Implementing an Efficient Speed Bump Detection System Using Adaptive Threshold Gaussian over Support Vector Machine for Improved Detection

R. Priyanka* and W. Deva Priya†

Department of Electronics and Communication Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, Tamil Nadu, 602105, India

Keywords: Adaptive Threshold, Support Vector Machine, Gaussian Thresholding, Intelligent Vehicle System, Roads, Novel Speed Bump Detection.

Abstract: This research endeavours to identify speed bumps from provided images using Adaptive Thresholding for enhanced detection. A total of 120 samples were divided equally into two groups. The first group, comprising 60 samples, underwent testing using the Support Vector Machine, while the second group was tested with the Adaptive Threshold-Gaussian. Each group underwent 10 iterations. The dataset, comprising 6000 images sourced from Kaggle.com, allocated 4800 images for training and the remaining for testing. With a G power roughly at 80%, the Gaussian Adaptive Threshold yielded an accuracy of 85.60%, surpassing the Support Vector Machine's 81.40%. A significance value of 0.002 ($p<0.05$) indicates that the results between the two groups are statistically significant. The Gaussian Adaptive Threshold, therefore, stands out for its superior accuracy.

1 INTRODUCTION

Adaptive thresholding is a type of image processing method extensively employed in digital image processing. In adaptive thresholding, image segmentation is accomplished by setting the threshold value equal to the sum of neighbouring values, where weights are a Gaussian window (KuKuXia 2018). Speed bumps are deemed a crucial component of the road traffic control system. They are designed and placed on roads to diminish vehicle speed and enhance neighbourhood safety (Kosakowska 2022). The project's most promising application is to offer safer navigation for drivers, mitigating accidents caused by overlooked speed bumps (Arunpriyan 2020). Moreover, human-caused abnormalities can diminish when drivers are warned of approaching speed bumps (Dewangan and Sahu 2021) (Palanivelu et al. 2022).

A compilation of approximately 200 articles from platforms such as Google Scholar, IEEE Xplore, and Springer has been amassed over the past five years. These articles propose distinct approaches to a specific issue. A notable strategy emphasised in the literature involves a multivariate genetic algorithm, as illustrated by Celaya-Padilla (2018). This algorithm capitalises on data from IoT devices fitted with accelerometers, GPS, and gyro sensors. Another significant paper by Dewangan and Sahu (2021) uses Raspberry Pi to detect speed bumps within an Intelligent Vehicle System, achieving commendable accuracy and precision (G. R et al. 2014).

In the quest to enhance the Advanced Driver Assistance System, the identification of speed bumps on well-marked roads is tackled by Devapriya, Babu, and Srihari (2016) using Gaussian filtering, median filtering, and connected component analysis. Intriguingly, a more straightforward method for speed bump detection and recognition using basic image processing methods, which simultaneously triggers driver alerts, is presented by Devapriya, Babu, and Srihari (2015). Notably, an engaging paper that utilises image processing techniques for speed bump recognition, combined with gyro and sensor inputs, is credited to Celaya-Padilla (2018). The research gap for novel speed bump detection lies in the

* *Research Scholar*

† *Research Guide, Corresponding Author*

inaccuracies when detecting a speed bump from an image taken from a long distance, and occasionally, there is false detection in the images. Existing methods seem to possess lesser accuracy in recognising speed bumps and require a significant amount of time to alert the drivers. The primary aim of this research is to achieve better accuracy by comparing two algorithms: Adaptive Gaussian Thresholding and Support Vector Machine for speed bump detection.

2 MATERIALS AND METHODS

The research for this project took place within the Compiler Design Laboratory at the Saveetha School of Engineering, part of the Saveetha Institute of Medical and Technical Sciences. This study comprises two distinct research groups: Adaptive Gaussian Thresholding and Support Vector Machine, with differentiation based on their ability to detect speed bumps in images. Of the 120 samples used, 60 samples each are assigned to these groups, as detailed by Arunpriyan (2020). Both groups undergo 10 iterative processes, ensuring reliable results. The dataset, sourced from Kaggle.com, contains 6000 images; 4800 for training and the rest for testing. The sample size was influenced by previous research, particularly that of Devapriya, Babu, and Srihari (2016). The study employs a 95% confidence interval and maintains a pretest power of 80% for robust statistical outcomes.

The research framework demands 4GB of RAM for prompt program processing. An Intel(R) CPU @ 1.10GHz or its superior version is recommended. The study used Windows 11 as its operating system. A storage space of 30GB is essential to house the collected dataset images, store the code, and accommodate necessary plugins. The Jupyter Notebook is utilised to operate the framework and test the program on images featuring speed bumps.

Adaptive Threshold- Gaussian

Adaptive thresholding, a facet of Image Binarisation, encompasses two primary methods: Adaptive Threshold Mean and Adaptive Threshold Gaussian. The following table 1 consists of accuracies of a sample size of 10 for both the Adaptive Threshold Gaussian (ATG) algorithm and the Support Vector Machine (SVM) algorithm.

Table 1.

S.No	ATG	SVM
1	83	84
2	84	81
3	89	83
4	87	80
5	85	79
6	87	85
7	84	78
8	89	80
9	80	83
10	88	81

The table below presents comprehensive statistics for two distinct groups, each comprising a sample size of N=10. The mean percentage accuracy achieved by the Adaptive Gaussian Thresholding method is documented at 85.60%, while the accuracy percentage attributed to the Support Vector Machine algorithm registers at 81.40%.

Table 2.

	Groups	**N**	**Mean**	**Std. Deviation**	**Std. Error Rate**
Accuracy	ATG	10	85.60	2.91357	0.92135
	SVM	10	81.40	2.27058	0.71802

This technique separates the desired object from the background based on the varying pixel intensities throughout the image. Instead of a manually specified threshold or fixed constraints, adaptive thresholding automatically determines the threshold value in relation to the image pixels, converting them into grayscale or a binary format. This method enables an automatic selection of the threshold value to differentiate the main subject from its backdrop, especially beneficial in scenarios with fluctuating lighting, colour, or contrast in the image.

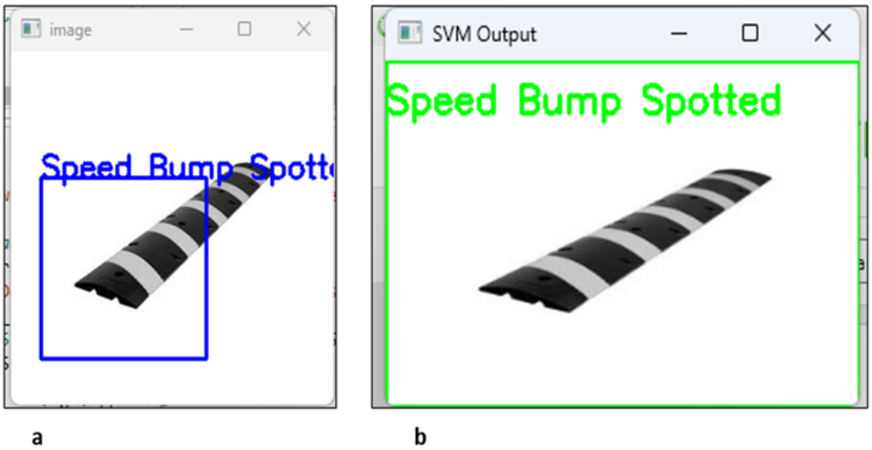

Fig. 1. and (a) The output of Adaptive Gaussian Threshold and (b) The output of Support Vector Machine.

Here's the Adaptive Threshold-Gaussian algorithm steps, rephrased in British English:

Step 1: Images are imported from the dataset for preprocessing, ensuring they're appropriately formatted for computational processing.
Step 2: During preprocessing, each image is resized to a standard dimension of 200x350, which trims unnecessary parts of the images.
Step 3: Once resized, RGB colour images are transformed into grayscale using the luminosity method. This colour conversion simplifies the computational demands.
Step 4: The Adaptive Threshold Gaussian algorithm is then applied to the grayscale images, turning them into binary images.
Step 5: A state-of-the-art speed bump detection system is then utilised, employing the Haar Cascade classifier, which has been trained on various speed bump images.
Step 6: Successfully identified speed bumps are emphasised with bounding boxes, signalling their detection.

Support Vector Machine

The Support Vector Machine (SVM) is pivotal in the suite of supervised learning algorithms, being especially prominent in both classification and regression tasks. Specifically, when diving into object recognition, which includes detecting speed bumps, SVM's expertise falls under the classification sphere. At its core, SVM focuses on pinpointing and earmarking the key vectors and boundary points in data distributions. These elements are critical in setting the hyperplane that segregates distinct classes in the data realm. As the name suggests, support vectors are these integral data points underpinning the SVM approach.

Steps for the Support Vector Machine algorithm:

Step 1: Employing deep learning techniques such as HOG, the dataset images are ingested and key features extracted.
Step 2: Resizing images enhances detection precision. Subsequently, the dataset is partitioned into training and testing subsets.
Step 3: With the data prepped, an SVM classifier is formulated using sklearn, and it's trained using the provided dataset.
Step 4: Object localisation is undertaken in two phases: initially via sliding windows and then through heatmapping. The sliding window outlines the potential detection area with a bounding box.
Step 5: Bounding boxes in static regions signal false detections, whilst those in recognised regions denote true detections.
Step 6: As mentioned, object localisation follows a dual process: firstly through sliding windows, and then supplemented by a heatmap to reaffirm bounding box demarcations.
Step 7: Depending on detection outcomes, a confusion matrix is developed. This matrix subsequently facilitates the calculation of system accuracy.

Statistical Analysis

For the statistical analysis of the results, IBM SPSS version 29 was utilised. The mean accuracies were compared using an independent sample t-test, with the confidence level set at 95% and the standard deviation error fixed at +/-2SD (Elliott and Woodward 2020). The research's independent variables encompass accuracy and the quantity of input images, whilst there isn't a dependent variable identified. The research utilised a sample size of (N=10), and the observed significance value stands at 0.002.

3 RESULTS

Both the Adaptive Threshold Gaussian and Support Vector Machine algorithms were executed multiple times using the Jupyter notebook platform. The results showed that Group1, employing the Adaptive Threshold Gaussian algorithm, achieved an accuracy of 85.60%. In contrast, Group 2, utilising the Support Vector Machine algorithm, registered an accuracy of 81.40%. This indicates that the Adaptive Threshold Gaussian algorithm outperforms the Support Vector Machine algorithm in terms of accuracy.

Table 1 enumerates the accuracy values derived from both the Adaptive Threshold Gaussian and Support Vector Machine algorithms. Table 2 delineates the mean accuracy values and standard deviation calculations for both algorithms. Meanwhile, Table 3 lists samples from the independent t-tests comparing the two algorithms. A graphical representation showcasing these values for both algorithms is provided.

Figure 1 showcases the outputs of both the Support Vector Machine and the Adaptive Gaussian Thresholding. While the Support Vector Machine detects the entire background along with the speed bump, the Adaptive Gaussian Thresholding zeroes in on a specific section of the speed bump.

Figure 2 visually contrasts the mean accuracies of the Adaptive Threshold Gaussian and Support Vector Machine algorithms. The concluding data infers that the Support Vector Machine is less effective than the Adaptive Threshold Gaussian algorithm, with the mean accuracy detection positioned at +/-2SD.

4 DISCUSSION

From the research results presented, it is unequivocally clear that the Adaptive Gaussian threshold algorithm outperforms the Support Vector Machine algorithm, thus validating the hypothesis. The accuracy percentage gleaned for the Adaptive Gaussian threshold algorithm stands at 85%, whereas the Support Vector Machine algorithm achieved an accuracy of 81%. The Adaptive threshold algorithm finds precedent in various related research pursuits, such as face detection, motion detection, and content recognition (Devapriya 2015). While the Support Vector Machine is renowned for its robustness as a classification and regression algorithm across myriad application fields (Cervantes 2020), the Adaptive Gaussian Threshold emerges as particularly suited for speed bump detection. This is attributable to its proficiency in converting RGB images into grayscale, enhancing the accuracy of detection (Celaya-Padilli 2018). Though sophisticated hardware like the NVIDIA GPU and Stereolabs ZED Stereo camera can facilitate the recognition and detection of speed bumps (Varma 2018), they invariably demand specific prerequisites and considerable investments. The approach of stereo vision, applied to local binary pattern images, presents a unique technique for detecting delineated speed bumps (Ballinas-Hernández, Olmos-Pineda, and Olvera-López 2022). However, the accuracy of such detection can sometimes leave room for improvement. Various machine learning algorithms, including Naive Bayes, Multi-Layer Perceptron, and Random Forest, have

Table 3: Independent Sample t-Test for Accuracy Comparison with 95% Confidence Interval and Equal Variance Assumption

		Levene's test for equality of variances		T-test for equality of means						
		F	Sig.	t	df	Sig. (2-tailed)	Mean difference	Std. Error difference	95% confidence Interval of the difference: Lower	95% confidence Interval of the difference: Upper
Accuracy	Equal Variance assumed	0.815	0.037	3.596	18	0.002	0.053	4.20	1.74	6.65
	Equal variance Not assumed			3.596	16.9	0.002	0.053	4.20	1.73	6.66

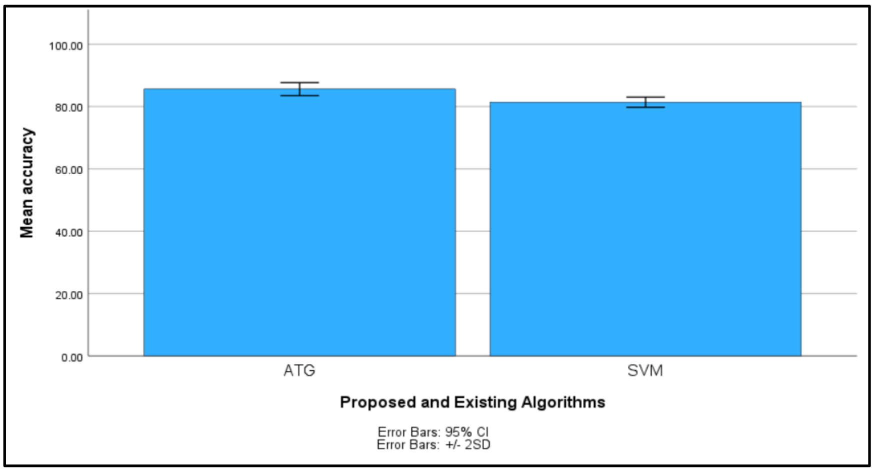

Fig. 2: Comparison of Mean Accuracies between Existing and Proposed Algorithms. This bar chart illustrates a comparison of mean accuracies, with the Y-axis representing accuracy values and the X-axis denoting the proposed and existing algorithms. The mean accuracy for the Adaptive Gaussian Thresholding is recorded at 85%, while the Support Vector Machine algorithm achieves an accuracy of 81%.

been trialled for detecting speed bumps using datasets sourced from GoPro cameras (Marques 2021). Nonetheless, the research indicates that deep learning algorithms consistently deliver the most accurate and superior performance.

This research is not without its limitations. In scenarios where images are captured under deficient lighting conditions, the speed bump detection deteriorates, leading to compromised accuracy. The research's overarching objective is multifaceted: it aims to detect speed bumps from various perspectives and discern unmarked speed bumps. However, challenges arise when an unmarked speed bump is used as input, resulting in a substantial drop in detection accuracy. One key feature that sets a speed bump apart from a crosswalk is its elevation. Yet, in certain images, this elevation isn't distinctly outlined, making detection arduous. Looking ahead, the future of novel speed bump detection holds promise in real-time video detection of both marked and unmarked speed bumps. As techniques evolve, advanced DNN methodologies could provide an avenue for even more precise speed bump detection on roads.

5 CONCLUSION

The conducted research clearly delineated the comparative efficiencies of the Adaptive Gaussian Threshold algorithm and the Support Vector Machine algorithm in the realm of speed bump detection. With the former securing an accuracy of 85.60% and the latter managing 81.40%, the Adaptive Gaussian Threshold algorithm unequivocally outshone its counterpart. This outcome was not merely a casual observation; it held statistical weight, given the achieved significance value of 0.002. This value, being less than the conventional threshold of 0.05, reinforced that the performance discrepancy between the two algorithms was not due to random chance but was indeed statistically significant. In essence, for those seeking to deploy an algorithm for detecting speed bumps with optimal accuracy, the Adaptive Gaussian Threshold algorithm emerges as the more promising choice over the Support Vector Machine algorithm, as corroborated by the empirical evidence presented in this study.

REFERENCES

Arunpriyan, J., V. V. Sajith Variyar, K. P. Soman, and S. Adarsh. 2020. “Real-Time Speed Bump Detection Using Image Segmentation for Autonomous Vehicles.” In *Advances in Intelligent Systems and Computing*, 308–15. Advances in Intelligent Systems and Computing. Cham: Springer International Publishing.

Ballinas-Hernández, Ana Luisa, Ivan Olmos-Pineda, and José Arturo Olvera-López. 2022. “Marked and Unmarked Speed Bump Detection for Autonomous Vehicles Using Stereo Vision.” *Journal of Intelligent & Fuzzy Systems* 42 (5): 4685–97.

Celaya-Padilla, Jose M., Carlos E. Galván-Tejada, F. E. López-Monteagudo, O. Alonso-González, Arturo Moreno-Báez, Antonio Martínez-Torteya, Jorge I. Galván-Tejada, Jose G. Arceo-Olague, Huizilopoztli Luna-García, and Hamurabi Gamboa-Rosales. 2018. “Speed Bump Detection Using Accelerometric Features, Gaussian Thresholding: A Genetic Algorithm

Approach." *Sensors* 18 (2). https://doi.org/10.3390/s18020443.

Cervantes, Jair, Farid Garcia-Lamont, Lisbeth Rodríguez-Mazahua, and Asdrubal Lopez. 2020. "A Comprehensive Survey on Support Vector Machine Classification: Applications, Challenges and Trends." *Neurocomputing* 408 (September): 189–215.

Devapriya, W., C. Nelson Kennedy Babu, and T. Srihari. 2015. "Advance Driver Assistance System (ADAS) - Speed Bump Detection." In *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 1–6.

———. 2016. "Real Time Speed Bump Detection Using Gaussian Filtering and Connected Component Approach." In *2016 World Conference on Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave)*, 1–5.

Dewangan, Deepak Kumar, and Satya Prakash Sahu. 2021. "Deep Learning-Based Speed Bump Detection Model for Intelligent Vehicle System Using Raspberry Pi." *IEEE Sensors Journal* 21 (3): 3570–78.

Elliott, Alan C., and Wayne A. Woodward. 2020. "Quick Guide to IBM® SPSS®: Statistical Analysis With Step-by-Step Examples." https://doi.org/10.4135/9781071909638.

G. Ramkumar and M. Manikandan, "Uncompressed digital video watermarking using stationary wavelet transform," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, Ramanathapuram, India, 2014, pp.

Kosakowska, Katarzyna. 2022. "Evaluation of the Impact of Speed Bumps on the Safety of Residents - Selected Aspects." *Transportation Research Procedia* 60 (January): 418–23.

Kishore Kumar, M. Aeri, A. Grover, J. Agarwal, P. Kumar, and T. Raghu, "Secured supply chain management system for fisheries through IoT," *Meas. Sensors*, vol. 25, no. August 2022, p. 100632, 2023, doi: 10.1016/j.measen.2022.100632.

KuKuXia. . *13_Adaptive_thresholding.py at Master · KuKuXia/OpenCV_for_Beginners*. Github. Accessed December 27, 2022. https://github.com/KuKuXia/OpenCV_for_Beginners.

Marques, Johny, Raulcezar Alves, Henrique C. Oliveira, Marco MendonÇa, and Jefferson R. Souza. 2021. "An Evaluation of Machine Learning Methods for Speed-Bump Detection on a GoPro Dataset." *Anais Da Academia Brasileira de Ciencias* 93 (1): e20190734.

Varma, V. S. K. P., S. Adarsh, K. I. Ramachandran, and Binoy B. Nair. 2018. "Real Time Detection of Speed Hump/Bump and Distance Estimation with Deep Learning Using GPU and ZED Stereo Camera." *Procedia Computer Science* 143 (January): 988–97.

Marques, Johny, Raulcezar Alves, Henrique C. Oliveira, Marco MendonÇa, and Jefferson R. Souza. 2021. "An Evaluation of Machine Learning Methods for Speed-Bump Detection on a GoPro Dataset in driver assistance system." *Anais Da Academia Brasileira de Ciencias* 93 (1): e20190734.

Palanivelu, J., Thanigaivel, S., Vickram, S., Dey, N., Mihaylova, D., & Desseva, I. (2022). Probiotics in functional foods: survival assessment and approaches for improved viability. Applied Sciences, 12(1), 455.

Rokhim, Abdul, Hanendyo Putro, and Dani F. Brilianti. 2022. "Automatic Speed Bump (ASeB) to Reduce the Rate of Traffic Accident in image processing" *RSF Conference Series: Engineering and Technology* 2 (2): 79–87.

Varma, V. S. K. P., S. Adarsh, K. I. Ramachandran, and Binoy B. Nair. 2018. "Real Time Detection of Speed Hump/Bump and Distance Estimation with Deep Learning Using GPU and ZED Stereo Camera." *Procedia Computer Science* 143 (January): 988–97.

V. P. Parandhaman, "An Automated Efficient and Robust Scheme in Payment Protocol Using the Internet of Things," 2023 Eighth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India, 2023, pp. 1-5, doi: 10.1109/ICONSTEM56934.2023.10142797.

Zheng, Zengwei, Mingxuan Zhou, Yuanyi Chen, Meimei Huo, and Dan Chen. 2019. "Enabling Real-Time Roads Anomaly Detection via Mobile Edge Computing." *International Journal of Distributed Sensor Networks* 15 (11): 1550147719891319.