Nigerian Journal of Physics (NJP)
ISSN online: 3027-0936
ISSN print: 1595-0611
DOI: https://doi.org/10.62292/njp.v34i3.2025.382
Volume 34(3), September 2025

A Computer Vision-Based Vehicle Speed Monitoring and Reporting System

*[1]Danladi, C., [2]Mohammed, A. S. and [2]Usman, A.

[1]Department of Electrical and Electronics Engineering, Federal University of Technology, Minna, Nigeria.
[2]Department of Telecommunications Engineering, Federal University of Technology, Minna, Nigeria.

* Corresponding Author Email: cl.danladi@gmail.com

Keywords:
Computer Vision,
Vehicle Speed Estimation,
License Plate Detection,
Traffic Management,
Over-Speeding Detection.

ABSTRACT
The detection and enforcement of over-speeding regulations remain a significant challenge due to limitations in existing vehicle speed estimation techniques. This study addresses this issue by developing a high-accuracy vehicle speed estimation system that incorporates a novel approach for license plate region extraction and over-speeding detection. The methodology employed involves the use of background subtraction methods to estimate vehicle speed, combined with a proprietary algorithm for vehicle plate region extraction. This information is then processed by a reporting system that identifies over-speeding vehicles. Results from extensive testing reveal a mean absolute error of 0.93 and a root mean square error of 1.40 in speed estimation, demonstrating high accuracy and precision of the developed system. Additionally, the mean absolute percentage error of 3.38% further substantiates the effectiveness of the system, leading to an overall accuracy of 96.62% in speed estimation. This advancement in traffic management technology has the potential to improve road safety, reduce traffic violations, and contribute to more efficient and streamlined urban planning.

INTRODUCTION

The recognition and monitoring of moving vehicles in traffic scenes have emerged as a crucial research area within the field of Intelligent Transportation Systems (ITS). This area of study aims to contribute to the development of efficient traffic surveillance, control, and information systems (Yang and Pun-Cheng, 2018). Vision-based technology has gained popularity in traffic surveillance and control systems due to its cost-effectiveness and the increasing computational power of hardware. By analysing surveillance videos, law enforcement agencies can conduct criminal investigations, monitor traffic conditions, and detect abnormal activities or events such as accidents (Liu et al., 2020). In recent years, image and video processing techniques have been extensively applied to analyse traffic data for various purposes, including estimating vehicle speed and extracting detailed vehicle information to issue traffic offense tickets (Fernandez Llorca et al., 2021).

Despite the involvement of traffic law enforcement agencies, the prevalence of over speeding and its significant contribution to fatal crashes and loss of lives remains a persistent issue (Fernandez Llorca et al., 2021). Existing hardware-based speed detection instruments have limitations, and there is a need to explore the potential of digital image processing techniques for accurate speed estimation. Additionally, the extraction of license plate numbers for identification and enforcement purposes poses a challenge. Therefore, the research problem revolves around developing a comprehensive solution that utilizes vision-based technology, advanced image processing algorithms, and automated workflows to accurately detect vehicle speed, extract license plate numbers, and enable timely enforcement actions against speed violations.

Given that speed is the leading cause of road traffic injuries and deaths, extensive research efforts are underway to develop effective methods for detecting vehicle speed (Fernandez Llorca et al., 2021). Various hardware-based speed detection instruments are available for this purpose. However, digital image processing (DIP) has gained prominence due to its wide range of applications, particularly in video surveillance systems (Wiley and Lucas, 2018). In this research project, a range of approaches and techniques, such as edge extraction, object tracking, motion vector technique, absolute difference, centroid method, and background image subtraction, will be utilized to determine and monitor vehicle speed using MATLAB.

These techniques leverage the power of image processing to analyse video data and extract relevant features that enable accurate speed estimation.

In addition to speed detection, the research project also focuses on extracting the license plate number of the vehicles under surveillance. The image processing toolbox in MATLAB offers a variety of techniques for license plate extraction. By applying these techniques, the project aims to accurately identify and extract the license plate numbers of vehicles. Furthermore, the extracted plate numbers will be integrated into a workflow that utilizes MATLAB web services to automatically send the image of the license plate to a designated email address. This automated process enhances the efficiency of traffic surveillance and provides a means for law enforcement agencies to track and identify vehicles involved in speed violations or other traffic offenses.

Several works exist in the area of vision-based vehicle speed detection. In Lahari et al., (2023), an over speeding vehicle detection system was developed using centroid tracking technique. The technique tracks a vehicle in an image frame and the Euclidean distance is evaluated between the detected vehicles in two successive frames. The results showed that the technique had an accuracy of 88%. This system, however, had no reporting feature.

In the study by Zhao et al., (2023), a novel approach to vehicle speed identification using time interpolation and feature point recognition was presented. The proposed method incorporated camera calibration within the detection area to enhance accuracy. By extracting key frames and employing image morphological processing, the system successfully identified the image coordinates of target vehicle feature points. Through coordinate conversion, the actual coordinates of these characteristic points were determined. Notably, the improved time interpolation method accounted for acceleration between frames, resulting in more reliable vehicle speed identification and a commendable reduction of 0.7% in error. This research contributes to advancing the field of vehicle speed measurement and demonstrates the effectiveness of the proposed technique. However, the absence of a reporting feature should be considered for potential future improvements.

The Vehicle Feature Detection System (VFDS) described in the study by Rajora et al., (2022) introduces a novel approach to vehicle feature detection, offering an alternative to radar-based techniques. By implementing various image processing methods in both online and offline modes, VFDS enables real-time calculation of vehicle speed, identification of vehicle features, and even determination of vehicle color. Notably, VFDS proves to be a cost-effective system compared to traditional radars, while maintaining a high level of accuracy or potentially surpassing it. The research highlights a new perspective on speed estimation and feature detection in vehicles. The study employed traffic movies captured by a mounted camera, with camera alignment and calibration determined through mathematical conditions and geometrical equations. Tensorflow and OpenCV were utilized for the entire process, facilitating effective object tracking based on object identities, states, and counts.

In addition, the paper by Khan and Srinivas, (2020), the authors present a vehicle speed detection system using Python. The study aims to predict vehicle speed based on data from recorded video sources. By employing techniques such as Gaussian blend models, DBSCAN, Kalman filter, and Optical flow, the authors propose an effective methodology for speed estimation. The integration of the Kalman filter algorithm enables accurate speed calculation, while the combination of optical flow and the Kalman filter enhances prediction performance even in scenarios with low image quality. The paper also addresses the importance of considering factors such as vehicle type, driving behavior, and vehicle position during video capture. The authors highlight the potential for future research, including improving the DBSCAN clustering technique for object recognition within vehicle clusters and utilizing adaptable pixel sets for speed detection based on vertical motion.

Furthermore, in the study by Khosravi et al., (2022), the authors propose an automated algorithm for calibrating on-road cameras to estimate vehicle speed and dimensions. By modeling the background, removing shadows, and identifying vehicle boundaries, the algorithm achieves accurate calibration. Popular vehicles are recognized using the bag of words method, and metric coefficients are calculated based on their dimensions. The algorithm tracks passing vehicles on the motion plane to determine speed and dimensions. Evaluation using a ground-truth dataset shows a mean error of 1.15 km/h, outperforming previous methods.

In the study by Chiranjeevi et al., (2023), a sensorless speed camera system is developed using image processing techniques with OpenCV in Python. The research focuses on tracking objects and estimating vehicle speeds through video analysis, eliminating the need for physical sensors. This approach offers cost-effectiveness, simplified installation, and wider monitoring coverage compared to traditional speed cameras. The study encompasses steps such as video input capture, object detection, object tracking, and speed estimation based on tracked object positions. The system also includes speed limit violation detection, visualizing the results in real-time. Challenges related to camera calibration, lighting conditions, occlusions, and precise speed estimation are acknowledged.

Additionally, in the study by Tourani et al., (2019), a motion-based vehicle speed measurement approach for Intelligent Transportation Systems (ITS) is presented. The proposed method utilizes video processing techniques to estimate vehicle speed without relying on specific visual features like license plates or windshields. By analyzing the motion parameters within a predefined Region of Interest (ROI), the system achieves real-time computing and outperforms feature-based approaches. The method comprises vehicle detection, tracking, and speed measurement modules. Moving objects are detected using the Mixture-of-Gaussian background subtraction method, followed by morphology transforms and filtration functions to identify potential vehicles. Blob tracking is employed for object tracking, and displacement between sequential frames is used to calculate vehicle speed. Experimental results demonstrate the system's acceptable accuracy compared to existing speed measurement systems.

In Mittal et al., (2021), the authors developed a system that can accurately track vehicles and estimate their speeds using image processing technology. The project utilizes video input to facilitate the flow of the system. While traditional methods like RADAR (Radio Detection and Ranging) are effective for speed estimation, they often require costly technologies, making them less feasible for implementation. In contrast, the proposed method leverages OpenCV for vehicle detection and tracking, providing a cost-effective alternative. The results show the system can effectively detect and track vehicles, enabling accurate speed estimation.

A vehicle speed estimation system using Haar classifier algorithm was presented in Mahalakshmi and Babu, (2019). The technique utilises Convolutional Neural Network and dlibs to track objects in real time and estimates the speed of moving vehicle objects. The results demonstrated the systems operability. However, no performance evaluation was carried out to determine its effectiveness.

In Kakde, (2023), the authors propose a unique and cost-effective method for accurately estimating vehicle speed without the need for expensive sensors. The approach involves analyzing the video stream of moving vehicles and includes four key components: determining the subject area, identifying car objects, utilizing a pixels per meter (PPM) algorithm for velocity estimation, and generating precise speed records and corresponding vehicle images. By combining these components, the proposed method offers an efficient solution for vehicle speed detection. The results demonstrated the system’s workability, however, no performance evaluation was carried out.

Based on the works reviewed, there are several research gaps that can be identified in the field of vehicle speed detection and estimation. Firstly, many existing systems and approaches rely on expensive sensors like radars or lidars, which can be costly to purchase and maintain. There is a need for alternative methods that utilize more affordable technologies, such as computer vision and image processing techniques, to accurately estimate vehicle speeds. Secondly, some of the reviewed works lacked reporting features, which are essential for generating comprehensive reports and facilitating traffic law enforcement. This highlights the importance of incorporating reporting capabilities in speed detection systems to enhance their practical usability.

MATERIALS AND METHODS

System Overview

In order for the objectives of this study to be achieved, the developed system comprises of three major modules. These modules are the vehicle speed estimation, license plate region extraction, and over-speeding vehicle reporting modules. The system takes in a video feed as its input. The video can be saved video or a live video. The video feed is converted into image frames and processed sequentially. The image frame is first passed to the vehicle speed estimation module which evaluates the speed of any moving vehicle detected within the image frame. After that process is completed, the system takes note of the evaluated speed and compares it against a pre-specified threshold. If the estimated speed is above the threshold, the license plate region extraction module analyses the image and crops out the region of the picture containing the license plate. After this is achieved, the over-speeding vehicle reporting module sends an email notification to the relevant authority containing the detected speed as well as the extracted license plate. Figure 1 presents the flowchart of the system operation.

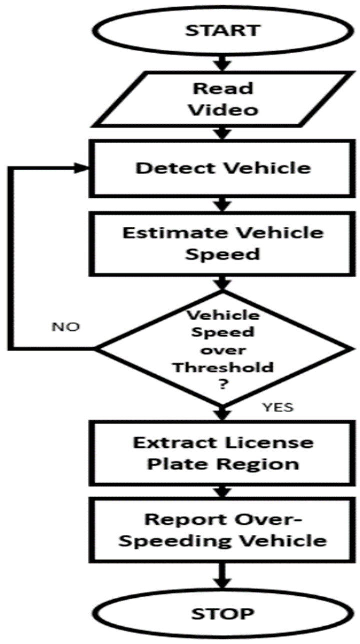

Figure 1: Flowchart of System Operation

Vehicle Speed Estimation

The process of vehicle speed estimation consists of various steps ranging from colour space conversion to evaluation of the vehicle speed. These steps are presented as follows:

i. Colour Space Conversion: An initial step in many image processing tasks, including this one, involves converting the given RGB image into a grayscale image. This process simplifies the image and allows for easier manipulation and analysis. The grayscale intensity I_gray is a weighted sum of the red (R), green (G), and blue (B) components of each pixel in the image, given by Equation 1.

$$I\_gray = 0.2990 * R + 0.5870 * G + 0.1140 * B \quad (1)$$

ii. Background Subtraction**:** In order to identify vehicles, background subtraction tech. This technique generates a foreground mask F that highlights the moving objects (vehicles in this case) by taking the absolute difference between the grayscale image I_gray and a background model I_background, then applying a detection threshold T as shown in Equation 2. The resulting foreground mask F is a binary image with the potential vehicle pixels being represented by 1 and all other pixels as 0.

$$F = |I\_gray - I\_background| > T \quad (2)$$

iii. Noise Removal: After the foreground mask is obtained, it is important to eliminate the noise in the mask. This is done using a morphological operation called "opening" that removes small white regions (noise) from the binary image, generating a filtered mask F'. The opening operation involves two major steps, namely: dilation and erosion.

The dilation operation is used to gradually enlarge the boundaries of the regions of foreground pixels. Mathematically, the dilation of a binary image A by a structuring element B can be defined as shown in Equation 3.

$$(A \oplus B)(x) = max\{A(x + y) : y \text{ in } B\} \quad (3)$$

Here max takes the maximum value of the pixel in the neighbourhood defined by B.

The erosion operation is used to erode away the boundaries of the foreground object (Always try to keep foreground in white). Mathematically, the erosion of a binary image A by a structuring element B can be defined as shown in Equation 4.

$$(A\, \Theta\, B)(x) = min\{A(x - y) : y \text{ in } B\} \quad (4)$$

Here (A Θ B) denotes the erosion of A by B, and ((A Θ B) ⊕ B) denotes the dilation of the eroded image by B.

iv. Connected Component Detection: Potential vehicles are identified as white connected regions in the binary image. Connected component labelling, also known as connected component analysis, blob extraction, region labelling, blob discovery, or region extraction, is an algorithmic application of graph theory that is used to determine the connectivity in an image. This task involves labelling the pixels in the image so that pixels with the same label share certain visual characteristics. Given an image I (a two-dimensional array) and a pixel p at coordinates (i, j) in the image, the connectivity C(p) of p can be defined in terms of its neighbourhood N(p) as shown in Equation 5.

$$C(p) = \{ q \in N(p) : I(q) = 1 \} \quad (5)$$

Here, N(p) is the set of pixels neighbouring p, and I(q) is the value of the pixel q in the image I. The connectivity C(p) is the set of neighbouring pixels q for which I(q) is 1, indicating that these pixels are connected. The exact definition of "neighbourhood" can vary. In a two-dimensional image, it typically includes either the four pixels orthogonally adjacent to p (known as 4-connectivity) or these four pixels plus the four pixels diagonally adjacent to p (known as 8-connectivity). After all pixels in the image have been processed, each connected component (set of connected pixels) in the image is assigned a unique label. These labels are used to identify potential vehicles and calculate their bounding boxes.

v. Centroid Calculation: For each potential vehicle identified, the centroid of its bounding box is calculated. Assuming the bounding box's top-left coordinate is (x, y) and its dimensions are represented as width w and height h, the centroid coordinates C are calculated as shown in Equation 6. The centroid provides a single point representation for each vehicle, which can be tracked across frames.

$$C = (x + w/2, y + h/2) \quad (6)$$

vi. Speed Estimation: The final step in this process is to calculate the vehicle's speed. To track a vehicle's movement, the vehicle's location in the previous frame and in the current frame are stored. These locations are updated based on the calculated centroid C. After that, the physical distance d_m that the vehicle moved between the two frames is evaluated, by converting the Euclidean pixel distance d_px between L_prev and L_curr to the real world distance using a scale factor distancePerPixel. Then, d_m is divided by the time interval between the frames, which is given by the distance between frames df divided by the frame rate fps, to obtain the speed S as shown in Equation 7 and Equation 8.

$$d\_m = d\_px * distancePerPixel \quad (7)$$

$$S = d\_m / (df / fps) \quad (8)$$

License Plate Region Extraction

This process starts with the detection of vehicles. The detection of the vehicle region makes it easier to localise the license plate. This is achieved using an object detection algorithm which is applied to grayscale frames to detect the vehicles. This could be seen as applying a detection function that takes a grayscale frame and a vehicle detector and returns a list of bounding boxes for detected vehicles. The extraction of the vehicle region involves the calculation of the sum of the width and height of each bounding box and selects the one with the maximum sum. This could be interpreted as finding the bounding box with the largest area as shown in Equation 9.

$$B_{\max} = \max\big(sum(B_i)\big), for\ i\ in\ [1, num\ of\ bounding\ boxes] \quad (9)$$

Where B_i is the ith bounding box and B_max is the bounding box with maximum sum of width and height. Then, the image inside this bounding box is cropped from the original frame. After that, the Sobel filter is used for edge detection. It is applied to a grayscale image I of the vehicle using a Sobel mask SM (Equations 10 and 11).

$$IS = I * SM \quad (10)$$

$$IS = IS\hat{\ }2 \quad (11)$$

Where IS is the resulting image after applying the Sobel mask. The image is then squared to highlight the edges and suppress weaker ones. The image is then normalized to scale the pixel values between 0 and 1. If min(IS) and max(IS) are the minimum and maximum pixel values in IS, the normalized image IN is computed as presented in Equation 12.

$$IN = (IS - min(IS)) / (max(IS) - min(IS)) \quad (12)$$

The Otsu method is used for thresholding the normalized image. This method finds a threshold level that minimizes the intra-class variance of the black and white pixels. If level is the threshold obtained by the Otsu method, the binary image IB is computed as shown in Equation 13.

$$IB = IN > level \quad (13)$$

A number of morphological operations are applied to the binary image IB to segment the license plate from the rest of the image. These operations involve dilation and erosion using structuring elements of different sizes and shapes. After morphological operations, connected component labelling is applied to the image, and the area of each connected component is calculated. If L is the labelled image and Areas is the vector of areas, this step is expressed as shown in Equation 14.

$$Areas_i = sum(L == i), \quad for\ i\ in\ [1, num\ of\ connected\ components] \quad (14)$$

Where Areas_i is the area of the ith connected component. The region corresponding to the largest area in the image is considered as the license plate region. The rectangle is extended by T pixels and this extended rectangle is considered as the final license plate region. Finally, the license plate image is extracted by cropping the original frame with the final license plate region.

Over-Speeding Vehicle Reporting

In the developed system, an automated email notification is dispatched when an over-speeding vehicle is detected. This process leverages Simple Mail Transfer Protocol (SMTP) for sending emails, specifically, through a Gmail account. The steps involved in this process are presented as follows:

i. Email Configuration: The email details including the sender's email, password, recipient's email, and the subject are first configured. Here, the email's subject is set as 'Overspeeding Vehicle Detected'. The message in the email consists of the speed of the overspeeding vehicle.
ii. SMTP Server Information: The SMTP server address and port number are defined. For Gmail, the SMTP server is typically `smtp.gmail.com`.
iii. Email Properties: Next, the system sets the email properties, including the email sender's address, SMTP server, SMTP username, and SMTP password. It then sets various properties of the Java Mail API to enable secure email sending via SMTP.
iv. Sending the Email: The system uses the `sendmail` function to send the email to the recipient. This function takes the recipient's email, subject of the email, and the message as arguments. If the email is sent successfully, a message 'Email notification sent for overspeeding vehicle' along with the speed of the vehicle is printed in the MATLAB console. If the email fails to send, an exception is caught and an error message 'Failed to send email notification' is printed along with the exception message.

RESULTS AND DISCUSSION

Vehicle Speed Estimation

In order to process the image, it first needs to be acquired. The system is capable of processing both live image frames and offline frames saved in a video. The system acquires RGB image frames from a camera (online) or a saved video (offline). The second step of the vehicle speed estimation process is the colour space conversion. Considering an RGB image has three channels, making it more complex to analyse, converting the image to a grayscale image makes it easier to process. Figure 2 shows the results of the image acquisition and process and Figure 3 shows the results of the colour space conversion process.

Figure 2: Image Acquisition

Figure 3: Colour Space Conversion

Figure 3 shows the results after colour space conversion. The image is gotten after the acquired image (Figure 2) is passed through the colour space conversion process highlighted by Equation 1. The image shows the output after converting the acquired image in RGB format to a grayscale format. The implication of this process is that it is easier to analyse a single channel image than a multi-channel image, due to the complexity associated with processing multiple pixel matrices simultaneously. After the colour space conversion, the background subtraction operation is carried out. Figure 4 depicts the frame difference of the background subtraction operation.

Figure 4: Frame Difference

In Figure 4, the frame difference image is provided. This was obtained after subtracting the background model from the grayscale image and taking the absolute value (Equation 2). The original acquired image was used as the background model in this instance. The frame difference is useful in generating the foreground mask as shown in Figure 5.

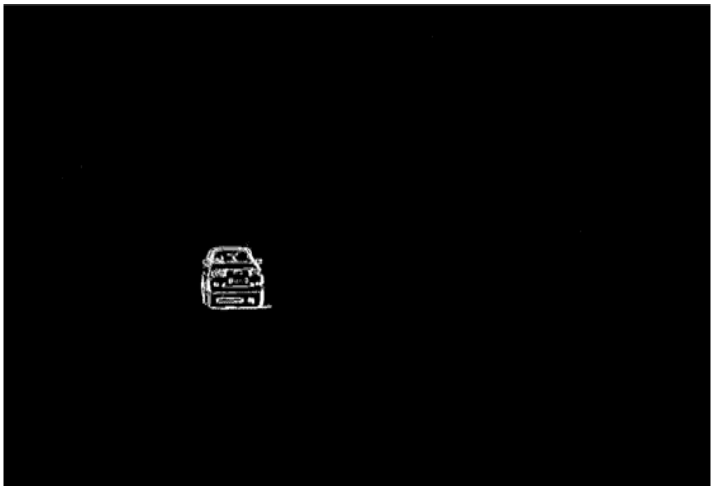

Figure 5: Foreground Mask

In Figure 5, the foreground mask is presented. This image is acquired after applying a detection threshold (Equation 2). From the image, it can be observed that applying the detection threshold on the frame difference generates a black and white image of the detected vehicle. This process is useful in identifying moving objects in an image frame.

After the foreground mask is acquired, the noise in the image needs to be eliminated before the potential vehicle is identified. This is achieved using morphological operations that involve dilation and erosion. After the noise removal, the potential vehicles are identified as white connected regions in the binary image. After all pixels in the image have been processed, each connected component (set of connected pixels) in the image is assigned a unique label. These labels are used to identify potential vehicles and calculate their bounding boxes. For each potential vehicle identified, the centroid of its bounding box is calculated. Figure 6 shows the original image with the bounding box around the vehicle.

Figure 6: Detected Vehicle Bounding Box

The concluding stage of this procedure involves determining the speed of the vehicle. By storing the vehicle's position from the last and the current frame, the motion of the vehicle is monitored. The updated locations are derived from the computed centroid and the vehicle speed is evaluated based on Equations 8 and 9. A screenshot of the estimated vehicle speed as it moves through the frame is presented in Figure 7.

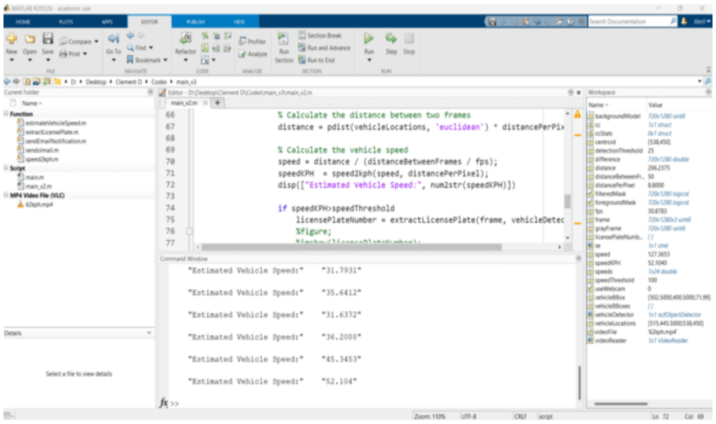

Figure 7: Estimated Vehicle Speeds

Vehicle License Plate Region Extraction

This process starts with the detection of vehicles. The detection of the vehicle region makes it easier to localise the license plate. This is achieved using an object detection algorithm which is applied to grayscale frames to detect the vehicles. In this process, a detection function uses a grayscale frame and a vehicle detector to identify vehicles and return a list of their bounding boxes. The largest bounding box, calculated by summing the width and height, is selected as it corresponds to the vehicle region. Figure 8 shows the cropped vehicle region.

Figure 8: Cropped Vehicle Image

This image inside the bounding box is cropped from the original frame and subjected to edge detection using a Sobel filter. The resulting image is squared, normalized and subjected to the Otsu method for thresholding. This creates a binary image to which morphological operations are applied, isolating the license plate from the rest of the image. Connected component labelling is then used to calculate the area of each component, the largest of which is considered the license plate region. This region is extended by a certain number of pixels and the resulting rectangle is deemed the final license plate region, from which the vehicle plate region image is extracted. Figure 9 shows the extracted image of the vehicle plate region.

Figure 9: Extracted Vehicle Plate Region

Over-Speeding Vehicle Reporting

The over-speeding vehicle reporting process begins with setting up essential email configurations, including sender's and recipient's details, and the aptly chosen subject, 'Overspeeding Vehicle Detected'. The body of the email displays the speed of the infringing vehicle and in addition, the image of the vehicle license plate region is attached to the email. Lastly, the crucial step of sending the email is executed. A successful dispatch results in a MATLAB console message confirming the sent notification with vehicle speed, while failure triggers an error message and exception detail. Figure 10 shows the received email message of the reporting system.

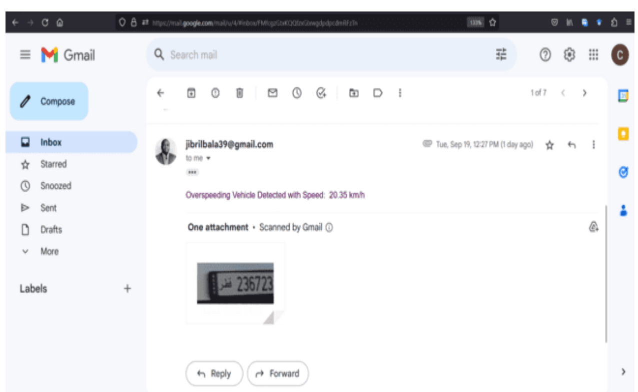

Figure 10: Email Reporting Over-Speeding Vehicle

Performance Evaluation

In order to evaluate the performance of the system, the vehicle dataset presented by Khorasani, (2020) was used. For night-time surveillance, an iPhone camera with 60 frames per second, 1080p resolution, an exposure duration of 1/120 seconds, and an active torch was utilised, while a Nikon D7000 camera with 24 frames per second, 1080p resolution, and an exposure time of 1/60 seconds was employed for daytime surveillance. Both cameras were mounted on an adjustable tripod near the road's edge, with an inclination of 0°, a horizontal angle of around 20° towards the road, and an elevation of approximately 2 feet from the ground. A total of ten test scenarios were run utilising three distinct vehicles with differing sizes and speeds ranging from 20kph to 70kph in a variety of ambient illumination situations spanning from morning to dusk. The obtained vehicle speeds were compared against the actual speeds provided in the dataset. Figure 11 shows a graphical comparison of acquired speeds and estimated speeds while Table 1 presents a comparison of the estimated speeds and the actual speeds based on the error values.

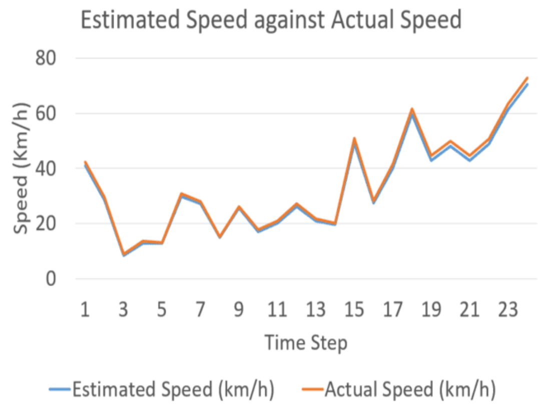

Figure 11: Estimated Speed Against Actual Speed

Table 1: Estimating Absolute Percentage Error

Estimated Speed (km/h)	Actual Speed (km/h)	Error	Square Error	Absolute Percentage Error
41.00	42.44	1.44	2.07	3.39%
28.50	29.56	1.06	1.13	3.59%
8.50	8.91	0.41	0.17	4.60%
12.92	13.54	0.62	0.38	4.57%
12.75	13.23	0.48	0.23	3.63%
29.93	30.73	0.80	0.64	2.60%
27.08	27.94	0.86	0.74	3.08%
14.87	15.33	0.46	0.21	3.00%
25.54	26.16	0.62	0.38	2.37%
17.13	17.90	0.77	0.59	4.30%
20.21	21.00	0.79	0.62	3.76%
26.28	27.22	0.94	0.88	3.46%
20.87	21.60	0.73	0.53	3.38%
19.59	20.08	0.49	0.24	2.44%
49.11	51.04	1.93	3.72	3.78%
27.41	28.15	0.74	0.55	2.63%
39.91	41.60	1.69	2.86	4.07%
59.58	61.75	2.17	4.71	3.52%
42.96	44.58	1.62	2.62	3.63%
48.16	49.86	1.70	2.89	3.40%
42.75	44.56	1.81	3.27	4.06%
48.93	50.70	1.77	3.13	3.49%
61.28	63.44	2.16	4.67	3.40%
70.41	72.72	2.31	5.34	3.18%
Sum		**22.21**	**47.59**	**81.12%**

From Table 1, the Root Mean Square Error can be evaluated as shown in Equation 15 while the Mean Absolute Error can be evaluated in Equation 16.

$$RMSE = sqrt((Sum\ of\ Square\ Error)\ /\ 24) = sqrt(47.59/24) = 1.40 \quad (15)$$

$$MAE = (Sum\ of\ Error)\ /\ 24 = 22.21\ /\ 24 = 0.93 \quad (16)$$

The RMSE, computed as the square root of the mean of squared errors, was found to be 1.40. This low RMSE value suggests that the standard deviation of prediction errors is relatively small, indicating a good fit of the estimation model to the actual speeds. On the other hand, the MAE, computed as the mean of absolute errors, was determined to be 0.93. This value denotes the average magnitude of errors in the estimation, disregarding their direction. The relatively low MAE implies that the model's predictions are, on average, close to the actual speeds. Taken together, these error metrics suggest a relatively high accuracy of the speed estimation system, though it's worth noting that the utility of these metrics may depend on the specific context and requirements of the estimation task.

The Mean Absolute Percentage Error (MAPE) is then the average of the absolute percentage errors and can be calculated as shown in Equation 17.

$$MAPE = (Sum\ of\ Absolute\ Percentage\ Errors)\ /\ n \tag{17}$$

$$MAPE = (81.12\%)\ /\ 24 = 3.38\%$$

So, the average absolute percentage error is about 3.38%. The accuracy can then be expressed as shown in Equation 19.

$$Accuracy = 100\% - MAPE = 100\% - 3.38\% = 96.62\% \tag{18}$$

The Mean Absolute Percentage Error (MAPE), derived via Equation 4.3, was calculated as the average of the absolute percentage errors across all data points. As such, the MAPE for the speed estimation system was found to be approximately 3.38%, indicating that the average error in the speed estimations, expressed as a percentage of the actual speeds, is relatively low. Consequently, the accuracy of the system, calculated using Equation 4.4, was determined to be approximately 96.62%. This high accuracy percentage signifies that, on average, the estimated speeds deviate very slightly from the actual speeds, suggesting that the speed estimation system performs its task effectively and with a high degree of precision.

CONCLUSION

The objective of this research was to develop a comprehensive speed monitoring and reporting system. This aim was successfully achieved through the design and implementation of a speed monitoring technique using background subtraction. This innovative method provided reliable vehicle detection and accurate speed estimation. Furthermore, a vehicle plate region extraction process was developed. This process, employing a series of image processing techniques, was crucial for accurately isolating and recognizing license plate information, which is essential for the unique identification of vehicles involved in speed violations. An over-speeding vehicle reporting technique was also developed, which utilized the SMTP protocol to dispatch automated email notifications. These notifications delivered detailed information about detected speed violations along with the corresponding license plate numbers. Finally, extensive testing and performance evaluations were conducted to assess the effectiveness and efficiency of the developed system. The results from these evaluations affirmed the robustness of the system in monitoring and reporting over-speeding vehicles. Future research works will focus on the use of Optical Character Recognition to extract the license plate numbers and send to relevant agencies.

REFERENCES

Chiranjeevi, K., Venkateswarlu, K., and Babu, E. C. (2023). Vehicle Monitoring and Speed Recognition System. International Journal of Creative Research Thoughts (IJCRT), 11(6), 886-890. IJRTI

Fernandez Llorca, D., Hernandez Martinez, A., & Garcia Daza, I. (2021). Vision-based vehicle speed estimation: A survey. IET Intelligent Transport Systems, 15(8), 987-1005. https://doi.org/10.1049/itr2.12079

Kakde, A., Sangode, L., Singh, S. K., and Ladekar, Y. (2023). Vehicle Speed Detection. International Journal of Advanced Research in Computer and Communication Engineering, 12(5), 1111-1116. http://dx.doi.org/10.17148/IJARCCE.2023.125189

Khan, M. M. and Srinivas, K. (2020). Vehicle Speed Detection using Python. International Research Journal of Engineering and Technology (IRJET), 7(5), 218-222. https://irjet.net/archives/V7/i5/IRJET-V7I546.pdf

Khorasani, M. (2020). Image Processing Based Vehicle Number Plate Detection and Speeding Radar. Medium. Retrieved online at https://pub.towardsai.net/image-processing-based-vehicle-number-plate-detection-and-speeding-radar-aa375952d0f6

Khosravi, H., Asgarian Dehkordi, R., & Ahmadyfard, A. (2022). Vehicle speed and dimensions estimation using on-road cameras by identifying popular vehicles. Scientia Iranica, 29(5), 2515-2525.

Lahari, S. V. S., Subha, G. S. S. K., Raju, V. B. M., and Vishnu, M. S. (2023). Vehicle Over Speed Detection using Centroid Tracking Technique. Journal of Emerging Technologies and Innovative Research, 10(4), 524-530.

Mahalakshmi, P. D., & Babu, D. M. (2019). Vehicle Speed Estimation using Haar Classifier Algorithm. International Journal of Trend in Scientific Research and Development (IJTSRD), 4(1), 243-246. 51 Vehicle Speed Estimation using Haar Classifier Algorithm

Mittal, A., Bansal, P., and Khushwah, A. (2021). Vehicle Detection and Tracking of Speed using Computer Vision. International Journal of Innovative Research in Technology, 8(2), 723-727. IJIRT152236_PAPER.pdf

Rajora, S., Chhonkar, R., Sharma, S., and Bhatia, S. (2022). Vehicle Feature Detection using Image Processing and OpenCV. International Journal of Research Publication and Reviews, 3(5), 482-485. Vehicle Feature Detection Using Image Processing and OPENCV

Tourani, A., Shahbahrami, A., Akoushideh, A., Khazaee, S., & Suen, C. Y. (2019). Motion-based vehicle speed measurement for intelligent transportation systems. International Journal of Image, Graphics and Signal Processing, 10(4), 42. https://doi.org/10.5815/ijigsp.2019.04.04

Wiley, V., & Lucas, T. (2018). Computer vision and image processing: a paper review. International Journal of Artificial Intelligence Research, 2(1), 29-36. http://dx.doi.org/10.17148/IJARCCE.2023.125189

Yang, Z., & Pun-Cheng, L. S. (2018). Vehicle detection in intelligent transportation systems and its applications under varying environments: A review. Image and Vision Computing, 69, 143-154. https://doi.org/10.1016/j.imavis.2017.09.008

Zhao, L., Tang, Y., & Wang, L. (2023). Research on vehicle speed identification method based on time interpolation method and feature point recognition. International Journal of Advanced Robotic Systems, 20(1), 17298806231153726. https://doi.org/10.1177/17298806231153726