be combined to spot and track the precise object through the trained model. The approach is employed to make a model with the assistance of two different ML approaches. The initial approach is that the implementation of the OpenCV library. With the assistance of this library, it becomes easier to maintain the record of the moving vehicles within the video stream which is further targeted to be achieved with the assistance of the training of the model. the idea of the second approach is that the implementation about the Inception-ResNetV2. it's one in every of the fastest algorithms available now.[12]

When AlexNet was presented, it absolutely was discovered that deeper neural networks are needed for further classification. However, processing it takes longer. So, it's trade-off between accuracy and speed. Google. Inc came up with new approach and divides the model in numbers of modules. This approach is also employed in the Inception-ResNetV2. It takes the input image of size 299*299*3 in stem cell of the model. it's further divided into two branches then that the processing may be divided. After that, the results are concatenated. To detect the article, primarily it is considered the Python-based machine learning libraries of TensorFlow, OpenCV Keras, Pandas, sklearn, Matplotlib, JSON, and PIL.

InceptionV4, Inception-ResNet, NumPy, Itertools, and shutil are getting used to attain the target output from this phase. Vehicle detection is very important for several fields, like military, civilian, and government applications. the govt needs real-time automatic vehicles relying on every road to manage the traffic. It either can use streams or pre-recorded video as classification and counting image objects. 2.1. Vehicle Detection and Counting in Traffic Video Based on OpenCV With the trendy social economy increasing, vehicle number in countries is growing rapidly, including China.

II. LITEARTURE SURVEY

[1] Kanrar & Siraj, 2010

The paper evaluates the performance of **multirate multicast** in distributed network environments. It investigates how throughput is affected as the number of receivers increases and compares the effectiveness of multirate versus unirate multicast sessions. Using queue mechanisms like **RED and Fair Queuing** and congestion control protocols at sender nodes, the study analyzes data transmission efficiency within clusters. The findings highlight the scalability and performance considerations necessary for real-time distributed applications such as video conferencing and multimedia delivery systems [1].

[2] Kanrar & Mandal, 2017

This work presents an approach to **video traffic analytics** in large-scale surveillance systems. It introduces a **game-theory-based topological space** to accelerate video delivery and addresses dynamic, real-time video interactions across distributed

systems. The study underlines the critical need for fast, efficient video data transfer to enable **real-time decision-making** in applications like terrorism response, city surveillance, and infrastructure protection. Video traffic is analyzed both for stored and incoming streams, with use cases in airports, metros, hospitals, and retail security [2].

[3] Kanrar & Mandal, 2016

Focusing on **interactive video-on-demand (VOD)** systems, this paper introduces a session-based **bandwidth management model** that supports real-time streaming with actions like pause, rewind, or fast-forward. The proposed architecture reduces search hop counts in distributed storage systems and optimizes bandwidth allocation, improving content delivery. The study emphasizes the hierarchical structure of multimedia storage networks and the importance of **multicast support and admission control** to handle high user loads and ensure seamless video access [3].

[4] Canny, 1986

Canny's seminal work introduces a computational method for **edge detection**, aiming to optimize edge localization and detection precision. The paper formulates mathematical criteria for edge definition and applies Gaussian smoothing to develop the **Canny edge detector**, known for its accuracy, low error rate, and resistance to noise. This detector became a foundational tool in computer vision, widely used in image processing tasks such as segmentation, object recognition, and biometric systems [4].

[5] Cheon et al., 2012

This study presents a **vision-based vehicle detection system** using **Histogram of Oriented Gradients (HOG)** and a novel HOG symmetry feature. The detection pipeline includes hypothesis generation based on shadow regions and hypothesis verification through a risk-aware classification method. Experimental results validate the system's robustness in accident-prone areas by prioritizing detection accuracy based on location relevance, making it suitable for real-time intelligent transportation systems [5].

[6] Lin et al., 2012

The authors propose a **blind-spot vehicle detection system** using integrated appearance and edge features for enhanced lane-change assistance. By combining part-based features and building multiple models from different viewpoints, the system successfully adapts to the variability in vehicle angles. The feature location information helps estimate vehicle presence with high accuracy, addressing a key challenge in automotive safety through real-time blind-spot monitoring [6].

[7] Wang & Lien, 2008

This paper introduces a **statistical approach to automatic vehicle detection** using local features across image subregions. It utilizes **PCA and ICA** for feature extraction and employs a likelihood model based on Gaussian Mixture Models. The use of position-based subregions enhances detection reliability, reduces false positives, and minimizes computational load. The system proves effective in varying lighting and occlusion conditions, making it suitable for robust traffic surveillance applications [7].

[8] Meena et al., 2020

The paper discusses the application of **machine learning** for **traffic prediction** in Intelligent Transportation Systems (ITS). By leveraging big data, GPS inputs, and environmental factors, the authors propose predictive models for congestion and routing. Techniques like **image processing, soft computing, and deep learning** are used to optimize traffic flow and minimize emissions. The work emphasizes the proactive capabilities of modern ITS in anticipating traffic patterns and improving urban mobility and safety [8].

III. PROPOSED METHOD

To create a vehicle tracking system using Python, you can follow these general steps:

Install Required Libraries: Begin by installing the necessary libraries, such as OpenCV and NumPy, which are commonly used for computer vision tasks. You can install them using pip:

- Capture Video: Use OpenCV to capture the video stream from a camera or load a pre-recorded video. This can be done using the cv2.VideoCapture() function.
- Preprocess Frames: To enhance the quality of the frames and simplify the tracking process, you can apply preprocessing techniques such as resizing, blurring, or converting the frames to grayscale.
- Detect Objects: Implement object detection algorithms such as Haar cascades, HOG + SVM, or deep learning-based methods like YOLO or SSD to identify vehicles within each frame. These algorithms provide bounding boxes around the detected vehicles.
- Track Vehicles: Apply a tracking algorithm to track the vehicles across frames. One common approach is to use the Centroid Tracking algorithm, which tracks objects based on their centroids (center points). Another option is to utilize object tracking algorithms like the Kalman Filter or optical flow methods.
- Display Results: Draw bounding boxes or labels around the tracked vehicles in each frame to visualize the tracking results. You can use OpenCV functions such as cv2.rectangle() or cv2.putText() for this purpose.
- Exit Condition: Determine the exit condition for the tracking process, such as reaching the end of the video or a specific number of frames. Once the exit condition is met, you can terminate the tracking process.

To propose a method for vehicle detection, one popular approach is to utilize deep learning-based object detection algorithms. These algorithms have demonstrated high accuracy and robustness in detecting vehicles in various scenarios. One widely used deep learning framework for object detection is the Single Shot MultiBox Detector (SSD). Here's a proposed method for vehicle detection using SSD:

- Dataset Preparation: Collect and annotate a dataset of images or videos that contain vehicles. Annotate the bounding boxes around the vehicles in each image or frame. Split the dataset into training and testing sets.
- Data Preprocessing: Preprocess the dataset by resizing the images to a fixed size, normalizing pixel values, and applying data augmentation techniques such as random cropping, flipping, and rotation. This step helps to improve the performance and generalization of the model.
- Model Training: Train an SSD model on the annotated dataset. This involves initializing a pre-trained convolutional neural network (CNN), such as VGG or ResNet, and fine-tuning it on the vehicle detection task using the annotated dataset. During training, the model learns to detect vehicles by optimizing the loss function, which combines localization and classification losses.
- Model Evaluation: Evaluate the trained SSD model on the testing dataset to measure its detection performance. Calculate metrics such as precision, recall, and mean average precision (mAP) to assess the model's accuracy and effectiveness.
- Inference and Detection: Utilize the trained model for vehicle detection in real-time or on new images/videos. Perform the following steps for each frame or image:
- Preprocess the frame by resizing and normalizing the image.
- Pass the preprocessed frame through the SSD model to obtain predictions for vehicle bounding boxes and their associated class labels.
- Apply a confidence threshold to filter out low-confidence detections.
- Optionally, perform non-maximum suppression to eliminate overlapping

bounding boxes and retain the most confident detections.

- Visualization and Post-processing: Visualize the detected vehicles by drawing bounding boxes around them in the frame. You can use libraries like OpenCV or Matplotlib for this purpose. Additionally, you can apply post-processing techniques, such as tracking algorithms, to track the detected vehicles across frames and maintain their identities.

IV. RESULTS

In video car / bus / truck / bike is there then count will be displayed. It will identify

1. Total cars
2. Total trucks
3. Total bikes
4. Total buses

Final count it will display for all vehicles separately

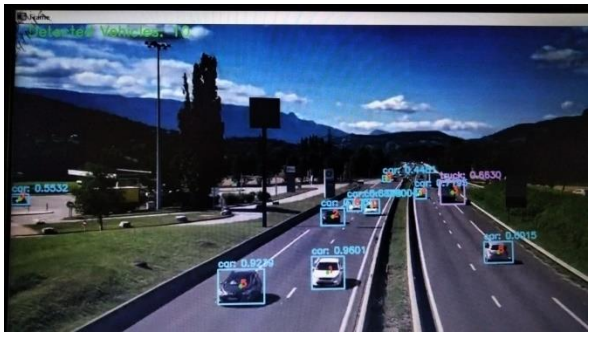

Fig. Detected Vehicle

Fig. Detected as a Truck

V. CONCLUSION

In this chapter vehicle detection, methodology is presented for object detection for the video surveillance. We have presented our model on InceptionV4 and Inception- ResNet that classify images and label videos supported the training it received. It has many applications within the field of medical diagnoses, defence, games, and virtualization and has high accuracy and high speed compare to other models presented up to now. Video detection within the captured video aims to acknowledge the moving vehicle within the captured video stream, to trace the movement of that concentrate on in whole video, and to identify the full count of the moving vehicles within the captured video. this is often exhibited with the assistance of OpenCV and TensorFlow. This includes background subtraction; it is a way of eliminating the background from image. to attain this, we extract the moving foreground from the static background. Afterwards, the thresholding is completed to achieve contrast, then noise removal is completed with the assistance of morphological transformation.

In future we can try our best to improve this system. Now it can detect any type of vehicles. In future we try to detect specific vehicles like cars, Buses, Trucks, Motor Bike and so on. And also try to detect the speed of these vehicles.

REFERENCES

[1] S. Kanrar, M. Siraj, “Performance of multirate multicast in distributed network.” Int. J. Commun. Netw. Syst. Sci. 3(6), 554–562, (2010) https://doi.org/10.4236/ijcns.2010.36074

[2] S. Kanrar, N.K. Mandal, Video traffic analytics for large scale surveillance. Multimedia Tools Appl. 76(11), 13315–13342 (2017). https://doi.org/10.1007/s11042-016-3752-0

[3] S. Kanrar, N.K. Mandal, Video traffic flow analysis in distributed system during interactive session, Adv. Multimedia. 2016, Article ID 7829570, pp 1–14 (2016), https://doi.org/10.1155/2016/7829570

[4] J. Canny, “A computational approach to edge detection,” IEEE Trans. Pattern Anal. Mach. Intell. PAMI 8(6), 679–698 (1986)

[5] Minkyu Cheon, Wonju Lee, Changyong Yoon, Vision-based vehicle detection system with consideration of the detecting location. IEEE Trans. Intell. Transp. Syst. 13(3), 1243–1252 (2012)

[6] B.F. Lin, Y.M. Lin, Fu Li-Chen, P.-Y. Hsiao, L.-A. Chuang, S.-S. Huang, M.-F. Lo, integrating appearance and edge features

for sedan vehicle detection in the blind-spot area. IEEE Trans. Intell. Transp. Syst. 13(2), 737–747 (2012)

[7] C.C.R. Wang, J.J.J. Lien, "Automatic vehicle detection using local features—a statistical approach," 9(1), 83–96 (2008)

[8] G. Meena, D. Sharma, and M. Mahrishi, "Traffic Prediction for Intelligent Transportation System using Machine Learning," Proc. 3rd Int. Conf. Emerg. Technol. Comput. Eng. Mach. Learn. Internet Things, ICETCE 2020, no. February, pp. 145–148, 2020.

[9] D. Li, B. Liang, and W. Zhang, "Real-time moving vehicle detection, tracking, and counting system implemented with OpenCV," ICIST 2014 - Proc. 2014 4th IEEE Int. Conf. Inf. Sci. Technol., pp. 631–634, 2014.

[10] A. P. Kulkarni and V. P. Baligar, "Real Time Vehicle Detection, Tracking and Counting Using Raspberry-Pi," 2nd Int. Conf. Innov. Mech. Ind. Appl. ICIMIA 2020 - Conf. Proc., no. Icimia, pp. 603–607, 2020.