Received: 17 March 2025 | Revised: 30 April 2025 | Accepted: 19 May 2025 | Published online: 5 June 2025

RESEARCH ARTICLE

Journal of Data Science and Intelligent Systems
2026, Vol. 4(2) 211–223
DOI: 10.47852/bonviewJDSIS52025712

Robust Histogram Signature-Driven Template Matching for Vehicle Tracking in Traffic Videos

BON VIEW PUBLISHING

Aaron Rasheed Rababaah[1,*]

[1]*College of Engineering and Applied Sciences, American University of Kuwait, Kuwait*

Abstract: Correlation-based template matching (CTM) is widely used for object detection because of its simplicity and effectiveness in scenarios where grayscale features are sufficient. However, CTM often underperforms when color is a crucial distinguishing factor. To address this limitation, we propose contextual hierarchical composite template matching (CH-CTM), a color-histogram-enhanced CTM algorithm that integrates color information into a traditional correlation framework. CH-CTM augments the correlation index with red, green, and blue histogram comparisons to improve robustness in color-sensitive contexts. We evaluated CH-CTM using five diverse traffic video datasets that include various lighting conditions, vehicle types, sizes, and colors. Twelve experiments were conducted using standard performance metrics. Results demonstrated notable improvements over baseline CTM: CH-CTM achieved a peak accuracy of 98.30%, an average accuracy of 92.43%, and average precision of 92%. These findings confirm the importance of incorporating color information into template matching, which expands CTM's applicability in complex real-world scenarios, particularly in traffic surveillance and object tracking.

Keywords: correlation-based template matching, color histogram, enhanced template matching, object detection and tracking, vehicle tracking, traffic videos, computer vision

1. Introduction

Vehicular movement monitoring is a fundamental component of modern transportation infrastructure and urban planning that supports applications such as traffic flow analysis, congestion management, and law enforcement operations. In this domain, vehicle tracking in traffic videos has become a crucial capability. Among the various methodologies used for vehicular movement monitoring, template-based approaches remain widely used because of their versatility, computational efficiency, and practical effectiveness in real-world traffic scenarios.

Template-based vehicle tracking involves constructing a reference model—or template—that encapsulates the key visual features of target vehicles in video frames [1]. Subsequent frames are analyzed by matching against these templates to detect and track vehicles over time. Traditional template-matching methods are valued for their relatively straightforward implementation, real-time processing capabilities, and adaptability to diverse environmental conditions [2].

Recent advancements have significantly enriched template-based tracking, with research exploring feature-enhanced templates, deep-learning-driven template generation, adaptive template update strategies, and hybrid approaches that combine template matching with techniques such as particle filtering and background subtraction [3, 4]. Despite the growth of deep learning in object detection and tracking, template-matching methods remain relevant, particularly in applications requiring interpretable, low-latency, and infrastructure-constrained solutions.

In this context, we introduce the contextual hierarchical composite template matching (CH-CTM) framework, which enhances traditional template-based tracking by incorporating hierarchical contextual information and composite template strategies. Unlike conventional methods that rely on static or single-scale templates, CH-CTM dynamically manages multiple templates across different spatial hierarchies, allowing for improved robustness to scale changes, occlusions, and complex urban environments. Furthermore, rather than replacing deep-learning-based detection methods, CH-CTM is positioned as a lightweight, complementary solution suited for scenarios where deep learning models are impractical because of computational or data constraints.

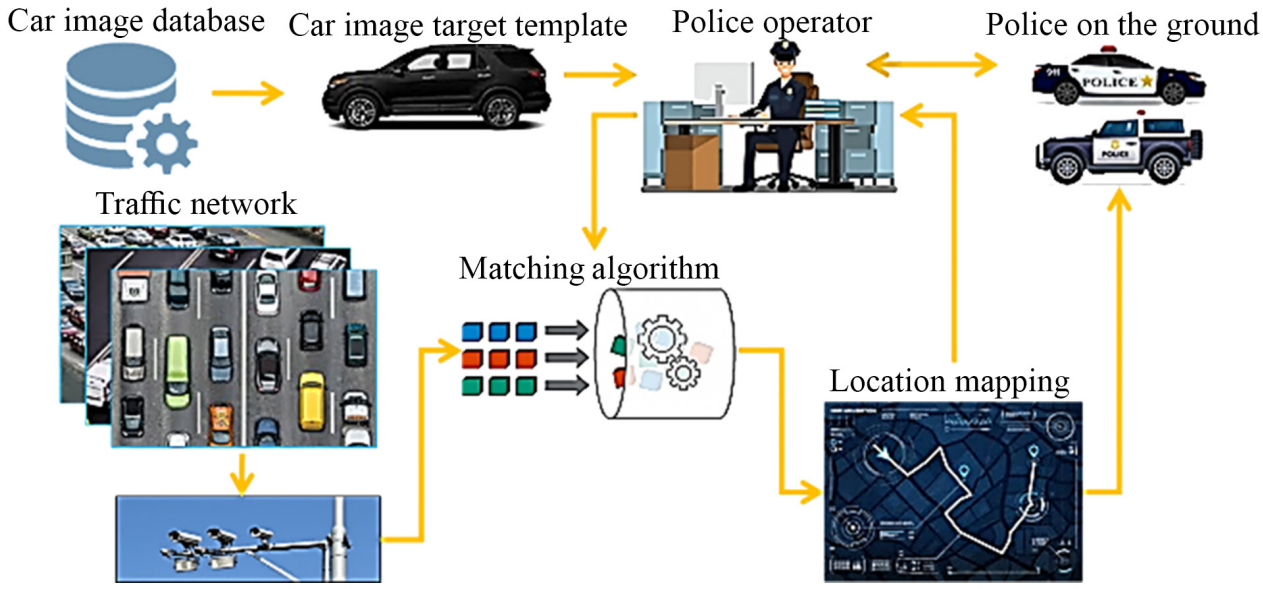
As shown in Figure 1, CH-CTM operates in a vehicular monitoring framework designed for real-time traffic surveillance. Templates of target vehicles are registered in a centralized database. An operator selects one or more templates, which are then processed by the CH-CTM algorithm using live video streams from fixed urban landmarks such as traffic lights, bridges, and roundabouts. CH-CTM continuously searches for matches in these streams. Once a positive match is detected, notifications are sent to registered clients, updating their operational maps and enabling the central police command to take appropriate actions, such as dispatching a patrol unit to intercept and verify the identified vehicle.

2. Literature Review

This section presents a review of several closely related works. Vehicle tracking in traffic videos has garnered significant interest due to its pivotal role in various applications such as traffic management, surveillance, and autonomous driving systems. Among diverse approaches, template-based methods have emerged as prominent techniques for vehicle tracking. This literature review aims to provide an extensive overview of template-based vehicle tracking techniques in traffic video analysis, focusing on their methodologies, strengths, limitations, and future directions.

*Corresponding author: Aaron Rasheed Rababaah, College of Engineering and Applied Sciences, American University of Kuwait, Kuwait. Email: arababaah@auk.du.kw

Figure 1
Framework of the proposed algorithm



Template matching is a fundamental approach in vehicle tracking, wherein a template representing a target object (car) is compared with each frame in a video sequence. Qureshi et al. [1] demonstrated the effectiveness of sum of squared differences (SSD) template matching for real-time vehicle tracking in traffic videos. For feature-based templates, Dallalzadeh and Guru [2] proposed a feature-based template-matching method that used color histograms and edge features, leading to improved performance in complex traffic scenarios with occlusions and illumination changes. In deep-learning-based templates, a deep-learning-based template learning method was introduced [3], where a convolutional neural network extracted informative features from car images for precise tracking in traffic videos. In the context of template update strategies, some studies proposed a dynamic template update mechanism based on online learning that allowed the template to adapt gradually to variations in car appearance caused by factors such as occlusions and viewpoint changes. Yang et al. [5] presented a multiview template fusion technique that integrated information from multiple cameras to create a holistic representation of the tracked car, enabling robust tracking across different viewpoints. The work of these authors represents multiview templates. For template matching in challenging conditions, Kawahara et al. [6] introduced a scale-adaptive template-matching algorithm based on contours that dynamically adjusted the size of the template to maintain accurate tracking performance under various scales. Real-time implementation may be necessary, as demonstrated by Shinde et al. [7], who developed an efficient template-based vehicle tracking system using parallel computing techniques. Their approach enabled real-time performance even on embedded platforms with limited computational resources. Hybrid approaches were reported by Orun [8]. The study presented a high-speed tracking method for overlapped vehicles using template matching based on contour information. The approach emphasized real-time performance and addressed challenges such as vehicle overlap, achieving robust vehicle tracking in challenging traffic conditions. Velazquez-Pupo et al. [9] presented a template-based vehicle tracking method with effective occlusion-handling mechanisms that enhanced tracking performance in congested traffic scenarios. Choi et al. [10] proposed a robust feature-based template-matching approach for vehicle tracking, demonstrating improved performance in scenarios with occlusions and various lighting conditions. Han et al. [11] proposed a template-based vehicle-tracking method with deep learning feature extraction, leveraging the discriminative power of deep neural networks for accurate tracking in challenging environments. Qureshi et al. [12] proposed a real-time vehicle-tracking system that integrated template matching with scale-invariant feature transform (SIFT) features, achieving robust tracking performance in traffic videos with occlusions and scale variations. The method presented by Patel and Brahmbhatt [13] used a template-based vehicle-tracking approach that combined optical flow with Shi-Tomasi corner detection and tracking, enabling effective performance in dynamic traffic scenes without relying on complex deep learning models. Sun et al. [14] proposed a data-fusion-based algorithm for multitarget tracking in digital videos. The authors primarily used histogram-based template matching to correctly resolve the association problem when multiple targets are present. This technique was reported to be effective in urban environments. Chantara et al. [15] proposed an adaptive scale template-matching algorithm for vehicle tracking that dynamically adjusted the template size to accommodate variations in target scale. Another work presented a template-based vehicle-tracking method with adaptive template updates using deep learning techniques, enhancing tracking robustness in challenging environments.

Template-based vehicle tracking in traffic video analysis is a dynamic and rapidly evolving research area with substantial advancements achieved in recent years. The integration of advanced features, deep learning models, and adaptive update strategies has shown promising results in enhancing tracking performance. However, challenges such as occlusions, scale variations, and real-time processing requirements persist [16, 17], warranting further research efforts. Future endeavors should focus on developing more robust and efficient template-based tracking algorithms to address the evolving demands of traffic surveillance and intelligent transportation systems.

3. Theoretical Framework

This section presents the technical foundation of the algorithms and techniques used.

3.1. Template matching

Correlation-based template matching (CTM) is a widely used technique for finding specific image patches (templates) in a large

image. It operates by calculating the similarity between the template and subregions of the large image, identifying locations where the similarity is highest. This makes it useful for tasks such as object detection, visual tracking, and image registration. The following presentation of the template-matching algorithm is based on the study reported by Gonzalez and Woods [18] as well as Aggarwal and Xia [19].

The core concept lies in comparing pixel intensities between the template and the image patches. The following two primary metrics were used by Rababaah [20]:

1) Cross-correlation (CC)

This measures the linear relationship between pixel intensities, as shown in Equation (1):

$$CC\left(dx,dy\right)=\sum_{m}\sum_{n}f\left(m+\mathrm{dx},n+\mathrm{dy}\right)\cdot G\left(m,n\right) \tag{1}$$

Where:

f = input image,
G = the template,
m,n = pixel coordinates,
dx, dy = delta increments in shift operations.

2) Normalized CC (NCC)

This normalizes CC to be less sensitive to variations in brightness and contrast, as shown in Equation (2):

$$\begin{aligned} R_f &= \sqrt{\sum_{m,n}f\left(m+dx,n+dy\right)-\bar{f}_{dx,dy}{}^2} \\ R_G &= \sqrt{\sum_{m,n}G\left(m,n\right)-\overline{G}^2} \\ NCC\left(dx,dy\right) &= \frac{CC\left(dx,dy\right)-MN\cdot\bar{f}_{dx,dy}\cdot\overline{G}}{R_f\cdot R_G} \end{aligned} \tag{2}$$

Higher values of both CC and NCC indicate higher similarity between the template and the image patch. The location with the highest value is considered the best match. The advantages of NCC include simplicity, computational efficiency, and invariance to changes in brightness and contrast (with NCC). The limitations of NCC include sensitivity to geometric transformations (scaling and rotation) and a tendency to produce false positives due to similar image features. Several techniques aim to address the limitations, including multiscale matching that searches for the template at different scales to overcome scaling variations. Phase-only correlation uses the phase information of Fourier transforms for rotation invariance, and feature-based matching extracts distinctive features from the template and image for better matching, but it may be computationally expensive.

3.2. Histogram similarity

Histogram similarity models are essential tools for measuring the likeness between histograms, commonly used in various fields such as image processing, computer vision, and data mining. Four of the most popular histogram similarity models include histogram intersection, chi-square distance, the Bhattacharyya coefficient, and Euclidean distance.

Histogram intersection measures the similarity between two histograms by computing the intersection of their respective histograms. Mathematically, histogram intersection similarity is calculated as the sum of the minimum values of the corresponding bins between two histograms [21], as shown in Equation (3):

$$\mathrm{HI}\left(f,G\right)=\sum_{i=1}^{N}\min\left(H_f\left(i\right),H_G\left(i\right)\right) \tag{3}$$

The chi-square distance quantifies the difference between two histograms by calculating the SSD between corresponding bins, normalized by the sum of the bin frequencies [22], as shown in Equation (4):

$$\chi^2\left(f,G\right)=\sum_{i=1}^{N}\frac{\left(H_f(i)-H_G(i)\right)^2}{H_f(i)+H_G(i)+\varepsilon} \tag{4}$$

The Bhattacharyya coefficient measures the overlap between two histograms. It is computed as the sum of the square root of the product of the frequencies in corresponding bins [23], as shown in Equation (5):

$$BC\left(f,G\right)=\sum_{i=1}^{N}\sqrt{H_f(i)\cdot H_G(i)} \tag{5}$$

Euclidean distance is another histogram similarity model that calculates the distance between two histograms in the multidimensional space. It is computed as the square root of the SSD between corresponding bin values [24], as shown in Equation (6):

$$ED\left(f,G\right)=\sqrt{\sum_{i=1}^{N}\left(H_f(i)-H_G(i)\right)^2} \tag{6}$$

These models provide different perspectives on histogram similarity, allowing for versatile applications in various domains.

3.3. The proposed algorithm

The schematic of the proposed algorithm is depicted in Figure 2 and described hereafter. Live video: in this stage, a video signal is fed to the system to search for a specific template. To ensure efficiency, each video feed should be assigned one processing machine, which may be a computer or an embedded microcontroller. The algorithm handles one video frame at a time. Therefore, the video stream is segmented into frames, and each individual frame is processed separately. The first operation applied on the frame is the grayscale conversion from RGB, as shown in Figure 3. The same operation is applied to the template image (shown in Figure 4) because CTM requires an input image of grayscale space.

For template matching, Equation (2) is used to calculate the similarity index of NCC between the current frame and the template image. This is shown in Figure 5, where the grayscale version of the resulting map appears on the left, and for better visual clarity, a Jit map is produced on the right. Furthermore, a 3D version of the same result is shown in Figure 6. The 3D version clearly shows the high peaks that represent the candidate locations of the target template.

The algorithm requires the correlation matching index (CMI) to be calibrated in the interval CMI ε [0, 1]. The CMI is the confidence threshold for detecting the presence of the template in the current frame. In this study, the CMI was set to 0.7.

Figure 2
The block diagram of the proposed algorithm

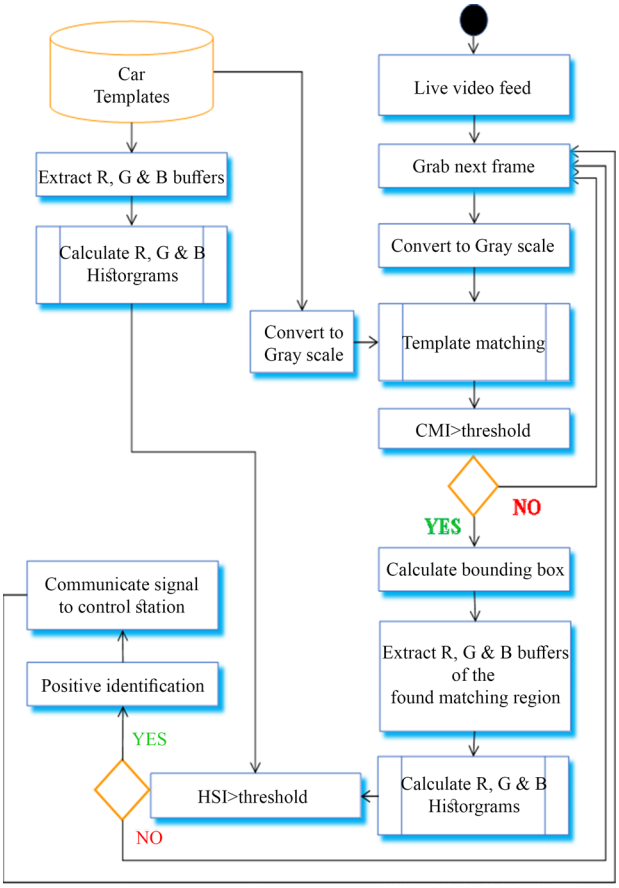


Figure 3
RGB to grayscale conversion of the raw input frame

Figure 4
RGB to grayscale conversion of the raw template image

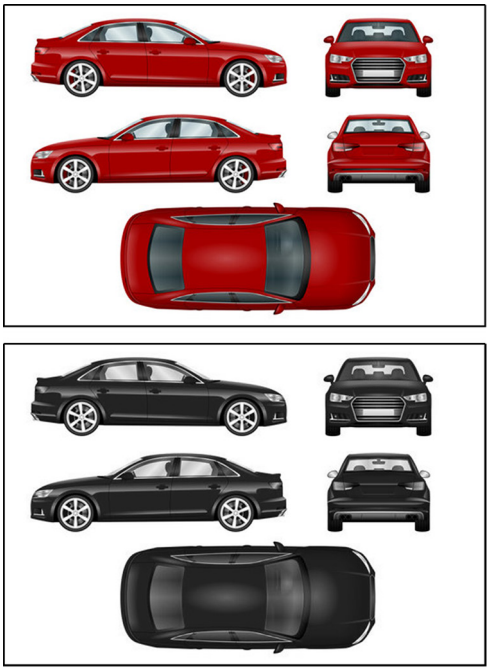
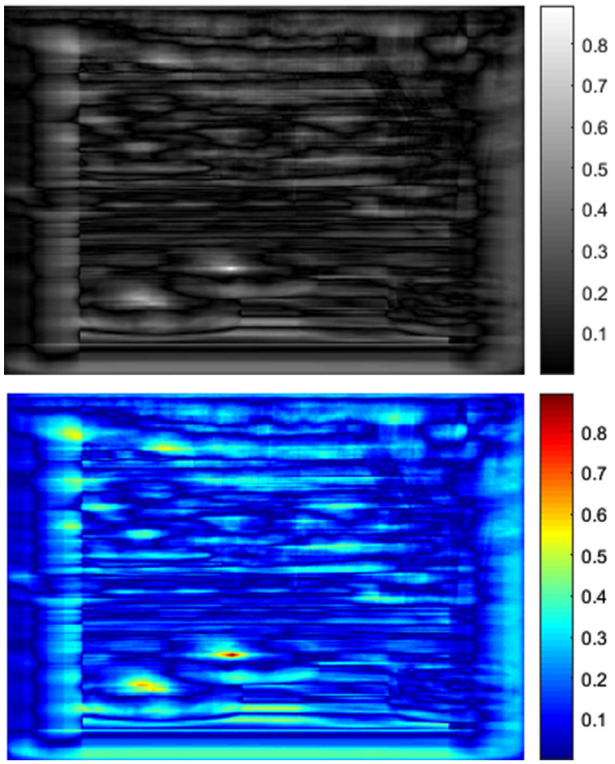


Figure 5
Top = grayscale version of the NCC map. Bottom = Jit map version of the NCC map



For a reliable detection, our algorithm further supports the initial detection using the traditional NCC by applying RGB histograms to verify the color distribution of the detected template.

Figure 6
3D version of the Jit map of the result shown in Figure 5

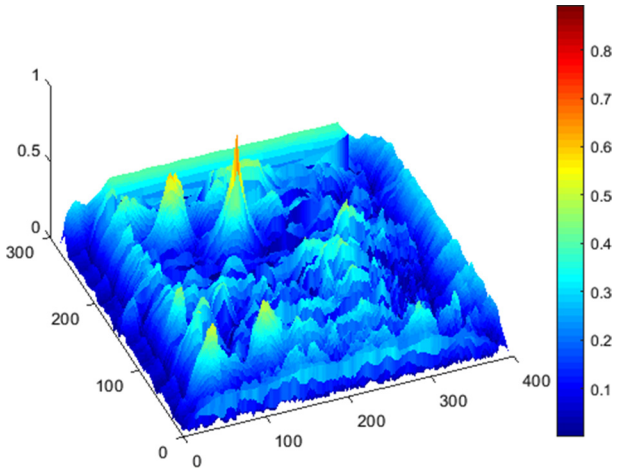
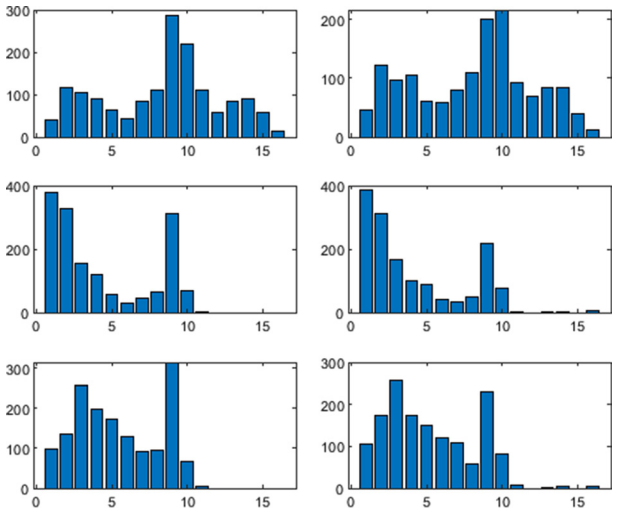


Figure 7
Left = R, G, and B histograms of the template. Right = R, G, and B histograms of the detected region in the frame



The RGB histograms of the template and the detected region in the current frame are normalized. This operation is shown in Figure 7. As shown in the presented sample histograms, we set the number of bins to 16, which was effective. Although a higher number of bins would produce a higher resolution of color scales, we chose to keep it at a manageable size to maintain processing efficiency.

The histogram similarity index (HSI) was then calculated between each pair of R, G, and B histograms of the template and the current frame using Equation (7).

$$HSI\left(h_1, h_2\right) = \frac{\text{mean}(|h_1(i) - h_2(i)|)}{H \cdot W} \tag{7}$$

Where:

HSI = histogram similarity index,

h1, h2 = the first and second calculated histograms of the template and current frame,

H = the height of the template image (pixels),

W = the width of the template image (pixels).

H*W denotes the size of the template, which is the same as that of the detected region. This plays an important role as a normalizing factor to map the calculated HSI to the interval [0, 1].

HSI must also be calibrated in the interval HSI ε [0, 1]. During our experiments, we found that a threshold of 0.025 was an effective level to avoid false positives. It is important to note that HSI is a distance-based index. Therefore, the lower it is, the better it is between two histogram vectors. As for the CMI, it is a correlation-based index. Therefore, the higher it is, the better it is between two images.

CH-CTM_Algorithm:

Input:

- Video stream V
- Template image T
- Target color histogram H_T (computed from T)
- Matching threshold τ_match
- Histogram similarity threshold τ_hist

Output:

- List of detected bounding boxes B over frames

Procedure:

1) Initialize
   a. Compute color histogram H_T for the template T.
   b. Initialize an empty list B to store detections.

2) For each frame F in video stream V
   a. Slide a window W of size T over F with a predefined stride.
   b. For each window W:
      i. Extract the image patch P corresponding to W.
      ii. Perform template matching:
      - Compute matching score S_TM between T and P (e.g., using normalized cross-correlation).
      iii. If S_TM ≥ τ_match:
      - Compute color histogram H_P of patch P.
      - Compute histogram similarity score S_HIST between H_T and H_P (e.g., using Bhattacharyya distance or histogram intersection).
      - If S_HIST ≥ τ_hist:
      - Add W to list B as a valid detection.

3) Post-processing (optional)
   a. Apply Non-Maximum Suppression (NMS) to remove overlapping detections.

4) Return B.

4. Experimental Work

This section has four subsections: Data, Experiments, Evaluation metrics, and Observations on the results.

4.1. Data

We used five datasets to evaluate the proposed algorithm. These datasets consist of archived video sequences collected under different real-world conditions. The five datasets collectively include 4,763 frames. Figure 8 presents representative samples from each dataset, and Table 1 summarizes their characteristics. The datasets were selected to reflect diverse environmental scenarios. The links to these datasets are given in the Data Availability Statement, later.

Figure 8
Samples of all five datasets used to test the proposed algorithm

4.2. Experiments

We conducted 15 different experiments and used typical performance metrics to evaluate CH-CTM. In each of the experiments and for each frame, two indices were calculated to verify positive detection, namely, correlation similarity index (CSI) and HSI. Each experiment represented a scenario from the videos in the datasets. The two indices were plotted on the same chart to compare their levels with the calibrated thresholds for positive detection. CSI was plotted without modifications, but HSI was amplified for a meaningful visual representation. For example, Figure 9 shows a scenario of approximately 400 frames. Frames 170–230 showed a positive detection of the target object. The 15 experiments are shown in Figures 9–23. The analysis of the experimental results will be presented and discussed in the following sections.

Table 1
Description of the datasets used

#	Dataset	Camera situation	Tested frames	Angle
1	Vehicle Detection Image Dataset	Fixed	404	Bird's eye
2	Car Detection and Tracking Dataset	Drone	451	Top view
3	Vehicle Detection Image Set	Drone	631	Top view
4	Top-View Vehicle Detection Image Dataset	Fixed	1,681	Bird's eye
5	UA-DETRAC_dataset	Fixed	1,506	Top view

4.3. Evaluation metrics

Before presenting the testing results, the evaluation metrics are introduced first. A confusion matrix (CM) is a typical analysis tool used to evaluate the performance of classification models and is a square matrix that contains performance results across all classes. CM measures the following four basic metrics: true positive (TP), the model accurately predicts a positive class; true negative (TN), the model accurately predicts a negative class; false positive (FP), the model inaccurately predicts a positive class; and false negative (FN), the model inaccurately predicts a negative class. On the basis of these four basic measures, five more metrics are computed as follows: accuracy measures the ratio of the correctly classified (P or N) samples to the overall population. Accuracy is calculated using Equation (8).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{8}$$

Precision measures the ratio of the correctly detected positives to all detected positive samples, as shown in Equation (9).

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

Recall (or sensitivity) measures the ratio of the correctly retrieved positive samples to the total number of actual positive samples, as shown in Equation (10).

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

Specificity measures the ratio of the correctly retrieved negative samples to the total number of actual negative samples, as shown in Equation (11).

$$Specificity = \frac{TN}{TN+FP} \tag{11}$$

F1-score measures the harmonic mean of precision and recall, which indicates the robustness of the classification model, as shown in Equation (12).

Figure 9
Testing results of Dataset_1 and Template_1

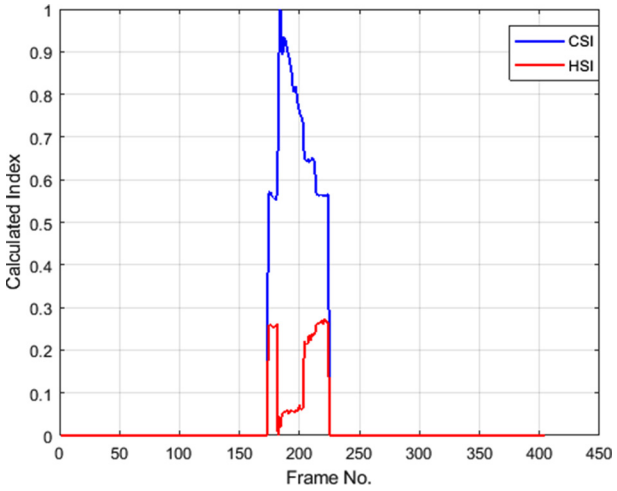


Figure 10
Testing results of Dataset_1 and Template_2

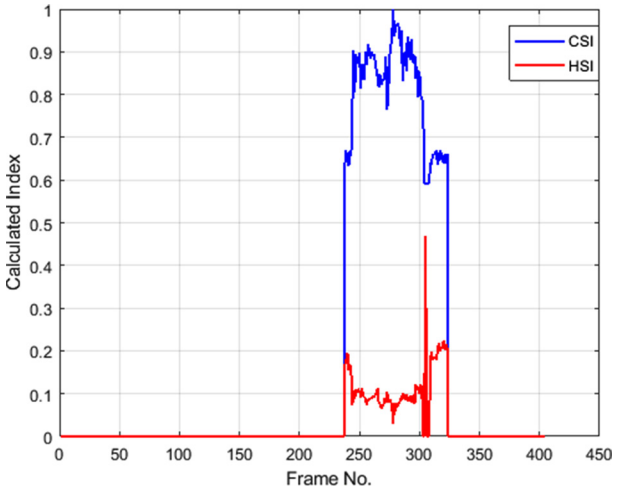


Figure 11
Testing results of Dataset_1 and Template_3

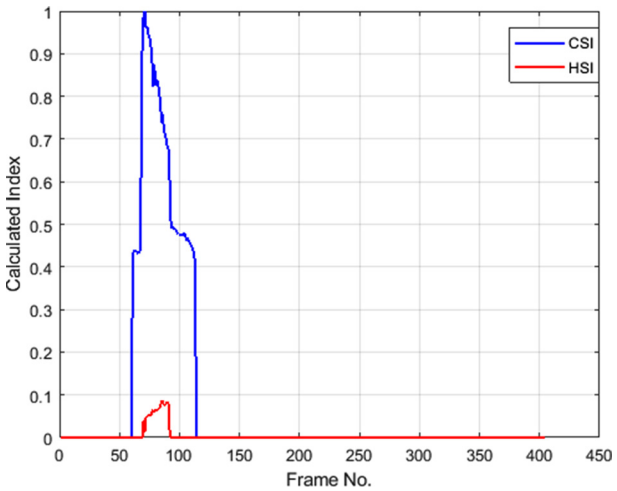


Figure 12
Testing results of Dataset_2 and Template_1

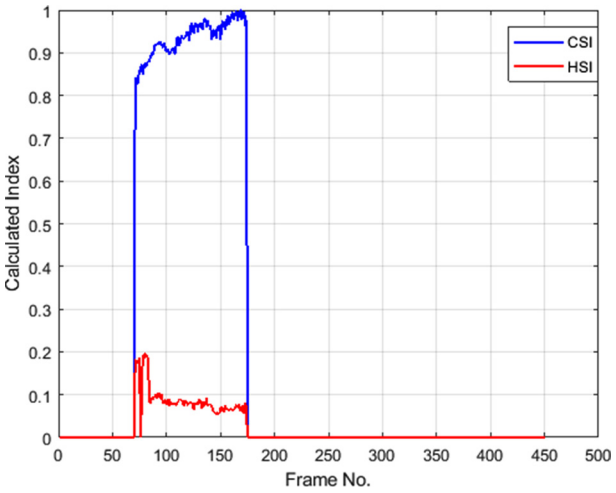


Figure 13
Testing results of Dataset_2 and Template_2

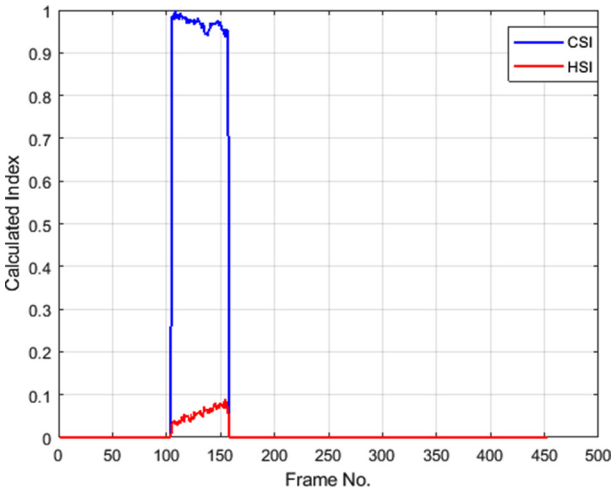


Figure 14
Testing results of Dataset_2 and Template_3

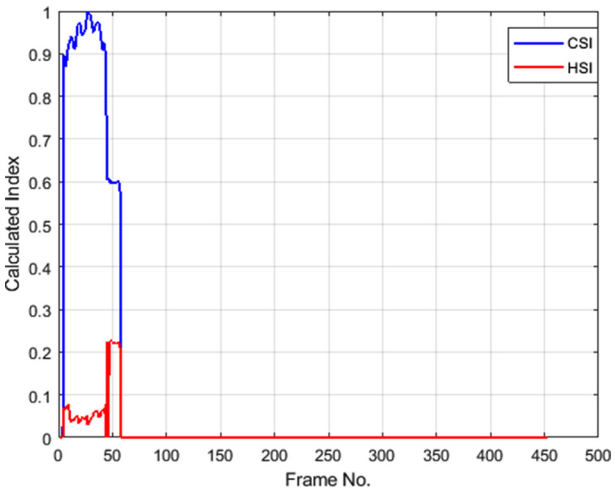


Figure 15
Testing results of Dataset_3 and Template_1

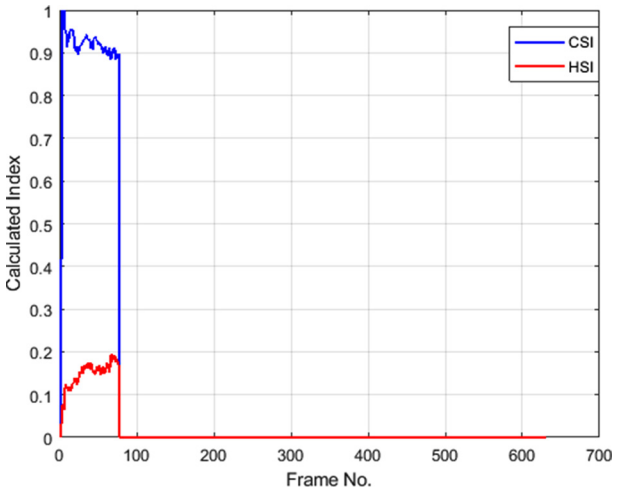


Figure 16
Testing results of Dataset_3 and Template_2

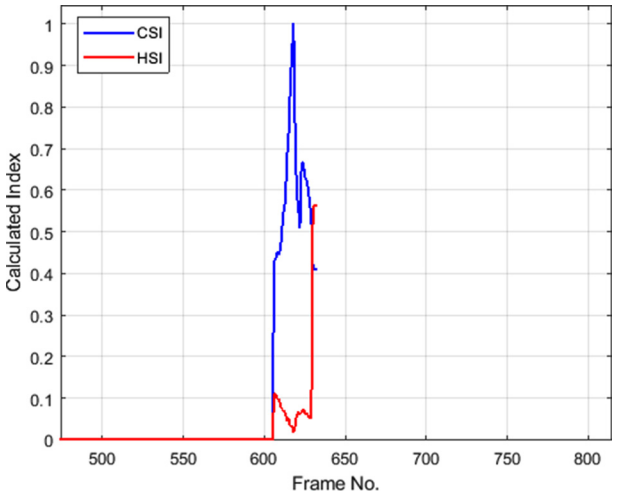


Figure 17
Testing results of Dataset_3 and Template_3

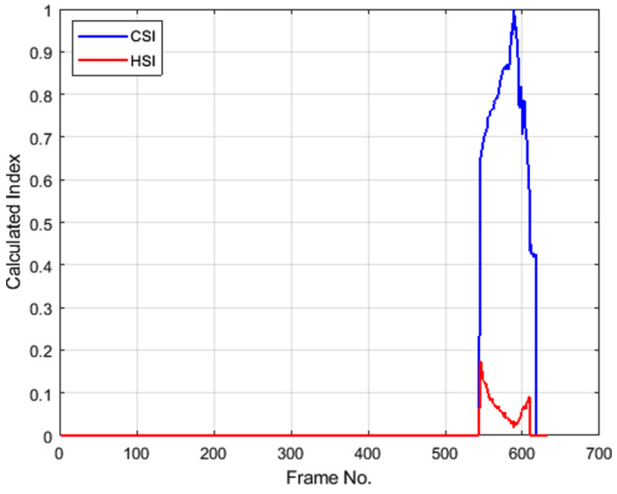


Figure 18
Testing results of Dataset_4 and Template_1

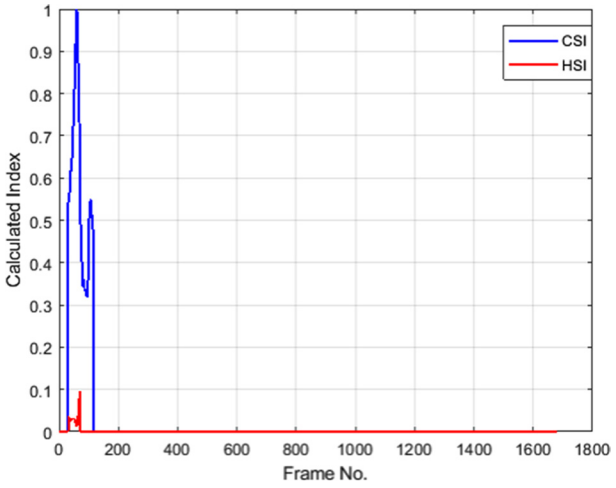


Figure 19
Testing results of Dataset_4 and Template_2

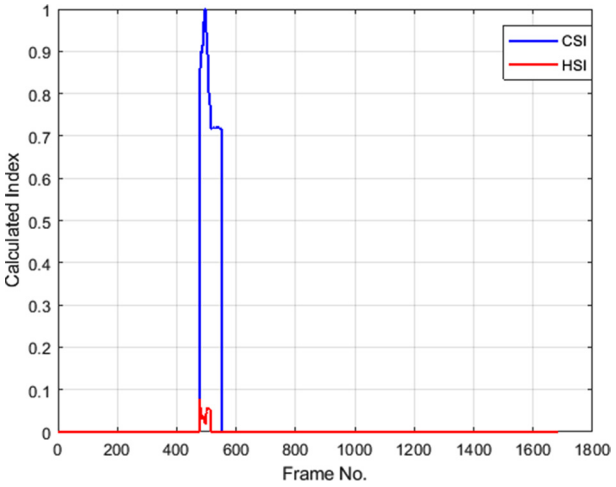


Figure 20
Testing results of Dataset_4 and Template_3

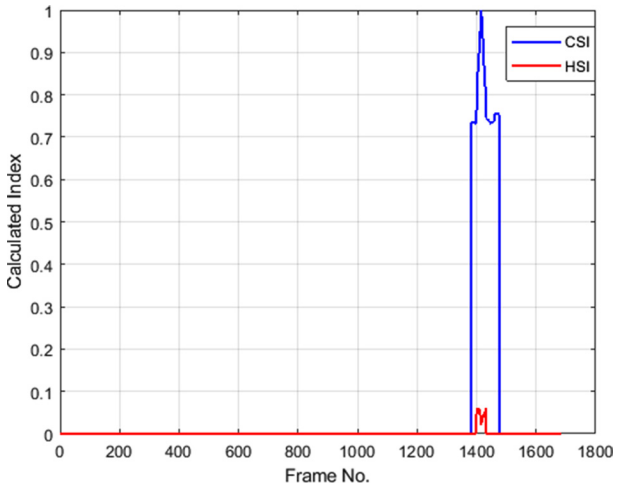


Figure 21
Testing results of Dataset_5 and Template_1

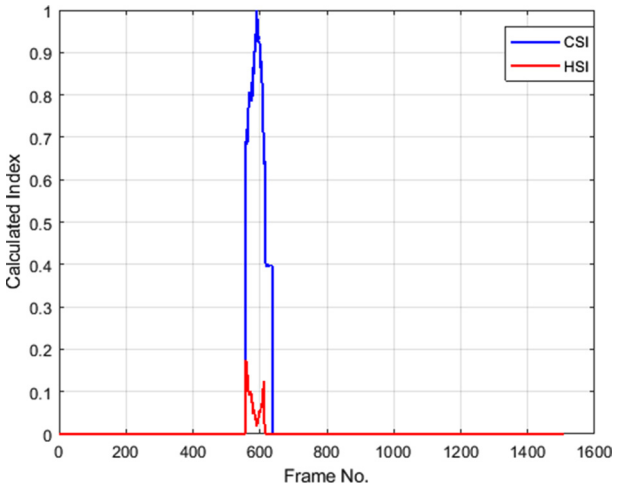


Figure 22
Testing results of Dataset_5 and Template_2

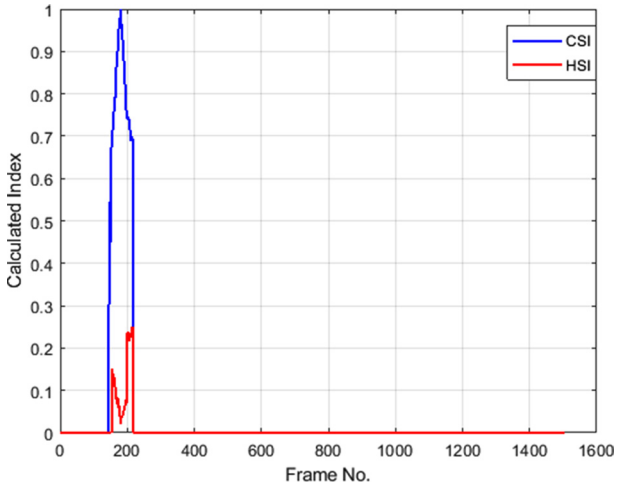


Figure 23
Testing results of Dataset_5 and Template_3

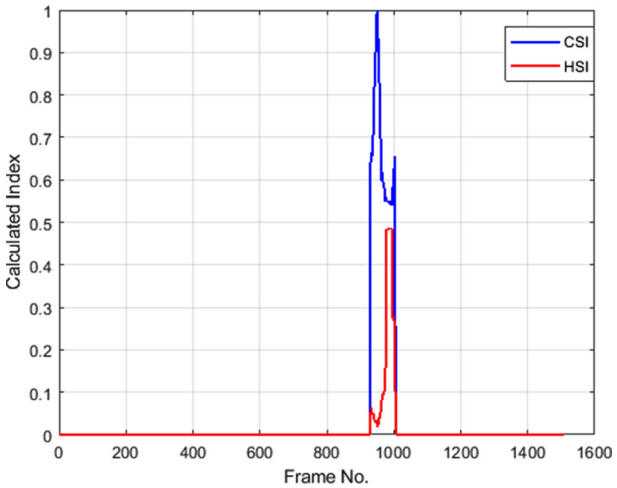
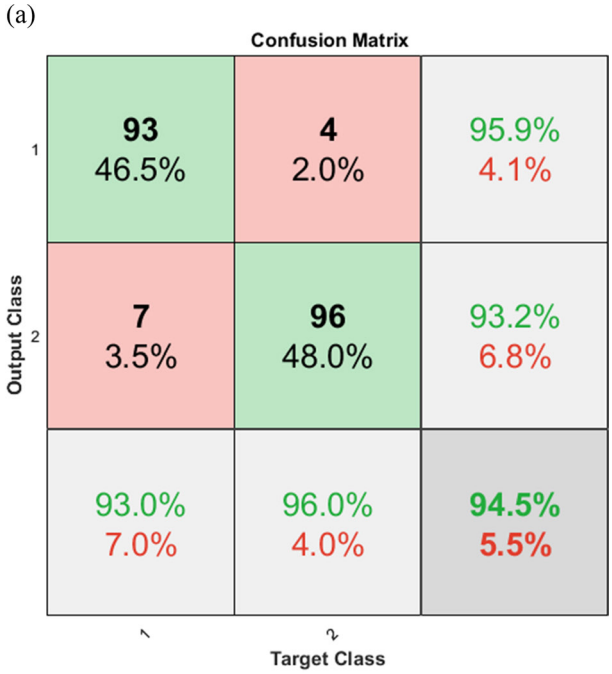
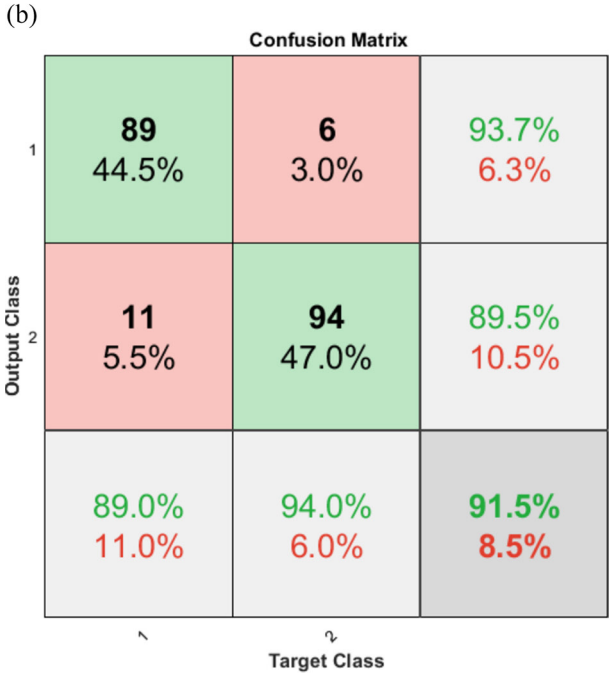


Figure 24
(a) Dataset_1 CM; (b) Dataset_2 CM

(a)



(b)



$$F1\_Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{12}$$

To calculate all of these performance metrics, a CM was generated for each experiment. Because there were six datasets, we produced six CMs, as shown in Figures 24–26. The metrics used in our study can be found in the study reported by Rababaah and Wolfer [25].

Tables 2 and 3 summarize the calculated performance metrics of all CMs in Figures 24–26. Table 1 presents the first three CMs (CM1–CM3), and Table 2 presents the remaining three CMs (CM4–CM6).

To highlight the detection and tracking accuracy and precision of the proposed algorithm CH-CTM, the results from all experiments

Figure 25
(a) Dataset_3 CM; (b) Dataset_4 CM

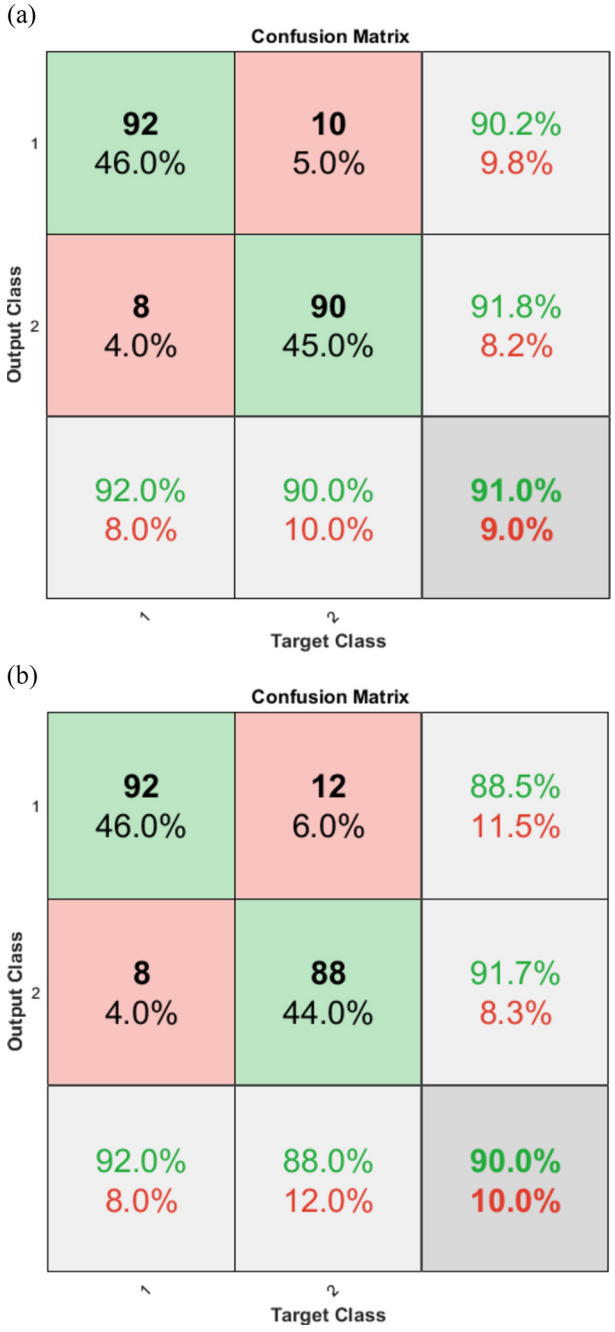
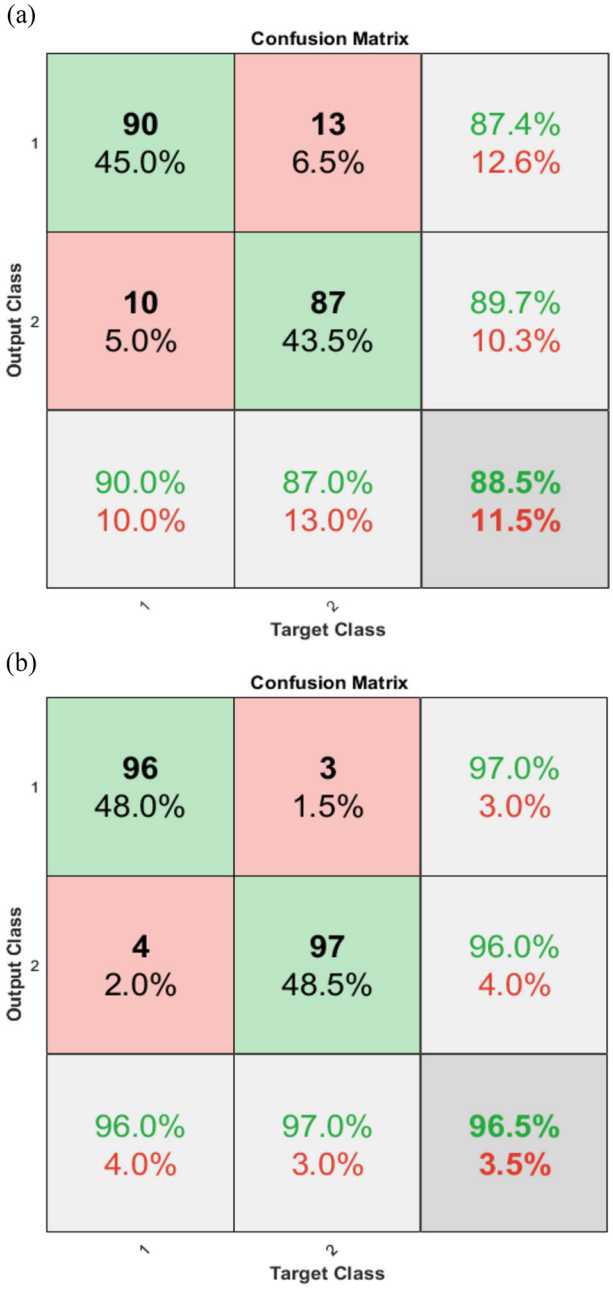


Figure 26
(a) Dataset_4 CM; (b) additional CM



are aggregated in Figures 27 and 28, respectively. Furthermore, the two figures show the following statistical measures of tracking accuracy: mean, STD, max, and min values.

4.4. Observations on the results

Several important observations regarding the performance and limitations of the proposed algorithm CH-CTM were documented throughout the experimental evaluation.

The algorithm demonstrated strong tracking reliability, achieving an average accuracy of 92.43% across diverse traffic scenarios. High precision (average, 92%) and favorable recall, specificity, and F1-score metrics further confirmed the consistency of the detection performance. However, a critical analysis revealed that although the overall accuracy was high, certain scenarios exposed the limitations of the approach.

Table 2
Performance metrics of the CMs of Dataset_1 to Dataset_3

Metric	CM1		CM2		CM3	
Accuracy	0.95	0.95	0.92	0.92	0.91	0.91
Precision	0.93	0.96	0.89	0.94	0.92	0.90
Recall	0.96	0.93	0.94	0.90	0.90	0.92
Specificity	0.93	0.96	0.90	0.94	0.92	0.90
F1-score	0.94	0.95	0.91	0.92	0.91	0.91

Table 3
Performance metrics of the CMs of Dataset_4 to Dataset_6

Metric	CM1		CM2		CM3	
Accuracy	0.90	0.90	0.89	0.89	0.97	0.97
Precision	0.92	0.88	0.90	0.87	0.96	0.97
Recall	0.88	0.92	0.87	0.90	0.97	0.96
Specificity	0.92	0.88	0.90	0.87	0.96	0.97
F1-score	0.90	0.90	0.89	0.88	0.96	0.97

Figure 27
Aggregated accuracy results of all 12 experiments

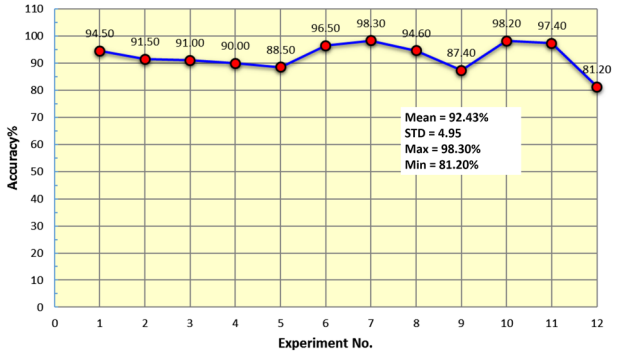
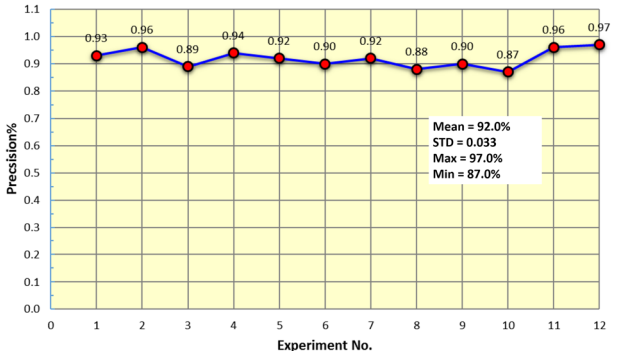


Figure 28
Aggregated precision results of all 12 experiments



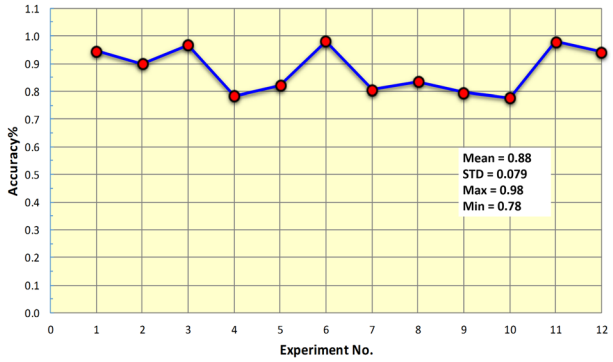
For example, the lowest recorded accuracy (81.20%) occurred in Scenario 9, where the target vehicle's white color was confounded by the presence of multiple similarly colored objects in the scene. This illustrates a key weakness of relying on simple color histograms (HSI) for verification: in complex or cluttered environments, color alone may be insufficient for robust disambiguation. Although vehicles with distinctive colors were more reliably tracked—as shown in several datasets—the approach struggled when color uniqueness was low, suggesting a need for more discriminative or multimodal feature descriptors in future work.

From a computational perspective, the algorithm maintained reasonable runtime efficiency across datasets of varying resolutions (336 × 256 to 854 × 480). The additional computational cost introduced by CH-CTM's enhancements (approximately 6% overhead) was relatively low. However, high-resolution inputs and the need for repeated histogram calculations may present challenges for real-time deployment, particularly on resource-constrained hardware. Therefore, although feasible for many real-time applications, deployment in large-scale or high-throughput traffic monitoring systems may require optimization or hardware acceleration.

Table 4
Performance metrics of Dataset_1 to Dataset_6 for gray-scale-based CTM

Metric	CM1		CM2		CM3	
Accuracy	0.95	0.90	0.97	0.78	0.82	0.98
Precision	0.95	0.98	0.70	0.94	0.92	0.90
Recall	0.92	0.85	0.94	0.89	0.81	0.89
Specificity	0.89	0.98	0.79	0.98	0.98	0.83
F1-score	0.97	0.90	0.88	0.89	0.91	0.91
Metric	**CM4**		**CM5**		**CM6**	
Accuracy	0.80	0.83	0.80	0.78	0.98	0.94
Precision	0.89	0.75	0.79	0.87	0.76	0.97
Recall	0.85	0.96	0.93	0.80	0.88	0.90
Specificity	0.65	0.88	0.83	0.90	0.86	0.97
F1-score	0.96	0.79	0.76	0.95	0.96	0.89

Figure 29
Aggregated results of all 12 experiments for grayscale-based CTM



Importantly, the fixed nature of the templates was another limitation. As the method relied on preselected target templates, it was less adaptable to dynamic changes such as significant vehicle appearance alterations, occlusions, or novel vehicle entries. Furthermore, performance under partial or full occlusion was not systematically evaluated, but qualitative observations suggested that tracking robustness degraded when targets were heavily occluded. Future work should consider integrating adaptive template update mechanisms or occlusion handling strategies.

Regarding generalizability, the current experiments covered a moderate range of conditions but primarily involved urban traffic environments with moderate vehicle densities. The performance of CH-CTM under more extreme conditions, such as heavy congestion, nighttime scenes, adverse weather, or nonurban settings, remains uncertain. Further testing using more diverse datasets is essential to comprehensively evaluate and enhance the model's generalization capabilities.

Finally, although color histograms provide a lightweight second-level confirmation in CH-CTM, their descriptive power is limited. Complex traffic scenes often involve lighting variations, shadows, reflections, and subtle color distortions, all of which can compromise histogram reliability. Future extensions could benefit from more sophisticated color models or the incorporation of texture, edge, or deep feature representations to enhance discrimination in challenging scenarios.

The results of the proposed algorithm were compared with those of the grayscale version of CTM. Table 4 presents the results of all six experiments. Figure 29 depicts the performance of grayscale CTM. The proposed algorithm outperformed grayscale CTM based on the comparison of their mean and STD accuracy. The results showed that the proposed algorithm achieved a 5% increase [(92.43 – 88)/88 × 100% = 5%] in accuracy compared with grayscale CTM. For stability, represented by STD, the proposed algorithm achieved a remarkable 59.6% reduction [(7.9 − 4.95)/4.95 × 100% = 59.6%].

Overall, although the CH-CTM algorithm demonstrated promising results and notable improvements compared with baseline template matching, its practical deployment should consider these limitations. Future work will aim to address these challenges to better align the system with the demands of real-world traffic monitoring applications.

5. Conclusion

This study proposed an enhancement to the traditional CTM algorithm. The enhanced algorithm, called CH-CTM, integrates color information into the standard CTM process, which originally relies solely on grayscale images. Specifically, CH-CTM computes color histograms for the red, green, and blue channels of both the target template and the candidate regions in video frames. These histograms serve as a secondary validation layer to reinforce detections initially identified by CTM. The proposed algorithm follows a two-stage validation process: first, a candidate region must meet a correlation threshold, and second, it must pass a histogram similarity threshold to be confirmed as a positive detection. To evaluate CH-CTM, five diverse traffic video datasets were utilized, and 15 experiments were conducted and analyzed.

The experimental results demonstrated that CH-CTM achieved reliable and consistent performance across all evaluated metrics. Notably, the method achieved an average accuracy of 92.43% and precision of 92%. Additional metrics, including specificity, recall, and F1-score, had satisfactory results. A recognized limitation of the proposed algorithm was its real-time processing capability. The average processing time per frame was approximately 0.5 s, tested using MATLAB 2018 on a system with Windows 7, an Intel i5 2.67 GHz processor (2 physical and 4 logical cores), and 8 GB RAM. This runtime limitation could potentially be mitigated by leveraging higher-performance computing environments or optimized implementations. Future research directions include a comparative analysis of CH-CTM against feature-based methods and deep-learning-based tracking approaches to further benchmark its strengths and identify areas for improvement. Enhancing runtime efficiency and exploring adaptive template update mechanisms are also promising avenues for extending this work.

Conflicts of Interest

The author declares that he has no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in Kaggle at https://www.kaggle.com/datasets/pkdarabi/vehicle-detection-image-dataset, in Kaggle at https://www.kaggle.com/datasets/amitkumargurjar/car-detection-and-tracking-dataset, in Kaggle at https://www.kaggle.com/datasets/brsdincer/vehicle-detection-image-set, in Kaggle at https://www.kaggle.com/datasets/farzadnekouei/top-view-vehicle-detection-image-dataset, and in Kaggle at https://www.kaggle.com/datasets/dtrnngc/ua-detrac-dataset.

Author Contribution Statement

Aaron Rasheed Rababaah: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration.

References

[1] Qureshi, A. M., Algarni, A., Aljuaid, H., Alatiyyah, M. H., Alnowaiser, K., & Jalal, A. (2025). Semantic segmentation based real-time traffic monitoring via Res-UNet classifier and Kalman filter. *SN Computer Science*, *6*(1), 30. https://doi.org/10.1007/s42979-024-03586-7

[2] Dallalzadeh, E., & Guru, D. S. (2010). Feature-based tracking approach for detection of moving vehicle in traffic videos. In *Proceedings of the First International Conference on Intelligent Interactive Technologies and Multimedia*, 254–260. https://dl.acm.org/doi/10.1145/1963564.1963609

[3] Prabakaran, P., Jayasudha, R., Sandhu, M., & Gupta, S. (2024). Deep learning-based image analysis to track vehicles. In *2024 2nd International Conference on Disruptive Technologies*, 601–606. https://doi.org/10.1109/ICDT61202.2024.10489331

[4] Lai, Y., He, M., Yao, C., Yin, Z., Zou, H., & Yang, Z. (2024). Dynamic update template for visual object tracking. In *Proceedings of the Fifth International Conference on Computer Vision and Data Mining*, *13272*, 132720Y. https://doi.org/10.1117/12.3048089

[5] Yang, H. F., Cai, J., Liu, C., Ke, R., & Wang, Y. (2023). Cooperative multi-camera vehicle tracking and traffic surveillance with edge artificial intelligence and representation learning. *Transportation Research Part C: Emerging Technologies*, *148*, 103982. https://doi.org/10.1016/j.trc.2022.103982

[6] Kawahara, H., Senoo, T., Ishii, I., Hirano, M., Kishi, N., & Ishikawa, M. (2022). High-speed tracking for overlapped vehicles using template matching based on contour information. *Transactions of the Society of Instrument and Control Engineers*, *58*(1), 21–30. https://doi.org/10.9746/sicetr.58.21

[7] Shinde, P. A., Mane, Y. B., & Tarange, P. H. (2015). Real time vehicle monitoring and tracking system based on embedded Linux board and android application. In *2015 International Conference on Circuits, Power and Computing Technologies*, 1–7. https://doi.org/10.1109/ICCPCT.2015.7159414

[8] Orun, A. (2022). *Automatic real-time vehicle classification by image colour component based template matching*. arXiv. https://arxiv.org/abs/2210.06586

[9] Velazquez-Pupo, R., Sierra-Romero, A., Torres-Roman, D., Shkvarko, Y. V., Santiago-Paz, J., Gómez-Gutiérrez, D., …, & Romero-Delgado M. (2018). Vehicle detection with occlusion handling, tracking, and OC-SVM classification: A high performance vision-based system. *Sensors*, *18*(2), 374. https://doi.org/10.3390/s18020374

[10] Choi, J.-H., Lee, K.-H., Cha, K.-C., Kwon, J.-S., Kim, D.-W., & Song, H.-K. (2006). Vehicle tracking using template matching based on feature points. In *2006 IEEE International Conference on Information Reuse & Integration*, 573–577. https://doi.org/10.1109/IRI.2006.252477

[11] Han, G., Jin, Q., Rong, H., Jin, L., & Zhang, L. (2023). Vehicle tracking algorithm based on deep learning in roadside perspective. *Sustainability*, *15*(3), 1950. https://doi.org/10.3390/su15031950

[12] Qureshi, A. M., Al Mudawi, N., Alonazi, M., Chelloug, S. A., & Park, J. (2024). Road traffic monitoring from aerial images using

template matching and invariant features. *Computers, Materials & Continua*, *78*(3), 3683–3701. https://doi.org/10.32604/cmc.2024.043611

[13] Patel, N., & Brahmbhatt, K. N. (2023). A video-based system for vehicle tracking based on optical flow and Shi-Tomasi corner detection algorithm. In *Proceedings of Fourth International Conference on Computing, Communications, and Cyber-Security*, 715–723. https://doi.org/10.1007/978-981-99-1479-1_53

[14] Sun, M., Xiao, J., Lim, E. G., Zhang, B., & Zhao, Y. (2020). Fast template matching and update for video object tracking and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10791–10799.

[15] Chantara, W., Mun, J.-H., Shin, D.-W., & Ho, Y.-S. (2015). Object tracking using adaptive template matching. *IEIE Transactions on Smart Processing and Computing*, *4*(1), 1–9. https://doi.org/10.5573/IEIESPC.2015.4.1.001

[16] Ali, A., Ghosh, A., & Chaudhuri, S. S. (2024). Real-time tracking of moving objects through efficient scale space adaptation and normalized correlation filtering. *Signal, Image and Video Processing*, *18*, 679–689. https://doi.org/10.1007/s11760-023-02758-x

[17] Rababaah, A. R. (2022). Deep learning solution for machine vision problem of vehicle body damage classification. *International Journal of Computational Vision and Robotics*, *12*(4), 426–442. https://doi.org/10.1504/IJCVR.2022.123853

[18] Gonzalez, R. C., & Woods, R. E. (2010). *Digital image processing*. UK: Pearson Education.

[19] Aggarwal, J. K., & Xia, L. (2014). Human activity recognition from 3D data: A review. *Pattern Recognition Letters*, *48*, 70–80. https://doi.org/10.1016/j.patrec.2014.04.011

[20] Rababaah, A. R. (2023). Comparative study of deep learning models versus machine learning models for wind turbine intelligent health diagnosis systems. *Arabian Journal for Science and Engineering*, *48*(8), 10875–10899. https://doi.org/10.1007/s13369-023-07810-z

[21] Swain, M. J., & Ballard, D. H. (1991). Color indexing. *International Journal of Computer Vision*, *7*(1), 11–32. https://doi.org/10.1007/BF00130487

[22] Zhang, D., & Lu, G. (2004). Review of shape representation and description techniques. *Pattern Recognition*, *37*(1), 1–19. https://doi.org/10.1016/j.patcog.2003.07.008

[23] Bhattacharyya, A. (1943). On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, *35*, 99–109.

[24] Larose, D. T., & Larose, C. D. (2014). *Discovering knowledge in data: An introduction to data mining*. USA: Wiley.

[25] Rababaah, A. R., & Wolfer, J. (2022). Convolution neural network model for an intelligent solution to crack detection in pavement images. *International Journal of Computer Applications in Technology*, *68*(4), 389–396. https://doi.org/10.1504/IJCAT.2022.125176

How to Cite: Rababaah, A. R. (2026). Robust Histogram Signature-Driven Template Matching for Vehicle Tracking in Traffic Videos. *Journal of Data Science and Intelligent Systems*, *4*(2), 211–223. https://doi.org/10.47852/bonviewJDSIS52025712