# Vehicle Detection under Tunnel using Background Subtraction Technique

**Prerna Rawat [1], Dr. Tanmoy Hazra [2], Dr. Bhupendra Singh [3]**

[1, 2, 3] Computer Engineering Department, Indian Institute of Information Technology, Pune, India

Email: [1] omprerna28@gmail.com, [2] tanmoyhazra316@gmail.com, [3] bhupi.pal08@gmail.com

**Abstract**

The safety and effectiveness of transportation systems are greatly dependent on vehicle detection in tunnel environments. Due to poor lighting, vehicle occlusions, and the unpredictable nature of traffic flow, this work is difficult. The development of reliable and precise vehicle identification techniques that are especially suited for tunnel conditions has been the subject of extensive research in recent years. This study provides an overview of vehicle detection methods developed specifically for tunnel conditions. This study evaluates various methodologies, such as conventional computer vision techniques and deep learning-based algorithms and discusses their advantages and disadvantages. In this study, the proposed methodology used bar filter and grate filter-based technique for vehicle detection and shows better outcome than the traditional methods. In tunnel settings, this study discusses different datasets and evaluation metrics commonly used for benchmarking vehicle detection algorithms. Traditional methods for detecting vehicles in tunnels often rely on manually designed features like color, texture, and motion-based cues, combined with traditional machine learning classifiers. These methods can produce satisfactory results but struggle in difficult lighting conditions and heavy traffic scenarios. However, in recent years, deep learning-based approaches, particularly convolution neural networks (CNNs), have demonstrated impressive performance in vehicle detection tasks. These approaches utilize large-scale annotated datasets and can learn complex representations directly from pixel data. As a result, they can effectively handle challenging lighting conditions, occlusions, and various vehicle orientations. Various datasets have been created specifically for assessing the effectiveness of vehicle detection algorithms in tunnel settings. These datasets encompass a range of lighting conditions, tunnel structures, and traffic scenarios, offering realistic and diverse testing scenarios. Common evaluation metrics, such as detection accuracy, false positives/negatives, and computational efficiency, are used to gauge the performance of different algorithms. The progress made in vehicle detection within tunnel environments has significant implications for enhancing traffic management, safety, and autonomous driving systems. Precise detection of vehicles in tunnels enables improved traffic flow optimization, incident detection, and real-time decision-making for autonomous vehicles. In this study, a proposed method utilized bar and grate filters based on mathematical calculations to evaluate vehicle detection in tunnel environments. The study demonstrates superior outcomes for vehicle detection in tunnel conditions compared to previous traditional methods.

Index Term - Autonomous driving, Convolution neural networks, Deep learning, Evaluation metrics, Tunnel environment, Traffic management, Vehicle detection.

## 1. Introduction

Vehicle detection has numerous uses in different fields. Some important applications of vehicle detection include managing traffic. It is crucial for traffic management systems as it enables the monitoring and analysis of traffic flow, congestion, and patterns in real-time [1]. This data can be utilized to enhance the timing of traffic signals, make real-time adjustments to lane configurations, and implement intelligent transportation systems to improve the flow of traffic. Intelligent transportation systems (ITS) heavily rely on vehicle detection to enable advanced functionalities such as adaptive cruise control, lane departure warning, collision avoidance systems, and automatic traffic surveillance [2]. These systems improve safety on the road and make traffic flow more smoothly [3]. In parking management, vehicle detection is used to monitor parking spaces and determine if they are occupied. This helps drivers find available parking spots and allows for better management of parking resources. Toll collection systems also utilize vehicle detection to identify vehicles and deduct toll fees automatically [4]. Additionally, it enables road pricing schemes where vehicles are charged based on factors like distance, time, or congestion. Finally, vehicle detection is essential for security and surveillance applications [5].

Vehicle detection technology plays a crucial role in safeguarding sensitive areas like airports, seaports, and border crossings. Vehicle detection technology plays a crucial role in monitoring and identifying vehicles in sensitive areas like airports, seaports, and border crossings. It not only helps track vehicles of interest and detect unauthorized ones but also enhances overall security measures. In the realm of autonomous driving systems, vehicle detection is a fundamental component. It enables autonomous vehicles to perceive and understand the surrounding traffic environment [6], detect and track other vehicles, and make informed decisions for safe and efficient navigation. Moreover, vehicle detection also contributes to pedestrian safety by detecting vehicles near crosswalks or pedestrian zones. This allows for the implementation of pedestrian detection systems that can provide warnings or intervene to prevent accidents involving pedestrians and vehicles [8]. These examples highlight the wide-ranging applications of vehicle detection technology. With the integration of artificial intelligence and deep learning, the advancements in vehicle detection techniques continue to expand its impact in various industries.

There are several approaches to background subtraction, including:

### A. Frame Difference

With this technique, successive frames are subtracted pixel by pixel. Foreground objects that have changed are highlighted in the ensuing difference image. The threshold technique is used to segregate the foreground.

### B. Running Average or Exponential Smoothing

In this method, the backdrop is calculated by computing a running average or increasingly weighted average over time. Then, to identify foreground items, the difference between the current frame and the background model is used.

### C. Gaussian Mixture Models (GMM)

Background subtraction models based on GMM assume that background pixel intensities follow a gaussian distribution. To represent the backdrop, the model learns a variety of gaussian, and pixels that deviate noticeably from the acquired dispersion are regarded as foreground.

### D. Adaptive Background Models

These models continuously update the backdrop estimation to account for slow changes in the image, such as camera motions or fluctuations in lighting. Researchers seek to deliver segmentation findings that are more reliable and precise. The selection of background subtraction models is determined by the specific needs and features of the application. Each approach has strengths and constraints. In addition, to improve the outcomes and extract accurate object representations, post-processing techniques like morphological procedures, contour evaluation, or object detection can be used. In applications like video surveillance, object tracking, interaction between humans and computers, and vehicle identification, where the precise extraction of foreground items is essential for subsequent analysis and decision-making, background subtraction is frequently employed.

### E. Motion Model

A motion model also represents the expected motion patterns of objects in a scene. It helps in predicting and tracking the movement of objects over time. Motion models can be simple, such as linear or constant velocity models, or more complex, considering factors like acceleration, direction changes, or object interactions. The motion model is employed in conjunction with the background subtraction model to refine the detection and tracking of moving objects. By considering the temporal aspect of object movement, the motion model helps in distinguishing between true moving objects and noise, or false detections caused by, for example, sudden changes in lighting or camera motion. The motion model provides information about object trajectories, speeds, and direction, which can be used to predict the future position of objects. It helps in associating object detections across frames, maintaining object identities, and compensating for gaps or occlusions in the detection process. By combining background subtraction and motion models, moving object detection systems can accurately identify and track objects in video sequences. These models are often used as the foundation for more advanced algorithms and techniques, such as optical flow, Kalman filtering, or deep learning-based methods, to achieve robust and reliable moving object detection in various applications like surveillance, autonomous driving, and activity recognition.

## 2. Literature Review

Kim [7] proposed the Yolo v2 method to detect vehicles under a tunnel environment. This method gives better results for vehicle detection but has limitations when a sudden light changes the condition. To improve accuracy, in this study the author used brightness smoothing and noise removal methods. Lee and Shin [10] proposed a deep learning-based framework for object detection and tracking. In this study, authors used a faster regional proposal network with a tracking algorithm. Authors assigned a unique id to the current frame and previous frame to track moving objects. This study gave a better outcome than the conventional methods. Espinosa, Velastin and Branch [12] proposed the pre-trained Faster R-CNN approach performed better with respect to the number of correct identifications and

Alex Net approach utilized for feature extraction. the gaussian mixture background subtraction method employed by the Alex Net framework and in this study the Faster R-CNN approach does not incorporate any dynamic features for identifying vehicles. Vehicles that are stopped and scenes that are obscured still present problems for background subtraction. Dong, Qn, Xu and Jiang [11] employed the background subtraction approach, the identification of edges technique, and unit split computation approach for vehicle detection. Then, based on the threshold value, identify whether vehicles are in the tunnel. The aim of this study was to reduce the ratio of vehicle identification errors in the specific tunnel environment, which is crucial for tunnel safety. Velazquez, Castaneda, Jelaca, Pizurica and Philips [13] proposed a morphological-based background removal method to produce a convex shape segmentation of the vehicles and used higher-order statistical filters for variations in brightness to identify vehicles. In terms of incorrect identification rate and overlay area identification, this strategy produced better results. Bertozzi, Broggi, Boccalini and Mazzei [14] used a vision method to identify impending tunnel entrances or exits. To prevent the tunnel blindness impact, the suggested approach enables various advanced driver assistance systems to respond based on the camera environment. The suggested method was evidently quick, it was suitable for usage as a background procedure for assisting other automatic driver assistance operations. Nguyen, Bella [15] proposed a trigger-based methodology within an area of interest to provide exact identification and safety alerts. A warning is given to the driver if it is determined that the vehicle is overweight and used to reduce the price of servicing, repairs, and examinations.

## 3. Comparative Analysis

The essential strategy for identifying vehicles is to erase the background, and numerous approaches have been put forth to do this. The following table contrasts various background removal techniques frequently used for vehicle detection:

### A. Gaussian Mixture Models (GMM)

Background subtraction techniques like GMM are frequently employed. It simulates the background's pixel brightness as a combination of Gaussian distributions. The Gaussian models are used to calculate each pixel's chances of being in the background or foreground. GMM handles progressive lighting changes well and can adjust to various background circumstances. Its problems with sluggish backdrop adaptation, and challenges representing complicated backgrounds.

### B. Codebook Model

The background is depicted by codebook models as a codebook containing words that describe its color and spatial properties. To establish whether a pixel is in the background or foreground, it is compared to the codewords. Effective codebook models can manage diverse backgrounds with various color ratios. It might have trouble with abrupt changes in illumination, need precise feature calibration, and experience trouble with occlusions.

### C. Eigen Background

The dominant eigenvectors discovered using Principal Component Analysis (PCA) on a collection of background images are used by the eigen background technique to represent the background as a subdomain. Eigen background is computationally effective, provides a simplified version of the background, and can manage slow illumination fluctuations. It struggled with abrupt changes in illumination, expects a stationary camera, and can be noise sensitive.

**D. Adaptive Methods**

The backgrounds model keeps being modified periodically by adaptive background subtraction techniques to handle gradual modifications and adjust to changing surroundings. Adaptive techniques can deal with changing lighting, shifting backgrounds, and long-term changes. These methods struggle with adapting to quick or abrupt changes, need exact parameter calibration, and can exhibit noise sensitivity and false detection.

**E. Deep Learning Based Methods**

Convolutional neural networks (CNNs) have demonstrated potential in background subtraction for vehicle detection. Using massive, annotated datasets, CNNs are trained to model the background and foreground of input images directly. Deep learning techniques can capture intricate patterns, dealing with difficult lighting, and adjusting to various backgrounds. Deep learning techniques need a lot of labeled training data and computer power, and their ability to interpret internal decision-making may be restricted.

The background's complexity, the lighting, the available computing power, and the desired accuracy and robustness all play a role in the background subtraction method selection. To choose an effective background subtraction method, it is crucial to consider the unique requirements and difficulties of the vehicle detection task in a particular scenario.

**Table No. 1.** Comparison Between Different Background Subtraction Methods

| Sr. no. | Technique | Advantage | Challenges |
|---|---|---|---|
| 1 | Gaussian Mixture Models (GMM) | Adaptive to verifying background condition | Need to improve for complex environment and sudden light change |
| 2 | Codebook | Good for complex environment condition | Occlusion condition effect results |
| 3 | Eigen background | Computationally efficient | Noise effects result |
| 4 | Adaptive Methods | Good for varying environment, complex scenario | Need to improve false detection |
| 5 | Deep Learning-Based Methods | Good for complex and different light condition | Processing time is high |

## 4. Dataset

Only a few publicly accessible datasets created exclusively for under tunnel vehicle detection are currently available. The following datasets can be used to detect vehicles in under tunnels: KITTI Dataset: For vehicle detection in surroundings like tunnels, the KITTI dataset contains a variety of scenarios, including urban environments. It offers precise 3D bounding box annotations for vehicles and pictures with excellent resolution. The Cityscapes dataset contains a variety of lighting conditions and traffic scenarios that may be important for tunnel vehicle detection, despite being largely focused on urban street images. It offers instance-level and pixel-level semantic annotations for vehicles. The Apollos cape dataset, which was created for study on autonomous vehicles, has a variety of settings, some of which resemble tunnels. It provides Lidar points, high-resolution images, and semantic segmentation labels. The UA-DETRAC collection is made up of a significant number of videos that were taken by various traffic surveillance cameras. It offers a variety of illumination settings and traffic scenarios that can be helpful for testing and refining vehicle identification models. BDD100K Dataset: The BDD100K dataset includes scenes from both urban and motorway environments. Although it doesn't concentrate on tunnels particularly, it offers a variety of lighting and traffic circumstances that can be useful for training and testing vehicle identification algorithms.

This study used the UA-DETRAC dataset. That UA-DETRAC Annotated dataset contains around 83000 training images and 53000 testing images. In the preprocessing stage, these images are reduced by 10% to prepare a new dataset. Which contain 8300 training images and 5300 testing images. The main aim of choosing the UA-DETREC dataset is the versatility scenario.

## 5. Proposed Methodology

The implementation of motion models and background removal models is crucial for the detection of moving objects. Background removal is a crucial technique for moving object detection. To distinguish between the foreground, which is made up of moving objects, and the backgrounds, which represent the scene's stationary components, is the essential idea of background subtraction. When there are no moving objects in the image, a series of frames over a set period are typically taken, and these frames are used to create the background subtraction model. A statistical representation of the background is created using this series of frames, and it may include color histograms, pixel intensity distributions, or complex models like Gaussian Mixture Models (GMM). Any noticeable differences between the current frame and the background model during object detection indicate the presence of foreground objects. These variations tend to serve as a threshold to distinguish between foreground and background objects, forming a binary mask that draws attention to the areas of interest that contain moving objects.

### A. Preprocessing

Gradient image preprocessing, which is common among computer vision and image processing, is a method for enhancing and extracting edge information from an image. It involves calculating the gradient of an image, which shows how quickly pixel intensities change across the image. Additional processing to carry out additional processing procedures is optional. Depending on the application, this may include morphological operations (like dilation or erosion) to refine the identified edges or post-processing techniques like edge connecting or contour segmentation to acquire continuous edge information.

The original input image is converted to a grayscale version. The conversion of the original or input image to a grayscale image is caused by data compression. By turning an image grayscale, processing time can be slashed. The gradient image preprocessing technique is used to transform the image to grayscale. The main goal of grayscale image conversion is to emphasize intensity values over color. The Sobel operator is used as an edge detection operator. It establishes the gradient's horizontal and vertical strengths. The Sobel operator convolves a small kernel with the image to determine the rate of change of intensity along these directions. The strength or size of the edge in that pixel is indicated by the gradient's magnitude. The gradient magnitude image is segmented, and its edges are retrieved using the threshold value. The threshold separates strong edges from background noise. Gradient image preparation methods are used in a variety of computer vision tasks, including object identification, image segmentation, and feature extraction. By looking at the gradient or edge information, it is easier to identify and separate items or areas of interest in an image.

**B. Implementation Details**

A statistical measure known as mathematical correlation assesses the association between two variables. It shows how strongly and in which direction the variables are correlated. The correlation coefficient is a common indicator of the magnitude and direction of a link. There are numerous different kinds of correlation coefficients; the Pearson, Spearman, and Kendall correlation coefficients are only a few examples. The range of these coefficients is from -1 to 1, where -1 indicates a strong negative correlation, 0 indicates no link, and 1 indicates a strong positive correlation. There are various mathematical techniques that can be employed to calculate the correlation coefficient depending on the type of link being evaluated.

The proposed method uses an approach called inter correlation that is based on mathematical equations and function pulse. The suggested method employs a grate filter to find high-frequency areas in an image. A matrix array is used for inter correlation computing in an image. Convolution and inter-correlation are analogous phenomena. The grate filter is used to filter out the high frequency region similarly to the convolution approach.

$$C(n) = O(n).\ M(n) = \Sigma_{i=-\infty}^{+\infty} O(i)\ .M(n-i) \qquad (1)$$

Equation (1) shows convolved image output. in (1) O(n) stands for the original image, M(n) for the matrix array, and C(n) for the convolution-added image. The same filtration process is considered in the suggested method.

Equation (2) shows the high frequency value on the horizontal and vertical axis. (x) on the horizontal axis denotes the X-axis, while (x-i) filters the function value. On the x-axis, the high frequency value is plotted. In (2) function $\delta(x-i)$ is used to filter out function $\lambda(x)$, means high frequency area.

$$\lambda(i) = \int_{-\infty}^{+\infty} \delta(x-i).\ \lambda(x)\ dx \qquad (2)$$

In order to display the brightness in a filtered image, choose the middle horizontal line after plotting high frequency values on the x-axis. In order to speed up data processing, a horizontal line on the x-axis that displays high brightness in a filtered image is chosen. A two-dimensional array is then converted into a one-dimensional array. An improved variant of the bar filter, a grate filter is utilized in this work to detect objects. In this study, the array matrix value is set to 1 only for the first and fourth columns, while the rest columns' values are zeros. The filter width is 8, it comprises 8 columns. Reduce calculation and processing.

time by modifying the filter. Results show that grate filters produce better outcomes for object detection with faster processing. The preprocessed gradient image is now transmitted to the grate filter. Following image processing, only the middle array line, which displays the brightness in a filtered image, gets selected from the horizontal axis. The object is then identified by comparing the horizontal axis brightness value with the threshold value. The threshold for this study is 0.5. The threshold value is compared to the width between points to determine whether an object has been detected; if the difference is greater, an object has been detected; otherwise, no object has been detected. The intensity level indicates whether an object is visible in the image based on the threshold value. When an object is present in an image an intensity plot graph indicates its presence; otherwise, previous images are used as the background image. Assessing and evaluating the brightness levels in an image can be done via brightness detection. The grate filter is made to draw attention to areas with various brightness levels, making it possible to identify and analyze bright or dark areas. After filtering, threshold approaches are frequently used to divide the image into various brightness levels. Determine and extract regions that have specific illumination characteristics by applying appropriate thresholds to the filtered image.

## 6. Results

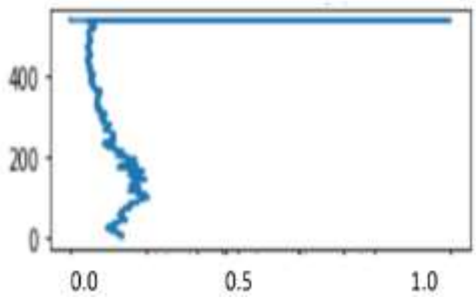
It's significant to remember that the filter selection for brightness detection depends on the intended application and the properties of the images. Furthermore, filters can be mixed or changed to meet the demands of the brightness detecting task specifically. Overall, intensity detection filters are essential for several image processing and computer vision tasks, including contrast amplification, identifying objects, and scene evaluation. Fig. 1 depicts the (a) original image, (b) grayscale image, (c) gradient image then intensity plot graph for these images. Fig. 2(a) shows the horizontal intensity plot for image 2(b) and 2(c) shows the vertical intensity plot for image 2(d). The result shows the intensity high if the object is detected in a region of interest otherwise its images as a background image.

(a) Original Image

(b) Gray Scale Image

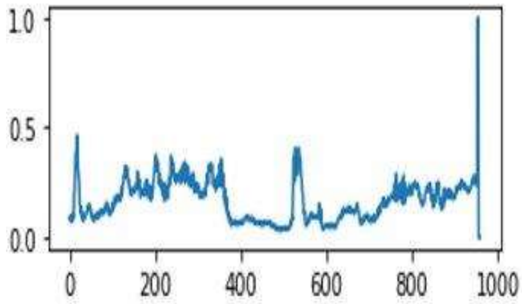
(c) Gradient Image

**Fig. 1**: Shows (a) Original image, (b) Grayscale image and (c) Gradient image.




(a) Intensity Plot for Horizontal cropped image

(b) Horizontal cropped Image

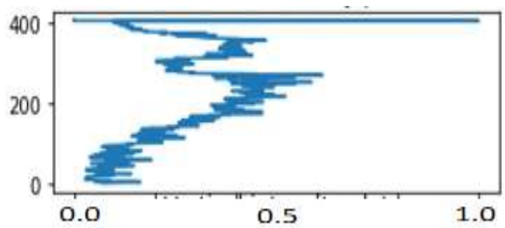



(c) Intensity Plot for vertical cropped image

(d) Vertically cropped image

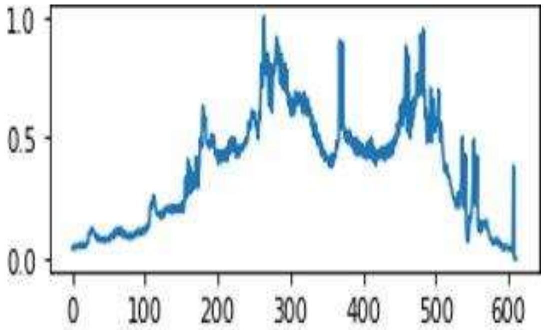
**Fig. 2:** Results for vehicle detection on horizontal intensity plot and vertical intensity plot

Fig. 3 depicts the intensity plot for another image under the tunnel, this image is taken from the internet for under tunnel light effects. Figure 3(a) shows the horizontal intensity plot for image 3(b) horizontal cropped image and figure 3(c) shows the vertical intensity plot for figure 3(d) vertical cropped image.




(a) Horizontal Intensity Plot

(b) Horizontal Cropped Image




(c) Vertical Intensity plot

(d) Vertical Cropped Image

**Fig. 3:** Result for under tunnel vehicle detection in horizontal intensity plot and vertical intensity plot.

## 7. Conclusion

As a result of poor lighting, obstructions, and changing traffic patterns, under tunnel vehicle detection is a difficult task. Deep learning and computer vision innovations, however, have shown promise in overcoming these difficulties. Background subtraction, motion-based strategies, and deep learning-based algorithms are just a few of the techniques that have been suggested for under tunnel vehicle detection. While motion models helped with tracking and predicting object movement, background subtraction models captured the stationary part background and isolated it from moving objects. By directly learning complicated structures from raw pixel data, deep learning techniques like convolution neural networks (CNNs) have exhibited amazing effectiveness in identifying vehicles in tunnel situations. Despite the lack of dedicated datasets for under tunnel vehicle detection, it is possible to modify or add existing general vehicle detection datasets to mimic tunnel-like conditions. Creating specialized datasets that capture tunnel scenarios would also be helpful for testing and refining under tunnel vehicle identification models. It has a huge impact on improving traffic management, safety, and autonomous driving systems when accurate under tunnel vehicle detection is successfully implemented. It improves overall transportation efficiency and safety in tunnel environments by enabling better event detection, traffic flow optimization, and real-time decision-making for self-driving vehicles. The efficiency and dependability of under tunnel vehicle detection systems will be significantly enhanced by ongoing research and improvements in algorithms, databases, and sensor technologies.

Reference

[1]. Sharma, P., Singh, A., Dhull, A. (2022). Applications of Deep Learning for Vehicle Detection for Smart Transportation Systems. In: Gupta, G., Wang, L., Yadav, A., Rana, P., Wang, Z. (eds) Proceedings of Academia-Industry Consortium for Data Science. Advances in Intelligent Systems and Computing, vol 1411. Springer, Singapore. https://doi.org/10.1007/978-981-16-6887-6_25

[2]. Chintalacheruvu, N., & Muthukumar, V. (2012). Video Based Vehicle Detection and its Application in Intelligent Transportation Systems. *Journal of Transportation Technologies*, *02*(04), 305–314. https://doi.org/10.4236/jtts.2012.24033

[3]. Samadi, Saeed, et al. "Performance Evaluation of Intelligent Adaptive Traffic Control Systems: A Case Study." *Journal of Transportation Technologies*, vol. 02, no. 03, 2012, pp. 248–259, https://doi.org/10.4236/jtts.2012.23027. Accessed 4 Dec. 2019.

[4]. Sathyanarayana, N. (2022). A Survey on Vehicle Detection and Classification for Electronic Toll Collection Applications. In: Majhi, S., Prado, R.P.d., Dasanapura Nanjundaiah, C. (eds) Distributed Computing and Optimization Techniques. Lecture Notes in Electrical Engineering, vol 903. Springer, Singapore. https://doi.org/10.1007/978-981-19-2281-7_10

[5]. Chen, Yi-Ling, et al. "Intelligent Urban Video Surveillance System for Automatic Vehicle Detection and Tracking in Clouds." *IEEE Xplore*, 2013, ieeexplore.ieee.org/document/6531838. Accessed 14 May 2021.

[6]. Galvao, L. G., Abbod, M., Kalganova, T., Palade, V., & Huda, M. N. (2021). Pedestrian and Vehicle Detection in Autonomous Vehicle Perception Systems—A Review. *Sensors*, *21*(21), 7267. MDPI AG. Retrieved from http://dx.doi.org/10.3390/s21217267

[7]. Kim, J. (2020). Vehicle Detection Using Deep Learning Technique in Tunnel Road Environments. *Symmetry*, *12*(12), 2012. MDPI AG. Retrieved from http://dx.doi.org/10.3390/sym12122012

[8]. Galvao, Luiz G., et al. "Pedestrian and Vehicle Detection in Autonomous Vehicle Perception Systems—a Review." *Sensors*, vol. 21, no. 21, 31 Oct. 2021, p. 7267, https://doi.org/10.3390/s21217267.

[9]. Moghimi, M., Nayeri, M., Pourahmadi, M., & Moghimi, M.K. (2018). Moving Vehicle Detection Using AdaBoost and Haar-Like Feature in Surveillance Videos. ArXiv, abs/1801.01698.

[10]. Lee, K. B., & Shin, H. S. (2019). An Application of a Deep Learning Algorithm for Automatic Detection of Unexpected Accidents Under Bad CCTV Monitoring Conditions in Tunnels. 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML). https://doi.org/10.1109/deep-ml.2019.00010

[11]. Dong, L., Qin, L., Xu, W., & Jiang, H. (2017). Detection method for vehicles in tunnels based on surveillance images. https://doi.org/10.1109/ictis.2017.8047865

[12]. Espinosa, J. E., Velastin, S. A., & Branch, J. W. (2017). *Vehicle Detection Using Alex Net and Faster R-CNN Deep Learning Models: A Comparative Study* (H. Badioze

Zaman, P. Robinson, A. F. Smeaton, T. K. Shih, S. Velastin, T. Terutoshi, A. Jaafar, & N. Mohamad Ali, Eds.). Springer Link; Springer International Publishing. https://doi.org/10.1007/978-3-319-70010-6_1

[13]. Frias-Velazquez, A., Jorge Oswaldo Niño-Castañeda, Vedran Jelaca, Aleksandra Pizurica, & Philips, W. (2011). *A mathematical morphology-based approach for vehicle detection in road tunnels*. https://doi.org/10.1117/12.894951

[14]. Bertozzi, M., Broggi, A., Gionata Boccalini, & Mazzei, L. (2011). *Fast Vision-Based Road Tunnel Detection*. https://doi.org/10.1007/978-3-642-24088-1_44

[15]. Nguyen, B. (2018). *Vision-Based Over-Height Vehicle Detection for Warning Drivers*. https://doi.org/10.17863/CAM.18902