Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.052582

ARTICLE

Semantic Segmentation and YOLO Detector over Aerial Vehicle Images

Asifa Mehmood Qureshi[1], Abdul Haleem Butt[1], Abdulwahab Alazeb[2], Naif Al Mudawi[2], Mohammad Alonazi[3], Nouf Abdullah Almujally[4], Ahmad Jalal[1] and Hui Liu[5,*]

[1]Faculty of Computing and AI, Air University, Islamabad, 44000, 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 System, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

[5]Cognitive Systems Lab, University of Bremen, Bremen, 28359, Germany

*Corresponding Author: Hui Liu. Email: hui.liu@uni-bremen.de

Received: 07 April 2024 Accepted: 18 June 2024 Published: 15 August 2024

ABSTRACT

Intelligent vehicle tracking and detection are crucial tasks in the realm of highway management. However, vehicles come in a range of sizes, which is challenging to detect, affecting the traffic monitoring system's overall accuracy. Deep learning is considered to be an efficient method for object detection in vision-based systems. In this paper, we proposed a vision-based vehicle detection and tracking system based on a You Look Only Once version 5 (YOLOv5) detector combined with a segmentation technique. The model consists of six steps. In the first step, all the extracted traffic sequence images are subjected to pre-processing to remove noise and enhance the contrast level of the images. These pre-processed images are segmented by labelling each pixel to extract the uniform regions to aid the detection phase. A single-stage detector YOLOv5 is used to detect and locate vehicles in images. Each detection was exposed to Speeded Up Robust Feature (SURF) feature extraction to track multiple vehicles. Based on this, a unique number is assigned to each vehicle to easily locate them in the succeeding image frames by extracting them using the feature-matching technique. Further, we implemented a Kalman filter to track multiple vehicles. In the end, the vehicle path is estimated by using the centroid points of the rectangular bounding box predicted by the tracking algorithm. The experimental results and comparison reveal that our proposed vehicle detection and tracking system outperformed other state-of-the-art systems. The proposed implemented system provided 94.1% detection precision for Roundabout and 96.1% detection precision for Vehicle Aerial Imaging from Drone (VAID) datasets, respectively.

KEYWORDS

Semantic segmentation; YOLOv5; vehicle detection and tracking; Kalman filter; SURF

1 Introduction

Intelligent Traffic monitoring has gained significant importance for traffic control and management. Due to recent technological advancements, many computers vision-algorithms have become popular in surveillance systems, i.e., traffic surveillance, crowd monitoring, anomaly detection, and many more. Researchers have designed many vision-based systems that can detect objects and track them efficiently and accurately [1,2]. With the popularity of mobile platforms, vast traffic videos can be obtained for analysis [3,4]. These means of data collection can provide a high viewing angle, thus covering a more distant road surface [5,6]. However, some limitations need to be addressed to develop an effective traffic monitoring system. These limitations include variable object size at a large viewing angle which lowers the overall detection and tracking accuracy of the model [7]. In this article, we focus on the above-mentioned issues and propose a viable solution for efficient traffic monitoring based on semantic segmentation and the You Only Look Once (YOLO) algorithm. As we are dealing with variable vehicle sizes, we need a robust method for object detection. We have combined the YOLO algorithm with the semantic segmentation technique to improve the overall results. The segmentation technique helps reduce the complexity of the images so that in return the computational cost as well as time complexity of the. The model also reduces [8–10]. YOLOv5 is a real-time object detection algorithm that is popular in many computer vision tasks for its capability to recognize objects of different sizes at a fast speed [11,12]. Our proposed traffic monitoring system consists of six stages. In the first stage, images are extracted from the traffic footage. These images are pre-processed to make them suitable for processing in the later stages. In the next stage, we applied semantic segmentation techniques to label and extract uniform regions in the image by labelling each pixel. We have used and compared three segmentation techniques in terms of their accuracy score and chose the best result to feed into the detection phase. For detection, YOLOv4 is implemented as it is a powerful object recognition method. All the detections are used for SURF feature extraction to allocate each vehicle a unique number so that they can be tracked simultaneously in the preceding frames. The Kalman filter is used to track vehicles by predicting their locations in the next frames. Finally, we estimate the route of tracked automobiles by plotting the predicted locations. We performed extensive experiments on two benchmark datasets to validate our model: Vehicle Aerial Imaging from Drone (VAID) and Roundabout. The major contributions of our proposed traffic monitoring system are as follows:

- To reduce the computational cost and time complexity, we implemented and compared three different semantic segmentation techniques in terms of their accuracy rate and used the best one for detection.
- We used the YOLOv4 algorithm for vehicle detection to increase the performance of the model.
- To enable multi-object tracking, we extracted SURF features from the detected vehicles and allocated them a unique number so that they can be identified in the succeeding frames.
- We used the Kalman filter for multi-object tracking across the frames extracted from the traffic scene videos.

The rest of the paper is divided into six sections as follows: The relevant research is included in Section 2. The proposed model architecture is detailed in Section 3. The experimentation phase's results are presented in Section 4, along with a comparison to the state-of-the-art techniques. Finally, Section 5 summarizes the main points and discusses potential directions for further research.

2 Literature Review

In the past 20 years, intelligent vehicle monitoring has drawn a lot of attention from the scientific community. The most current developments in the field of vehicle monitoring are briefly covered in this section. We divide the literature into two streams as shown below.

2.1 Traffic Monitoring Using Conventional Methods

Traditional machine learning methods have been used for developing traffic monitoring applications. In [13], the merging of the SIFT and SVM allows for vehicle detection. Pyramid pooling, sliding windows, and NMS are integrated to further increase classification ability, which significantly improves the outputs provided for vehicle detection. Zhou et al. [14] implemented an adaptive method for background estimation. Further, Principal Component Analysis (PCA) is used to create a low-dimensional feature from the two histograms of each candidate, and a support vector machine classifier is used to decide whether or not it belongs to a real vehicle. After combining all of the categorized findings, a parallelogram is created to depict each vehicle's shape. Also, in [15], optical flow algorithm is used to determine the vehicles' moving direction flow along with a distance factor calculation. For tracking, feature templates for each vehicle were generated. An approach to context-aware recognition and tracking in urban aerial data is presented in [16]. The foreground mask is used in [17] to extract the blobs for vehicle detection. The overlapping blobs are separated by assuming that the width of the vehicle is not bigger than the width of the lane. The spatial pyramid context-aware feature-based tracker, motion prediction-based tracking, and motion detection-based tracking were the two types of tracking systems that the model was based on. The standard machine vision method can identify the vehicle more quickly, but it does not perform well in situations where the brightness of the image is changing, the background is moving irregularly, or there are slow-moving cars or complicated scenarios. Consequently, it is challenging to apply these algorithms to real scenarios and to obtain the accuracy and robustness required for practical application. The performance of deep learning algorithms is better for detection tasks especially when the object varies in the images. Therefore, our proposed algorithm uses YOLOv5 to detect vehicles in aerial images. Also, to reduce the overall computational complexity we used, the SURF feature combined with the Kalman filter for tracking.

2.2 Traffic Monitoring Using YOLO Algorithm

The use of You Only Look Once (YOLO) to analyze traffic vision data is one of the most well-known examples [18]. Based on its capacity to process multiple photos more quickly than traditional region-based convolutional neural networks (RCNNs), YOLO is highly used in real-time traffic monitoring. Because of the great detection accuracy and speed of YOLOv3, a new and effective detector called YOLO-ACN is proposed in [19]. An attention mechanism, a soft-NMS, CIoU (complete intersection over union) loss function, and depth-wise separable convolution are added to this method to make it better. The slow detection speed issue was solved in [20] using an enhanced YOLOv4-based video stream vehicle detection algorithm. The YOLOv4 algorithm is initially described theoretically in this study, followed by an algorithmic method for accelerating detection speed, and finally, actual road tests the algorithm in this work is utilized to make decisions for safe vehicle driving and can improve recognition speed without compromising accuracy. In another study [21], YOLOv3 was used for vehicle detection. The tracking was accomplished by using the centroid points of each vehicle bounding box. Experiments show that the YOLOv3 algorithm beats the conventional method in terms of detection accuracy and rate. A vehicle detection and counting method based on the combination of YOLOv4 and Convolutional Neural Network (CNN) is presented in [22]. They used the DeepSORT tracker to track vehicles effectively. For object detection, researchers

proposed an enhanced YOLOv3 transfer learning-based deep learning system [23]. The network is trained on a challenging data set for this work, and the output is quick and accurate, which is advantageous for applications that require detecting objects.

To reduce the computational complexity and increase the efficiency of the overall model, this paper combines the YOLOv5 algorithm with a segmentation technique for vehicle detection.

3 Materials and Methods

In this article, we proposed an effective traffic monitoring model. The model consists of six phases. First of all, all the images are pre-processed. These pre-processed images are segmented where each pixel is assigned a unique label to extract uniform regions from the images. To recognize vehicles in each segmented image, we employed the YOLOv5 algorithm which is fast and robust in detecting objects of variable sizes. To accomplish tracking across the different image frames, we extracted SURF features, based on which, a unique number was assigned to ease the identification of the vehicle in the succeeding frames during multi-vehicle tracking. These vehicles were tracked using the Kalman filter which predicts the location based on the detections. In the end, the path followed by each vehicle is estimated using the centroid points of the rectangular bounding box. Fig. 1 describes the overall architecture of a vision-based vehicle detection and tracking system.

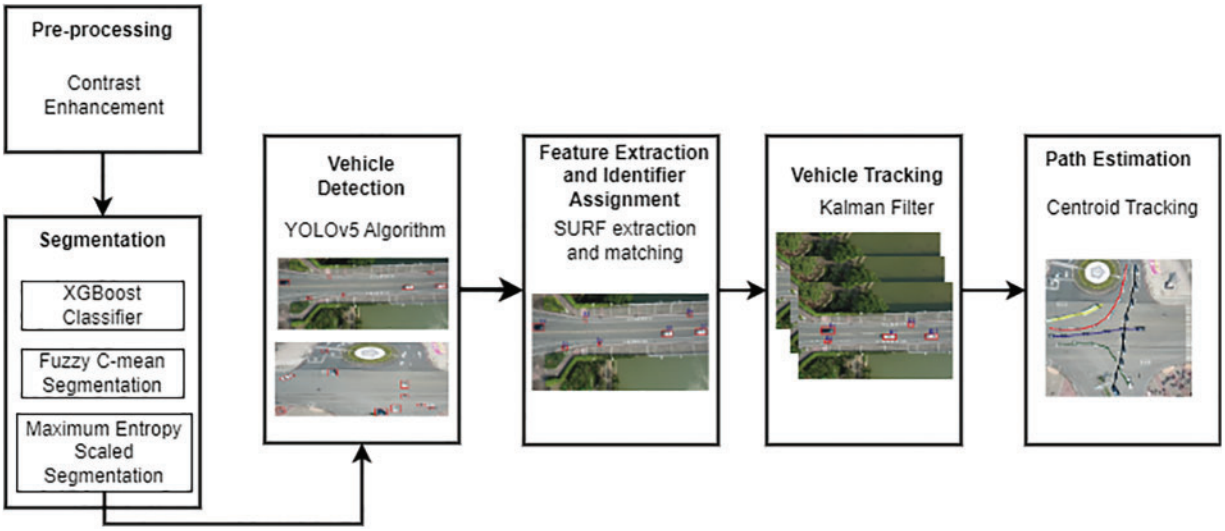

Figure 1: Block diagram of the proposed vision-based vehicle detection and tracking system

3.1 Preprocessing

Pre-processing is done to decrease noise and undesired distortion. This enhances the appearance of images that are required for further processing. To ensure consistency in the images, we resized all images to 768 × 768-pixel dimensions. To enhance the brightness level, we applied gamma correction [24,25] which is given by Eq. (1).

$$S = K^{1/T} T = \log \frac{0.5 \times 255}{P} \tag{1}$$

where K represents the input image and T denotes the computed gamma value. S is for the output image that has shrunk to [0,255]. The brightness enhancement output is visualized in Fig. 2.

(a) (b)

Figure 2: Contrast level enhancement using gamma correction (a) original images, (b) pre-processed images

3.2 Semantic Segmentation Framework

To reduce the computational complexity of the model, we applied semantic segmentation to the images before passing it to the YOLOv5 algorithm. For this purpose, we applied three different segmentation techniques and compared them in terms of their error rate. The best results were used for further processing. The details of each segmentation technique are given below.

3.2.1 XGBoost Segmentation

XGBoost is a Gradient-boosted trees method. It uses a supervised learning strategy that utilizes regularisation and specialized loss function optimization, as well as function approximation [26,27]. Therefore, to extract meaningful features on which we can train the XGBoost classifier, we used the Canny edge features [28], Scharr [29], Gaussian [30], and Sobel edge detection [31] filters. The classifier training process utilizes the retrieved feature set. Based on a combination of Classification and Regression Trees (CART) techniques, XGBoost improves and optimizes the algorithm to deliver superior outcomes. If $g(x)$ represents the single tree output, then it can be denoted using Eq. (2).

$$g(x) = d_r\,(x_j) \tag{2}$$

where dq stands for the matching leaf r's score and x for the input feature vector, Consequently, the ensemble tree's output is given in Eq. (3).

$$h_j = \sum\nolimits_{c=1}^{c} g_c(x_j) \tag{3}$$

The objective function stated in Eq. (4) is minimised using the XGBoost algorithm in each iteration.

$$\mathrm{W}\,(\mathrm{t}) = \sum\nolimits_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{w}\left(\mathrm{h_j}, \hat{\mathrm{h}}_{\mathrm{j}}^{\mathrm{t}-1} + \mathrm{g_t}(\mathrm{x_j})\right) + \sum\nolimits_{\mathrm{i}=1}^{\mathrm{t}} \Omega(\mathrm{g_i}) \tag{4}$$

where $\mathrm{g_i}$ is the regularisation term that controls the model's complexity and prevents overfitting, and $w\left(h_j, \hat{h}_j^{t-1} + g_t(x_j)\right)$ signifies the training loss between the actual label h_j and the output label $\hat{h}_j^{t-1}$ for the n number of samples. The complexity can be calculated by using Eq. (5).

$$\Omega\,(g) = \gamma M + \frac{1}{2}\lambda \sum\nolimits_{i=1}^{M} d_i^2 \tag{5}$$

where M stands for the number of leaves, γ is the dataset-dependent pseudo-regularization hyperparameter, and λ denotes the L2 norm for leaf weights. XGBoost uses gradients to approximate the loss function to find the ideal value of the objective function. The segmentation result using the XGBoost classifier is shown in Fig. 3.

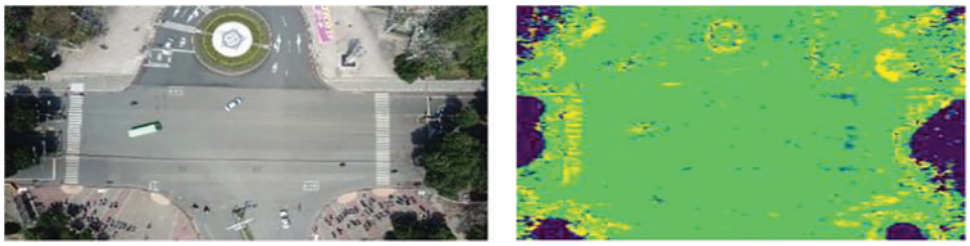

Figure 3: Semantic segmentation using XGBoost

3.2.2 FCM Segmentation

FCM is a powerful segmentation technique that groups the pixels in such a way that a single pixel may belong to more than one cluster. This multi-occurrence of the same element can be referred to as fuzzy. In the FCM segmentation process, the objective function is optimised in several iterations to get the final output. Throughout the iterative procedure, the clustering centres and membership degrees have been continuously updated [32,33]. Performance index H_{FCM} is formulated using Eq. (6).

$$\boldsymbol{H_{FCM}} = (\boldsymbol{Q}, \boldsymbol{S}) = \sum_{i=1}^{r} \sum_{b=1}^{N} \boldsymbol{z_{ib}^{t}} \|\boldsymbol{q_b} - \boldsymbol{s_i}\|^2, \mathbf{1} < \boldsymbol{t} < \boldsymbol{\infty} \tag{6}$$

where r is the number of clusters, N is the number of pixels, q_b is the bth pixel, s_i is the centre of the ith cluster, and t is the blur exponent. Each cluster centre and membership function are updated using Eqs. (7) and (8).

$$\mathbf{z_{ib}^{t}} = \frac{\mathbf{1}}{\sum_{j=1}^{c} \left(\frac{\mathbf{l_{ib}^{2}}}{\mathbf{l_{jb}^{2}}} \right)^{\frac{1}{t-1}}} \tag{7}$$

$$\boldsymbol{s_j} = \frac{\sum_{k=1}^{N} \boldsymbol{z_{ib}^{t}} \boldsymbol{q_b}}{\sum_{k=1}^{N} \boldsymbol{z_{ib}^{t}}} \tag{8}$$

where $\mathbf{l_{ib}^{2}}$ represents the distance between pixel q_b and cluster centroid s_j and z_{ib}^{t} stands for the membership matrix that belongs to [0, 1]. The FCM objective function is minimized by assigning pixels with high membership to pixels' values when they are adjacent to the center of their respective class and small membership values when they are far from them. The segmentation result using FCM is given in Fig. 4.

3.2.3 Maximum Entropy Scaled Segmentation

We also used the maximum entropy scaled segmentation technique. In this segmentation, edge weights are used to describe similarities using a similarity matrix and are used to depict an image over a graph (G = K, E) with vertices that represent image pixels [33]. To ensure that the final equation (G = K, A) is correct, we must select an edge group (A) to a limit of E. Although every edge in the graph maintains a self-loop, this fact alone is insufficient to solve the graph partition problem. When the edge of the related vertices is excluded from A, we increase the edge weight for the self-loop so that each

linked vertex has a constant incidence. We also employ the entropy scale of the random walk (RW) criteria on the graph to obtain dense and reliable clusters for segmentation purposes. The probability set functions prob u,v are denoted using Eq. (9) when the RW distribution remains constant.

$$\boldsymbol{prob}_{\boldsymbol{u,v}}(\mathfrak{A}) = \begin{cases} \dfrac{\boldsymbol{I}_{\boldsymbol{u,v}}}{\boldsymbol{W}_{\boldsymbol{u}}} & \text{if } \boldsymbol{u} \neq \boldsymbol{v} \text{ and } \boldsymbol{e}_{\boldsymbol{u,v}} \in \boldsymbol{A} \\ \mathbf{0} & \text{if } \boldsymbol{u} \neq \boldsymbol{y} \text{ and } \boldsymbol{e}_{\boldsymbol{u,v}} \notin \boldsymbol{A} \\ \mathbf{1} - \dfrac{\sum_{\boldsymbol{v}} \boldsymbol{e}_{\boldsymbol{u,v}} \in \boldsymbol{A}\boldsymbol{W}_{\boldsymbol{u,v}}}{\boldsymbol{W}} & \text{if } \boldsymbol{u} = \boldsymbol{v} \end{cases} \tag{9}$$

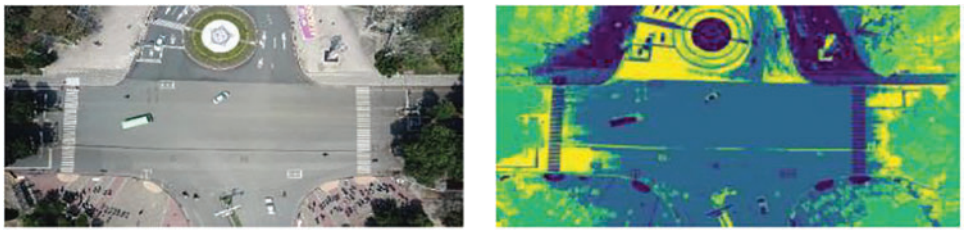

Figure 4: Semantic segmentation using FCM

As a result, the function given in Eq. (10) may be used to represent the entropy scale of the RW on (G = K, $\mathfrak{A}$).

$$\mathfrak{S}(\mathfrak{A}) = -\sum_{\boldsymbol{u}} \boldsymbol{\mu}_{\boldsymbol{u}} \sum_{\boldsymbol{y}} \boldsymbol{prob}_{\boldsymbol{i,j}}(\mathfrak{A}) \mathbf{log}\left(prob_{\boldsymbol{u,v}}(\mathfrak{A})\right) \tag{10}$$

We use a balancing function to generate clusters that have equal-sized dimensions. Then, the $Z_{\mathfrak{A}}$. The diffused cluster's membership function can be written using Eq. (11).

$$\boldsymbol{p}_{\boldsymbol{Z}_{\mathfrak{A}}}(\boldsymbol{i}) = \frac{|\boldsymbol{S}_{\boldsymbol{i}}|}{|\boldsymbol{K}|}, \boldsymbol{i} = \mathbf{1}, \ldots, \boldsymbol{N}_{\mathfrak{A}} \tag{11}$$

where $S_{\mathfrak{A}} = S_{1,}S_2, \ldots, S_{N\mathfrak{A}}$ represents the set of edges. Also, the balancing function is given in Eq. (12).

$$\boldsymbol{B}(\mathfrak{A}) = \mathfrak{H}(\mathfrak{A}) - \boldsymbol{N}_{\mathfrak{A}} = -\sum_{\boldsymbol{i}} \boldsymbol{p}_{\boldsymbol{Z}_{\mathfrak{A}}}(\boldsymbol{i}) \boldsymbol{log}\left(\boldsymbol{p}_{\boldsymbol{Z}_{\mathfrak{A}}}(\boldsymbol{i})\right) - \boldsymbol{N}_{\mathfrak{A}} \tag{12}$$

while $\boldsymbol{N}_{\mathfrak{A}}$ supports a single cluster, entropy $\mathfrak{S}(\mathfrak{A})$ supports clusters with similar sizes. Clusters that are flexible, symmetric, and organized are encouraged by the objective function, which is the combination of the composition of the entropy rate and the balance function. The clustering is achieved by optimizing the objective function about the edges set as given in Eq. (13).

$$\mathbf{max}_{\mathfrak{A}}\, \boldsymbol{H}(\mathfrak{A}) + \boldsymbol{B}(\mathfrak{A}) \tag{13}$$

when $\boldsymbol{N}_{\mathfrak{A}}$ is more than or equal to n and subject to A is a subset of E, the weight of the balancing term is assumed to be 0. The segmentation result using the maximum entropy scaled segmentation is shown in Fig. 5.

The three implemented segmentation techniques are compared based on the error rate to select the best results for further processing. It can be seen from Table 1 that the error rate for maximum entropy scaled segmentation is low as compared to the other two methods. Therefore, the results from this technique were passed to the vehicle detection phase based on YOLOv5. The Comparison of Error Rate of Semantic Segmentation Techniques over Roundabout and VAID Dataset.

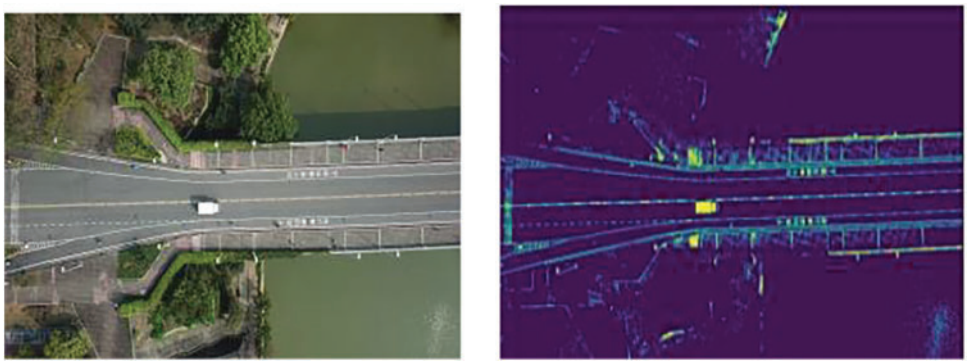

Figure 5: Entropy scaled segmentation

Table 1: Comparison of error rate of semantic segmentation techniques over Roundabout and VAID datasets

Datasets	Error rate (%)		
	XGBoost	FCM	Maximum entropy
Roundabout	17.6	14.4	10.8
VAID	15.3	14.5	9.2

3.3 Vehicle Detection Using YOLOv5

YOLO algorithms are one-stage detectors that are extensively employed in object identification systems, particularly for vehicle detection tasks, due to their high-performance characteristics. YOLO focuses on global information for target detection as it inputs the full image and returns the position of the object bounding box [34,35].

The processing time of YOLO deeper networks is drastically decreased using the single-stage detector mechanism. Additionally, it works better when detecting small targets [36–38]. The architecture of YOLOv5 consists of four main components as shown in Fig. 5. The backbone mainly contains Cross-Stage Partial (CSP) networks and Spatial Pyramid Pooling (SPP) to extract the best region of the image for further feature extraction. The SPP module makes the detection invariant to object sizes. The Neck module creates feature pyramids using Path Aggregation (PANet) and Feature Pyramid Network (FPN). The bottom-up path and low-level feature propagation are enhanced by the FPN structure. Additionally, localization features are transmitted via the PAN framework from lower feature maps to higher feature maps. The final prediction of the object detection and its corresponding bounding box is detected using the three convolutional layers in the Head module. YOLOv5 employs the Sigmoid Linear Unit (SiLU) activation function in its hidden layers, while the output layer's convolution process makes use of the Sigmoid activation function. Both these activation functions are given in Eqs. (14) and (15).

$$SiLU\,(u) = u \times \sigma\,(u) \tag{14}$$

where $\sigma\,(u)$ represents the logistic sigmoid.

$$S\,(u) = \frac{1}{1 + e^{-u}} \tag{15}$$

where $S(u)$ denotes the sigmoid activation function. The detection results from the YOLOv5 algorithm are shown in Fig. 6.

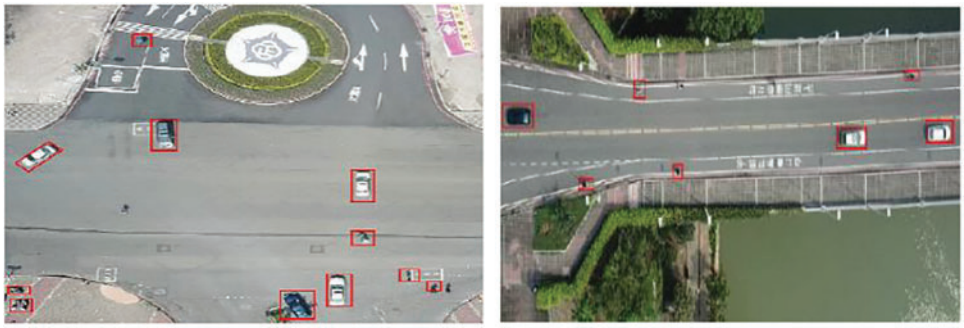

Figure 6: Vehicle detection using YOLOv5

3.4 Feature Extraction Unique Number Allocation

Each frame contains more than one vehicle. To simultaneously track these vehicles across all the frames, we must allocate them a unique number which eases each vehicle identification. Also, these unique allotted numbers must be retrieved based on some appearance feature. For this purpose, we subjected all the detected vehicles to Speeded-Up Robust Feature (SURF) extraction [39–41]. It is efficient for image comparison. Since SURF features can be quickly computed, they are reliable for object recognition in real-time scenarios. It included a description and an interest point detector. For the interesting locations that the detector had discovered, the descriptor created feature vectors. Using Eq. (16), the integral and interest point has been calculated.

$$S_{\Sigma}(\mathbf{k}) = \sum_{u=0}^{u \leqslant x} \sum_{v=0}^{v \leqslant y} S(u, v) \tag{16}$$

where $S_{\Sigma}(\mathbf{k})$ is the input image and $S(u,v)$ represents the integral image at the position $k = (x, y)^T$. Additionally, the Hessian matrix as shown in Eq. (17) was used to calculate the blob structure in the image.

$$\mathcal{H}(\mathrm{x}, \sigma) = \begin{bmatrix} L_{xx}(\mathrm{k}, \sigma) & L_{xy}(\mathrm{k}, \sigma) \\ L_{xy}(\mathrm{k}, \sigma) & L_{yy}(\mathrm{k}, \sigma) \end{bmatrix} \tag{17}$$

where $L_{xx}(k, \sigma)$, $L_{xy}(k, \sigma)$, and $L_{yy}(k, \sigma)$ represent the second-order derivative of the Gaussian function convoluted with image S. The number assigned to each vehicle is recovered by using the thresholding method based on the number of feature matching. If no matches are found, then the vehicle is registered as a new one. The number allocated to each vehicle for tracking is shown in Fig. 7.

3.5 Multi-Vehicle Tracking via Kalman Filter

To track multi-vehicles in the succeeding frames, we implemented the Kalman filter. Based on flawed and untrustworthy data, the Kalman filter produces hidden variable estimations. Additionally, the Kalman filter predicts the state of the system using earlier calculations [42–46]. When employing the Kalman filter, we consider a tracking system in which c_s is the state vector describing the dynamic behaviour of the object and subscript s is the discrete time. Eq. (18) can be used to define the state transition from time $s-1$ to time s.

$$c_s = Fc_{s-1} + Bu_{s-1} + w_{s-1} \tag{18}$$

where B symbolizes the control input matrix subjected to the control vector, i.e., $\boldsymbol{u}_{s-1}$, and F represents the state transition matrix applied to the prior state vector $\boldsymbol{c}_{s-1}$. Additionally, the process noise vector $\boldsymbol{w}_{s-1}$ is made up of white Gaussian noise having a zero mean and covariance A as stated in Eq. (19).

$$p(w) \sim M(0, A) \tag{19}$$

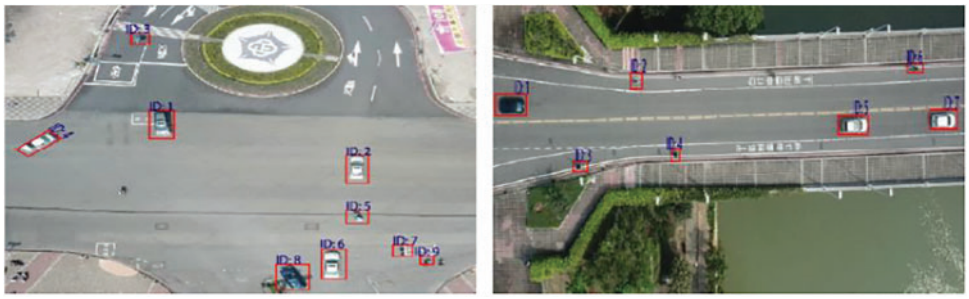

Figure 7: ID assignment to each detected vehicle based on SURF feature extraction

To show how state and measurement relate to one another at the present time step K, represented by Eq. (20), the measurement model and the process model are integrated.

$$z_s = Nc_s + v_s \tag{20}$$

where v_s represents the measurement of the noise vector with a covariance N having probability distribution $p(v) \sim M(0, J)$. N is the measurement matrix and zs denotes the measurement vector. The Kalman filter has two stages: prediction and correction. The prediction step moves the present state forward and provides an a priori evaluation of the state $\hat{c}^{+}_{s-1}$. Eqs. (21) and (22) can be used to estimate the projected state and predicted error covariance.

$$\hat{c}^{-}_{s} = F\hat{c}^{+}_{s-1} + Bu_{s-1} \tag{21}$$

$$R^{-}{}_{s} = FR^{+}{}_{s-1}F^{T} + \mathrm{A} \tag{22}$$

Feedback is given throughout the corrective phase. Actual measurement is added to the a priori estimate provided in Eq. (23) to enhance the a posteriori estimate $\hat{c}^{+}_{s}$.

$$\hat{c}^{+}_{s} = \hat{c}^{-}_{s} + K_s\tilde{y} \tag{23}$$

where $\tilde{y}$ is the measurement residual and K_s denotes the Kalman gain. Both are equivalent to Eqs. (24) and (25), respectively.

$$K_s = F^{-}{}_{s}N^{T}\left(J + NF^{-}{}_{s}N^{T}\right)^{-1} \tag{24}$$

$$\tilde{y} = z_s - N\hat{c}^{-}_{s} \tag{25}$$

Eq. (26) can be used to obtain the updated error covariance.

$$F^{+}{}_{s} = (I - K_sN)F^{-}{}_{s} \tag{26}$$

In the equations above, the operator ^ stands for an estimation of a variable. The superscripts − and +, respectively, denote prior and posterior estimations. The projected state estimate was built on top of the previously revised state estimate. State error covariance is the name given to the symbol F.

Using Eq. (27), covariance may be calculated for any variable x.

$$cov\,(x) = \mathbb{E}[(x - \hat{x})\,(x - \hat{x})^{T}]^{T} \tag{27}$$

where $\mathbb{E}$ stands for the expected mean value. The initialization phase of the Kalman filter, which is necessary for its use, calls for the initial guess of the state estimate, $\hat{c}_0^+$, and the error covariance matrix,

$F^+{}_0$ along with matrices A and J. The updated error covariance is lower than the anticipated error covariance following the use of the measurement in the update stage. The prediction and updated stages are used to implement the Kalman filter for each time step K. The results of the Kalman filter tracking are given in Fig. 8.

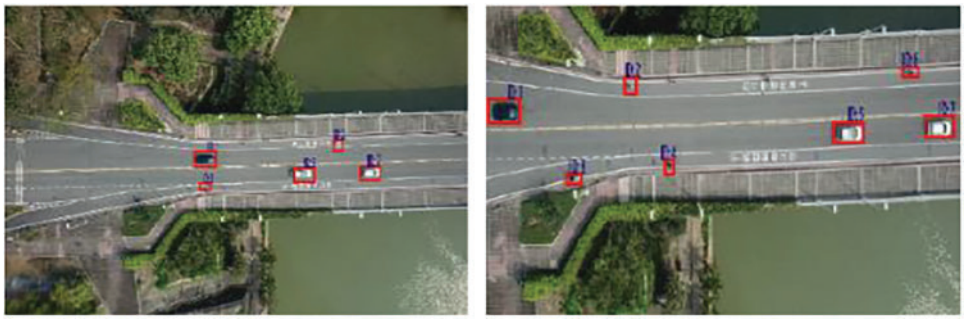

Figure 8: Tracking vehicles across the image frames using Kalman filter

3.6 Path Estimation via Centroid Point

To compute the path followed by each tracked vehicle as far as the tracking is being done. We calculated the centre points of the estimated rectangular bounding box obtained from the estimated position by the Kalman filter. These points were plotted and joined for each car. The steps involved in path estimation are given in Algorithm 1. The algorithm takes segmented images and corresponding detected vehicles as input. All detected vehicles are subjected to SURF feature extraction. For every image frame, if the feature matches with the previously detected vehicle and the number of matches is greater than 6 then the ID is retrieved and assigned to the newly detected car. All the detected cars are then tracked using the Kalman filter and rectangular coordinates are extracted to plot the trajectories as shown in Fig. 9 where the plotted paths of each vehicle are represented using a different color.

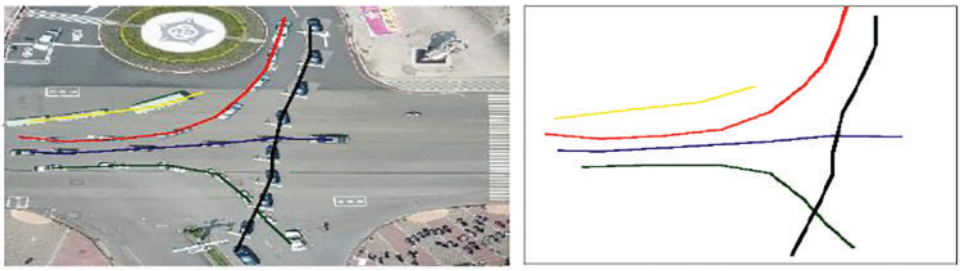

Figure 9: Path estimation of vehicles being tracked across image frames

Algorithm 1: Path Estimation of Tracked Vehicles

Input: I $M = \{I^1, I^2, \ldots, I^z\}$
// segmented images
Output: $\{T^r\}_{r=1}^{N}$ // vehicle trajectories
D=$\{D^1, D^2, \ldots, D^N$ ←YOLOv5(IM) //Detected Vehicles
IDNumber=0
fv=[]//feature vector
For j = 1 to the length of D
new_vehicle←SURF(Dj)
If fv is empty then
Increment IDNumber by 1
fv←new_vehicle,IDNumber
Else
match← (new_vehicle, fv)
If matches>6
RetrieveIdentificationNumber(Di)
Else
Increment IDNumber by 1
fv←new_vehicle,IDNumber
end for
IdentificationNumbers=$\{id^1, id^2, \ldots, id^m\}$
(D,IdentificationNumbers)←KalmanFilter(D) //Predicted locations of vehicles by kalman filter
For i = 1 to the length of D
$x_1^i, x_2^i, y_1^i, y_2^i$ ←ExtractRectangularCoordinatesofVehicle()
$x_{centroid} \leftarrow \frac{(x_1^i + x_2^i)}{2}$
$y_{centroid} \leftarrow \frac{(y_1^i + y_2^i)}{2}$
$T^r \leftarrow [x_{centroid}, y_{centroid}]$
end for
return vehicle trajectories

4 Experiments and Results

Our proposed vision-based vehicle detection and tracking model has been built over a laptop having Intel Core i5-7200U 2.50 GHz of processing power, 6 GB RAM, and Python tools. The proposed model performed well when tested over the two datasets: Roundabout and VAID dataset. Furthermore, the dataset and experimentation details are discussed below. A contrast with the other state-of-the-art techniques has also been drawn.

4.1 Datasets Description

The dataset used to test our proposed algorithm is described in detail below.

4.1.1 Roundabout Traffic Dataset

The Roundabout aerial image dataset [47] contains 15,474 RGB images. All these images are in .jpg format. The dataset consists of different traffic locations having different traffic patterns. Each

image is combined with an XML file in the Visual Object Classes (PASCAL VOC) format to indicate the locations of the cars in each of these pictures.

4.1.2 VAID Dataset

The VAID (Vehicle Aerial Imaging from Drone) dataset [48] has 5985 aerial photographs of Taiwan. The photos are in the .jpg format having a resolution of 1137 × 640 pixels. All the traffic sequences are captured by using a drone under variable lighting and traffic conditions dataset.

4.2 Performance Measurement and Result Analysis

We implemented three different segmentation methods, i.e., by using XGBoost classifier, FCM, and maximum entropy scaled semantic segmentation. For comparison, we used the accuracy and sensitivity measure as shown in Tables 2 and 3. Entropy-scaled segmentation has the highest accuracy of 89.2 and 90.8 for both datasets.

Table 2: Accuracy comparison over Roundabout and VAID datasets

Datasets	Accuracy (%)		
	XGBoost	FCM	Maximum entropy
Roundabout	82.4	85.6	89.2
VAID	84.7	85.5	90.8

Table 3: Sensitivity comparison over Roundabout and VAID datasets

Datasets	Sensitivity		
	XGBoost	FCM	Maximum entropy
Roundabout	0.812	0.819	0.857
VAID	0.787	0.801	0.901

The segmented images were subjected to vehicle detection. Table 4 shows the results of the detection algorithm over the two benchmark datasets namely Roundabout and VAID datasets, respectively.

Table 4: Vehicle detection performance over Roundabout and VAID datasets

Datasets	Precision	Recall	F1 score
Roundabout	94.1	88.2	91.1
VAID	96.3	90.7	93.4

After detection, as our model tracks multiple vehicles across the image frames, therefore, we used Multi-Object Tracking Accuracy (MOTA) and Multi-Object Tracking Precision (MOTP) to assess the

performance of the tracking algorithm calculated as follows:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSwitches_t)}{\sum_t Ground\ Truth_t} \tag{28}$$

$$MOTP = \frac{\sum_{t,i} Bounding\ Box\ Overlap_{t,i}}{\sum_t No.of\ Matches_t} \tag{29}$$

where t is the number of frames. The algorithm achieved a MOTA of 87.2% and a MOTP of 92.3% on the Roundabout dataset, while on the VAID dataset, it achieved a MOTA of 91.7% and an MTP of 95.6%. These scores demonstrate the algorithm's effectiveness in tracking objects accurately across different environments. Table 5 shows the results for the tracking algorithm on the two benchmark datasets.

Table 5: Vehicle tracking performance over Roundabout and VAID datasets

Datasets	MOTA	MTP
Roundabout	87.2	92.3
VAID	91.7	95.6

4.3 Comparison with State-of-the-Art Techniques

In this section, we compared our proposed vehicle detection and tracking algorithm with other state-of-the-art methods. Table 6 shows the comparison of the proposed detection algorithm whereas Table 7 shows the comparison of tracking algorithms.

Table 6: Vehicle detection comparison with SOTA techniques over Roundabout and VAID datasets

Datasets	Methods	Precision
Roundabout	Optical Flow [49]	56.1
	Frame Differencing [50]	79.5
	Proposed	**94.1**
VAID	YOLOv4 [48]	83.12
	Proposed	**96.3**

Table 7: Vehicle tracking comparison with other state-of-the-art techniques over Roundabout and VAID datasets

Datasets	Methods	Precision
Roundabout	Template matching [51]	71.3
	Centroid tracking [51]	81.3
	Proposed	**87.2**

(Continued)

Table 7 (continued)

Datasets	Methods	Precision
VAID	Template matching [50]	78.8
	Centroid tracking [21]	84.6
	Proposed	**91.7**

5 Conclusion

In conclusion, this paper has presented a vision-based vehicle detection and monitoring system. Firstly, all the images are pre-processed to enhance the contrast level. These images were segmented into uniform regions using the semantic segmentation technique by applying FCN. The remarkable results of the proposed model show that it outperforms the SOTA remote sensing scene classification techniques. Vehicle detection is done by employing the YOLOv5 algorithm. All the detected vehicles are subjected to SURF feature extraction based on which a number is assigned to each of them. These unique numbers were retrieved in the succeeding frames using the feature-matching method to locate multiple vehicles. The tracking is accomplished using the Kalman filter. Path estimation of each tracked vehicle is done by calculating and plotting the centroid points of the rectangular bounding box. The model is experimented on two publicly available datasets, i.e., roundabout and VAID datasets. Our proposed algorithm achieved a detection precision of 94.1% and 96.3% and a tracking accuracy of 87.2% and 91.7%, respectively. The results show that the proposed system outperformed other state-of-the-art techniques. In the future, we will perform additional research on pattern recognition methods and other feature extraction techniques to enhance the outcomes of both the detection and tracking algorithms.

Acknowledgement: This research is supported and funded by Princess Nourah bint Abdulrahman University Researchers, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: The APC was funded by the Open Access Initiative of the University of Bremen and the DFG via SuUB Bremen. This research was supported by the Deanship of Scientific Research at Najran University, under the Research Group Funding Program Grant Code (NU/RG/SERC/12/30). This research is supported and funded by Princess Nourah bint Abdulrahman University Researchers Supporting Project Number (PNURSP2024R410), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. This study is supported via funding from Prince Sattam bin Abdulaziz University Project Number (PSAU/2024/R/1445).

Author Contributions: Study conception and design: Asifa Mehmood Qureshi, Nouf Abdullah Almujally, data collection: Naif Al Mudawi, and Abdul Haleem Butt; analysis and interpretation of results: Abdulwahab Alazeb, and Hui Liu; draft manuscript preparation: Asifa Mehmood Qureshi, Ahmad Jalal, and Mohammad Alonazi. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets that support the findings of this study are openly available at https://zenodo.org/records/6407460 and https://vision.ee.ccu.edu.tw/aerialimage/ (accessed on 7 April 2024).

Ethics Approval: Not applicable.

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

References

[1] A. Ahmed, A. Jalal, and A. A. Rafique, "Salient segmentation basedobject detection and recognition using hybrid genetic transform," in *2019 Int. conf. Appl. Eng. Math., ICAEM 2019*, Taxila, Pakistan, Aug. 2019, pp. 203–208. doi: 10.1109/ICAEM.2019.8853834.

[2] M. Alarfaj *et al.*, "An intelligent framework for recognizing social human-object interactions contextual scene understanding: Template objects detector and feature descriptors for indoor/outdoor scenarios view project solar photovoltaic power forecasting view project an intelligent framework for recognizing social human-object interactions," *Comput. Mater. Contin.*, vol. 1, no. 1, pp. 1207–1223, 2022. doi: 10.32604/cmc.2022.025671.

[3] M. Qureshi, A. H. Butt, and A. Jalal, "Highway traffic surveillance over UAV dataset via blob detection and histogram of gradient," in *2023 4th Int. Conf. Adv. Comput. Sci.*, Lahore, Pakistan, Feb. 2023, pp. 1–5. doi: 10.1109/ICACS55311.2023.10089709.

[4] M. Qureshi and A. Jalal, "Vehicle detection and tracking using Kalman filter over aerial images," in *2023 4th Int. Conf. Adv. Comput. Sci.*, Lahore, Pakistan, Feb. 2023, pp. 1–6. doi: 10.1109/ICACS55311.2023.10089701.

[5] 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.

[6] N. Ammour, H. Alhichri, Y. Bazi, B. Benjdira, N. Alajlan and M. Zuair, "Deep learning approach for car detection in UAV imagery," *Remote Sens. 2017*, vol. 9, no. 4, pp. 312, Mar. 2017. doi: 10.3390/RS9040312.

[7] M. Qureshi, A. Almujally, S. Alotaibi, H. Alatiyyah, and J. Park, "Intelligent traffic surveillance through multi-label semantic segmentation and filter-based tracking," *Comput. Mater. Contin.*, vol. 76, no. 3, pp. 3707–3725, 2023. doi: 10.32604/cmc.2023.040738.

[8] T. Beheim, "Multi-vehicle detection and tracking in aerial imagery sequences using deep learning algorithms," Doctoral dissertation, Tech. Univ. of Munich, Germany, Nov. 2021.

[9] S. Kapania, D. Saini, S. Goyal, N. Thakur, R. Jain and P. Nagrath, "Multi object tracking with UAVs using deep SORT and YOLOv3 RetinaNet detection framework," in *ACM Int. Conf. Proc. Ser.*, Bangalore, India, Jan. 2020, pp. 1–6. doi: 10.1145/3377283.3377284.

[10] S. Zhao and F. You, "Vehicle detection based on improved YOLOv3 algorithm," in *2020 Int. Conf. Intell. Transp. Big Data Smart City, ICITBS 2020*, Vientiane, Laos, Jan. 2020, pp. 76–79. doi: 10.1109/ICITBS49701.2020.00024.

[11] S. Chen, C. Wang, and Y. Zho, "A pedestrian detection method based on YOLOv5s and image fusion," *Electron. Opt. Control*, vol. 29, pp. 96–101, 2022.

[12] L. Shao, H. Wu, C. Li, and J. Li, "A vehicle recognition model based on improved YOLOv5," *Electronics*, vol. 12, no. 6, pp. 1323, 2023. doi: 10.3390/electronics12061323.

[13] Y. W. Tu and J. X. Guo, "Research on vehicle detection technology based on SIFT feature," in *Proc. 2018 IEEE 8th Int. Conf. Electron. Inf. Emerg. Commun., ICEIEC 2018*, Beijing, China, Sep. 2018, pp. 274–278. doi: 10.1109/ICEIEC.2018.8473575.

[14] J. Zhou, D. Gao, and D. Zhang, "Moving vehicle detection for automatic traffic monitoring," *IEEE Trans. Veh. Technol.*, vol. 56, no. 1, pp. 51–59, Jan. 2007. doi: 10.1109/TVT.2006.883735.

[15] Y. Liu, Y. Lu, Q. Shi, and J. Ding, "Optical flow based urban road vehicle tracking," in *9th Int. Conf. Comput. Intell. Secur., CIS 2013*, Sichuan, China, 2013, pp. 391–395. doi: 10.1109/CIS.2013.89.

[16] 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 Signal Based Surveill., AVSS 2017*, Lecce, Italy, 2017, pp. 1–6. doi: 10.1109/AVSS.2017.8078504.
[17] R. Velazquez-Pupo *et al.*, "Vehicle detection with occlusion handling, tracking, and OC-SVM classification: A high performance vision-based system," *Sensors*, vol. 18, no. 2, pp. 374, 2018. doi: 10.3390/s18020374.
[18] J. P. Lin and M. Te Sun, "A YOLO-based traffic counting system," in *2018 Conf. Technol. Appl. Artif. Intell., TAAI 2018*, Taichung, Taiwan, Dec. 2018, pp. 82–85. doi: 10.1109/TAAI.2018.00027.
[19] Y. Li, S. Li, H. Du, L. Chen, D. Zhang and Y. Li, "Focusing on small target and occluded object detection," *IEEE Access*, vol. 8, pp. 227288–227303, 2020. doi: 10.1109/ACCESS.2020.3046515.
[20] X. Hu, Z. Wei, and W. Zhou, "A video streaming vehicle detection algorithm based on YOLOv4," in *2021 IEEE 5th Adv. Inf. Technol., Electron. Autom. Control Conf. (IAEAC)*, Chongqing, China, 2021, pp. 2081–2086. doi: 10.1109/IAEAC50856.2021.9390613.
[21] S. Ali and A. Jalal, "Vehicle detection and tracking from aerial imagery via YOLO and centroid tracking," in *2023 4th Int. Conf. Adv. Comput. Sci.*, Lahore, Pakistan, Feb. 2023, pp. 1–6.
[22] T. N. Doan and M. T. Truong, "Real-time vehicle detection and counting based on YOLO and DeepSORT," in *2020 12th Int. Conf. Knowl. Syst. Eng., KSE 2020*, Can Tho City, Vietnam, Nov. 2020, pp. 67–72. doi: 10.1109/KSE50997.2020.9287483.
[23] G. S. R. MacHiraju, K. A. Kumari, and S. K. Sharif, "Object detection and tracking for community surveillance using transfer learning," in *Proc. 6th Int. Conf. Inven. Comput. Technol. ICICT 2021*, Coimbatore, India, Jan. 2021, pp. 1035–1042. doi: 10.1109/ICICT50816.2021.9358698.
[24] K. V. Najiya and M. Archana, "UAV video processing for traffic surveillence with enhanced vehicle detection," in *Proc. Int. Conf. Inven. Commun. Comput. Technol. (ICICCT 2018)*, Coimbatore, India, 2018, pp. 662–668. doi: 10.1109/ICICCT.2018.8473204.
[25] G. Xu, J. Su, H. Pan, Z. Zhang, and H. Gong, "An image enhancement method based on gamma correction," in *ISC. 2009—2009 Int. Symp. Comput. Intell. Des.*, Changsha, China, 2009, vol. 1, pp. 60–63. doi: 10.1109/ISCID.2009.22.
[26] A. Hashmi, A. H. Osman, "Brain tumor classification using conditional segmentation with residual network and attention approach by extreme gradient boost," *Appl. Sci.*, vol. 12, no. 21, pp. 10791, Oct. 2022. doi: 10.3390/app122110791.
[27] H. N. Pham *et al.*, "Lesion segmentation and automated melanoma detection using deep convolutional neural networks and XGBoost," in *Proc. 2019 Int. Conf. Syst. Sci. Eng., ICSSE 2019*, Dong Hoi, Vietnam, Jul. 2019, pp. 142–147.
[28] L. Yuan and X. Xu, "Adaptive image edge detection algorithm based on canny operator," in *2015 4th Int. Conf. Adv. Inf. Technol. Sens. Appl., AITS 2015*, Harbin, China, Feb. 2016, pp. 28–31. doi: 10.1109/AITS.2015.14.
[29] Kumar, N. Lal, and R. N. Kumar, "A comparative study of various filtering techniques," in *Proc. 5th Int. Conf. Trends Electron. Inform., ICOEI 2021*, Tirunelveli, India, Jun. 2021, pp. 26–31. doi: 10.1109/ICOEI51242.2021.9453068.
[30] N. Khalid, Y. Y. Ghadi, M. Gochoo, A. Jalal, and K. Kim, "Semantic recognition of human-object interactions via gaussian-based elliptical modeling and pixel-level labeling," *IEEE Access*, vol. 9, pp. 111249–111266, 2021. doi: 10.1109/ACCESS.2021.3101716.
[31] G. N. Chaple, R. D. Daruwala, and M. S. Gofane, "Comparisons of robert, prewitt, sobel operator based edge detection methods for real time uses on FPGA," in *Int. Conf. Technol. Sustain. Dev., ICTSD 2015*, Mumbai, India, Apr. 2015.
[32] J. Miao, X. Zhou, and T. Z. Huang, "Local segmentation of images using an improved fuzzy C-means clustering algorithm based on self-adaptive dictionary learning," *Appl. Soft Comput.*, vol. 91, no. 2–3, pp. 106200, Jun. 2020. doi: 10.1016/j.asoc.2020.106200.
[33] Rafique, M. Gochoo, A. Jalal, and K. Kim, "Maximum entropy scaled super pixels segmentation for multi-object detection and scene recognition via deep belief network," *Multimed. Tools Appl.*, vol. 82, no. 9, pp. 13401–13430, 2022. doi: 10.1007/s11042-022-13717-y.

[34] Dewi, R. C. Chen, Y. C. Zhuang, and H. J. Christanto, "Yolov5 series algorithm for road marking sign identification," *Big Data Cogn. Comput.*, vol. 6, no. 4, pp. 149, 2022. doi: 10.3390/bdcc6040149.
[35] J. Yao, J. Qi, J. Zhang, H. Shao, J. Yang and X. Li, "A real-time detection algorithm for kiwifruit defects based on YOLOv5," *Electronics*, vol. 10, no. 14, pp. 1711, 2021. doi: 10.3390/electronics10141711.
[36] Neupane, T. Horanont, and J. Aryal, "Real-time vehicle classification and tracking using a transfer learning-improved deep learning network," *Sensors*, vol. 22, no. 10, pp. 1–21, 2022. doi: 10.3390/s22103813.
[37] J. Wang, Y. Dong, S. Zhao, and Z. Zhang, "A high-precision vehicle detection and tracking method based on the attention mechanism," *Sensors*, vol. 23, no. 2, pp. 724, 2023. doi: 10.3390/s23020724.
[38] J. H. Kim, N. Kim, Y. W. Park, and C. S. Won, "Object detection and classification based on YOLO-V5 with improved maritime dataset," *J. Mar. Sci. Eng.*, vol. 10, no. 3, pp. 377, 2022. doi: 10.3390/jmse10030377.
[39] G. Du, F. Su, and A. Cai, "Face recognition using SURF features," in *MIPPR 2009: Pattern Recognit. Comput. Vis.*, Oct. 2009, vol. 7496, pp. 593–599.
[40] Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, Barcelona, Spain, 2011, vol. 98, pp. 2564–2571. doi: 10.1109/ICCV.2011.6126544.
[41] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Underst.*, vol. 110, no. 3, pp. 346–359, Jun. 2008. doi: 10.1016/j.cviu.2007.09.014.
[42] P. R. Gunjal, B. R. Gunjal, H. A. Shinde, S. M. Vanam, and S. S. Aher, "Moving object tracking using Kalman filter," in *2018 Int. Conf. Adv. Commun. Comput. Technol., ICACCT 2018*, Haryana, India, Nov. 2018, pp. 544–547.
[43] S. K. Weng, C. M. Ku, and S. K. Tu, "Video object tracking using adaptive Kalman filter," *J. Vis. Commun. Image Represent.*, vol. 17, no. 6, pp. 1190–1208, 2006. doi: 10.1016/j.jvcir.2006.03.004.
[44] Soule, K. Salamatian, A. Nucci, and N. Taft, "Traffic matrix tracking using Kalman filters," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 3, pp. 24–31, Dec. 2005. doi: 10.1145/1111572.1111580.
[45] S. Park, S. Yu, J. Kim, S. Kim, and S. Lee, "3D hand tracking using Kalman filter in depth space," *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, pp. 1–18, Feb. 2012. doi: 10.1186/1687-6180-2012-36/FIGURES/21.
[46] X. Li, K. Wang, W. Wang, and Y. Li, "A multiple object tracking method using Kalman filter," *2010 IEEE Int. Conf. Inf. Autom., ICIA 2010*, Harbin, China, 2010, vol. 1, no. 1, pp. 1862–1866. doi: 10.1109/ICINFA.2010.5512258.
[47] Puertas, G. De-las-heras, and J. Fern, "Dataset: Roundabout aerial images for vehicle detection," *Data*, vol. 7, pp. 47, 2022.
[48] 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.
[49] Y. Chen and Q. Wu, "Moving vehicle detection based on optical flow estimation of edge," in *2015 11th Int. Conf. Nat. Comput. (ICNC)*, Zhangjiajie, China, Jan. 2015, pp. 754–758.
[50] H. Zhang and K. Wu, "A vehicle detection algorithm based on three-frame differencing and background subtraction," in *2012 5th Int. Symp. Comput. Intell. Des. Isc.*, Washington, DC, USA, 2012, vol. 1, pp. 148–151.
[51] R. Shahzad and A. Jalal, "A smart surveillance system for pedestrian tracking and counting using template matching," in *2021 Int. Conf. Robot. Autom. Ind., ICRAI 2021*, Xi'an, China, 2021.