Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.043611

ARTICLE

Road Traffic Monitoring from Aerial Images Using Template Matching and Invariant Features

Asifa Mehmood Qureshi[1], Naif Al Mudawi[2], Mohammed Alonazi[3], Samia Allaoua Chelloug[4] and Jeongmin Park[5,*]

[1]Department of Creative Technologies, Air University, Islamabad, 46000, Pakistan

[2]Department of Computer Science, College of Computer Science and Information System, Najran University, Najran, 55461, Saudi Arabia

[3]Department of Information Systems, College of Computer Engineering and Sciences, Prince Sattam bin Abdulaziz University, Al-Kharj, 16273, Saudi Arabia

[4]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

[5]Department of Computer Engineering, Tech University of Korea, Gyeonggi-do, 15073, South Korea

*Corresponding Author: Jeongmin Park. Email: jmpark@tukorea.ac.kr

Received: 07 July 2023 Accepted: 04 December 2023 Published: 26 March 2024

ABSTRACT

Road traffic monitoring is an imperative topic widely discussed among researchers. Systems used to monitor traffic frequently rely on cameras mounted on bridges or roadsides. However, aerial images provide the flexibility to use mobile platforms to detect the location and motion of the vehicle over a larger area. To this end, different models have shown the ability to recognize and track vehicles. However, these methods are not mature enough to produce accurate results in complex road scenes. Therefore, this paper presents an algorithm that combines state-of-the-art techniques for identifying and tracking vehicles in conjunction with image bursts. The extracted frames were converted to grayscale, followed by the application of a georeferencing algorithm to embed coordinate information into the images. The masking technique eliminated irrelevant data and reduced the computational cost of the overall monitoring system. Next, Sobel edge detection combined with Canny edge detection and Hough line transform has been applied for noise reduction. After preprocessing, the blob detection algorithm helped detect the vehicles. Vehicles of varying sizes have been detected by implementing a dynamic thresholding scheme. Detection was done on the first image of every burst. Then, to track vehicles, the model of each vehicle was made to find its matches in the succeeding images using the template matching algorithm. To further improve the tracking accuracy by incorporating motion information, Scale Invariant Feature Transform (SIFT) features have been used to find the best possible match among multiple matches. An accuracy rate of 87% for detection and 80% accuracy for tracking in the A1 Motorway Netherland dataset has been achieved. For the Vehicle Aerial Imaging from Drone (VAID) dataset, an accuracy rate of 86% for detection and 78% accuracy for tracking has been achieved.

KEYWORDS

Unmanned Aerial Vehicles (UAV); aerial images; dataset; object detection; object tracking; data elimination; template matching; blob detection; SIFT; VAID

1 Introduction

As urbanization has intensified, traffic in metropolitan areas has increased significantly, as they have highways or motorways that connect urban areas. Real-time traffic monitoring on freeways could provide advanced traffic data that could be used to make better plans and strategies for transportation in both cities and suburbs [1–3]. Traffic control and flow management demand a quick response to the current event for which traffic data must be acquired in real-time circumstances. There are several ways to collect traffic data which include stationary cameras mounted on roadsides or inductive loops. However, the limitations of these methods include the need for special equipment to be installed. Also, they cannot cover a large road network such as highways [4,5]. On the contrary, Unmanned Aerial Vehicles (UAVs) are powered aircraft that can broadcast real-time, high-resolution aerial images from remote locations at a low cost. Moreover, UAVs provide the flexibility of on-demand traffic monitoring. Methods based on aerial images must be able to deal with low frame rates, different altitudes, and camera movement to produce efficient and accurate results with a low computational cost [6,7]. Although deep learning-based methods have made a lot of progress these methods require specialized hardware along with a large amount of data for training and testing. The computational and memory requirements of deep learning models are very high. Moreover, these models are not completely understandable and controllable which makes it difficult to understand the features being learned to propose future improvements. Therefore, we proposed a novel technique that uses lightweight algorithms and can be used for different traffic scenarios with little modifications.

In this paper, a novel technique for the detection and tracking of vehicles based on aerial images of the highway is presented. For model development, two publicly available datasets, i.e., Traffic flow A1 Beekbergen Deventer and VAID were used. To reduce the overall computational cost, irrelevant data elimination has been done using the masking technique. Then, noise reduction is performed by using a combination of different techniques, i.e., Sobel edge detection to detect edges, Canny edge detection for edge refinement, and then applying the Hough line transform to detect the road lane markings. These edge-detection methods are still widely used in object-detection applications. They can detect refined edges by focusing on the minor details of the object [8,9]. The detected lines are then removed from the images leaving behind only the vehicles. The vehicles are then detected by using the blob detection technique based on the convexity, color, size, and inertia of the blobs. After vehicles were found in the image, each object found in the next frame was matched to a template using the template matching algorithm to find out where the vehicle might be. Furthermore, Scale Invariant Feature Transform (SIFT) was applied to the vehicle model to find SIFT features and its corresponding patches were found across image frames to get the best possible match and to reduce false positives. The main contribution of our proposed model includes:

- The blob detection technique has been combined with different thresholding schemes to detect vehicles of varying size and having a minute color difference between the background and vehicles.
- Template matching combined with SIFT features is used to track vehicles across bursts of frames.
- The proposed model is lightweight and computationally less expensive with the least memory requirements which makes it efficient for aerial data.
- The models provide a clear insight of all the techniques being implemented which allows the flexibility to improve further.

The paper has been divided into sections as it includes sections on related work, materials, and methodologies, which illustrate an overview of our proposed technique and a comprehensive study including data elimination, vehicle detection, matching, and finally vehicle tracking. Then, the results section presents the experiments conducted and the outcomes of the system, and finally, the paper concludes with an overall synopsis.

2 Related Work

A variety of advanced vehicle detection and tracking analysis approaches have been studied. Table 1 presents a literature review of existing traffic monitoring models which are mainly based on image processing and machine learning techniques.

Table 1: Literature review for existing traffic monitoring models

State-of-the-art models	Main contributions	Limitations
[5]	A traffic monitoring model has been proposed which is based on the background image subtraction method. The detection of the vehicle is done using the center, length, and width of the vehicle. Whereas the ground control points are used to track them.	The background image subtraction method is considered a limitation of the generalizability of the model.
[6]	Using readily available UAV systems and a standard laptop computer, a framework has been developed that can accurately detect, track, and classify multiple moving objects in real time. The framework can also provide useful information about the detected objects, such as their coordinates and velocities.	Although this study was restricted to a particular height, it demonstrated a useful and reliable strategy for identifying and following moving objects in moving films.
[7]	Multiple unmanned aerial vehicles were used in the design of a system to monitor traffic. In addition, a technique for creating adaptive UAV trajectories was created, and it is based on tracking moving objects in the UAV's field of view. The same mobility models that are often used to model the mobility of vehicles were also employed to construct the UAV's trajectory.	The identification of a single low-speed vehicle serves as the foundation for the analysis of congestion events in this research. By defining a traffic gridlock, identification can be improved.

(Continued)

Table 1 (continued)

State-of-the-art models	Main contributions	Limitations
[10]	A technique to detect unwanted traffic flows in the IoT network. The model CorrAUC focuses on the appropriate selection of features to detect malicious traffic flows.	However, the accuracy of the model is good, but the computational time of the model is high as it needs Area Under Curve (AUC) to be calculated for each feature and selecting the best features.
[11]	The system exploits the Red, Green, and Blue (RGB) as well as the grayscale properties of the aerial images. Colors having different colors were detected using a color differencing approach. Whereas cars that have an appearance similar to road color were detected by dynamic thresholding. Detected vehicles were tracked by using a shape-based matching algorithm.	The color differencing approach is not suitable where the background is complex.
[12]	The research presents a vehicle detection algorithm in aerial images. The detection algorithm implements Harris corner extraction. Feature density was estimated by searching through a rectangular window based on a thresholding scheme. Further, the cars were classified using a histogram of gradient and a histogram of Gabor filters which were passed to Support Vector Machine (SVM), K Nearest Neighbor (KNN), and Random Forest classifiers.	The algorithm does not contain any method to deal with the dimensionality reduction of features when the density of vehicles on the road is high.

(Continued)

Table 1 (continued)

State-of-the-art models	Main contributions	Limitations
[13]	The paper proposed a model for vehicle detection and tracking in urban aerial images. A spatial pyramid context-aware (SPCT) feature-based tracker, motion prediction based on the Kalman filter, and motion detection tracker based on a median motion mask were used to implement the model. Moreover, feature fusion techniques were used for target localization.	The method proposed cannot be applied to large datasets. Also, the SPCT tracker implements the color of histogram technique which does not produce favorable results for colored vehicles.
[14]	A new method of estimating traffic flow parameters with improved vehicle recognition from aerial recordings was put forth. The Kanade-Lucas-Tomasi (KLT) tracker, Support Vector Machine (SVM), and connected graphs are all used in the suggested solution.	The dataset's size was only 100 photos. When compared to deep learning techniques, handcrafted features reduce the system's accuracy, which results in inadequate learning.
[15]	This paper presented a method for highway traffic monitoring using medium-resolution satellite images. After road masking, a median filter was applied to the image and then the image difference with the background image was calculated. Then a gray level of the resulting image was computed. In the last step, bright and large regions were detected by using a thresholding scheme which was classified as cars.	The thresholding scheme based on bright regions is not suitable for images having low illumination conditions.
[16]	To find the traffic density on the road a model based on temporal variance and distance transform was presented. The model produces reasonable results even at night time. The model complexity is low.	High-density regions were not marked on the image. Also, car counts were not given to make comparisons of different density ratings.

(Continued)

Table 1 (continued)

State-of-the-art models	Main contributions	Limitations
[17]	Multiple vehicle tracking models based on extended Kalman filter, HOG, and SVM for vehicle detection and viewpoint estimation were presented. Pose estimation was done using the Region-based Convolutional Neural Network (R-CNN) deep learning model.	Viewpoint estimation is not feasible when the altitude of the camera increases or when the illumination condition changes.
[6]	The authors proposed a traffic monitoring method. For vehicle detection, the model used image differencing background updation, road division, and lane marking estimation methods on video frames. The tracking of the vehicle was done using a Kalman filter.	An image differencing method is not an efficient approach for large and complex datasets as it requires the intermediate result to be stored in memory and also results in the information loss of stationary cars.
[18]	The model proposed in this research article used SIFT features for vehicle detection and tracking. The detection of the vehicle was done using the difference of aligned images and then calculating the SAD value. The tracking of the vehicle was accomplished by matching the SIFT features in different frames.	The calculation of the sum of absolute difference values can falsify the detection of vehicles when there is a large number of vehicles or in case of occlusion.
[6]	A multi-UAV-based crowd-monitoring system employing UAVs to routinely track individuals in motion has been developed. To efficiently monitor all targets, the suggested system can divide the UAVs' target monitoring responsibilities across multiple drones.	The proposed method has only been tested through simulation, does not support diverse tasks, and does not address practical perceptual concerns such as target recognition, target tracking, and re-identification, or practical behavior concerns such as collision avoidance and obstacle avoidance.

(Continued)

Table 1 (continued)

State-of-the-art models	Main contributions	Limitations
[19]	An approach for vehicle detection and tracking implements color, edge, and motion information to detect lane markings, road lanes, and other vehicles on the road whereas cars were detected using template matching. Tracking of the vehicle was performed by using the recursive least square filter.	The proposed model utilized template matching for car detection which is not an efficient and robust approach when the image contains a large number of different car models.
[20]	They proposed a method based on k-mean clustering and image differencing techniques. Vehicle tracking was based on centroidal position coordinates and different values.	The background information was updated if a change occurred in it which limits the applicability of the model to multiple road scenarios.
[21]	In this article, the authors presented a real-time object detection and tracking algorithm. the model combines am improved RefineDet based detection module. Also, it uses harmony search algorithm with a twin support vector machine model is used for classification.	The model lacks pre-processing of the data to reduce the overall computational complexity of the model.

Table 1 shows that traffic monitoring is still a challenging task. There are various methods implemented using machine learning and deep learning methodologies. Deep learning models are hardware as well as computationally expensive. Therefore, we combined different machine learning models to improve the accuracy of vehicle detection and tracking. Multiple vehicle detection and tracking using UAV video frames on complex roads with different backgrounds and illumination conditions, heavy traffic, and low image resolution make it hard for the model to be as accurate as it could be.

3 Proposed Method

Data has been collected from two publicly available datasets, namely, the VAID dataset and the Traffic flow A1 Beekbergen Deventer dataset. After data collection, image pre-processing techniques which include gray scale conversion and geospatial referencing have been applied followed by irrelevant data elimination techniques. After eliminating irrelevant data from the images using the mask elimination technique and Sobel, Canny, and Hough transform, vehicles were detected using the blob detection algorithm. The detected vehicles were matched across the image frames to get the possible match of each vehicle. To increase the tracking algorithm efficiency and to reduce the false positive rate for tracking purposes. The matching vehicles were also subjected to SIFT feature matching to get improved results. Fig. 1 shows the overall workflow methodology of the proposed model.

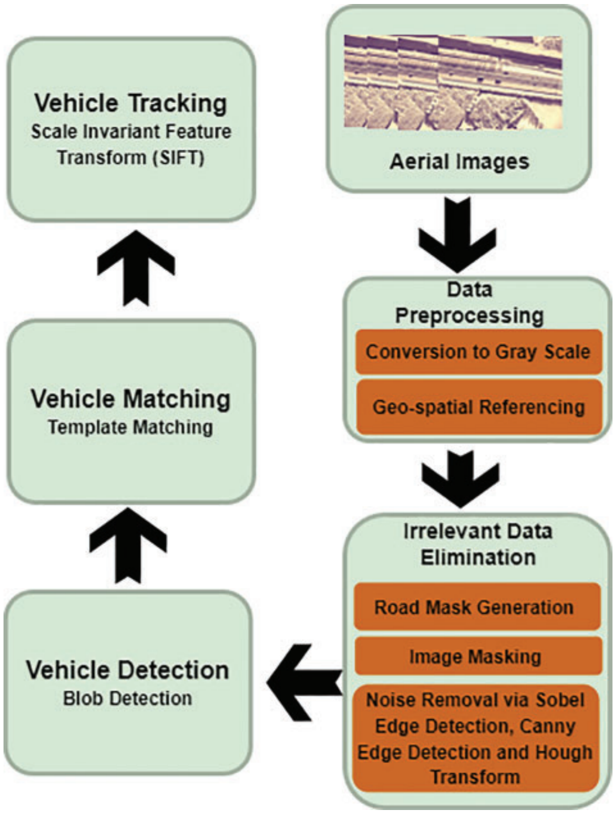


Figure 1: Flowchart demonstrating the proposed traffic surveillance model

3.1 Image Pre-Processing

The raw images must undergo pre-processing to remove noise, irrelevant data, and other shortcomings before passing it to the detection phase. For this, all the images were converted to a grayscale, and then georeferencing was performed.

3.1.1 Gray-Scale Conversion

As the original images of the dataset contain only gray shades; therefore, before detecting the vehicles all the images were converted to a gray-scale image. Each pixel's value in a grayscale image is a single sample containing data about the intensity of the light [22]. The weighted method was used for the conversion as described in Eq. (1).

$$GrayScale = 0.299R + 0.587G + 0.114B \tag{1}$$

where R is Red, G is green, and B is blue.

3.1.2 Noise Removal

Georeferencing is the process of assigning coordinates to geographical objects inside a frame of reference [23]. The coordinates of each image were approximated by using the Keyhole Markup

Language (KML) file of the dataset. It was implemented by using the Geospatial Data Abstraction Library (GDAL) Python. The images with coordinates information embedded were saved with the .tiff extension. Georeferencing of images was performed by using Eqs. (2) and (3).

$$x = a + bi + cj \tag{2}$$

$$y = d + ei + fj \tag{3}$$

where image coordinates *(i, j)* are transformed into map coordinates *(x, y)* using *(a-f)* ground control points (GCPs). The result can be visualized in Fig. 2.

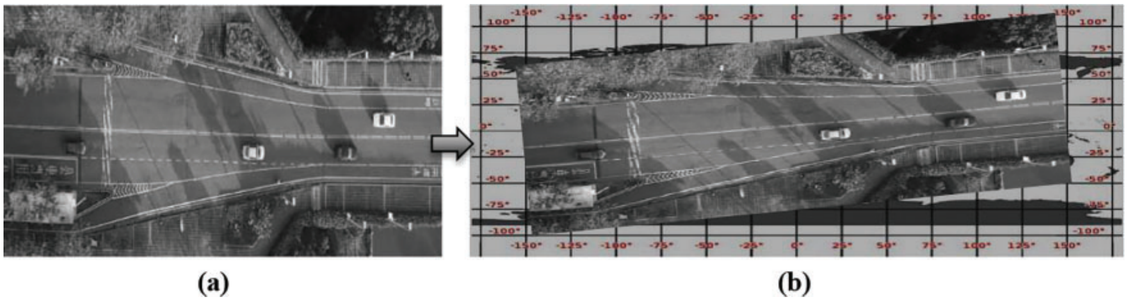


Figure 2: Visualization (a) gray-scale image and (b) georeferenced image

3.2 *Irrelevant Data Elimination*

The pre-processed data contains a lot of areas that are not useful for further processing and will also increase the computational complexity of the proposed model. The road masking technique was used to address this issue and to pass only the relevant area to the next phase, the road masking technique was used. Further, the noise was removed using a combination of the Sobel edge detector and Hough line transform.

3.2.1 Road Mask Generation

To eliminate areas other than the road, the masking technique was used. A binary image of zero- and non-zero values is referred to as a mask. For this purpose, manual masks were generated as an image which was used for further image processing.

3.2.2 Image Masking

In the image masking technique, when a mask is applied to another binary or grayscale image of the same size, all the pixels that are zero in the mask are set to zero in the output image. The rest of the pixels remain unaltered [24]. The output image can be obtained by simply multiplying the two input images as shown in Eq. (4).

$$Q(i,j) = P_1(i,j) \times P_2(i,j) \tag{4}$$

where $P_1(i,j)$ represents the original image, $P_2(i,j)$ denotes image mask, and $Q(i,j)$ is the resultant image. The result of the image masking technique can be seen in Fig. 3.

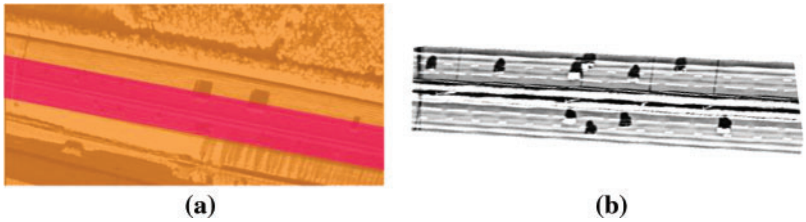


Figure 3: Visualization (a) manually created image masks to eliminate irrelevant areas and (b) resultant images after irrelevant areas are removed

3.2.3 Noise Reduction

To remove the noise from the image, i.e., road markings and lanes Sobel edge detection, Canny edge detection, and Hough line transform techniques were applied. First of all, a median filter was applied to eradicate unnecessary details from the image A median filter is a nonlinear filter that computes each output sample as the median value of the input samples inside the window; the outcome is the middle value following the sorting of the input values as seen in Eq. (5).

$$I'(u,v) \leftarrow \text{median}\,\{I(u+i,v+j) \mid (i,j) \in R\} \tag{5}$$

where (u,v) are the image coordinates and (i,j) are the filter co-efficients in moving region R. The gradient magnitude can be approximated using the Sobel method. The Sobel operator can recognize edges and their directions [25]. It helped retain the information about very small-sized vehicles and bikes in the image frames. The approximation of the gradient magnitude in this cross operator is believed to make it easy to recognize edges and their orientations as calculated by using Eqs. (6) and (7).

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{6}$$

$$\Theta = tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{7}$$

where the partial derivatives in the y and x directions are represented by G_y and G_x.

To find edges, the Sobel edge filter employs a horizontal and vertical filter. After applying both filters to the image, the results were merged. Eq. (8) shows the convolution filters.

$$x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} and\; y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{8}$$

For edge refinement, Canny edge detection was applied. A multi-stage method is used by the Canny edge detector to find a variety of edges in images. The false edges are suppressed by using non-maxima suppression and double thresholding methods [26]. The edges are calculated by using Eq. (9).

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}} \tag{9}$$

The next step, after edge detection, involves the removal of road lanes and markings to get all the vehicles left behind. To detect the lines, the Hough line transform was applied. To focus on certain shapes (vehicles) inside an image, the Hough transform was utilized. Since the desired features must be provided in some parametric form, the classical Hough transform is usually used for the identification of regular curves like lines, circles, ellipses, etc. [27]. For the Hough line transform, the lines must be expressed in the polar system as shown in Eq. (10).

$$y = \left(-\frac{\cos\theta}{\sin\theta}\right)x + \left(\frac{r}{\sin\theta}\right) \tag{10}$$

For each given line values of ρ and Θ are determined. Then, any x, y point along this line fulfills Eq. (11).

$$\rho = x\cos\Theta + y\sin\Theta \tag{11}$$

where Θ is the angle formed by the line's perpendicular to the x-axis and ρ is the line's distance from the origin.

The steps of the noise reduction algorithm can be seen in Fig. 4.

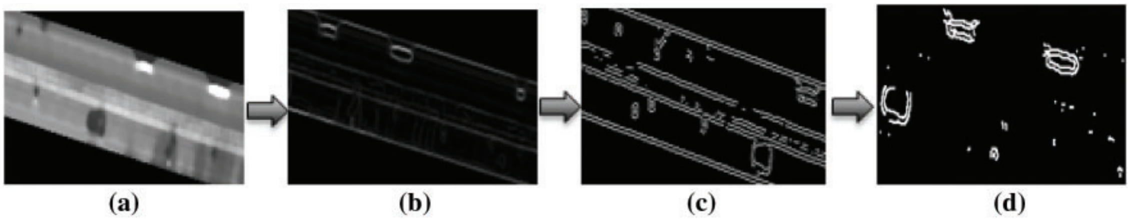


Figure 4: The result of noise reduction algorithm (a) output of the median filter (b) Sobel edge detection (c) Canny edge detection (d) Hough line transforms for noise reduction

3.3 *Blob Detection*

Before blob detection, the image frames were divided into bursts of 5 images. The first image of each burst was used for detection while tracking was done on the other 4 images. The detection of vehicles was accomplished by a blob detection algorithm [28]. The images were dilated to improve the visibility of the blobs by using Eq. (12). Dilation increases the boundaries of the regions of the foreground pixels [29].

$$X_k = \text{ dilate } (X_{k-1}, J) \cap A_{\text{not}} \tag{12}$$

where J is the structuring element, A_{not} is the boundary's negative, and X_k is the region that fills the boundary after convergence.

Then, the blob detection algorithm was applied with a minimum threshold value of area equal to 40 and the maximum threshold was set to 300. Blobs can be defined as a collection of pixel values that resembles a colony or as a sizable object that stands out against the background [30]. The blobs were also filtered by circularity, circularity, inertia, and distance between the blobs. The output of the blob detection algorithm is shown in Fig. 5.

3.4 *Template Matching*

Before performing vehicle tracking, the vehicles detected in the burst's initial image are geometrically set up. It should be noted that not all cars can be designed geometrically. The position of the vehicle is determined using the explicit shape models, which are then utilized in template matching to find the possible match in the other two images of the burst. The template matching process comprises panning the template across the entire image and assessing how much it resembles the covered window in the image. To evaluate the similarity, the normalized correlation coefficient was used as shown in Eq. (13).

$$R\left(x,y\right)=\frac{\Sigma_{x',y'}\left(T'\left(x',y'\right).I'\left(x+x',y+y'\right)\right)}{\sqrt{\Sigma_{x',y'}\left(T'\left(x',y'\right)^{2}.I'\left(x+x',y+y'\right)^{2}\right)}} \tag{13}$$

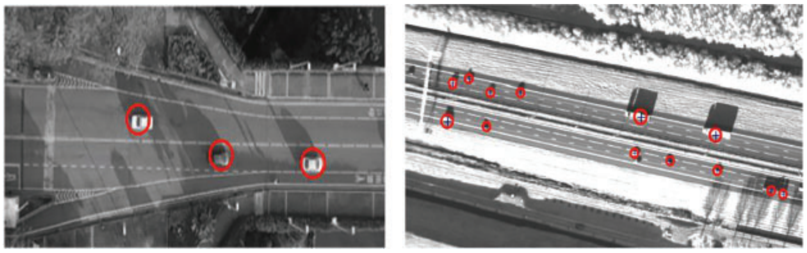


Figure 5: Vehicle detection using a blob detection algorithm

The matrix R contains the computed result metric for each location T' over I'. The result matrix R has an entry for each (x, y)-coordinate in the original image which shows the degree of matches. Every time the similarity measure exceeds a predetermined threshold, a match is discovered. Different thresholding settings, i.e., 0.5 and 0.7 were employed in the case of a car or truck respectively to reduce the number of false positives. The output of template matching can be visualized in Fig. 6.

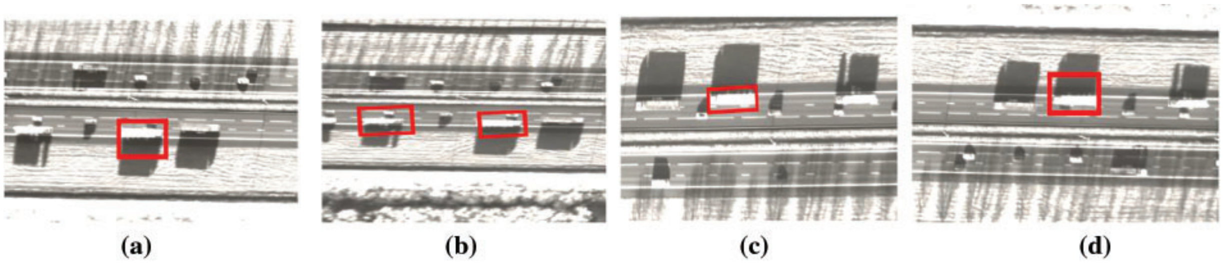


Figure 6: Vehicle tracking based on template matching algorithm across image bursts (a) Frame 1 (b) Frame 2 (c) Frame 3 and (d) Frame 4

3.5 Vehicle Tracking

The output of the template matching algorithm contains more than one matching for a large number of vehicles. To improve the accuracy and to get only one best-match Scale, Invariant Feature Transform (SIFT) features were extracted for motion estimation and filtering. To get a reliable match of the object within the images, SIFT was designed to extract highly identifiable invariant features from image bursts [31]. The SIFT method primarily involves four phases.

3.5.1 Scale-Space Extrema Prediction

The difference of Gaussians used by the SIFT method approximates the Laplacian of Gaussian. The feature points are found by using the difference of Gaussian (DoG) at different scales of the image to look for local maxima. If the input image is $f(x, y)$, then the associated scale space is defined as shown in Eq. (14).

$$L(x, y, \sigma) = G(x, y, \sigma)^{*}f(x, y) \tag{14}$$

When an image is blurred with two different σ, let them be kσ and σ, the difference of Gaussian is obtained by using the formula in Eq. (15). This procedure is carried out for several octaves of the Gaussian Pyramid image. Images are then explored for local extrema oversize and space.

$$G(x, y, k\sigma) = \frac{1}{2\pi(k\sigma)^2}e^{-\frac{x^2+y^2}{2k^2\sigma^2}} \tag{15}$$

where x, y represents the image coordinates.

3.5.2 Keypoint Localization

To acquire more precise findings, possible key point locations must be revised after being discovered. They expanded the scale space using the Taylor series (Eq. 16) to locate the extrema with more accuracy, and if the intensity at these extrema is less than a certain threshold, it is rejected.

$$\sum\nolimits_{n=0}^{\infty}\frac{f^{(n)}(a)}{n!}(x-a)^n \tag{16}$$

3.5.3 Orientation Assignment

Now, each key point has an orientation given to it to achieve image rotation invariance. Depending on the scale, a neighborhood is selected around the keypoint position, and the gradient's amplitude and direction are determined as given in Eqs. (17) and (18).

$$|I| = \sqrt{I_x^2 + I_y^2} \tag{17}$$

$$\Theta = tan^{-1}\left(\frac{I_y}{I_x}\right) \tag{18}$$

where I_x and I_y represent the two descriptor coordinates.

3.5.4 Keypoint Descriptor

Around the key point, a 16 × 16 neighborhood was taken. It was divided into 16 subblocks of 4 × 4 size each. An 8-bin orientation histogram was produced for each sub-block. There are 128 possible bin values in all. To create a key point descriptor, it can be visualized as a vector.

3.5.5 Keypoint Matching

Keypoint between two images are matched by measuring the Euclidean distance as calculated in Eq. (19). A thresholding scheme described in Eq. (20) is used to get the best possible match.

$$d(p, q) = \sqrt{\sum\nolimits_{i=1}^{n}(q_i - p_i)^2} \tag{19}$$

where q_i and p_i are the keypoints of the images.

$$if\,\frac{distance\ to\ first\ match}{distance\ to\ second\ match} < 0.8 \text{ then a match was found} \tag{20}$$

The SIFT feature matching can be visualized in Fig. 7.

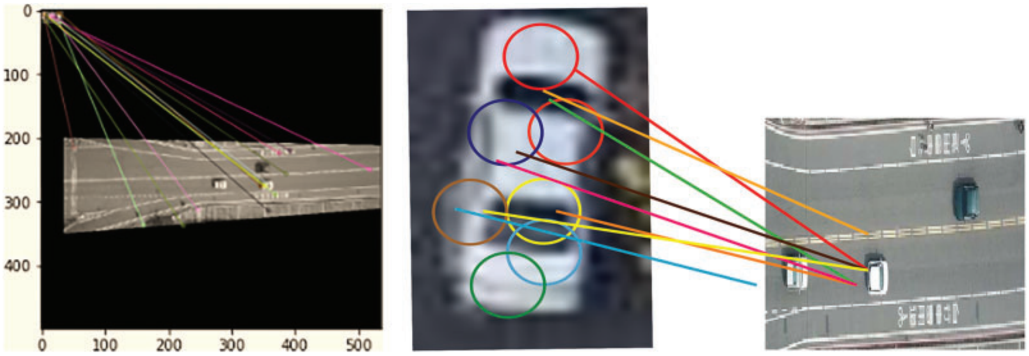


Figure 7: SIFT feature matching for vehicle tracking

4 Performance Evaluation

4.1 Dataset Description

The Traffic flow A1 Beeksbergen Deventer dataset has been collected along with geolocation, latitude, and longitude. Each video contains 2000 frames which were extracted in .png format. It has been created to monitor the traffic on Highway A1 (91.7 km) in the Netherlands. 2000 image frames were used to evaluate the proposed model.

The Vehicle Aerial Imaging from Drone (VAID) dataset is based on 5985 aerial images taken from drones at different angles and illumination conditions in Taiwan. All the images have a resolution of 1137 × 640 pixels in .jpg format [32]. The frames from both datasets were divided into bursts of 5 images. Detection was done on the very first image of each burst whereas the tracking was done in the succeeding four images.

4.2 Experimental Settings and Results

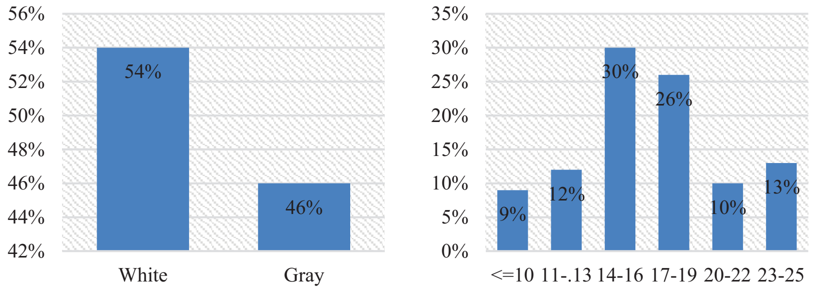
The system was trained and assessed using a PC with an Intel Core i7 and 64-bit Windows 10, and Python (3.7). The computer has a 5 Giga Hertz (GHz) CPU and 16 GB of Random Access Memory (RAM). The effectiveness of the system is tested by using the above-mentioned datasets.

4.2.1 Experiment I: Vehicle Count and Type Analysis over Image Frames

An analysis of both datasets with respect to vehicle color and vehicle count in each image frame is presented. It helps to select and improve appropriate light-weight method to detect vehicles as per the color difference from the background and number of vehicles in each frame with an acceptable precision rate. Fig. 8 illustrates vehicle color type and vehicle count of the dataset Traffic flow A1 Beekbergen Deventer.

4.2.2 Experiment II: Comprehensive Analysis of Selected Datasets

Both the Traffic Flow A1 Beekbergen Deventer and the VAID datasets have been used in the experiment to test the outcomes in terms of true positives, false positives, false negatives, precision, recall, and F1-scores. False positives are non-vehicle detections, true positives are the number of vehicles marked, and false negatives are missed vehicle detections. For tracking, the number of successfully tracked cars is known as the true positives, the number of inaccurately tracked vehicles is known as the false positives, and the number of detected but untracked vehicles is known as the false negatives (vehicles not tracked due to occlusion are not counted as false positives. Table 2 shows the outcomes for the detection algorithm over both the Traffic flow A1 Beekbergen Deventer and VAID datasets.



(a) **(b)**

Figure 8: Distribution of traffic flow A1 Beekbergen Deventer (a) vehicle color type (b) vehicles per image

Table 2: Detection precision, recall, and F1-score results

Datasets	Images evaluated	TP	FP	FN
Traffic flow A1 Beekbergen Deventer	400	1109	162	386
VAID	144	871	146	174

Table 3 represents the results of the tracking algorithm over both datasets.

Table 3: Tracking precision, recall, and F1-score results

Datasets	Images evaluated	Cars geometrically configured	Precision	Recall	F1-score
Traffic flow A1 Beekbergen Deventer	1600	896	0.80	0.77	0.78
VAID	576	720	0.78	0.79	0.78

4.2.3 Experiment III: Comparison with Other State-of-the-Art Models

In this experiment, we compared our model with state-of-the-art models. For the region of interest extraction from the images, frame differencing or background image differencing techniques produce reasonable results, but these techniques are not flexible and do not produce consistent results on other datasets. Moreover, differencing methods are not available for stationary vehicles. Also, these techniques increase the computational and memory requirements of the overall model. For vehicle detection, a comparison of k-mean clustering and blob detection was used. K-mean clustering is sensitive to noise which hinders the accuracy of the model. Blob detection implements thresholding criteria which helps to overlook any noise left behind after the pre-processing phase. Template matching is the simplest and most commonly used tracking algorithm. but evaluating its preciseness, it results in more than one match of a car with a low threshold value or no match if the threshold value for the matching score increases. Therefore, to get reasonable results for matching and to further reduce the number of false positives, a combination of template matching with Histogram of Gradient (HOG) and SIFT was evaluated. HOG was not suitable for the motion configuration of very small-sized vehicles or bikes. However, this limitation can be minimized by using invariant features i.e., SIFT algorithm. However, the Convolutional Neural Network (CNN) has a better preciseness rate than our proposed method, but its computational time and memory consumption have significantly increased. Table 4 presents a comparison of our proposed model results over Traffic flow A1 Beekbergen Deventer with the other state-of-the-art systems.

Table 4: Comparison with the other conventional algorithms over A1 traffic flow A1 Beekbergen Deventer dataset

Pre-processing techniques	Detection algorithm	Tracking algorithm	Detection preciseness (%)	Tracking preciseness (%)
Masking with street segments	K-mean clustering	Template matching	80	66.6
Consecutive frame differencing	K-mean clustering	Template matching	75	68.2
Masking	CNN	Centroid tracking	89	74
Masking	**Blob detection**	**Template matching + SIFT**	**87**	**80**

Table 5 presents a comparison of VAID dataset results with other techniques. It can be seen that our proposed model tracks the vehicle more precisely when compared with other state-of-the-art algorithms.

Table 5: Comparison with the other conventional algorithms over VAID

Pre-processing techniques	Detection Algorithm	Tracking algorithm	Detection preciseness (%)	Tracking preciseness (%)
Background image subtraction	Blob detection	Template matching + HOG	82	72
Masking	Blob detection	Template matching + HOG	86	74.9
Masking	CNN	Centroid tracking	90	77
Masking	**Blob detection**	**Template matching + SIFT**	**86**	**78**

5 Conclusion and Future Works

The proposed traffic monitoring model is based on masking, noise reduction, blob detection for pattern recognition, and template matching combined with SIFT for vehicle tracking. To evaluate the performance of the proposed model, two datasets, namely, the traffic flow A1 Beekbergen Deventer dataset and VAID were used. To remove the irrelevant area and noise masking a state-of-the-art technique was proposed along with the Hough line transform method. For the tracking of vehicles, the research built and employed car models in template matching. Besides, SIFT, features helped get the best possible match for each vehicle, which makes the results more accurate. Future work includes methods to deal with different illumination conditions for higher accuracy.

Acknowledgement: The authors are thankful to Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R54), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: This work was supported by a grant from the Basic Science Research Program through the National Research Foundation (NRF) (2021R1F1A1063634) funded by the Ministry of Science and ICT (MSIT), Republic of Korea. The authors are thankful to the Deanship of Scientific Research at Najran University for funding this work under the Research Group Funding Program Grant Code (NU/RG/SERC/13/40). Also, the authors are thankful to Prince Satam bin Abdulaziz University for supporting this study via funding from Prince Satam bin Abdulaziz University project number (PSAU/2024/R/1445). This work was also supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2023R54), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: Study conception and design: Asifa Mehmood Qureshi, Jeongmin Park, data collection: Mohammed Alonazi; analysis and interpretation of results: Asifa Mehmood Qureshi, Naif Al Mudawi and Samia Allaoua Chelloug; draft manuscript preparation: Asifa Mehmood Qureshi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: All publicly available datasets are used in the study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. H. Yu, J. W. Hsieh, Y. S. Chen, and W. F. Hu, "An automatic traffic surveillance system for vehicle tracking and classification," *Lect. Notes Comput. Sci.*, vol. 2749, pp. 379–386, 2003. doi: 10.1007/3-540-45103-X.

[2] A. Azim and O. Aycard, "Detection, classification and tracking of moving objects in a 3D environment," in *2012 IEEE Intell. Vehicles Symp.*, Alcala de Henares, Spain, 2012, pp. 802–807.

[3] A. Ahmed, A. Jalal, and K. Kim, "Multi-objects detection and segmentation for scene understanding based on Texton forest and kernel sliding perceptron," *J. Electr. Eng. Technol.*, vol. 16, no. 2, pp. 1143–1150, 2021. doi: 10.1007/s42835-020-00650-z.

[4] A. Angel, M. Hickman, P. Mirchandani, and D. Chandnani, "Methods of analyzing traffic imagery collected from aerial platforms," *IEEE. Trans. Intell. Transp. Syst.*, vol. 4, no. 2, pp. 99–107, 2003. doi: 10.1109/TITS.2003.821208.

[5] M. Schreuder, S. P. Hoogendoorn, H. J. van Zulyen, B. Gorte, and G. Vosselman, "Traffic data collection from aerial imagery," *IEEE. Trans. Intell. Transp.*, vol. 1, pp. 779–784, 2003. doi: 10.1109/ITSC.2003.1252056.

[6] R. S. de Moraes and E. P. de Freitas, "Multi-UAV based crowd monitoring system," *IEEE. Trans. Aero. Electron. Syst.*, vol. 56, no. 2, pp. 1332–1345, Apr. 2020. doi: 10.1109/TAES.2019.2952420.

[7] M. Elloumi, R. Dhaou, B. Escrig, H. Idoudi, and L. A. Saidane, "Monitoring road traffic with a UAV-based system," in *IEEE Wirel. Commun. Netw. Conf. (WCNC)*, Barcelona, Spain, Jun. 2018, pp. 1–6.

[8] D. R. D. Varma and R. Priyanka, "Performance monitoring of novel iris detection system using sobel algorithm in comparison with canny algorithm by minimizing the mean square error," in *2022 3rd Int. Conf. Intell. Eng. Manage. (ICIEM)*, London, UK, 2022, pp. 509–512.

[9] J. S. Yadav and P. Shyamala Bharathi, "Edge detection of images using prewitt algorithm comparing with sobel algorithm to improve accuracy," in *2022 3rd Int. Conf. Intell. Eng. Manage. (ICIEM)*, London, UK, 2022, pp. 351–355.

[10] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A malicious Bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3242–3254, 2021. doi: 10.1109/JIOT.2020.3002255.

[11] S. Hinz, D. Lenhart, and J. Leitloff, "Detection and tracking of vehicles in low framerate aerial image sequences," *Image*, vol. 24, pp. 1–6, 2006.

[12] J. Gleason, A. V. Nefian, X. Bouyssounousse, T. Fong, and G. Bebis, "Vehicle detection from aerial imagery," in *2011 IEEE Int. Conf. Robot. Autom.*, Shanghai, China, 2011, pp. 2065–2070.

[13] M. Poostchi, K. Palaniappan, and G. Seetharaman, "Spatial pyramid context-aware moving vehicle detection and tracking in urban aerial imagery," in *2017 14th IEEE Int. Conf. Adv. Video and Signal Based Surveill. (AVSS)*, Lecce, Italy, 2017, pp. 1–6.

[14] K. V. Najiya and M. Archana, "UAV video processing for traffic surveillence with enhanced vehicle detection," in *2018 Second Int. Conf. Inven. Commun. Comput. Technol. (ICICCT)*, Coimbatore, India, 2018, pp. 662–668.

[15] S. Drouyer and C. Franchis, "Highway traffic monitoring on medium resolution satellite images," in *IGARSS 2019-2019 IEEE Int. Geosci. Remote Sens. Symp.*, Yokohama, Japan, 2019, pp. 1228–1231.

[16] D. Sitaram, N. Padmanabha, S. Supriya, and S. Shibani, "Still image processing techniques for intelligent traffic monitoring," in *2015 Third Int. Conf. Image Inform. Process. (ICIIP)*, Waknaghat, India, 2015, pp. 252–255.

[17] R. J. López-Sastre, C. Herranz-Perdiguero, R. Guerrero-Gómez-olmedo, D. Oñoro-Rubio, and S. Maldonado-Bascón, "Boosting multi-vehicle tracking with a joint object detection and viewpoint estimation sensor," *Sens.*, vol. 19, no. 19, pp. 1–24, 2019.

[18] K. Mu, F. Hui, and X. Zhao, "Multiple vehicle detection and tracking in highway traffic surveillance video based on sift feature matching," *J. Inform. Process. Syst.*, vol. 12, no. 2, pp. 183–195, 2016. doi: 10.3745/JIPS.02.0040.

[19] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Mach. Vision Appl.*, vol. 12, no. 2, pp. 69–83, 2000. doi: 10.1007/s001380050126.

[20] A. N. Rajagopalan and R. Chellappa, "Vehicle detection and tracking in video," in *Proc. 2000 Int. Conf. Image Process.*, Vancouver, Canada, 2000, vol. 1, pp. 351–354.

[21] M. F. Alotaibi, M. Omri, S. Abdel-Khalek, E. Khalil, and R. F. Mansour, "Computational intelligence-based harmony search algorithm for real-time object detection and tracking in video surveillance systems," *Mathematics.*, vol. 10, no. 5, pp. 733, Feb. 2022. doi: 10.3390/math10050733.

[22] C. Saravanan, "Color image to grayscale image conversion," in *2nd Int. Conf. Comput. Eng. Appl. ICCEA*, Bali, Indonesia, 2010, vol. 2, pp. 196–199.

[23] A. Hackeloeer, K. Klasing, J. M. Krisp, and L. Meng, "Georeferencing: A review of methods and applications," *Annals GIS.*, vol. 1, no. 20, pp. 61–69, 2014. doi: 10.1080/19475683.2013.868826.

[24] S. Sen, M. Dhar, and S. Banerjee, "Implementation of human action recognition using image parsing techniques," in *Emerg. Trends in Electron. Dev. Comput. Tech.*, Kolkata, India, Jul. 2018, pp. 1–6.

[25] A. A. Kumar, N. Lal, and R. N. Kumar, "A comparative study of various filtering techniques," in *2021 5th Int. Conf. Trends in Electron. Inform. (ICOEI)*, Tirunelveli, India, Jun. 2021, pp. 26–31.

[26] L. Yuan and X. Xu, "Adaptive image edge detection algorithm based on canny operator," in *2015 4th Int. Conf. Adv. Inform. Technol. Sens. Appl. (AITS)*, Harbin, China, Feb. 2016, pp. 28–31.

[27] D. H. H. Tingting, "Transmission line extraction method based on Hough transform," in *Proc. 2015 27th Chin. Control Decis. Conf. CCDC 2015*, Qingdao, China, Jul. 2015, pp. 4892–4895.

[28] X. Luo, Y. Wang, B. Cai, and Z. Li, "Moving object detection in traffic surveillance video: New MOD-AT method based on adaptive threshold," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 11, pp. 742, Nov. 2021. doi: 10.3390/ijgi10110742.

[29] A. Soni and A. Rai, "Automatic colon malignancy recognition using sobel & morphological dilation," in *2020 Third Int. Conf. Multimed. Process., Commun. & Inform. Technol. (MPCIT)*, Shivamogga, India, Dec. 2020, pp. 63–68.

[30] A. Goshtasby, "Template matching in rotated images," *IEEE. Trans. Pattern. Anal.*, vol. 7, no. 3, pp. 338–344, 1985. doi: 10.1109/TPAMI.1985.4767663.

[31] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, "SIFT features tracking for video stabilization," in *14th Int. Conf. Image Anal. Process. (ICIAP)*, Modena, Italy, 2007, pp. 825–830.

[32] H. Y. Lin, K. C. Tu, and C. Y. Li, "VAID: An aerial image dataset for vehicle detection and classification," *IEEE Access*, vol. 8, pp. 212209–212219, 2020. doi: 10.1109/ACCESS.2020.3040290.