JOURNAL OF APPLIED SCIENCE, TECHNOLOGY AND COMPUTING
e-ISSN: 3036-0250

Vol. 1 No. 2 (2024) 28-38
https://publisher.uthm.edu.my/ojs/index.php/jastec

JASTEC

Video Analysis Based on Gaussian Mixture Model for Traffic Video Surveillance System

Voon Zi Yi[1], Nor Afiqah Mohd Aris[1]*, Siti Suhana Jamaian[1]

[1] *Department of Mathematics and Statistics, Faculty of Applied Sciences and Technology, Universiti Tun Hussien Onn Malaysia, Muar, 84600, MALAYSIA*

*Corresponding Author: norafiqahmohdaris@gmail.com

DOI: https://doi.org/10.30880/jastec.2024.01.02.004

Article Info

Received: 25 October 2024
Accepted: 2 December 2024
Available online: 11 December 2024

Keywords

Video analysis, Gaussian Mixture Model, traffic surveillance, Python Programming, vehicle detection

Abstract

The implementation of a smart transportation system aims to enhance safety and optimize traffic flow within cities. One critical aspect of traffic flow analysis is accurately identifying moving vehicles. This paper presents an approach for traffic video analysis using the Gaussian Mixture Model (GMM) for the traffic surveillance system. The problem of detecting moving vehicles in complex traffic scenes is challenging due to varying environmental conditions, shadows, and background noise. To address this, the GMM is refined by optimizing the number of Gaussian components, adjusting the learning rate, and the examination of diverse threshold values for pixel classification. The optimal settings for these parameters were determined through extensive testing, resulting in improved detection performance. Specifically, fine-tuning these parameters led to an increase in detection accuracy and a reduction in false positives, demonstrating their significant impact on the robustness of the model. The methodology involves the application of GMM on real-world traffic video datasets sourced from ChangeDetection.NET, with performance evaluated through precision, recall, F-measure, and similarity metrics. Results demonstrate significant improvements in both accuracy and reliability of vehicle detection, contributing valuable insights to the field of traffic surveillance. These findings hold potential for enhancing urban traffic management systems, ultimately contributing to more efficient traffic flow and safety.

1. Introduction

In Malaysia, approximately 93% of households own at least one car, with most individuals relying on single-occupant vehicles for their daily commutes [1]. As urban areas grow, the volume of traffic increases significantly, making it essential to develop efficient traffic surveillance systems. Video surveillance has become a key tool in managing traffic flow, detecting incidents, and ensuring road safety within intelligent transportation systems (ITS) [2]. These systems provide vital data that help traffic authorities make better decisions, reduce congestion, and improve overall transportation efficiency [3]. Accurate and reliable vehicle detection plays a crucial role in these systems, and extensive research has focused on different methods for analyzing traffic video footage. In these systems, the analysis generally starts with basic descriptions of the traffic scene, which evolve into more detailed insights [4].

One popular approach for traffic video analysis is the Gaussian Mixture Model (GMM). This method is widely used for background subtraction and detecting moving objects, such as vehicles, against dynamic and complex backgrounds. GMM relies on the normal distribution, a statistical model often used in machine learning for background subtraction in object detection [5]. It works by modelling pixel intensity variations over time, allowing the detection of moving vehicles even in challenging environments. While GMM has been successfully applied to many traffic surveillance applications, it does face challenges, especially in situations involving changing lighting, shadows, and moving backgrounds, which can affect its performance. Despite these challenges, GMM's ability to accurately detect vehicles significantly aids higher-level image processing tasks [6].

Although much research on GMM has focused on optimizing its parameters, few studies have thoroughly examined how adjustments to specific settings—such as the number of Gaussian components, learning rate, and pixel classification threshold—impact overall vehicle detection performance. Gaussian components refer to the number of distributions used to model the background and foreground in the video. Each component helps the model represent different parts of the scene, such as static background areas and moving vehicles. Meanwhile, the learning rate (denoted as α) controls how quickly the model adapts to changes in the video. A higher learning rate allows the model to react faster to new objects or changes, while a lower rate results in slower adaptation, which can sometimes miss fast-moving objects. The threshold value determines how sensitive the model is when classifying pixels as foreground (e.g., vehicles) or background. A low threshold may result in too many false positives (misclassifying background as vehicles), while a high threshold may miss detecting smaller or partially obscured vehicles.

This gap is addressed in the current study, which proposes an improved algorithm for traffic video analysis using GMM. The study aims to refine the GMM approach through parameter optimization, evaluate its performance using real-world traffic video datasets, and analyze the results using precision, recall, F1 score, and similarity metrics. The use of GMM offers several benefits, including enhanced detection accuracy, adaptability to diverse video content, ease of implementation, and computational efficiency, making it a suitable choice for real-time vehicle detection [7]. This study also explores various GMM modifications that have been suggested to improve performance and overcome challenges in background subtraction techniques [8].

For implementation, the study uses Python, which provides a comprehensive set of libraries for machine learning and image processing. This allows for efficient and flexible development. Visual Studio Code (VS Code) was chosen as the development environment due to its user-friendly interface, strong debugging tools, and seamless integration with Python, which simplifies the implementation process. The structure of the paper is as follows: Section 2 discusses the methodology and experimental setup, Section 3 presents the results and discussion, and Section 4 concludes the paper with recommendations for future research.

2. Methods

In this study, the successful exploration of video analysis in traffic surveillance heavily relies on the strategic selection and utilization of specific materials, equipment, and resources. The primary elements include the GMM as a statistical representation of pixel intensity characteristics, the Python programming language as the computational framework, a comprehensive traffic surveillance video dataset obtained from ChangeDetection.NET [9], and the Visual Studio Code (VSC) integrated development environment for coding and execution. The video analysis is focused on the GMM for traffic video surveillance systems considered video footage used in this research is obtained from ChangeDetection.NET [9]. The ChangeDetection.NET platform was chosen for obtaining the dataset because it offers a large, diverse collection of video sequences that are ideal for the task of background subtraction and object detection in traffic surveillance. The dataset comprises nearly 90,000 frames across 31 video sequences, covering six different categories, including thermal and visible spectrum videos. This variety is crucial for training and evaluating algorithms like GMM, which require video data with varied environmental conditions, lighting, and background complexities to ensure robustness. The comprehensive dataset provided by ChangeDetection.NET ensures that the results are both generalizable and reproducible across different traffic scenarios.

The video footage selected for this study, specifically the highway.zip file from the Baseline category, was carefully chosen due to its relevance to urban traffic scenarios, making it representative of typical environments encountered in traffic surveillance. The frames from this file underwent a series of preprocessing steps. First, frame selection was performed to focus on the most relevant portions of the footage. Next, pixel intensity normalization was applied to account for lighting variations across frames, ensuring consistent input for the GMM algorithm. Finally, the frames underwent background subtraction, utilizing the pre-annotated background and foreground labels already present in the dataset. This pre-annotation allowed for the direct application of GMM, eliminating the need for additional annotation or manual intervention, thus streamlining the evaluation process for detecting moving objects, such as vehicles. By utilizing ChangeDetection.NET, this research benefits from a well-established and extensively used dataset in the field of traffic surveillance.

3. Gaussian Mixture Model for Video Analysis

GMM employs a combination of Gaussian distributions to represent the characteristics of a single pixel across different time instances. The Gaussians within the model are categorized as either background or foreground, depending on their persistence and variance [10]. This process involves estimating the parameters based on maximum likelihood, considering the information obtained from a well-trained prior model. The random variable X is assigned as a pixel and modelled as a mixture of K Gaussian distributions, and the GMM can be written as

$$P\left(X_t \middle| \theta\right) = \sum_{k=1}^{K} W_{k,t} \cdot \eta\left(X_t, \mu_k, \sum\nolimits_{k,t}\right) \tag{1}$$

where $W_{k,t}$ is the weight of k^{th} Gaussian in the mixture of time *t*. θ is defined as $\left\{\mu_k, \sum_{k,t}, W_{k,t} : k=1, \ldots\ldots, K\right\}. \eta\left(X_t, \mu_k, \sum_{k,t}\right)$ is the Gaussian probability density function with mean μ_k and covariance matrix $\sum_{k,t}$, which is given as

$$\eta\left(X_t, \mu_k, \sum\nolimits_{k,t}\right) = \frac{1}{\sqrt{(2\pi)^n \left|\sum_{k,t}\right|}} exp\left\{-\frac{1}{2}\left(X_t - \mu_{k,t}\right)^T \sum\nolimits_{k,t}^{-1}\left(X_t - \mu_{k,t}\right)\right\} \tag{2}$$

where *n* is the dimension of the pixel intensity. To accommodate computational considerations, Stauffer and Grimson [10] assumed that the red, green, and blue color components are independent of each other and have equal variances. Therefore, the covariance matrix is assumed to be as

$$\sum\nolimits_{k,t} = \sigma_{k,t}^2 \tag{3}$$

This information is then compared to a set of *K* Gaussian distributions, each with a specific matching condition denoted by $\left|x_t - \mu_{k,t-1}(x,y)\right| \leq 2.5\sigma_{k,t-1}$.where x_t is the mean of the new distribution and has a large variance initial (which is 50) with a smaller value of weight [11].

New Gaussian distributions need to be incrementally if the matching conditions are not satisfied. The mean and variance unmatched remain unchanged, whereas if the parameter of the i^{th} Gaussian distribution is matched, the following update procedure can be implemented.

$$\mu_{k,t} = (1-\theta)\,\mu_{k,t-1} + \theta x_{k,t} \tag{4}$$

$$\sigma_{k,t}^2 = (1-\theta)\,\sigma_{k,t-1}^2 + \theta\left(x_{k,t} - \mu_{k,t-1}\right)^T\left(x_{k,t} - \mu_{k,t-1}\right) \tag{5}$$

$$\sum\nolimits_{k,t} = (1-\theta)\sum\nolimits_{k,t-1} + \theta \times diag\left[\left(x_{k,t} - \mu_{k,t-1}\right)^T\left(x_{k,t} - \mu_{k,t-1}\right)\right] \tag{6}$$

$$\theta = \alpha\eta(x_{k,t}|\mu_{k,t}, \Sigma_{k,t}) \tag{7}$$

where α is the learning rate, $0 < \alpha < 1$ and it decides to update the speed of the background.

If the i^{th} Gaussian distribution matches with $x_{k,t}$, $G_{k,t} = 1$, it undergoes an update; otherwise, $G_{k,t} = 0$ remains the same. Once all the Gaussian distribution weights have finished updating, normalization is performed. Once the Gaussian model completes its update, the *K* Gaussian distributions corresponding to every pixel can be described as

$$B = arg\,\min_b\left(\sum_{k=1}^{b} W_k > T\right);\ 0.5 < T < 1 \tag{8}$$

These distributions obey in front of the *B* Gaussian distributions arranged according to the priority specified by $\varphi_{k,t} = W_{k,t} / \left|\sum_{k,t}\right|$. The matching inspection is performed iteratively for each $x_{k,t}$, comparing it with the *B* Gaussian distributions. If $x_{k,t}$ matches any of the points within the previous *B* Gaussian distributions, it is classified as a background point. Otherwise, it is classified as a foreground point. This process continues until the moving object detection is complete.

3.1 Parameters of Gaussian Mixture Model

Learning rate (α) from equation (7), the threshold value (T) and the number of Gaussian component (K) are GMM parameters for exact selection [12]. A Gaussian mixture model has been constructed for each pixel, and its learning rate can be adaptively adjusted based on the scene change inferred from the frame difference [13]. The learning rate of the parameter estimate determines the weight of the current pixel's impact on the model's parameters. It also influences the convergence speed of the models. A small learning rate is suitable for slowly changing backgrounds, allowing gradual adaptation of the model.

A higher threshold is selected for complicated scenes with variations in lighting or cluttered backgrounds, reducing false positives. In simpler scenes, a lower threshold is used for the accurate detection of moving objects. The determination of the number of Gaussian components (K) used to model each background pixel depends on the highest number of modes observed in the pixel's background distribution. In other words, the value of K is chosen randomly to accommodate the maximum number of modes present in the background distribution of a pixel. This ensures that the model captures the complexity and variations in the pixel's background accurately. In this research, different settings of these parameters for video analysis will improve the detection accuracy of vehicles.

3.2 Performance Metric Calculations

For calculating key performance metrics, including precision, recall, F1 score, and similarity. In the realm of traffic surveillance, the ability to accurately identify and classify pixels as foreground or background is paramount for the system's effectiveness [14].

In the evaluation process, four fundamental metrics are considered: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) [15]. These metrics serve as the foundation for subsequent calculations. Precision, a vital metric that measures the accuracy of positive predictions, offering insights into the reliability of foreground identifications., is calculated as the ratio of TP to the sum of TP and FP Mathematically, precision is expressed as

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

Recall gauges the model's ability to identify all relevant instances, providing a measure of sensitivity is calculated as the ratio of TP to the sum of TP and FN

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

F1 Score offers a balanced assessment of model performance and providing a balanced assessment of precision and recall, is calculated using

$$F1\ Score = \frac{2(Precision \times Recall)}{Precision + Recall} \tag{11}$$

Similarity provides an overall measure of correct classifications, encompassing both foreground and background identifications. Similarity is calculated as the ratio of the sum of TP and TN to the total number of instances.

$$Similarity = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Performance metric calculation ensures a comprehensive and insightful evaluation of the highway frame processing system. The metrics collectively contribute to a nuanced understanding of the algorithm's strengths and areas for improvement, guiding the optimization of parameters for enhanced traffic surveillance accuracy.

3.3 Computational Algorithm

To achieve effective video traffic analysis using the Gaussian Mixture Model (GMM) algorithm, a series of computational steps have been meticulously designed as follows:

Step 1 Initialization:
Set parameters Number of Gaussian components alpha and threshold number for the algorithm.

Step 2 K-means Initialization:
Use the K-means algorithm to initialize Gaussian means for each color channel and employ K-means clustering to assign pixels to initial clusters.

Step 3 Parameters Initialization:
Initialize mean, variance, and probability for each Gaussian component for each pixel and initialize foreground and background images.

Step 4 Variable Parameters:
Adjust the variable parameters such as number of gaussians, alpha and threshold value for analysis.

Step 5 Pixel-wise Processing:
Update means, variances, and probabilities based on the pixel value and current parameters. Normalize probabilities (π) for each pixel. Classify the pixel as foreground or background based on the threshold value. Update the background pixel value based on the means and classification.

4. Results and Discussion

This research focuses on examining the key parameters that influence the performance of a highway frame processing system. Specifically, it explores how changes in the number of Gaussian components, the learning rate (α in Python), and different threshold values for pixel classification affect the performance of the GMM algorithm. The study uses Python software for this analysis. Figure 1 illustrates the original image used in the study.

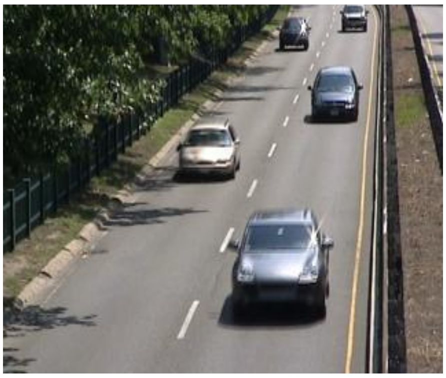


Fig. 1 *Image input of original frame (#1062)*

4.1 Related Parameters for Analysing the GMM

Three main settings were explored: the number of Gaussian components, the learning rate, α and the threshold value to analyse the image by using the Gaussian Mixture Model.

4.1.1 Effects of the Number of Gaussian Components on the GMM

The number of Gaussian components in a GMM plays a crucial role in the accuracy of object detection, especially in dynamic traffic environments. As shown in Fig. 2, the quality of segmentation improves as the number of Gaussian components increases from one to three, with a constant learning rate (α = 0.001) and threshold of 16. When only a single Gaussian component is used, the model simplifies the scene, resulting in rough segmentation. In this case (Fig. 2a), the background is poorly modelled, making it difficult to distinguish moving vehicles from the background.

Increasing the number of Gaussian components to two (Fig. 2b) leads to better separation between the foreground (vehicles) and background. The second Gaussian component allows the model to capture more variations in pixel intensities, improving its ability to handle subtle lighting changes and background shifts. However, while segmentation improves, there are still some misclassifications, especially in regions with complex motion or shadows.

With three Gaussian components (Fig. 2c), the model achieves the best segmentation, providing more detailed and accurate results. The increased complexity allows the GMM to account for a broader range of pixel behaviours, such as gradual lighting changes or occasional background movement. This leads to more accurate foreground detection and less interference from the background. However, adding more components also increases the computational cost and makes the model more sensitive to noise. This sensitivity can cause minor background fluctuations to be detected as part of the foreground, highlighting the need to balance model complexity with practical performance considerations.

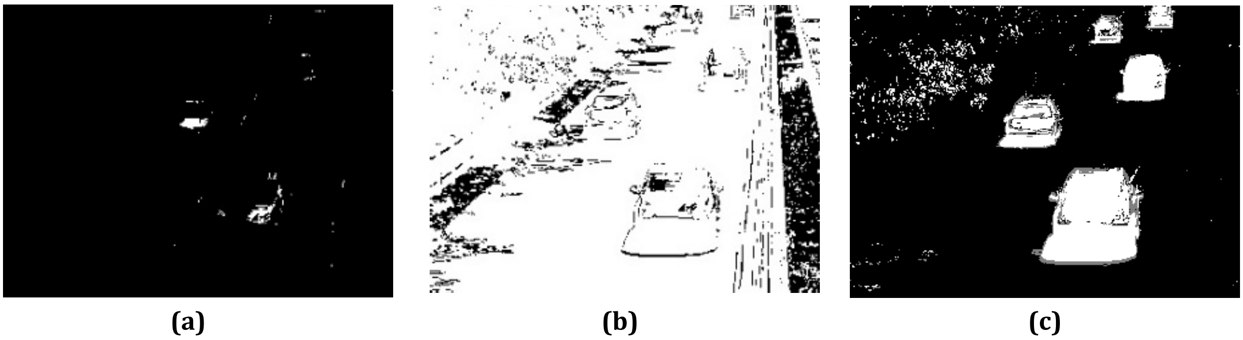


Fig. 2 *Image output with number of Gaussian (a) One; (b) Two; (c) Three*

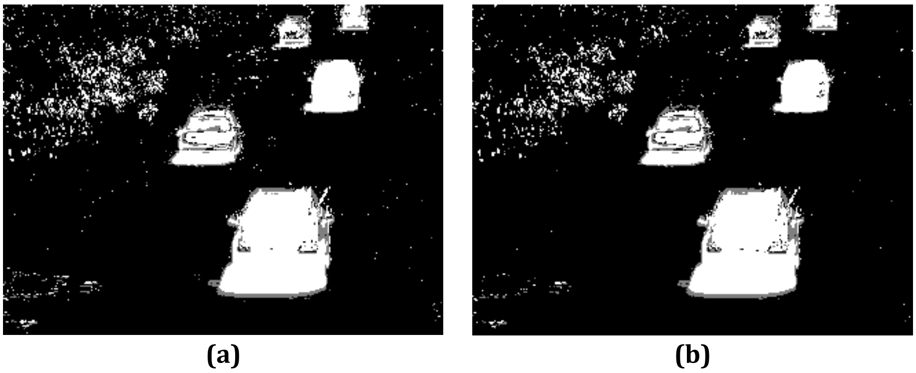
4.1.2 Effects of Learning rate, α on the GMM

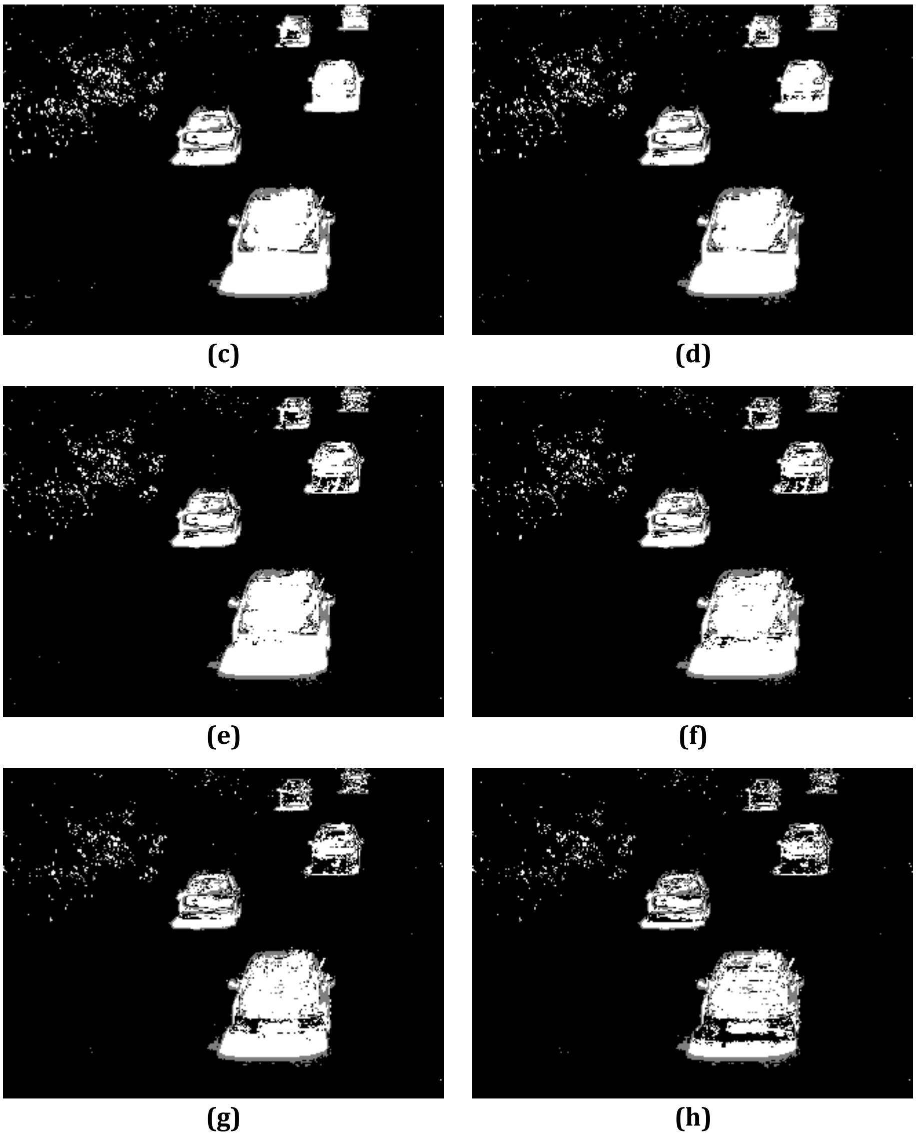
The learning rate (α) plays a crucial role in the performance of the GMM by controlling how quickly the model adjusts to changes in the scene. A properly tuned learning rate allows the model to capture dynamic traffic elements, such as vehicle movement and lighting changes, without becoming overly sensitive to noise or background fluctuations. Fig. 3 demonstrates how different learning rates affect GMM performance, with rates ranging from 0.0005 to 0.028, while keeping the number of Gaussian components at 3 and the foreground pixel threshold at 16.

At the lowest learning rate of 0.0005 (Fig. 3a), the model adapts very slowly to changes in the scene. This results in a stable background model but causes a delay in detecting new or fast-moving objects, making it unsuitable for real-time traffic surveillance. The slow adaptation means the model lags in detecting vehicles, which is impractical in dynamic traffic situations. As the learning rate increases to 0.005 and 0.008 (Fig. 3c and 3d), the model becomes more responsive, detecting vehicles more accurately while still maintaining a stable background. These moderate learning rates provide a good balance, allowing the model to quickly adapt to new objects without overreacting to transient changes in the environment. For example, at a learning rate of 0.005 (Fig. 3c), vehicle detection is noticeably improved, with clearer foreground segmentation and fewer background artefacts.

At higher learning rates, such as 0.015 and 0.025 (Fig. 3e and 3g), the model adapts more rapidly to changes in the scene. While this quick adaptation can be useful in dynamic environments, it also increases the model's sensitivity to noise. At a rate of 0.025 (Fig. 3g), for instance, the model overreacts to minor changes in pixel intensity, misclassifying parts of the background as foreground. This results in cluttered segmentation, where noise and lighting changes are wrongly interpreted as moving vehicles, reducing accuracy.

The optimal learning rate for this dataset is 0.005 (Fig. 3c), where the model strikes the best balance. At this rate, the GMM achieves clear and accurate segmentation of vehicles, adapting efficiently to changes in traffic scenes without overfitting to background noise. These results emphasize the importance of choosing an optimal learning rate: too low a rate leads to slow adaptation and missed detections, while too high a rate causes instability and increased noise sensitivity.





(c) (d)

(e) (f)

(g) (h)

Fig. 3 *Image output with learning rate (a) 0.0005; (b) 0.0008; (c) 0.005; (d) 0.008; (e) 0.015; (f) 0.018; (g) 0.025; and (g) 0.028*

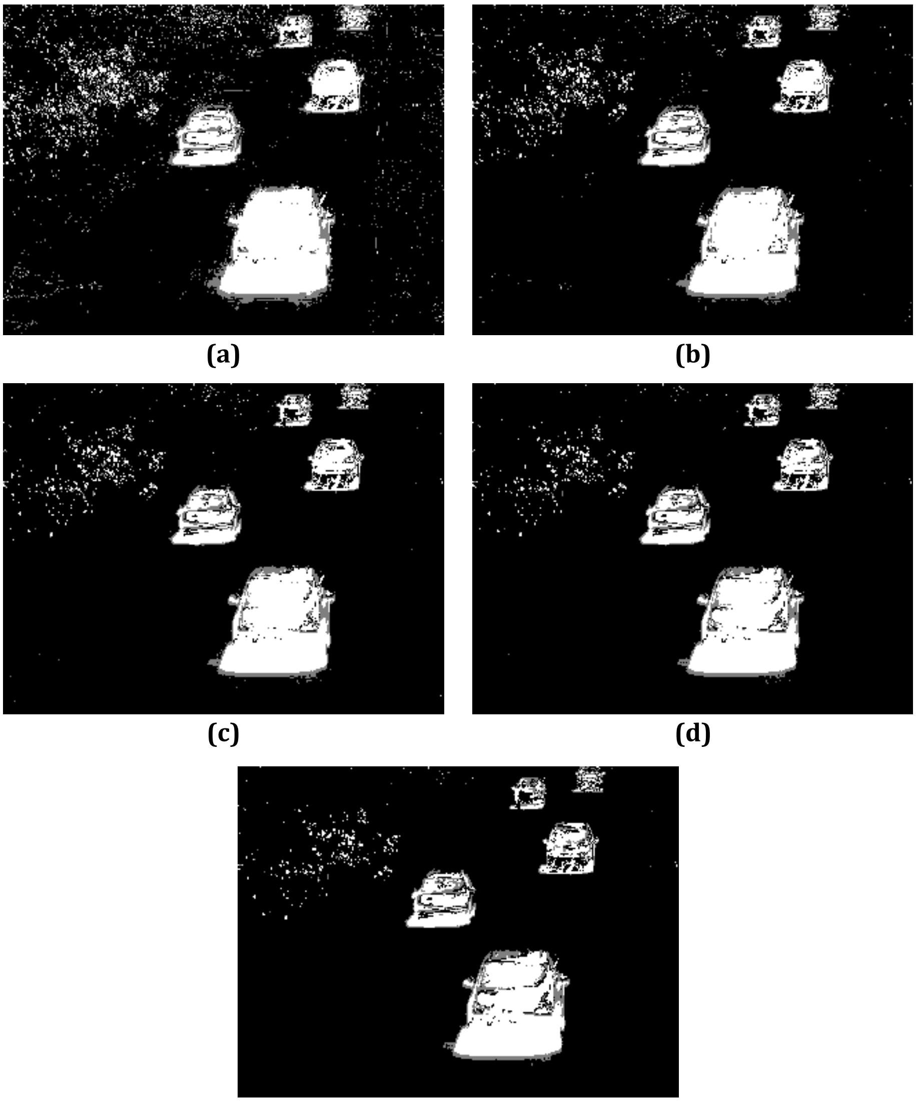
4.1.3 Effects of the Threshold Value on the GMM

The threshold value in the GMM plays a crucial role in determining how pixels are classified as either part of the foreground (vehicles) or the background. This parameter directly affects the accuracy of vehicle detection, as it controls the model's sensitivity to changes in pixel intensity. Fig. 4 illustrates how varying the threshold value, from 5 to 25, with a constant number of Gaussian components (3) and a learning rate of 0.005, affects the balance between detection flexibility and the risk of false positives.

At the lowest threshold of 5 (Fig. 4a), the GMM is overly permissive, classifying many background pixels as foreground. This leads to over-segmentation, where noise, shadows, and small lighting changes are mistakenly identified as moving objects. While this setting captures some minor details of vehicles, the high rate of false positives makes it impractical for real-world traffic monitoring, as it reduces precision. Increasing the threshold to 10 (Fig. 4b) makes the model more selective, filtering out some of the background noise. However, at this threshold, the GMM still includes non-vehicle objects, showing that further refinement is needed to better distinguish between the foreground and background.

The optimal performance is achieved at a threshold of 15 (Fig. 4c). At this value, the GMM accurately classifies most vehicle pixels while minimizing false positives from the background. This balance is important for traffic video analysis, as it ensures that the system reliably detects vehicles without being distracted by irrelevant background elements. The segmentation output at this threshold is clean, with a clear distinction between vehicles and the surrounding environment, making it the most effective setting for this application.

When the threshold is increased to 20 and 25 (Fig. 4d and 4e), the model becomes too conservative, classifying fewer pixels as foreground. This results in under-segmentation, where parts of vehicles are missed and incorrectly classified as background. While this reduces false positives, it also compromises the model's ability to detect smaller or partially hidden vehicles, which are important for real-time traffic monitoring. A threshold that is too high may miss crucial foreground details, lowering the system's overall reliability. The results highlight the need for selecting an appropriate threshold value. A low threshold introduces too much noise, while a high threshold may cause important vehicle details to be overlooked.



(a) **(b)**

(c) **(d)**

(e)

Fig. 4 *Image output with threshold value (a) 5; (b) 10; (c) 15; (d) 20; and (e) 25*

4.2 Performance Analysis

In the analysis of the image segmentation performance, several key parameters were systematically investigated to understand their impact on the accuracy and efficacy of the algorithm. This included testing different values for the number of Gaussian components, learning rates, and threshold values, all of which impacted the segmentation results.

4.2.1 Effects of the Number of Gaussian on the GMM

Table 1 shows the impact of the number of Gaussian components on the performance of the Gaussian Mixture Model (GMM) in vehicle detection. When only one Gaussian component is used, the model struggles, with a recall of 0.6246 and a precision of 0.6557. This indicates that it frequently misclassifies background pixels as vehicles. The F-measure and similarity scores are also low, reflecting the model's limited reliability at this stage.

When two Gaussian components are used, the performance improves significantly. Recall increases to 0.6713, and precision rises to 0.8691, meaning the model becomes better at identifying vehicles and reducing background noise. The F-measure score of 0.7575 and similarity of 0.6097 show a more balanced and accurate detection compared to the one-component setup.

With three Gaussian components, the model achieves the best performance. Recall rises to 0.8192, and precision stays high at 0.8547. The F-measure score of 0.8366 and similarity of 0.7191 indicate the clearest and most accurate vehicle detection. However, increasing the number of components also increases the computational load, which could be a concern for real-time traffic monitoring systems.

In summary, while increasing the number of Gaussian components improves detection accuracy, it also increases computational complexity, which needs to be balanced for real-time applications.

Table 1 *Result of performance based on number of Gaussian*

Number of Gaussian	Recall	Precision	F-measure	Similarity
1	0.6246	0.6557	0.6398	0.4704
2	0.6713	0.8691	0.7575	0.6097
3	0.8192	0.8547	0.8366	0.7191

4.2.2 Effects of the Learning rate, α on the Performance Index

Table 2 shows how different learning rates affect the GMM's ability to detect vehicles accurately. At the lowest learning rate (α = 0.0005), the model adapts slowly, resulting in a precision of 0.8702 and recall of 0.7547. While the background model remains stable, the model struggles to quickly detect all vehicles, as reflected by the lower recall. The F-measure score of 0.8083 and similarity of 0.6783 indicate that while the model is reliable, it is not responsive enough to fast changes in the traffic scene.

As the learning rate increases to 0.005, the performance improves significantly. Recall increases to 0.8590, and the F-measure score rises to 0.8428, showing that the model is better at detecting vehicles while maintaining strong precision (0.8271). The similarity metric also improves to 0.7283, indicating better alignment with the actual traffic flow. However, when the learning rate exceeds 0.015, the model becomes overly sensitive to small changes in the scene, leading to instability. This results in more false positives, reducing precision and introducing noise, which causes performance to decline.

The optimal learning rate is found in the range of 0.005 to 0.008, offering the best balance between accurate detection and stable performance. A lower learning rate is too slow to capture real-time changes, while a higher rate makes the model too sensitive to minor fluctuations, decreasing its overall accuracy.

Table 2 *Result of performance based on learning rate*

Learning Rate, α	Recall	Precision	F-measure	Similarity
0.0005	0.7547	0.8702	0.8083	0.6783
0.0008	0.8018	0.8604	0.8301	0.7095
0.005	0.8590	0.8271	0.8428	0.7283
0.008	0.8631	0.8185	0.8402	0.7245
0.015	0.8735	0.7807	0.8245	0.7014
0.018	0.8767	0.7615	0.8151	0.6879
0.025	0.8760	0.7261	0.7940	0.6584
0.028	0.8667	0.6523	0.7443	0.5928

4.2.3 Effects of Threshold Value on the Performance Index

Table 3 shows the impact of different threshold values on the model's performance, including recall, precision, F-measure score, and similarity. At a low threshold of 5, the GMM detects a high number of vehicles, as seen by the recall of 0.9600. However, the precision is much lower at 0.1219, indicating that many background elements are incorrectly identified as vehicles. This leads to a low F-measure score of 0.2164, suggesting that while the model is sensitive, it lacks accuracy due to a high number of false positives.

As the threshold increases to 15, the model becomes more selective. Precision improves, and the F-measure score rises, showing a better balance between recall and precision. The GMM is better at filtering out background noise while still detecting most vehicles accurately. This threshold offers a good compromise between detecting vehicles and reducing false positives.

At higher thresholds (above 20), the model becomes too conservative. While precision continues to improve, recall drops significantly, meaning the model misses many vehicles. The F-measure score decreases, and the similarity metric shows a decline in overall performance. At these higher thresholds, the model struggles to detect smaller or less obvious vehicles, reducing its effectiveness.

In conclusion, a moderate threshold value (around 10 to 15) strikes the best balance for traffic video analysis. A lower threshold leads to too many false positives, while a higher threshold causes the model to miss key vehicles, lowering its accuracy.

Table 3 *Result of performance based on threshold value*

Threshold Value	Recall	Precision	F-measure	Similarity
5	0.6862	0.8587	0.7628	0.6166
10	0.8288	0.8158	0.8222	0.6981
15	0.8693	0.7853	0.8252	0.7024
20	0.8875	0.7636	0.8209	0.6962
25	0.8948	0.7417	0.8111	0.6822

5. Conclusion

This study has successfully demonstrated the use of a GMM for traffic video surveillance, with a focus on optimizing key parameters such as the number of Gaussian components, learning rate, and threshold values. The results show that parameter tuning plays a critical role in enhancing vehicle detection accuracy. Specifically, the optimal settings, such as using three Gaussian components, a learning rate of 0.005, and a threshold value of 15, yielded the most accurate segmentation of moving vehicles. This approach offers a balanced solution between detection accuracy and computational efficiency, which is crucial for real-time traffic monitoring applications. The findings contribute valuable insights into the improvement of traffic surveillance systems, helping to address challenges like complex backgrounds, lighting variations, and noise interference. By optimizing the GMM algorithm, this study aids in the development of more reliable and efficient urban traffic management systems, contributing to safer and more organized road environments.

Despite the promising results achieved in this study, several areas remain open for further exploration. Further studies could focus on testing the GMM approach in more complex traffic scenarios, such as high-speed highways or urban intersections, to assess its adaptability in challenging environments. Additionally, the development of self-optimizing algorithms that can adjust parameters in real-time based on evolving traffic conditions would increase the robustness and efficiency of the system, ensuring long-term performance improvements. These directions offer exciting opportunities for advancing the scalability and accuracy of traffic monitoring systems.

Acknowledgement

The authors would also like to thank the Faculty of Applied Sciences and Technology, Universiti Tun Hussein Onn Malaysia for its support.

Conflict of Interest

The authors declare that there is no conflict of interests regarding the publication of the paper. No financial or personal relationships with individuals or organizations have inappropriately influenced our work. The research presented in this paper has been conducted in an unbiased manner, and the findings are reported objectively.

Author Contribution

The authors confirm contribution to the paper as follows: ***study conception and design:*** *Voon Zi Yi, Siti Suhana Jamaian;* ***analysis and interpretation of results:*** *Voon Zi Yi, Nor Afiqah Mohd Aris;* ***validation of results:*** *Voon Zi Yi, Nor Afiqah Mohd Aris, Siti Suhana Jamaian;* ***draft manuscript preparation:*** *Voon ZI Yi. All authors reviewed the results and approved the final version of the manuscript. They have agreed to be accountable for all aspects of the work, ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.*

References

[1] Ramakrishnan, S, (2014, November 11). The Malaysian transport system is going nowhere. *Malaysia Kini*

[2] Boukerche, A., Tao, Y., & Sun, P. (2020). Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems. *Computer Networks*, 182, 107484.

[3] Chen, W., Tian, Y., Wang, Y., & Huang, T. (2016). Fixed-point Gaussian Mixture Model for analysis-friendly surveillance video coding. *Computer Vision and Image Understanding*, 142, 65–79.

[4] Nicolas, H., & Brulin, M. (2014). Video traffic analysis using scene and vehicle models. *Signal Processing: Image Communication*, 29(8), 807–830.

[5] Goyette, N., Jodoin, P.-M., Porikli, F., Konrad, J., & Ishwar, P. (2012). Changedetection.net: A new change detection benchmark dataset. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*. IEEE. https://doi.org/10.1109/cvprw.2012.6238919

[6] Hsieh, J., Yu, S., Chen, Y., & Hu, W. (2006). Automatic Traffic Surveillance System for Vehicle Tracking and Classification. *IEEE Transactions on Intelligent Transportation Systems,* 7(2), 175–187.

[7] Jagannathan, P., Rajkumar, S., Frnda, J., Parameshachari, B. D., & Prabu, S. (2021). Moving Vehicle Detection and Classification Using Gaussian Mixture Model and Ensemble Deep Learning Technique. *Wireless Communications and Mobile Computing,* 2021, 1–15.

[8] Alandkar, L., & Gengaje, S. R. (2016). Dealing Background Issues in Object Detection using GMM: A Survey. *International Journal of Computer Applications,* 150(5), 50–55.

[9] Rakesh, S., Hegde, N. P., Gopalachari, M. V., Jayaram, D., Madhu, B., Hameed, M. A., Vankdothu, R., & Kumar, L. (2023). Moving object detection using modified GMM based background subtraction. Measurement: Sensors, 30, 100898. https://doi.org/10.1016/j.measen.2023.100898

[10] Stauffer, C., & Grimson, W. (1999). Adaptive background mixture models for real-time tracking.

[11] Ma, Y., Zhang, Z. P., & Li, Y. (2015). Moving Vehicles Detection Based on Improved Gaussian Mixture Model.

[12] Chavan, R. (2021). Object detection by automatically tuning GMM training parameters using Cuckoo Search Optimization technique.

[13] Ying-Hong, L., Hong-Fang, T., & Yan, Z. (2010). An improved Gaussian mixture background model with real-time adjustment of learning rate.

[14] Mishra, S. (2022). Baffling concept of true positive and true negative. *Medium*.

[15] Yao, J., & Shepperd, M. (2021). The impact of using biased performance metrics on software defect prediction research. *Information & Software Technology*, 139, 106664.