Int. J. Advanced Networking and Applications
Volume: 15 Issue: 04 Pages:100-200 (2023) ISSN: 0975-0290 (Special Issue – ASCIS 2023)

An Efficient Algorithm for Vehicle Detection and Counting

Puhsp B. Patil[a] **and Samreen Mulla**[b]

[a]BLDEA's V.P. Dr. PGHCET Vijayapura , Karnataka,India
[b]BLDEA's V.P. Dr. PGHCET Vijayapura ,Karnataka,India
[a]Samulla.indi1999@gmail.com
[b]Pushpapatil2008@gmail.com

Abstract: The implementation of detection technology in captured videos extends across various domains. 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. An accident occurred anywhere which may rather be detected. This paper looks into the difficulty of current existing problems the fields of unsupervised surveillance, security, support from traffic police etc. Improved algorithm Single Shot MultiBox Detector (SSD) and machine learning frameworks Open CV, Tensor Flow are utilized for the implementation of detection for automobiles. Different methods are employed to find and tally the quantity of automobiles. Even the count of automobiles being counted from night time videos is made easier with this method.

Keywords: Single Shot MultiBox Detector (SSD), unsupervised sur veillance,Vehicle detetction,autonomous driving.

1 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 same time. That being said, new topics 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 hu- man kind is trying achieve. One among these challenges is traffic in highways and cities. It is crucial that new technologies object detection and tracking be developed in order to use automated camera surveillance to deliver facts that could set a decision- making process in context.
The objective of this project aims to offer information about the development of a successful and accurate vehicle identification system using Python, so contributing to developments in computer vision and its applications in transportation, surveillance, and autonomous driving.

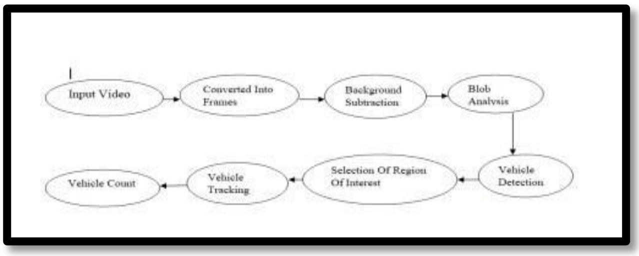

Fig 1: block diagram of proposed method

In Figure 1, the block diagram illustrates the method being suggested, encompassing various steps. Initially, it takes a video as input and converts it into frames. Next, morphological operations and vehicle detection using SSD are applied, followed by vehicle tracking. Finally, the method determines the count of objects present in each frame.

2. Related Work

S. Kanrar et al. in [1-3] this work is required a particular mechanism within the sur- veillance and security. Efficient identification of a vehicle in live stream video re- quires massively enhancement within the surveillance techniques to counter the ter- rorism. In high speed stream handle, the live video stream is additionally simultane- ous copied to the storage server to explore the insights into the chunk of video frames for further process [1– 3]. J. Canny et.al in [4] This object detection and identification within The purpose of the recorded video is to acknowledge the mobile vehicle within the chunk video frames to monitor the motion of that exact the subject throughout the entire video stream [4]. Technologies related to machine learning 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 the count of vehicles help to reinforce the surveillance technique for the captured live stream. Computer vision may be employed in object detection, object tracking, and object classification, video surveillance, and background modelling [2]. Y.M. Lin, Fu Li-Chen et.al in [5-6] 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]. J.J.J. Lien et.al in [7] we concentrate on the vehicle detection within the recorded video stream order that the track of the overall vehicles can be maintained and also the total moving vehicles within the recorded video stream has the potential to be calculated [7]. 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 Open CV 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 mod- el. The concept behind the second approach is to utilize the Inception-ResNetV2 im-

Int. J. Advanced Networking and Applications
Volume: 15 Issue: 04 Pages:100-200 (2023) ISSN: 0975-0290 (Special Issue – ASCIS 2023)

plementation, which happens to be one of the most efficient algorithms currently available [12].**Paper organization :** section 2 depicts the proposed approach. Section 3 depicts the Experimental results. Section 4 depicts the conclusion.

3. Proposed Method

Sequence of steps to adhere for the proposed method:

- **Capture Video:** Use OpenCV to capture the video stream from a camera or load a pre-recorded video. This is achievable using the cv2.VideoCapture() function.
- **Preprocess Frames:** To improve frame quality and make tracking easier,You can use preprocessing techniques such as scaling, blurring, or turning the frames to grayscale during the process.
- **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 conclusion of the video or a specific number of frames. Once the ex- it condition is met, you can terminate the tracking process.

3.1 Algorithm employed in the project: The advancement of deep learning has led to significant enhancements in object detection, making it a widely applied technology in various real-world scenarios, including self-driving cars, surveillance systems, and object tracking. In these applications, object detection algorithms commonly employ pre-trained deep neural networks, such as convolutional neural networks (CNNs), that have been trained on extensive datasets of annotated images to acquire the necessary features and representations for accurate object detection.

SSDs, or Single Shot Multibox Detectors, have emerged as a popular and effective approach to object detection. A key attribute of SSDs is their utilization of CNNs to predict both bounding boxes and class labels for objects within an image. This stands in contrast to alternative object detection algorithms, like the well-known R-CNN family of methods, which involve multiple networks and stages in the detection process. The use of a single network enables SSDs to achieve higher speed and efficiency compared to these alternative methods.

Int. J. Advanced Networking and Applications
Volume: 15 Issue: 04 Pages:100-200 (2023) ISSN: 0975-0290 (Special Issue – ASCIS 2023)

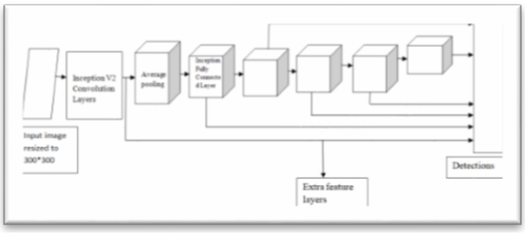

Fig 2 : The framework of the learning system based on SSD.

In Figure 2, The framework of SSD is depicted, comprising a classifier and additional feature layers. In the general structure of an SSD, you'll find a foundational network like VGG or ResNet, initially trained on an extensive image classification dataset like ImageNet. Following this base network, there are supplementary layers referred to as "extra layers," which are appended above the base network. These additional layers play a crucial role in identifying objects at various scales and typically consist of a combination of convolutional and pooling layers.

3.2 Advantages and Disadvantages of the SSDs:A primary benefit of SSDs lies in their rapid and efficient performance. Their utilization of a singular network enables real-time object detection, rendering them well-suited for applications like self-driving cars and surveillance systems. Furthermore, by employing a pre-trained base network, SSDs can capitalize on extensive labeled data used in image classification tasks. This capability enables them to attain high accuracy even when trained on comparatively modest datasets.

On the flip side, SSDs come with certain limitations. Firstly, their accuracy falls short compared to alternative methods like the R-CNN family of techniques. This discrepancy arises from the use of a single network, preventing SSDs from leveraging the supplementary context and information offered by multiple networks.Another constraint of SSDs pertains to their sensitivity to the scale of objects within an image. Given that the extra layers are tailored to detect objects at diverse scales, SSDs may encounter challenges when dealing with objects substantially smaller or larger than those present in the training dataset.

3.3 The Single Shot Detectors (SSDs) Flow

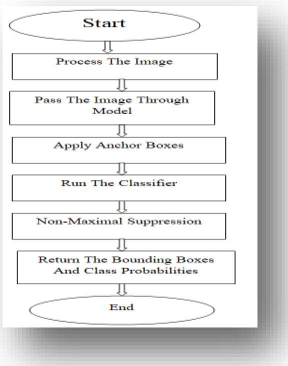

Fig 3: The Single Shot Detectors (SSDs) Flow

Int. J. Advanced Networking and Applications
Volume: 15 Issue: 04 Pages:100-200 (2023) ISSN: 0975-0290 (Special Issue – ASCIS 2023)

The flowchart delineates the primary processes essential for executing the SSD task. Initially, the input image undergoes preprocessing to ready it for model processing, involving steps like resizing, normalizing, and data augmentation. Subsequently, the preprocessed image traverses the model, generating feature maps that encapsulate a high-level abstraction of the image.
Following the model's processing, anchor boxes (or default boxes) are applied to the feature maps, creating a collection of potential bounding boxes for objects in the image. Subsequently, a classifier is employed on each bounding box to ascertain the object's class enclosed within it. To eliminate redundant boxes and refine the ultimate set of bounding boxes, Non-Maximal Suppression is employed. Ultimately, the algorithm yields a set of bounding boxes along with their corresponding class probabilities as the output.

3.4 Training Objective

The training objectives of SSD are well-suited for handling various subject categories. Let xp_i,j = 1 indicate a match for category p's i-th default box and j-th ground truth box. In the matching strategy mentioned above, we can have i xi,j = 1. The overall objective loss function consists of weighted sums of localizationloss (Lloc) and confidence loss (Lconf):

L(x,c,l,g)=1/N(Lconf (x,c)+αLloc (x,l,g)) (1)

where N represents how many default boxes were found to match. We set the loss to 0 if N equals 0. The localization loss is a Smooth L1 loss between the parameters of the predicted box (l) and the ground truth box (g). The regression involves calculating offsets for the center (cx, cy), width (w), and height (h) of the default bounding box, following a similar approach to Faster R-CNN.

$$L_{loc}(x,l,g) = \sum_{i\in Pos}^{N} \sum_{m\in cx,cy,w,h} x_{ij}^{k} smooth_{L1}\left(l_i^m - \hat{g}_j^m\right)$$
$$\hat{g}_j^{cx} = \frac{g_j^{cx} - d_i^{cx}}{d_i^w} \qquad \hat{g}_j^{cy} = \frac{g_j^{cy} - d_i^{cy}}{d_i^h}$$
$$\hat{g}_j^{w} = log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^{h} = log\left(\frac{g_j^h}{d_i^h}\right) \qquad (2)$$

The soft max loss over many classes of confidences is the measure of confidence loss(c).

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{i,j}^{p} log(\hat{c}_i^p) - \sum_{i\in Neg} log(\hat{c}_i^0) \quad (3)$$

Where $\hat{c}_i^p = \frac{expc_i^r}{\sum_p exp(c_i^p)}$

The weight term α is determined to be 1 through cross-validation.

One frequent way for proposing a solution utilize for vehicle detection is deep learning-based object detection algorithms. These algorithms have proven to be highly accurate and resilient in detecting automobiles in a variety of settings. The Single Shot MultiBox Detector (SSD) is a widely used framework for deep learning object detection. Here's a method for detecting vehicles that uses 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 to testing and training.
- **Data Preprocessing**: Preprocess the dataset by resizing the images to a fixed size, normalizing pixel values, and utilizing data augmentation methods like random cropping, flipping, and rotation. This step contributes to enhancing 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 detection of vehicles task using the anno- tated dataset. During training, the model learns to detect vehicles by optimiz- ing 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.
- **Inference and Detection:** Use the trained model to detect vehicles in real- time or on new images/videos. For each frame or image, follow these steps:
- Preprocess the frame by resizing and normalizing the image.
- Pass the preprocessed frame through the SSD model to obtain predictions for vehicle bounding bounding boxes and the related class labels.
- Apply a confidence threshold to filter out low-confidence detections.
- Optionally, Apply non-maximum suppression to remove overlapping bound- ing boxes and keep only the most confident detections.
- **Visualization and Post-processing:**Draw boxes around the observed vehi- cles inside the frame to visualize them. For this reason, libraries such as OpenCV or Matplotlib can be used.

4. Experimental results and discussion

The system is utilized to detect, recognize and track the vehicles in the sequence of video frames, after that classification of vehicles is done which are detected in accordance with their size in different classes. Edges are being counted to show how many areas of particular size which have particular vehicles like car locate the points and count the vehicles in the traffic domain and monitoring over it on highways. 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

Int. J. Advanced Networking and Applications
Volume: 15 Issue: 04 Pages:100-200 (2023) ISSN: 0975-0290 (Special Issue – ASCIS 2023)

moving vehicle is being maintained. The initial approach is to identify and count the entire number of moving vehicles within the captured video stream with the assistance of ML libraries like OpenCV.

The input video is served to process, the frames from the video, after which frames are transformed into the blobs, the pixels are then read and the object detection model is called for each blob the trained model detects the object and creates the bonding boxes for each detected object. Once the tracking of vehicles is completed, the detection line method is used to achieve vehicle counting.

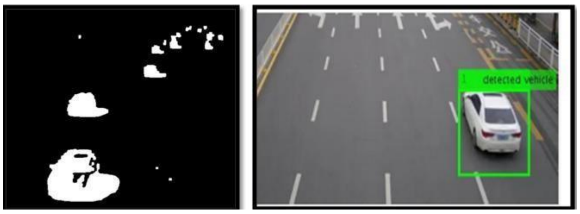

Fig 4 : Results Of Applying Morphological

Figure 4 illustrates the morphological operations, encompassing background subtraction and blob analysis.

Object detection is done using contours, followed by tracking and total counting of the moving vehicle. TensorFlow is used to complete the item identification.

Fig 5 : Vehicle Detection And Counting

In Figure 5, the objects such as cars, bikes, and persons detected during the daytime are depicted, along with a count of these objects.

This project also detect the vehicles during night time. There is no comparison between day and night , because this project equally detects and tracks the vehicles during day and night.

Fig 6: Night Vehicle Detection and Counting

Figure 6 illustrates the objects detected during the nighttime and also provides a count of these objects.

Table 1: RESULTS OF DETECTIONS

Sl .No	Vehicle Type	True Probability	Detected Probability
1	Car	1	0.997
2	Bus	1	0.3686
3	Truck	1	0.5623
4	Petted Plant	1	0.5241
5	Traffic Light	1	0.667
6	Person	1	0.8164
7	Motor Bike	1	0.7976

The table 1 predicts that the system stabilizes giving more accurate results as video progresses.

5. Conclusion

This chapter discusses the methods employed in video surveillance, specifically object detection for vehicle detection. We showcase our model using SSD, which identifies objects in images and labels videos based on the training data. Single Shot Detectors (SSDs) stand out as a widely embraced and effective approach to object detection. Employing a solitary convolutional neural network (CNN), they predict both bounding boxes and class labels for objects within an image, imparting a notable speed and efficiency advantage over alternative methods.It has several ap- plications in the fields of medical diagnosis, defense, game and virtualization, and it has great accuracy and speed when compared to other models now available. The objective of video detection in the recorded video is to identify the moving vehicle present in the video stream, track its movement throughout the entire video, and as- certain the total count of moving cars in the captured footage. This is frequently demonstrated with the help of OpenCV and TensorFlow. This involves backdrop subtraction, a technique used to eliminate the background from an image. We achieve this by distinguishing the moving foreground from the stationary background. After-

Int. J. Advanced Networking and Applications
Volume: 15 Issue: 04 Pages:100-200 (2023) ISSN: 0975-0290 (Special Issue – ASCIS 2023)

ward, thresholding is applied to enhance contrast, and morphological transformations are utilized to remove noise.

In the coming times, we are committed to enhancing this system's capabilities. While it currently possesses the ability to identify a wide range of vehicle types, our future endeavors will be focused on fine-tuning its detection capabilities for specific vehicle categories such as cars, buses, trucks, motorcycles, and more. Additionally, we aspire to incorporate speed detection functionality for these vehicles.

References

[1] S. Kanrar and M. Siraj, "Performance Of Multirate Multicast In Distributed Network," Int. J. Commun. Netw. Syst. Sci., vol. 3, no. 6, pp. 554-562, 2010.

[2] S. Kanrar, N.K. Mandal, "Video Traffic Analytics For Large Scale Surveillance", Multi- media Tools Appl. 76(11), 13315–13342 (2017).

[3] S. Kanrar, N.K. Mandal, "Video Traffic Flow Analysis In Distributed System During Inter-active Session", Adv. Multimedia. 2016, Article ID 7829570, pp 1–14 (2016).

[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, "Inte- grating 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.

[11] W. Cai, J. Li, Z. Xie, T. Zhao, and K. Lu, "Street Object Detection Based On Faster R- CNN," Chinese Control Conf. CCC, vol. 2018-July, pp. 9500–9503, 2018.

[12] F. Ekonomi, J. Akuntansi, and U. S. Ratulangi, ," Analysis Of The Effectiveness Of Park- ing Tax And Its Contribution To Regional Original Income (Pad) Kotamobagu City Jurnal Emba", J. Emba J. Ris. Ekon. Manajemen, Bisnis Dan Akunt., vol. 8, no. 4, pp. 381–388, 2020.