Optimizing Vehicle Detection using Concurrent Multiple Virtual Lattice Layers

Manipriya Sankaranarayanan and K. Siva Sai

Department of Computer Science and Engineering,
Indian Institute of Information Technology, Sricity, Andhra Pradesh India
manipriya.s@iiits.in

Abstract. Accurate vehicle detection in Intelligent Transportation Systems (ITS) is critical for various applications utilizing traffic videos. Traditional methods often analyze entire frames, resulting in high computational complexity, increased false positives, and extensive memory usage. To address these challenges, this paper introduces Virtual Lattice Layers (VL2), a novel framework that uses computer vision techniques for vehicle detection in traffic videos. VL2 employs a method where video frames are segmented into structured lattice layers. Each layer is processed independently using parallel computing techniques, which helps to lower computational costs. The system optimizes vehicle detection by applying heuristic strategies and strategically focuses its processing within specific Regions of Interest (ROIs). Moreover, to enhance detection across diverse motion directions, this work extends VL2 to Multiple Virtual Lattice Layers (MVL2), enabling concurrent processing. Experimental results across multiple traffic video datasets shows improved overall detection accuracy of 95% in a well illuminated traffic video. The concurrent processing capability shows the framework's robustness in detecting vehicles from varying angles within traffic videos.

Keywords: Lattice layers, Computer vision, Intelligent Transportation system, Vehicle Detection, Optimized Vehicle detection

1 Introduction

Vehicle detection is a fundamental element of intelligent transportation systems, pivotal in applications ranging from traffic monitoring and surveillance to autonomous driving. Traditional approaches to vehicle detection is to process the entire image of the video that leads to computationally intense, increased false positives, reduced efficiency, scalability issues and higher memory usage. To mitigate these drawbacks, modern vehicle detection systems often employ techniques such as region-based approaches (like region proposal networks or selective search) or object detection frameworks (such as Faster R-CNN [2], YOLO [6] [7], or SSD) that focus computational efforts on specific ROIs [8] or use more efficient processing strategies tailored for real-time applications. These methods aim to balance accuracy, efficiency, and scalability in vehicle detection tasks., such as sliding window techniques [10] and Region-Based Convolutional Neural Networks (R-CNNs) [13] [3], confront significant challenges concerning computational efficiency and accuracy. However, sliding window methods [10] entail scanning the entire image using a fixed-size window, a process that can incur substantial computational costs. Even when limited to Regions of Interest (ROIs), these methods often struggle with maintaining high accuracy across varying scales and orientations of vehicles within the scene. On the other hand, while R-CNNs offer improved efficiency compared to sliding windows, they necessitate intricate network architectures and substantial computational resources, potentially limiting their suitability for real-time applications.

A significant drawback identified in current research is the sensitivity of detection systems to the specific angle from which traffic video is captured. This limitation significantly affects the reliability and effectiveness of detection algorithms used in intelligent traffic management systems. Cameras positioned at narrow angles often fail to capture the entire scene or obscure critical areas due to perspective limitations and obstructions, thereby hindering the accurate identification of objects such as vehicles, pedestrians, and road signs. Moreover, varying angles can distort the appearance and size of objects in the video, making it challenging for algorithms to consistently and accurately detect them. Furthermore, objects can be partially or fully obscured by other elements in the scene, further complicating detection tasks. The performance of existing detection system is thereby heavily dependent on the placement and orientation of cameras, posing practical challenges in ensuring consistent and comprehensive monitoring across diverse real-world environments.

For vehicle detection algorithms, traditional methods like background subtraction and frame differencing [4] have been widely used because of their simplicity and effectiveness in static camera conditions. Background subtraction [4] involves creating a static model of the background and detecting vehicles. While effective, this method can be sensitive to slight changes and requires some background modeling techniques to handle dynamic environments.

Frame differencing, on the other hand, detects motion by computing the absolute difference between consecutive frames. This method is straightforward and computationally efficient, making it suitable for real-time applications. However, it may struggle with detecting slow-moving vehicles. Convolutional Neural Networks (CNNs) [3] and their variants, such as R-CNNs and Faster R-CNNs [2], have shown good performance in vehicle detection . These models uses large datasets to learn features of vehicles, achieving high detection accuracy. However, their complex architectures and high computational demands causes challenges for real-time uses in environments.

Altogether these challenges demands the need for innovative approaches in vehicle detection that can balance computational efficiency with high accuracy, especially in dynamic and real-world scenarios typical of intelligent transportation systems [19]. Hence this paper introduces a novel approach for vehicle detection using Virtual Lattice Layers (VL2) within a Selected Region of Interest (ROI). This method uses computer vision techniques to process traffic videos more efficiently. By focusing detection efforts on a specific ROI and processing Multiple VL2 concurrently, it is aimed to optimize the computational complexity while maintaining high detection accuracy and robustness of the angle of traffic videos. The approach involves splitting the ROI into smaller grids and processing each grid independently, enabling processing and thus improving the overall efficiency of the detection system.

This paper outlines the limitations of existing vehicle detection methods and presents our concurrent lattice layer approach as a more efficient alternative. Section 2 presents the detail of the proposed MVL2 . Section 3 shows the detection results and effective robustness. Finally the paper is concluded followed by references.

2 Multiple Virtual Lattice Layer (MVL2)

In recent years, advancements in Intelligent Transportation Systems (ITS) have increasingly relied on sophisticated video analysis techniques to enhance traffic management, safety, and efficiency. Among the critical tasks within ITS is accurate vehicle detection from traffic videos, which serves as a foundational element for applications ranging from real-time traffic monitoring to autonomous vehicle navigation. Traditional methods often face challenges such as high computational complexity, increased false positives, and inefficiencies in handling diverse traffic scenarios. To address these issues, this paper proposes an innovative frameworks known as Virtual Lattice Layers (VL2) and its extension, Multiple Virtual Lattice Layers (MVL2). This paper explores the principles, methodologies, and potential applications of VL2 and MVL2 in the subsequent sections for enhancing vehicle detection capabilities within ITS, aiming to contribute to the advancement of smarter and more effective transportation systems.

Virtual Lattice Layers (VL2) is a novel framework designed to enhance vehicle detection accuracy in traffic videos within ITS applications. VL2 divides video frames into structured lattice layers as shown in Figure 1, where each layer functions as a separate processing unit. This segmentation allows for parallel processing of each layer, using multi-core or distributed computing resources to optimize computational efficiency. By focusing processing efforts within defined Regions of Interest (ROIs) and applying heuristic techniques, VL2 minimizes false positives while maintaining high detection accuracy. Multiple Virtual Lattice Layers (MVL2) extends the capabilities of VL2 by introducing concurrent processing of multiple lattice layers. This extension enables simultaneous analysis of different perspectives or aspects of the traffic scene, enhancing the framework's robustness in detecting vehicles from various angles and under varying environmental conditions. MVL2's ability to handle multiple viewpoints in real-time makes it particularly suitable for dynamic traffic environments where quick and accurate decision-making is essential. The overall process and the corresponding modules of optimizing vehicle detection using MVL2 is shown in Figure 2. The details of the modules are discussed in the subsequent sections.

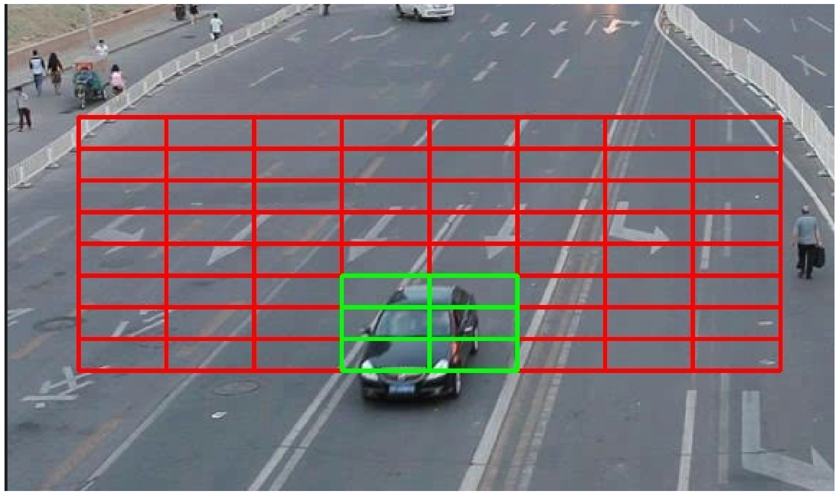

Fig. 1. Virtual Lattice Layer (VL2)

2.1 Direction of motion Detection

Determining the direction of vehicle motions plays a crucial role in the placement and configuration of Multiple Virtual Lattice Layers (MVL2) in a traffic video. By analyzing the directionality of vehicle movements using techniques like optical flow [18], MVL2 identifies distinct motion patterns and their corresponding coordinates within traffic video frames. These directional insights guide the strategic placement of Multiple VL2 across the traffic video. Layers are positioned to effectively capture and analyze vehicles moving in various directions and lanes within the monitored traffic scene. This methodical approach ensures comprehensive coverage and enhances MVL2's ability to accurately detect and track vehicles across different viewpoints and environmental conditions. By aligning the placement of VL2 with observed motion directions, MVL2 optimizes its performance in real-time traffic monitoring and management applications, contributing to more effective and reliable ITS solutions. However, detailed explanation of this topic exceeds the page limit constraints of this paper.

2.2 Detection Region Selection and Dividing into VL2

The process of dividing the Region of Interest into smaller lattice layers is an essential part. This method allows for more precise analysis of of vehicles and improves the overall accuracy of detection algorithms. From the previous step the Region of Interest (ROI) within the video frame are identified. We define the ROI by giving the dimensions of x co-ordinate , y-cordinate, grid height and width. Once the ROI is defined, it is divided into smaller, equally sized lattice as shown in Figure 1. The size of the grid (e.g., 8x8 or 16x16) is chosen based on factors such as the resolution of the video, the expected size of the vehicles,

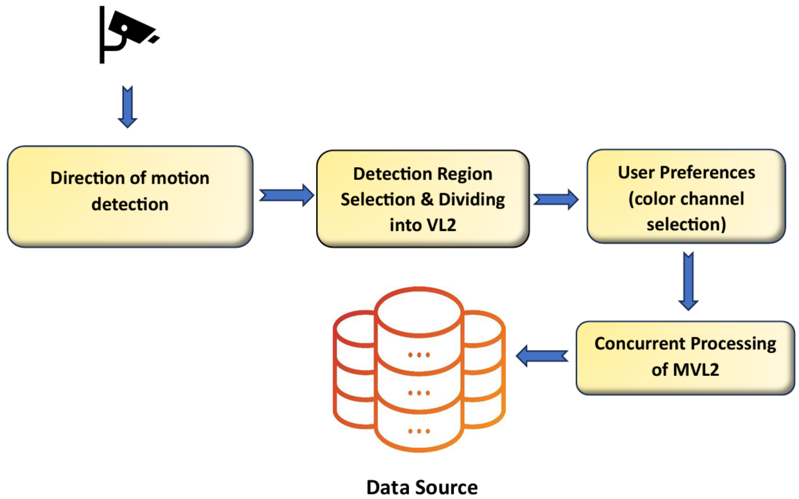

Fig. 2. Overall Work Flow for MVL2 Vehicle Detection

and computational constraints. Assume the ROI has a width W and height H. If the chosen grid size is $n \times n$, the number of grids along the width and height can be calculated as in Equation 1 and 2. Using the dimensions, the coordinates for a grid located at a position can be determined as in Equation 3 to 6. After dividing the ROI into lattice, the vehicle detection algorithm is applied to each grid independently. The same technquies are followed for multiple VL2 based on the number of direction of motion detected from the previous step.

$$\text{grid\_width} = \frac{\text{roi\_width}}{\text{num\_cols}} = \frac{W}{n} \tag{1}$$

$$\text{grid\_height} = \frac{\text{roi\_height}}{\text{num\_rows}} = \frac{H}{m} \tag{2}$$

$$\text{start\_grid\_row} = \max\left(0, \frac{y - \text{roi\_y}}{\text{grid\_height}}\right) \tag{3}$$

$$\text{end\_grid\_row} = \min\left(\text{num\_rows} - 1, \frac{y + h - \text{roi\_y}}{\text{grid\_height}}\right) \tag{4}$$

$$\text{start\_grid\_col} = \max\left(0, \frac{x - \text{roi\_x}}{\text{grid\_width}}\right) \tag{5}$$

$$\text{end\_grid\_col} = \min\left(\text{num\_cols} - 1, \frac{x + w - \text{roi\_x}}{\text{grid\_width}}\right) \tag{6}$$

2.3 User Preference

In this study, users have the flexibility to choose the lattice size based on the video type and specific requirements. Based on the preference the user is allowed to optimize the detection accuracy. Another innovative aspect of employing MVL2 for detection involves selecting the appropriate color channel for analysis. Color channel processing plays an important role in MVL2 vehicle detection system by enhancing the detection capabilities through different representations of color channels.This work uses two primary methods for processing the color information in video frames which is grayscale and HSV channels. Grayscale processing simplifies the computational complexity by reducing the image to a single channel, making it easier to detect motion of the vehicle. On the other hand, HSV processing allows us to use the color information more effectively. By splitting the video frame into its HSV components, the hue (H), saturation (S), and value (V) channels are independantly processed [19]. This separation helps in detecting vehicles more under varying lighting conditions, as different channels can highlight different aspects of the moving vehicles. Based on the user's choice of grayscale or any combination of color channel, the model converts the image accordingly. The detection in lattice is done by the operation of the channels as in Figure 3 and based on user choice as shown below.

```
choices = {
    'H': [0],
    'S': [1],
    'V': [2],
    'H+S': [0, 1],
    'H+V': [0, 2],
    'S+V': [1, 2],
    'H+S+V': [0, 1, 2],
    'gray': 'gray'
}
```

2.4 Concurrent processing of MLV2

After the Region of Interest is selected, it is divided into an n*n lattice structure, creating smaller regions known as lattice layers. This lattice layer approach allows for more precise analysis of vehicles within ROI. Each lattice layer is analyzed independently to detect the vehicles [19]. The detection process involves many image processing techniques applied to each lattice, including color channel processing, Gaussian blurring, thresholding, and contour detection. By dividing the detection region into smaller lattice, the system can process these regions concurrently, enabling parallel processing and improving the overall efficiency and speed of the detection algorithm.

Based on the diversity of motion directions present in traffic video data, the framework of Multiple Virtual Lattice Layers (MVL2) processes multiple instances of Virtual Lattice Layers (VL2) concurrently. This approach enables

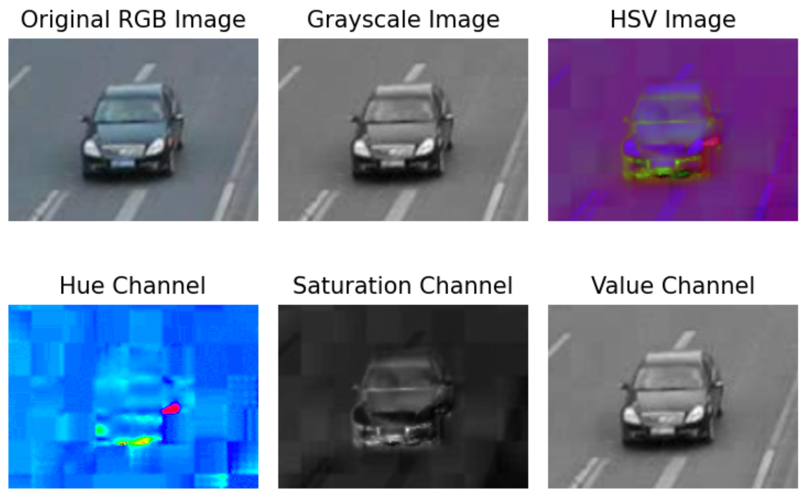

Fig. 3. Color Channels

the system to handle and analyze various movement trajectories simultaneously within the video frames. Each VL2 instance focuses on detecting vehicles moving in specific directions, utilizing parallel processing techniques to efficiently manage computational resources. By segmenting the video frames into structured lattice layers and processing them concurrently, MVL2 enhances the framework's ability to detect vehicles from multiple angles and directions of motion in real-time. This concurrent processing strategy not only improves the accuracy and efficiency of vehicle detection but also enhances the framework's overall robustness in capturing dynamic traffic scenarios effectively.

The techniques involved:

Blurring Image To reduce the noise of the image and smoothening the image blur is applied by doing this,the channel is more effective for thresholding and contour detection.

Thresholding of image Thresholding converts the channel into binary,the pixel values are either black or white (0,255)

Dilation of Image This operatiopn helps in expanding the white regions (foreground) in the binary image.This helps in closing the gaps between the contours.

Contours detection This will find the object in the lattice and gives the boundaries of the detected part in the lattice.

Processing Techniques Vehicle detecction algorithm are implemented acroos the coordinates of the lattice independantly.There are two choice to process the lattice : one is sequncially and other is concurrent processing. In sequential processing, the coordinates of lattice,and these lattice are processed to detect a vehicle or part of vehicle in that layer by executing one by one to store the result in a result matrix. The execution takes place on lattice after completing the detection on before lattice. As sequential execution of these lattices takes more time, which increases the execution time complexity, so there is a need to use ThreadPools to execute these lattice. All the grids are passed to ThreadPoolExecutor, the coordinates and channels of the grid cell is passed to processing function and results of that gridcell as 1 or 0 indicating the detection

In this work,further processing of the results from lattice layer object detection to create a binary matrix representing the presence of detected objects within each lattice are done as matrix shown below. This matrix is updated for each video frame, providing a real-time representation of vehicle locations within the ROI. The binary matrix, M, is created with dimensions num_grids_height $\times$ num_grids_width. Each element M[i][j] in the matrix corresponds to a grid and is set to 1 if an object is detected in that grid, and 0 otherwise. I n this 2D array the results are stored in middle where there is aggregate of ones gives the vehicle. This matrix can also be used to calculate the accuracy results and used to futhur advancements to the work.

$$\begin{matrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{matrix}$$

3 Evaluation of Results

The performance of MVL2 are evaluated using the traffic videos from Detrac dataset. The DETRAC dataset is a widely used benchmark in the field of computer vision and intelligent transportation systems (ITS). It comprises extensive video sequences captured from urban traffic environments, featuring diverse scenarios such as varying weather conditions, lighting, and traffic densities. The dataset includes annotations for vehicles, pedestrians, and other relevant objects, making it invaluable for training and evaluating algorithms aimed at tasks like vehicle detection, tracking, and behavior analysis. Images of typical and interstate road scenes are included in this dataset. The screenshot of few videos are shown in Figure 4.

Using MVL2 results in the matrix as discussed in the previous section. Figure 5 shows the vehicle detection in MVL2 which is represented in green and non detected lattice represented in red.

Fig. 4. Screenshot of Images from Detrac Dataset

From the analysis it is seen that for a 30 seconds video the lattices are executed sequencially the execution time is approximately 52.88 seconds and memory usage is approximately 106.51 MB. Whereas for concurrent execution of the lattices it is taking execution time of approximately 36.99 seconds memory usage of approximately 112.57 MB. The system configuration used for the evaluation is System Type: 64-bit operating system, x64-based processor Processor:11th Gen Intel(R) Core(TM) i5-1155G7 @ 2.50GHz, RAM:8.00 GB.

The accuracy of vehicle detection across the detrac dataset are frame by frame for the movement in the lattice. The execution time in (milisecond) and Memory used in (MB) is shown in Table 1.

Video Category	**Accuracy**	**Precision**	**Execution Time**	**Memory used(MB)**
Sunny	94.69	94.4	115	113.99
Night	88	76	36.09	125.22
Complex Environment	65	48	53.71	126.27

Table 1. Table with Four Rows and Five Columns

The metrics presented in Table 1 provide a comprehensive assessment of the performance of vehicle detection algorithms across diverse scenarios within the DETRAC dataset. These findings are crucial for evaluating algorithmic strengths and weaknesses, guiding future optimizations, and enhancing the reliability and efficiency of vehicle detection systems in real-world applications. The accuracy of detection are analysed for specific videos categorized as "Sunny Video," "Night Video," and "Complex Environment."

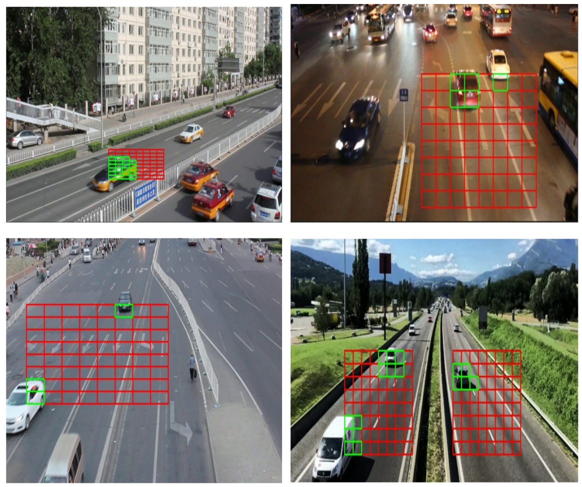

Fig. 5. Screenshot of Vehicle Detection using MVL2

Sunny videos consistently demonstrate higher accuracy and precision compared to night videos and those featuring background motion. For instance, the accuracy ranges from 92.89% to 96.5% in sunny conditions, indicating robust performance in well-lit environments where clear visibility and high contrast between vehicles and backgrounds contribute to more accurate detections. Precision levels are also notably high, with values exceeding 90% in some instances, reflecting the algorithms' ability to drastically minimize false positives and reliably identify vehicles. In contrast, night videos exhibit lower accuracy, ranging around 88%, and comparatively lower precision, which can be attributed to reduced visibility and increased noise inherent in low-light conditions. The challenges posed by night scenes, including diminished contrast and higher levels of visual artifacts, contribute to these lower metrics, highlighting the difficulties algorithms face in accurately detecting vehicles under such circumstances. Complex environment videos present additional challenges, with the lowest accuracy at 65% and precision at 48%. These videos involve complex scenes with multiple moving objects besides vehicles, such as pedestrians or moving trees, which pose challenges for algorithms focused on vehicle-specific detection. The higher execution times and memory usage in these scenarios further underscore the computational complexities involved in distinguishing vehicles amidst dynamic background movements. The overall accuray in detection of vehicle irrespective of the video category is 85%.

The variations in concurrent execution time across different video categories also reflect the computational demands associated with processing varying levels of scene complexity. Sunny videos, despite their higher accuracy, require more computational resources, as indicated by longer execution times compared to night videos. Conversely, night videos demonstrate quicker execution times, suggesting potential optimizations or simplified scene processing under reduced lighting conditions. The number of VL2 in each of the video depends on the number of different direction of motion available on the video and the execution time is dependant on the same.

Overall, these findings demands the need for adaptive algorithms that can effectively handle diverse environmental conditions encountered in real-world applications of vehicle detection. Future research efforts will be focused on refining algorithms to improve accuracy and precision in challenging scenarios like nighttime and complex, while optimizing computational efficiency to support real-time operations in intelligent transportation systems. By addressing these challenges, advancements in vehicle detection technology can significantly enhance safety, efficiency, and reliability across urban and highway environments

Overall, the values presented in Table 1 illustrate the trade-offs between accuracy, precision, execution time, and memory usage across different video types within the DETRAC dataset. These metrics are essential for evaluating and optimizing vehicle detection algorithms for efficient and reliable performance in real-world ITS applications.

You

4 Conclusion

This paper has introduced Virtual Lattice Layers (VL2) and its extension, Multiple Virtual Lattice Layers (MVL2), as innovative frameworks designed to enhance vehicle detection accuracy in Intelligent Transportation Systems (ITS) using traffic videos. Traditional methods often face challenges of high computational complexity, increased false positives, and extensive memory usage when analyzing entire frames. VL2 addresses these issues by segmenting video frames into structured lattice layers and processing each independently through parallel computing techniques, thereby reducing computational costs while optimizing detection within specific Regions of Interest (ROIs). The extension to MVL2 further improves performance by enabling concurrent processing, particularly beneficial for capturing vehicles across diverse motion directions. Experimental results across multiple traffic video datasets demonstrate a significant improvement in overall detection accuracy, achieving up to 95% in well-illuminated traffic scenarios. The framework's capability to detect vehicles effectively from varying angles underscores its robustness and potential utility in real-world applications of traffic surveillance and management within ITS.

Acknowledgment

This publication is an outcome of the R&D work undertaken in the project under TiHAN Faculty Fellowship of NMICPS Technology innovation Hub on Autonomous Navigation Foundation being implemented by Department of Science & Technology National Mission on Interdisciplinary Cyber-Physical Systes (DST NMICPS) at IIT Hyderabad.

References

1. JISU KIM. SUNGJUN HONG, and EUNTAI KIM, "Novel On-Road Vehicle Detection System Using Multi-Stage Convolutional Neural Network," June 30, 2021
2. Ahmad Arinaldi, Jaka Pradana, and Arlan Arventa Gurusinga, "Detection and classification of vehicles for traffic video analytics," Digital Services Division, PT Telekomunikasi Indonesia, Jakarta Pusat 10110, Indonesia.
3. Mallikarjun Anandhalli, Vishwanath Baligar,Pavana Baligar, Vol. 10, No. 1, March 2021, pp. 66 73.
4. Haiying Zhang1, Kun Wu1, "A Vehicle Detection Algorithm Based on Three-Frame Differencing and Background Subtraction," Oct 1, 2023
5. Hasliza A Rahim, Usman Ullah Sheikh, AR.Badlishah Ahmad, and Aini Syuhada Md Zain, "Vehicle speed detection using frame differencing for smart surveillance system," ISSPA 2010, Kuala Lumpur, Malaysia, 10-13 May, 2010
6. Nils Defauw,Marielle Malfante, Olivier Antoni, and Tiana Rakotovao, "Vehicle Detection on Occupancy Grid Maps: Comparison of Five Detectors Regarding Real-Time Performance," 2 Febraury 2023
7. R.Anandhi,G. Sekar, C. Kalaivani, and N. Jayalakshmi, "Vehicle Detection And System Tracking Using Yolo V3 Model, A Computer Vision Technique," November 2, 2022
8. Agustritus Pasrah Hati Telaumbanua1,Tri Putra Larosa, Panji Dika Pratama, Ra'uf Harris Fauza and Amir Mahmud Husein, "Vehicle Detection and Identification Using Computer Vision Technology with the Utilization of the YOLOv8 Deep Learning Method," Oct 1, 2023
9. Abderrahim EL BOUZIADY,Rachid OULAD HAJ THAMI, Mounir GHOGHO, Omar BOURJA and Sanaa EL FKIHI, "Vehicle Speed Estimation usingextracted SURF features from StereoImages," 2018 IEEE
10. Puguh Budi Prakoso,Yuslena Sar, "Vehicle detection using background subtraction and clustering algorithms," June 2019, pp.1393 1398
11. R. Rajakumar, M. Charan, R. Pandian, T. Prem Jacob, A. Pravin and P. Indumathi , "Lane Vehicle Detection and Tracking Algorithm Based on Sliding Window," 28 February 2022 pp 905–919
12. Nicholas A. Mandellos,Iphigenia Keramitsoglou,Chris T. Kiranoudis, "A background subtraction algorithm for detecting and tracking vehicles," March 2011, Pages 1619-1631
13. Genyuan Cheng[1], Yubin Guo[2], Xiaochun Cheng[3], Dongliang Wang, Jiandong Zhao2, "Real-time Detection of Vehicle Speed Based on Video Image," 2020
14. Cheng-Jian Lin ,2 Shiou-Yun Jeng,3 and Hong-Wei Lioa1, "A Real-Time Vehicle Counting, Speed Estimation, and Classification System Based on Virtual Detection Zone and Y," 2021 pp 10
15. Khushi Gupta1, Trishul Shrivas2, Siddharth Sharma3, Uma Raj, "Traffic Management system using Deep Learning ," 2023
16. Shuang Li[1], Faliang Chang[1], Chunsheng Liu1, Nanjun Li[1] , "Vehicle counting and traffic flow parameter estimation for dense traffic scenes," 2023
17. KEATTISAK SANGSUWAN AND MONGKOL EKPANYAPONG, "Video-Based Vehicle Speed Estimation Using Speed Measurement Metrics," 2024
18. Yanfeng Chen and Qingxiang Wu, "Moving vehicle detection based on optical flow estimation of edge," June 30, 2021

19. Sankaranarayanan, Manipriya, C. Mala, and Samson Mathew. "Efficient vehicle detection for traffic video-based intelligent transportation systems applications using recurrent architecture." Multimedia Tools and Applications 82.25, pp. 39015-39033, 2023