This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3524453

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.Doi Number

R-CNN based Vehicle Object Detection via Segmentation capabilities in Road Scenes

Bisma Riaz Chughtai[1], Haifa F. Alhasson[2], Mohammed Alnusayri[3], Mohammed Alatiyyah[4], Hanan Aljuaid[5], Ahmad Jalal [1, 6], and Jeongmin Park[7]

[1] Faculty of Computing and AI, Air University, E-9, Islamabad 44000, Pakistan
[2] Department of Information Technology, College of Computer, Qassim University, Buraydah 52571, Saudi Arabia
[3] Department of Computer Science, College of Computer and Information Sciences, Jouf University, Sakaka 72388, Saudi Arabia
[4] Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 16278, Saudi Arabia
[5] Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah Bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia
[6] Department of Computer Science and Engineering, College of Informatics, Korea University, Seoul, 02841, South Korea
[7] Department of Computer Engineering, Tech University of Korea, 237 Sangidaehak-ro, Siheung-si 15073, Gyeonggi-do, Korea

Corresponding author: Jeongmin Park (e-mail: jmpark@tukorea.ac.kr).

This work was supported by the IITP (Institute of Information & Communications Technology Planning & Evaluation)-ICAN(ICT Challenge and Advanced Network of HRD)(IITP-2024-RS-2022-00156326), 50) grant funded by the Korea government (Ministry of Science and ICT). The authors acknowledge Princess Nourah Bint Abdulrahman University Researchers supporting Project number (PNURSP2024R54), Princess Nourah Bint Abdulrahman University, Riyadh 11671, Saudi Arabia.

ABSTRACT. In the realm of intelligent transportation systems, vehicle detection and classification stand as pivotal tasks. An effective traffic monitoring system should be capable of detecting, counting, and categorizing moving vehicles. Vehicle classification is a crucial task that can offer insights into road users and help make decisions to reduce congestion, for instance. This paper delves into advanced methodologies for detecting and classifying vehicles on roadways, addressing the limitations of traditional techniques that are often computationally intensive and data acquisition-sensitive. We propose a novel approach that leverages state-of-the-art machine learning and deep learning algorithms to enhance accuracy and efficiency. The proposed model comprises five stages. Initially, all images undergo preprocessing to reduce noise and enhance brightness. In the second stage, the foreground elements are extracted using segmentation techniques. The YOLOv8 algorithm is then used to process these segmented images to identify the vehicles inside them. Next, in the feature extraction phase, the detected vehicles are analyzed using Maximally Stable Estimated Features (MSER), Geometric features and Binary Robust Invariant Scalable Keypoints (BRISK) features. We employ the Recurrent Convolutional Neural Network (R-CNN) classifier for classification. Experimental results from two datasets demonstrate superior performance, with the model achieving an accuracy of 0.94% on the BITVehicle dataset and 0.98% on the Vehicle-OpenImage dataset. Additionally, a comparative analysis was conducted, showcasing the model's performance against the latest techniques in the field.

INDEX TERMS segmentation, BRISK, MSER, vehicle recognition, YOLOv8, recurrent convolutional neural network (RNN), Object detection, Deep learning.

I. INTRODUCTION

Intelligent Transportation Systems (ITS) have become vital for efficient traffic management in today's rapidly expanding urban environments. As cities grow in size and population, they face challenges such as traffic congestion, pollution, and safety concerns. ITS solutions aim to address these challenges by minimizing environmental impacts, enhancing road safety, optimizing traffic flow, and enabling real-time data collection and analysis. For urban roadways to function efficiently, these systems must integrate cutting-edge technologies that support dynamic, data-driven decision-making [1].

A core component of ITS is vehicle detection and classification, which plays a crucial role in monitoring traffic, reducing congestion, preventing accidents, and informing policy decisions. Accurate real-time detection and classification enhance vehicle flow, mitigate congestion, and enable dynamic adjustments to traffic signals [2]. Moreover, vehicle data collected by ITS can help predict and reduce

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3524453

high-risk scenarios, contributing to overall road safety. Additionally, this data supports decision-makers in shaping traffic laws and planning infrastructure, ultimately fostering safer, more efficient urban environments. Consequently, advancements in vehicle detection technologies are integral to the success of ITS frameworks [3].

However, current vehicle detection systems face several challenges. These include high computational costs, the need for extensive datasets, and vulnerability to external factors such as weather, lighting conditions, and occlusions. Traditional methods, while effective in controlled environments, often struggle under real-world conditions and require significant processing power. Recent advancements in computer vision and deep learning offer new opportunities to overcome these limitations, enabling vehicle detection systems to operate effectively across various environmental conditions [4][5].

To address the limitations of traditional approaches, researchers have increasingly turned to machine learning and deep learning techniques for vehicle detection and classification. Among these techniques, the YOLO (You Only Look Once) model in its various versions, Convolutional Neural Networks (CNNs), Single Shot Multibox Detectors (SSD), and Faster R-CNN have proven effective in tackling challenges such as low resolution and complex environments. These approaches have improved detection accuracy, making them suitable for applications like surveillance, parking systems, and traffic monitoring.

Despite their effectiveness, existing methods face several key challenges. Deep learning models such as YOLO are computationally intensive, limiting their real-time application in complex environments. Furthermore, these models often struggle with occlusion, where vehicles are partially hidden, leading to detection errors. Additionally, traditional R-CNN models have limitations in effectively segmenting vehicles from the background, particularly in cluttered or low-contrast road scenes. Although these models have been instrumental in advancing vehicle detection, there is a gap in the literature regarding the integration of advanced segmentation techniques with CNN-based architectures to overcome these challenges.

This research proposes a hybrid approach to fill this gap by combining robust segmentation methods with state-of-the-art deep learning models, specifically YOLOv8 and R-CNN. This approach aims to improve vehicle detection and classification accuracy while reducing computational costs and enhancing performance in environments with occlusions and other complexities. The solution includes a five-stage process, beginning with image preprocessing to reduce noise and enhance brightness, followed by segmentation to extract foreground elements. YOLOv8 is then used to detect vehicles in the segmented images, and advanced feature extraction techniques—including Maximally Stable Extremal Regions (MSER), Binary Robust Invariant Scalable Keypoints (BRISK), and geometric features—are applied to the detected vehicles. The final stage employs a Recurrent Convolutional Neural Network (R-CNN) to classify the vehicles based on the extracted features. This integration of techniques has led to superior accuracy, as demonstrated by the model's performance across multiple datasets, contributing to the advancement of ITS and traffic monitoring systems [6], [7].

This paper is structured as follows: Section II reviews the literature on vehicle detection and classification methods. Section III outlines the proposed vehicle detection and recognition system. Section IV discusses vehicle recognition via R-CNN. Section V details the experimental setup and analysis. Section VI presents research limitations and future work, while Section VII concludes the paper with key findings and suggestions for future research.

II. LITERATURE REVIEW

Numerous scholars have investigated object detection and classification through traditional methods. These conventional systems assess different features to categorize images and identify objects. Researchers have employed a diverse array of techniques for vehicle detection and recognition.

A) Traditional Vehicle Detection Algorithm

To identify vehicles, traditional detection algorithms often require manual feature design, relying on elements like lines, shadows, and edges. For example, detection methods based on prior knowledge use the contrast between vehicle shadows and surrounding pixels to extract vehicle position. However, these methods are significantly impacted by varying lighting conditions, which can alter the gray values of images, leading to false positives when shadows from non-vehicle objects mimic the size and shape of vehicles. Additionally, approaches that detect vehicles through brake lights are limited by similar lighting dependencies, resulting in poor detection performance. Consequently, these prior knowledge-based methods struggle with accuracy and are inadequate for high-precision detection tasks, especially in complex or variable environments.

Traditional object detection methods also include those founded on basic machine learning, which integrates machine learning algorithms with manually designed object features. These approaches use predefined characteristics to identify and classify objects, blending computational techniques with expert knowledge. However, their effectiveness can be limited by how well the features are designed and the complexity of the detection environment [8, 9]. Detection training can be improved by Decision tree (DT), Support Vector Machine (SVM) and HOG features. There are algorithms accessible for the tracking and detection tasks. This different approach not only boosts detection accuracy through a simple cascading mechanism but also expands the model size and computational demands. Although integrating shallow machine learning enhances vehicle detection accuracy somewhat, it remains highly reliant on feature selection. The significant modeling required for complex and ever-changing road conditions is a key factor that constrains its advancement. [10-13]. Support Vector Machines (SVM) have demonstrated effectiveness in multi-class vehicle classification, offering good generalization capability and performance for small, nonlinear datasets Cui et al. [14]. Multi-Layer Perceptron (MLP) combined with SMOTE resampling achieved 86.02% accuracy in vehicle make and model recognition A. Ahmad et al. [15]. D. T., Munroe, and M. G. Madden, [16] proposed classification using both single and multiple classes techniques have been explored, with features extracted from frontal vehicle views yielding high accuracy in determining vehicle make and model. M. A. Al-Shaher [17] these methods have applications in intelligent transportation systems, automatic monitoring, and autonomous driving. However, challenges remain, such as long training times for some models. D. Kleyko et al. [18] A comparison of logistic regression, neural networks, and SVM for vehicle classification using road surface vibrations and magnetic field disturbances found logistic regression to be the most effective, with a 93.4% classification rate. Another approach utilized the J48 decision tree algorithm to classify vehicles using data from a 3-axis magnetometer sensor, achieving nearly 100% accuracy K. Ying., et al [19]. Under certain circumstances, this kind of algorithm can recognize objects to some extent, but it has many drawbacks and is easily influenced by external occurrences [20-22].

B) Vehicle Recognition using Deep Learning

Recent research has focused on using deep learning methods for vehicle detection and classification in intelligent transportation systems. For instance, N. Rao, et al. [23] the study report focuses on the development of self-driving vehicles, which has sparked significant interest from various sectors, including automotive businesses, technology corporations, government agencies, and investors. The article presents an object detection system for road scenarios that employ deep learning techniques, notably You Only Look Once (YOLO) versions 3, 4, and 5. The system is designed to identify objects including on-road vehicles, traffic signals, cars, trucks, and buses by examining road data. The paper presents [24] a novel vehicle detection technique that makes use of a Multi-level Feature Pyramid Network (MFPN) for enhanced layer interaction and a multi-scale detector depending on the YOLOv7 network to effectively address challenges in real-world road scenes, such as low-resolution and complicated environments. To further increase localization accuracy, the technique additionally uses a new Focaler-SIoU bounding box loss function. On the UADETRAC dataset, this approach achieved an average precision of 73.9%, making it appropriate for real-time applications in challenging road conditions. To provide automatic vehicle tracking and identification, which is crucial for traffic surveillance systems, the study devised [25] efficient approaches. Vehicle detection in both stationary and moving situations has been effectively achieved with deep learning models such as YOLOv5 and SSD. Focus loss was added to the SSD model, significantly increasing feature acquisition speed and enhancing detection efficiency.

The paper emphasizes [26] how deep learning has significantly advanced the technology for classifying different types of vehicles, especially when considering contemporary intelligent transportation systems. The accuracy and effectiveness of vehicle detection have increased with the integration of pattern recognition and image processing. A dataset of 3000 images that included a balanced representation of four vehicle classes was used to train the model. The YOLOv5 model demonstrated remarkable performance metrics, as demonstrated by the trial results: 98.2% precision, 94.9% recall, 97.9% mean Average Precision (mAP) at 0.5, 92.8% mAP at 0.5:0.95, and 95.3% total accuracy. These scores highlight how well the model can identify and categorize various vehicle kinds. The suggested vehicle classification has two vehicle attributes [27] types and colors. Colors have seven classes, whereas types have four classes. Next, vehicle image classification is done using CNN. The experimental findings demonstrate CNN's strong performance in practical applications. To recognize and find items, we built a utilizing a current convolution neural network founded on fast regions to create a convolution neural network from scratch that identifies and recognizes them. In this study, [28] they evaluated three types of vehicles for classification and detection: buses, cars, and bicycles. Their method will take the complete take an image as input and produce a bounding box as the outcome with feature class probability estimates. The experiment results showed that the projected system can significantly increase detection accuracy. This study examines [29] three primary models: YOLOv4 (You Only Look Once v4), Improved SSD (Single Shot Multibox Detector), and Faster R-CNN (Faster Region-based Convolutional Neural Network). The analyses highlight the models' progressive gains in detection performance and accuracy. They also introduce EnsembleNet, a model that combines YOLOv5 and Faster R-CNN to attain higher average precision values. Another method combines CNN models with edge features to produce a quicker and more precise vehicle recognition. This study [30] increases image quality by correcting and eliminating extraneous information. Second, we appropriately prepare the training and validation sets using a waterfall strategy. Third, they use a Fourier transformation to extract object descriptors that represent the vehicle object's features. Finally, they use deep learning to localize and recognize car model items. The experimental findings show that their method has a high accuracy and identification rate regarding to vehicle detection. K. Vishal [31] examined several algorithms for identifying cars in aerial photos of UAVs, including CNN-SVM, Faster R-CNN, YOLO, and SSD. To recognize vehicles in traffic surveillance systems, V. Murugan et al. [32] developed a Region-based CNN

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3524453

(RCNN) strategy that included background estimation and removal approaches. These experiments show how deep learning works well for classifying and detecting vehicles, which has applications in surveillance, parking systems, and traffic monitoring. To evaluate performance and efficiency, the study [33] used a variety of criteria, including accuracy, sensitivity, and precision. Convolutional neural networks, or CNNs, especially Faster R-CNN, have demonstrated encouraging outcomes in this field. Research has indicated that for tasks involving vehicle identification and categorization, the accuracy ranges from 93% to 97.32%. To maximize performance, researchers have experimented with various CNN topologies, such as RCNN, AlexNet, VGGNet, and GoogLeNet. Vehicle area detection and vehicle categorization, which may include detecting vehicle types, brands, and license plates, are the two basic steps in these techniques [34-35]. X. Wang, et al. [36] These methods offer advantages in handling complex traffic data, adapting to various conditions, and learning from large datasets. S. U. Hassan et al. [37] On their own-generated dataset, the authors report reaching an outstanding accuracy of 97.3%. This high degree of accuracy suggests that their method is quite successful at identifying and categorizing cars, which is an important part of the study. Applying deep learning methods has demonstrated superior accuracy and robustness in contrast with typical machine learning techniques. These advancements contribute to improved traffic management, road safety, and the development of smart city infrastructure. After the vehicle is detected by the CNN recognize it by its category.

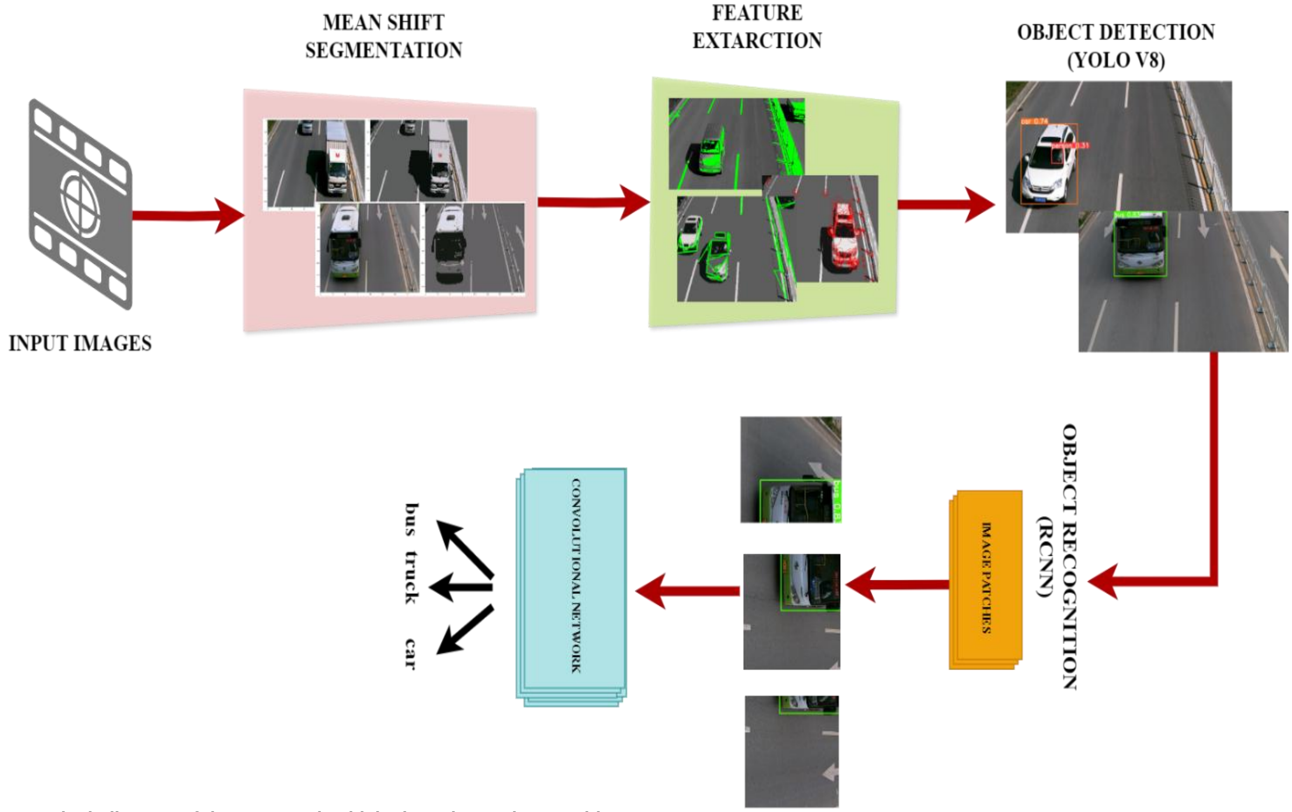


FIGURE 1. Block diagram of the proposed vehicle detection and recognition system.

III. THE PROPOSED VEHICLE DETECTION AND RECOGNITION SYSTEM

This article, introduces a model designed for detecting and recognizing vehicles. Our method begins with pre-processing, where we prepare all the images for analysis. Following this, we segment the images by labeling each pixel to identify and extract uniform regions. Next, we extract spatial features from these regions using various descriptors. These features are then combined to enhance the information used for vehicle detection. Finally, we use the extracted features to detect and recognize vehicles. Figure 1 offers a visual summary of our proposed model and its components.

A) Pre-processing

We have used the median filters a non-linear digital filtering technique to eliminate noise from the vehicle dataset, helping to enhance the image quality by reducing unwanted artifacts while maintaining important features like the edges and contours in the images. It works by moving a window (or kernel) across the image, and for each pixel in the image, it replaces the pixel value with the median value of the pixel intensities within the window. Median filters are widely used for removing impulse noise from digital images due to their simplicity and effectiveness [38-39]. Various types of median filters have been developed, including standard, weighted, iterative, recursive, directional, switching, and

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

adaptive median filters [40]. These filters aim to reduce noise while preserving image details and quality [41]. The refined pictures are displayed in Figure. 2 Equation (1), (2), (3), and (4) are used to describe the smoothed image resulting from the application of the median filter.

An input image J and a filter window (Kernel) of size $k \times k$:

$$J_k(i,j) = \left\{ J(m,n) \,\middle|\, m \in \left[i - \frac{k-1}{2}, i + \frac{k-1}{2}\right], n \in \left[j - \frac{k-1}{2}, j + \frac{k-1}{2}\right] \right\} \quad (1)$$

Where $J_k(i,j)$ represents the $k \times k$ submatrix of centered at pixel (i,j) in the original image J. The submatrix $J_k(i,j)$ includes all pixel values within the window, bounded by the indices m and n.

$$J_{k,sorted}(i,j) = sort(flatten(J_k(i,j))) \quad (2)$$

Here, flatten $(j_k(i,j))$ converts the submatrix $J_k(i,j)$ into a 1D vector, an $(.)$ sorts this vector in ascending order.

$$M(i,j) = median(J_{k,sorted}(i,j)) \quad (3)$$

Where $M(i,j)$ represents the median value of the sorted vector $J_{k,sorted}(i,j)$.

$$J_{filtered}(i,j) = M(i,j) \quad (4)$$

The pixel value at position (i,j) in the output filtered image $J_{filtered}$ is set to the median vale $M(i,j)$, calculated from the local neighborhood around (i,j).

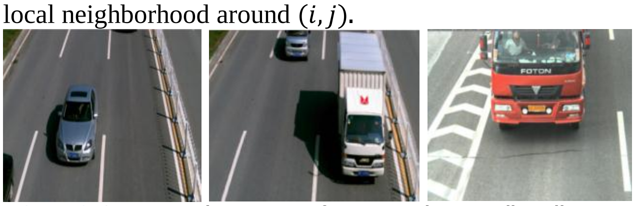


FIGURE 2. Contrast Enhancement of Images Using a Median Filter.

B) Segmentation

To minimize the computational complexity of the model, we applied semantic segmentation to the images before passing it to the CNN algorithm. For this purpose, we applied the combination 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.

1) Shift Mean Segmentation

Mean shift segmentation has proven effective for vehicle detection and tracking in image datasets. It can segment color vehicle images into candidate regions for number plate detection W. Jia, et al. [42]. When applied to Mean Shift and semi-dense varinace maps, S. Lefebvre and S. Ambellouis, [43] allow for the simultaneous detection and tracking of numerous vehicles in 3D, even with occlusions. Recent advancements include using anisotropic range kernels based on global image attributes to improve accuracy across different range spectra and incorporating local attributes to reduce over segmentation. Combined with a region adjacency graph for faster merging, these modifications have shown significant improvements in segmentation accuracy and processing speed compared to traditional mean shift methods H. Cho et al. [44]. All things considered, mean shift segmentation performs admirably various vehicle tracking and detection applications.

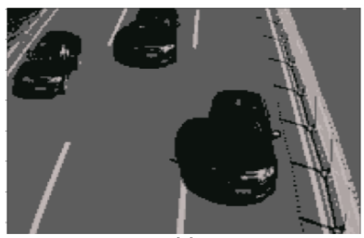
A technique dubbed "Modified Shift Mean," intended for picture segmentation, was utilized by us. This technique improves on the conventional mean shift method by iteratively altering pixel intensities according to how similar they are to nearby pixels, thus increasing the accuracy of image segmentation. For each pixel, we first compute a Gaussian-weighted kernel. This kernel calculates the effect that neighboring pixels have on the target pixel. In terms of math, this is represented as:

$$k(y,x) = exp\left(-\frac{\|I(y,x) - I(\acute{y},\acute{x})\|^2}{h^2}\right) \quad (5)$$

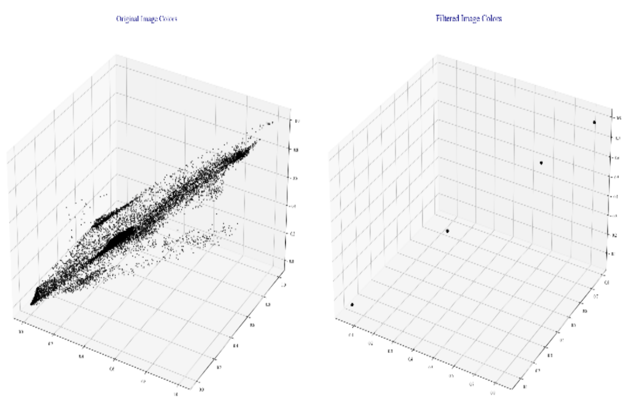
Here, $k(y,x)$ is the kernel, $I(y,x)$ is the intensity of the pixel at position (y,x), and h is a bandwidth parameter that controls the extent of this influence. Next, we compute a gradient vector $\nabla(y,x)$ by summing the weighted differences between the target pixel and its neighbors:

$$\Delta I(y,x) = \frac{\sum_{\acute{y},\acute{x}} (I(\acute{y},\acute{x}) - I(y,x)).k(\acute{y},\acute{x})}{\sum_{\acute{y},\acute{x}} k(\acute{y},\acute{x})} \quad (6)$$

This gradient then updates pixel's intensity, effectively shifting its value towards a mean that better represents its local neighborhood. This process is repeated for a set number of iterations, allowing the pixel values to converge to a stable state that highlights the segmented regions of the image. The outcome is a segmented image where regions with similar intensities are grouped together, providing clear and well-defined segments. Figure. 3 presents some examples of the resultant images.



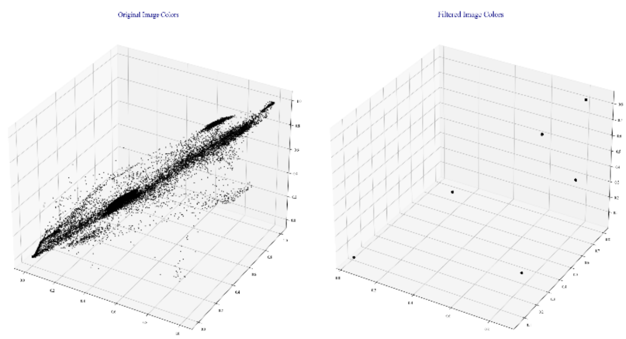
(a)



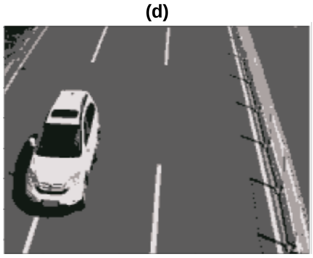
(b)

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3524453

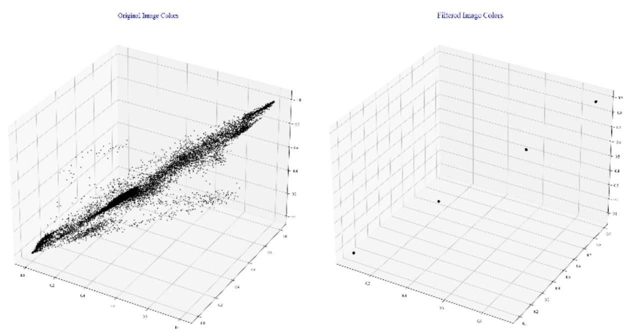
(c)



(d)



(e)



(f)

FIGURE 3. **Results of Shift Mean Segmentation with graph representation. (a) shows the segmented Car class result, with (b) displaying the corresponding graph visuals. (c) shows the segmented Truck class result, while (d) presents its graph visuals. (e) the segmented SUV class result is displayed, and (f) shows the associated graph visuals.**

C) Vehicle Detection

To improve the vehicle detection accuracy, they augment the YOLOv8 network [45], to locate and detect vehicles in image [46]. YOLOv8 is a major step forward in object recognition, providing better resilience and efficacy than earlier versions. It incorporates attention mechanisms and dynamic convolution, particularly improving small object detection [47]. YOLOv8 outperforms other mainstream models in multiple metrics, addressing issues such as sample imbalance and optimizing non-maximum suppression [48]. YOLOv8 improvements have demonstrated encouraging outcomes in some applications, such as the fast and accurate detection of moving objects [49]. Additional improvements have been made, like the addition of Context Attention Block (CAB) and changed Spatial Attention, especially concerning small item detection.

The YOLOv8 model, which has gained popularity for its speed and accuracy in real-time object detection tasks, was employed to identify vehicles. We employed the pre-trained YOLOv8 model, which is well suited to our requirements as it has been tuned for smaller and faster detections. Initially, we arranged the images into a folder and then configured the model to analyze every image within that folder. Bounding boxes are drawn around the predicted positions of cars in each image by the model. Following processing, images containing the identified vehicles were saved along with the results. Using a pre-trained YOLOv8 model, we could capitalize on the model's past learnings and optimize it for successful vehicle detection in the image. The resulting YOLOv8 image. Resultant mages of YOLOv8 are shown in Figure. 4.

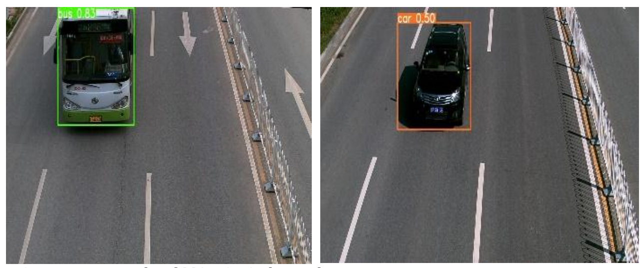


FIGURE 4. **Result of YOLOv8 detection.**

D) Feature Extraction

This section focuses on applying different machine-learning techniques to extract important features from different divided objects. These methods assist us in focusing on the most crucial information while eliminating extraneous or unimportant data. We improve the object recognition process by doing this. It is now simpler to recognize items in the photos thanks to the feature vector representation of the extracted features. Several feature extractors were used in this work, including Maximally Stable External Region (MSER), Geometric Features, and Binary Robust Invariant Scalable Keypoints (BRISK), to capture the essential features of the objects.

1) Maximally Stable External Regions (MSER)

This collection of studies examines the use of maximally stable extremal regions (MSER) for vehicle identification and classification across a range of imaging modalities [50]. WignerMSER, a variation of MSER for thermal infrared pictures, was introduced by A. Akula et al. [51] and showed better item categorization ability than standard MSER. Low resolution and poor visual quality were issued that S. Karim et al. [52] addressed in their examination of MSER for vehicle detection in satellite imagery. They achieved accuracies of 93% and 85.7%, respectively, when comparing MSER with dense sampling in a bag-of-features architecture for classifying vehicle types in thermal infrared imaging. Support Vector Machines (SVM) and MSER were coupled by J. B. Kim [53] to recognize and identify license plate information.

We have used Maximally Stable Extremal Regions (MSER) algorithm to detect and highlight stable regions in grayscale images from a specified input folder, saving the processed images with detected regions to an output folder. The MSER algorithm identifies regions that remain stable across varying intensity thresholds in the image, which are then marked as green dots on the image. The image converting to grayscale, detecting regions using the MSER algorithm, and then drawing and saving the highlighted regions as output images. The mathematical foundation of MSER involves identifying connected components in binary images that remain stable as the intensity threshold changes, with the regions of minimal area change over the threshold range being considered maximally stable. It is computed as by Equation (7) and (8).

$$Min(I(p)) > max\ (I(q)) \tag{7}$$

Where $I(p)$ represents the intensity of pixel at particular region. $I(q)$ shows the intensity of pixel at neighboring or surrounding region. When the region p has consistently higher intensities than its surroundings, it is identified as a maximally stable region as the threshold increases.

$$Min(I(p)) < max\ (I(q)) \tag{8}$$

when the region p has intensities consistently lower than its surroundings, it remains stable as the threshold decreases. Results are shown in Figure. 5.

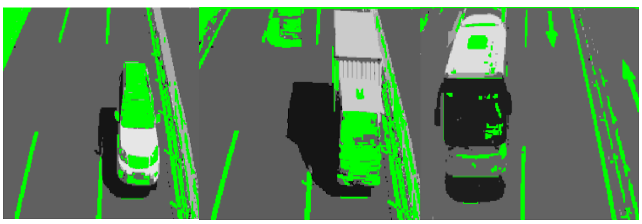


FIGURE 5. Extracted features through MSER

2) Geometric Features

Local geometric features are computed over the segmented objects [54]. Several approaches have been proposed, including 3-D feature extraction from 2-D SAR imagery using shadow information [55] and identified objects are categorized as geometrical shapes such as squares, rectangles, triangles, and circles [56]. Additionally, stochastic marked point processes have been employed to describe images using geometric objects, offering shorter computing times and broader applicability [57].

To processes an image to detect and analyze geometric features, specifically polygons. The image is first converted to grayscale and then blurred to reduce noise. Canny edge detection is applied to identify edges in the image. Contours are found based on these edges, and the code approximates these contours as polygons using the `cv2.approxPolyDP` function. Only significant polygons with more than three points and a considerable area are considered. The number of sides is stored for each detected polygon, and the centroid is calculated using image moments. The polygons are then drawn on the original image, and the image is displayed alongside a scatter plot that visualizes the centroids of the polygons. The scatter plot color-codes these centroids based on the number of sides of the polygons, allowing for a visual analysis of the geometric features in the image. It is calculated as in Equation (9), (10), (11) and (12).

$$P_{approx} = approxPolyDP(C, \epsilon) \tag{9}$$

Polygon approximation (Ramer-Douglas-Peucker algorithm) is used where P_{approx} is the approximated polygon in the object. C is original contour, a set of points representing an object's boundary. ϵ is the parameter that will control the approximation accuracy, the smaller the ϵ, the closer the approximation will be to the original contour.

$$cX, cY = \left(\frac{M_{10}}{M_{00}}, \frac{M_{01}}{M_{00}}\right) \tag{10}$$

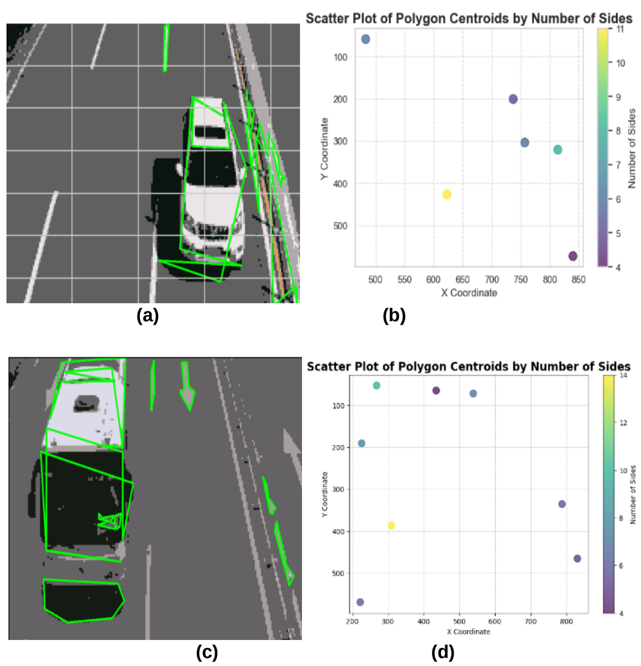
To calculate centroid of the (cX, cY) are the coordinates of the centroid of polygon. M_{00} is the area (zeroth moment), representing the polygon total area (total mass) of the polygon. M_{10} and M_{01} are the first along the x and y axes, respectively. The centroid is computed as the pixel coordinates weighted average, with the pixel intensities serving as the weights.

$$A = \left(\sum_{i=1}^{n-1}(x_i\, y_{i+1} - y_i x_{i+1}) + (x_n y_1 - y_n x_1)\right) \tag{11}$$

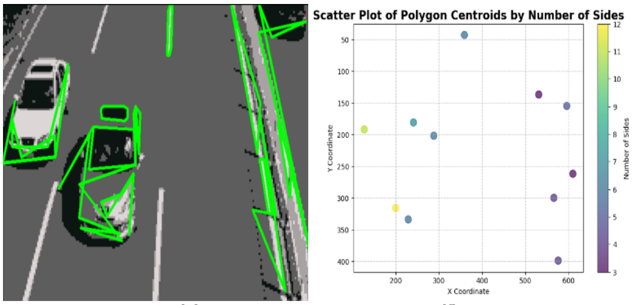
Where A is the area of polygon, (x_i, y_i) are the coordinates.

$$DrawPolygons(P_{approx}, color, thickness) \tag{12}$$

The function draws the approximated polygon P_{approx}, on the image with a specified color and line thickness. Extracted features and graph visuals are shown in Figure 6.



This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3524453



(e) (f)

FIGURE 6. Extracted geometric polygons of the SUV, Bus and Sedan are shown in (a),(c) and (e) while (b),(d) and (f) represent the centroids of the detected geometric shape.

3) Binary Robust Invariant Scalable Keypoints (BRISK)

Different researchers applied BRISK features [58-60]. For object recognition [61] directly, they have considered five different namely SURF (S), MINEIGEN (M), BRISK (B), FAST (F) and HARRIS (H). The main topics of this study are the robustness and invariance to picture noise, scale, translation, and rotation changes [62], the effects accuracy and recall values of blur, viewpoints, and illumination fluctuation have been studied [63], different methods for keypoints extraction [64], builds a BRISK-like sampling pattern fundamental to the scene surface by using a local parametrization of the scene surface generated from depth information. [65]. Mathematically, it is calculated by using Equation (13), (14), (15) and (16). Extracted keypoints are shown in Figure. 7.

$$K = \{(x_i, y_i)\} \tag{13}$$

Whew K is the set of keypoints and (x_i, y_i) are the coordinates of the ith keypoints.

$$D_i = f(I, (x_i, y_i)) \tag{14}$$

D_i is the descriptor for the ith keypoints, f is the function that computes the descriptor, and I is the image.

$$K_{sorted} = sort(K, key = response) \tag{15}$$

Their response sorts the keypoints K in the descending order.

$$K_{top} = \{K_{sorted}[i]\} \tag{16}$$

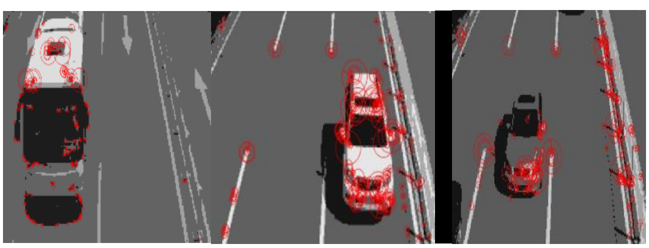


FIGURE 7. Result of BRISK KEYPOINT feature extractor

E) Vehicle Detection

The proposed approach incorporates several strategies to reduce computational costs, ensuring its viability for real-time vehicle detection and classification. The YOLOv8 algorithm is specifically chosen for its efficiency in balancing speed and accuracy. YOLOv8 integrates advanced attention mechanisms and dynamic convolution operations, which optimize feature extraction and reduce processing time, particularly for small object detection.

Moreover, the preprocessing stage minimizes data complexity by employing median filtering and shift mean segmentation, which enhance image quality while reducing noise and irrelevant background elements. This pre-segmentation step ensures that the subsequent stages of feature extraction and classification operate on simplified, high-quality inputs.

In addition, the use of targeted feature extraction techniques such as Maximally Stable Extremal Regions (MSER), Binary Robust Invariant Scalable Keypoints (BRISK), and geometric features eliminates redundant data processing, focusing computational resources on the most significant elements of the image. These optimizations collectively contribute to reducing computational overhead, as demonstrated by the low training and inference times reported in the experimental results.

IV. VEHICLE RECOGNITION via RCNN

The application of region-based Convolutional Neural Networks (RCNN) for vehicle recognition in traffic surveillance systems has been studied recently. Numerous tasks, including vehicle detection, logo recognition, and license plate identification, have been addressed using RCNN approaches [66-68]. These approaches outperform conventional machine learning techniques in identifying various vehicle classes and logos with high accuracy and efficiency [69]. A hierarchical RCNN technique that employs many layers of RCNNs for vehicle identification, character recognition, and license plate recognition has been described to handle complex scenarios. [70]. To classify vehicle types, M. I. Lakhal et al. [71] presented a novel end-to-end trainable framework combining convolutional and recurrent networks, exhibiting state-of-the-art effectiveness using open datasets. By considering the complete visual context, T. K. Cheang et al. [72] addressed the shortcomings of sliding window techniques and produced a unified ConvNet-RNN model for vehicle license plate recognition that led to noticeably better performance. Additionally, [73] concentrated on Convolutional neural networks for vehicle produce and model identification to assess frontal vehicle photos and achieved an astounding 98.7% accuracy. Earlier work [74] introduced the use of recursive neural networks to classify vehicles in image sequences by organizing tracking data into directed acyclic graphs. Faster Recurrent CNN [75] techniques have been developed to improve region selection and categorization efficiency. These methods have shown promising results across various datasets, highlighting the potential of neural network-based approaches for vehicle classification task in traffic monitoring application

We employed the R-CNN (Recurrent Convolutional Neural Network) architecture, renowned for its effectiveness in both object detection and classification tasks, to achieve high accuracy in vehicle classification. To accurately assess the framework's efficiency, we carefully sorted our dataset into separate folders for each class of vehicle and divided the photographs into training and validation sets. We normalized the pixel values to the range [0, 1] before training the model. This is a common preprocessing step that enhances stability during training. The ResNet50 backbone, which is pre-trained and has a deep structure of 50 layers organized in residual blocks to assist in alleviating the vanishing gradient problem and

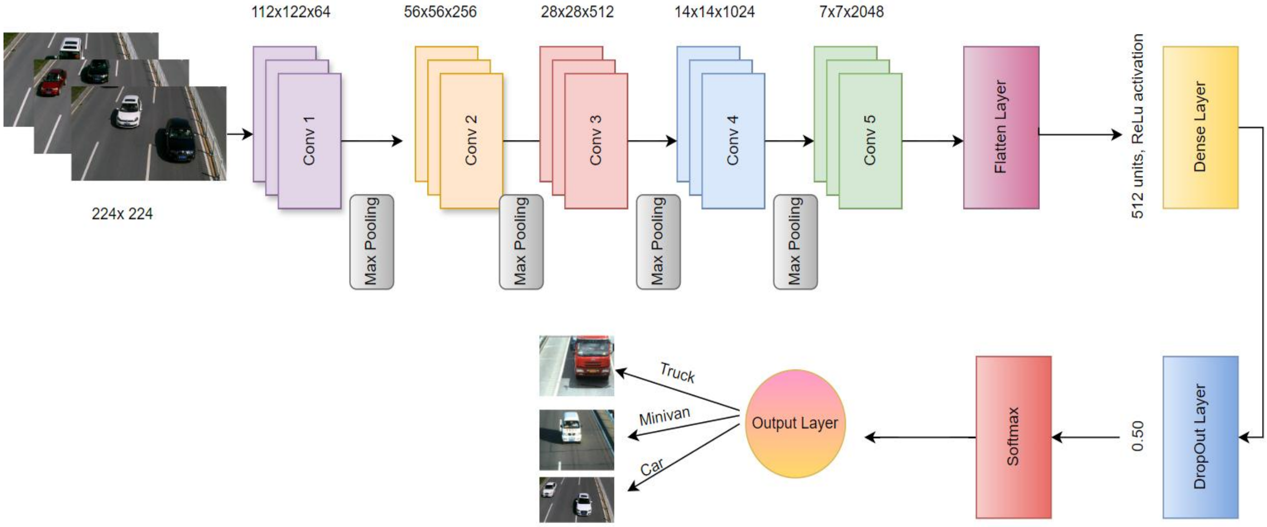


FIGURE 8. The architecture of R-CNN for vehicle recognition

detect complicated patterns, was used by the R-CNN model. We included a Recurrent Proposal Network (RPN) to create bounding boxes, or regions of interest (ROIs), inside the input images to modify the R-CNN for our objective. After extracting high-level features from these ROIs using the ResNet50 backbone, a RoI Pooling layer shrunk the feature maps in preparation for the final classification and bounding box regression tasks. A dense layer of 512 units with ReLU activation and an output layer with a softmax activation function was added to the R-CNN's personalized classification head, which These generated odds for the various car categories. For bounding box regression, a separate dense layer was added to predict the coordinates of the bounding boxes, enabling precise vehicle localization. We initially froze the ResNet50 backbone weights during training to preserve the features learned from the ImageNet dataset while training the newly added layers on our specific dataset. This fine-tuning approach allowed the model to retain general features while adapting to the specific characteristics of vehicle images (See Table 1). The model was compiled using the Adam optimizer, which dynamically adjusts the learning rate, with loss functions tailored for both classification (using sparse categorical cross-entropy) and bounding box regression. Early stopping was implemented to monitor validation loss, and halting training when no improvement was detected over several epochs to further enhance performance and prevent overfitting. Additionally, model checkpointing was utilized to preserve the top-performing model according to validation loss, assuring the retention of the most optimal configuration. Figure 8 shows the diagram of R-CNN. Algorithm 1 depicts the whole vehicle recognition architecture.

TABLE 1. Training/ Testing details of used datasets

Datasets	Total images	Training data (70%)	Testing data(30%)
BITVehicle	9852	6897	2955
Vehicle-OpenImage	627	440	187

Algorithm 1:

Pseudo-code for the proposed model
Input: RGB Images # Preprocessing the Image

```
# Apply Median Filter
preprocessed_image = preprocess_image(image)

# Apply Shift Means Segmentation
segmented_image = apply_shift_means(preprocessed_image)

# Feature Extraction
# Extract Geometric Features
geometric_features = extract_geometric_features(segmented_image)

# Extract BRISK Features
brisk_features = extract_brisk_features(segmented_image)

# Extract MSER Features
```

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3524453

```
mser_features = extract_mser_features(segmented_image)


# Object Detection using YOLOv8
yolo_model = load_yolo_v8_model('path/to/yolov8/model')
detected_objects = yolo_model.detect_objects(segmented_image)

# Extract Regions of Interest (RoIs) from Detected Objects
rois = extract_rois(detected_objects)

# Classification using R-CNN
# Define the R-CNN Model Architecture
def build_rcnn_model(input_shape, num_classes):
    model = Sequential()
    model.add(ResNet50(include_top=False, input_shape=input_shape))
    model.add(Flatten())
    model.add(Dense(512, activation='relu'))
    model.add(Dense(num_classes, activation='softmax'))
    model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])
    return model
rcnn_model = build_rcnn_model(input_shape=(224, 224, 3),
num_classes=6)

# Train R-CNN Model on Extracted RoIs
rcnn_model.fit(rois, labels, epochs=16, validation_split=0.2)

# Evaluate the Model
evaluation_results = rcnn_model.evaluate(test_data, test_labels)
```

V. EXPERIMENTAL SETUP AND ANALYSIS

On an Intel Core i7 PC running 64-bit Windows 11, Python 3.7 was used for the system evaluation and training. 16 GB of RAM and a 1.90GHz CPU power the machine. This section gives a summary of the study's experiments and findings in order to highlight the significance of the suggested model.

The proposed system is designed to mitigate the dependency on extensive datasets through the effective utilization of transfer learning. Pre-trained models, such as YOLOv8 and ResNet50 in the R-CNN architecture, leverage prior knowledge from large, general-purpose datasets. This transfer learning approach allows the model to achieve high accuracy even with moderate-sized datasets like BITVehicle and Vehicle-OpenImage.

A) BITVehicle DATASET

The BITVehicle dataset, created by the Beijing Institute of Technology (BIT), is a widely used resource in computer vision for vehicle detection, classification, and recognition tasks. It comprises pictures taken in various traffic situations, showing numerous kinds of vehicles, including cars, trucks, and buses, seen from different perspectives and in a range of illumination and environmental circumstances. It contains images captured in various traffic environments, featuring six vehicle classes: Bus (558), Microbus (883), Minivan (476), Sedan (5922), SUV (1392), and Truck (822). Bounding boxes and class labels are applied to every image, resulting in a complex and diversified dataset that is perfect for building models in applications such as autonomous driving, traffic monitoring, and intelligent transportation systems.

Furthermore, the use of data augmentation techniques ensures that the training dataset covers a wide range of variations, including scaling, rotation, and brightness adjustments. These augmentations simulate real-world conditions, increasing the robustness of the model. Future work will include expanding the dataset with additional synthetic samples generated under diverse weather and lighting conditions to further enhance model performance.

B) Vehicle-OpenImage DATASET

The Vehicle-OpenImage, is well-known. A thoughtfully selected portion of the Open Images collection, the Vehicles-OpenImage collection is designed especially for computer vision applications such as vehicle recognition and classification. The Vehicle-OpenImage dataset, which is accessible to the public, is used to assess our research. This dataset includes 627 RGB images with dimensions of 416 × 416, covering five classes: buses, cars, trucks, ambulances, and motorbikes. Figure 10 shows a few examples from the Vehicle-OpenImage collection.

C) EXPERIMENT 1: EXPERIMENTAL RESULTS USING PROPOSED APPROACH

We conducted experiments using the publicly available BITVehicle and Vehicle-OpenImage datasets. Our results, presented in the tables, show the accuracy of object recognition. We conducted 16 epochs for our experiments. The computational time for the BITVehicle dataset was 13,367.49 seconds, while for the Vehicle-OpenImage dataset, it took 279.01 seconds to complete the 16 epochs. Classification accuracy was extracted from both datasets, with BITVehicle containing six classes and Vehicle-OpenImage containing five classes. Our analysis reveals that the proposed method outperforms other leading object recognition algorithms with accuracy rates of 0.94% and 0.98%. The R-CNN model delivered exceptional performance on both datasets. Figure 9 and 10 provide the confusion matrix for object recognition on the BITVehicle and Vehicle-OpenImage datasets, respectively.

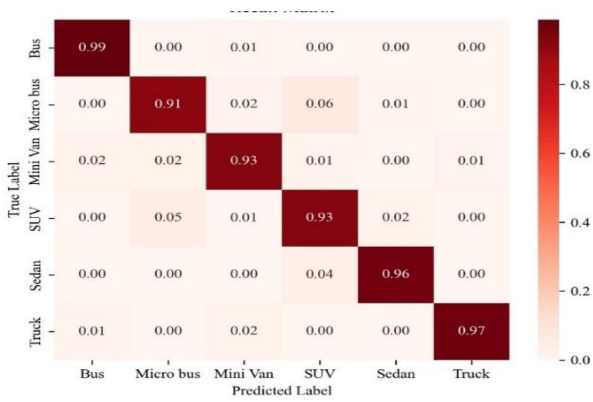


FIGURE 9. Confusion matrix for individual class accuracies over BITVEHICLE dataset using RCNN

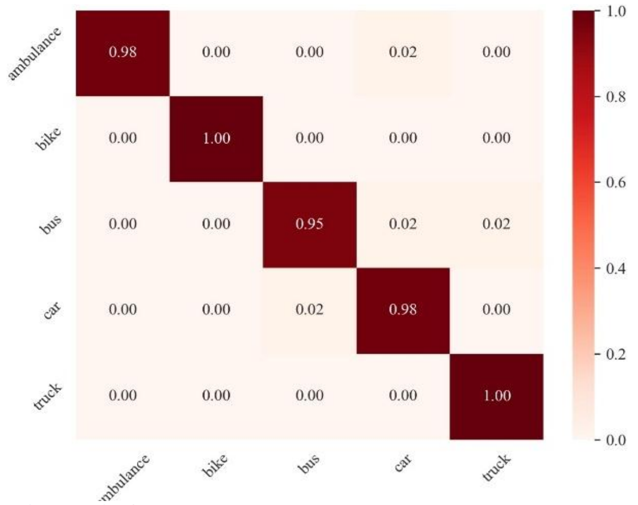


FIGURE 10. CONFUSION MATRIX FOR INDIVIDUAL CLASS ACCURACIES OVER VEHICLE-OPENIMAGE DATASET USING RCNN

D) EXPERIMENT 2: EXPERIMENTAL RESULTS FOR PRECISON, RECALL AND F1 SCORE

This section, presents precision, recall, and F1 score for each class in the datasets. These results highlight the high accuracy of our recognition algorithm in identifying complex objects. We calculated the precision, recall, and F1 scores for each object class using Equations (17), (18), and (19). The F1 score, also known as the F measure, is calculated as a weighted average of precision and recall, with values ranging from 0 to 1, where 1 indicates the highest precision

$$Precision = \frac{TP}{TP+FP} \tag{17}$$

$$Recall = \frac{TP}{TP+FN} \tag{18}$$

$$F1score = \frac{2(Pr*Rcl)}{Pr+Rcl} \tag{19}$$

In this case, Rcl corresponds to Recall and shows the percentage of true positive predictions to the total number of actual positives. Pre stands for Precision and indicates the accuracy of positive predictions. Here, FN stands for False Negatives and TP is for True Positives. The Precision, Recall, and F1 score examination metrics are included in Tables 5 and 6, along with the computation times for the datasets employed. Tables 2 and 3 demonstrate the F1, recall for each of the two datasets.

TABLE 2. PRECISION, RECALL, F1 SCORE OVER BITVEHICLE DATASET

Classes	Precision	Recall	F1 Score	Support
BS	0.98	0.99	0.98	332
MB	0.92	0.91	0.92	529
MV	0.93	0.93	0.93	270
SV	0.94	0.93	0.94	713
SN	0.83	0.96	0.89	85
TK	0.98	0.97	0.98	224
Acc			0.94	2153
Macro avg	**0.93**	**0.95**	**0.94**	2153
Weighted avg	**0.94**	**0.94**	**0.94**	2153

TABLE 3. PRECISION, RECALL, F1 SCORE OVER VEHICLE-OPENIMAGE DATASET

Classes	Precision	Recall	F1 Score	Support
AM	1.00	0.98	0.99	43
BK	1.00	1.00	1.00	37
BS	0.95	0.95	0.95	43
CR	0.98	0.98	0.98	98
TK	0.97	1.00	0.98	31
Acc			0.98	252
Macro avg	**0.98**	**0.98**	**0.98**	252
Weighted avg	**0.98**	**0.98**	**0.98**	252

E) EXPERIMENT 3: ROC (RECEIVER OPERATING CHARACTERISTIC) CURVES

The ROC curve is a graphical tool used to evaluate a classification model's performance, especially for binary classification problems. With different threshold settings, it aids in assessing the trade-off between the True Positive Rate (TPR) and False Positive Rate (FPR). The TPR (sensitivity) is plotted on the y-axis of the ROC curve against the FPR (1 specificity) on the x-axis. Usually, the curve begins at (0,0) and finishes at (1,1). The AUC shows the classifier's overall performance. Perfect categorization is shown by an AUC of 1, while no discrimination capacity (random guessing) is suggested by an AUC of 0.5. Equations (20) and (21) have been used to assess ROC curves for both datasets. The ROC curves are represented visually in Figures 11 and 12.

$$TPR = \frac{True\ Positive}{True\ Positive+False\ Negative} \tag{20}$$

$$FPR = \frac{False\ Positive}{False\ Positive+True\ Negative} \tag{21}$$

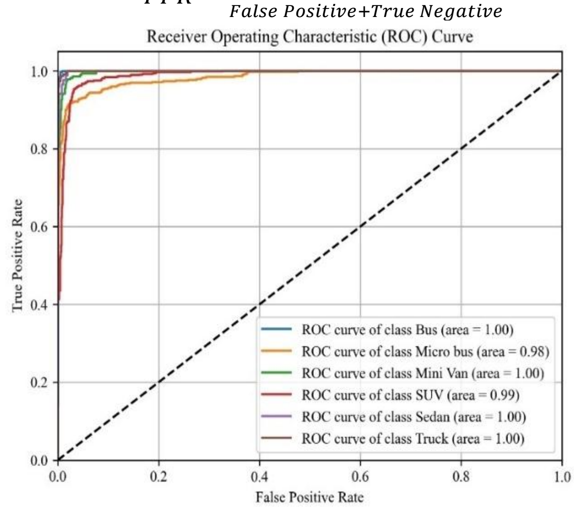


FIGURE 11. ROC Curve over BITVehicle Dataset

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3524453

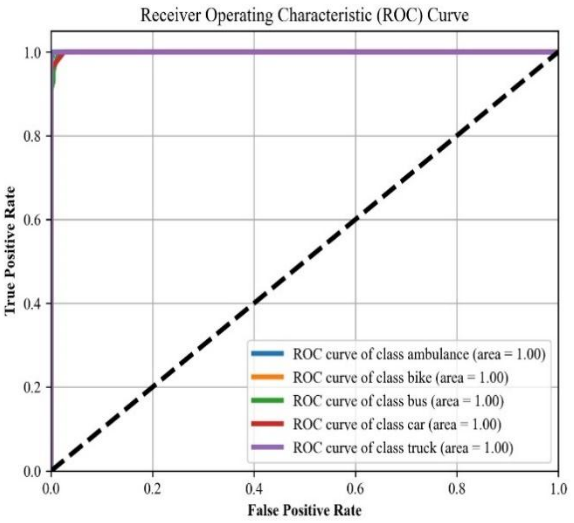


FIGURE 12. **ROC Curve over Vehicle-Open Image Dataset**

F) EXPERIMENT 4: COMPUTATIONAL COMPLEXITY OF TIME AND SPACE

In our study, we have thoroughly evaluated our model's time and space complexity of our model, documenting these details in the corresponding table. Additionally, we have calculated the Intersection over Union (IoU) values for the BITVehicle and Vehicle-OpenImage datasets, which are also presented in a separate table. The tables provide a clear comparison of the computational demands and the performance of our model across the two benchmark datasets, offering valuable insights for further optimization and application in real-world scenarios. Table 4 shows the time and space complexity for each dataset, and Table 5 and Table 6 illustrate the Intersection over union over both datasets.

TABLE 4. **COMPUTATIONAL COMPLEXITIES OF TIME AND SPACE**

Dataset	Time Complexity	Space Complexity	Computational time
BITVehicle	O(n^2)	O (1)	13,367.49
Vehicle-OpenImage	O(n^2)	O (1)	279.01

TABLE 5. **INTERSECTION OVER UNION OVER BITVEHICLE**

Classes	IOU	Classes	IOU
BS	0.89	**SV**	0.85
MB	0.92	**SN**	0.87
MV	0..85	**TK**	0.91
MEAN IOU=0.88|%			

TABLE 6. **INTERSECTION OVER UNION OVER VEHICLE-OPENIMAGE**

Classes	IOU	Classes	IOU
AM	0.93	**CR**	0.90
BK	0.92	**TK**	0.89
BS	0..87		
MEAN IOU=0.90|%			

G) DISCERNING OUR APPROACH TO CURRENT SYSTEMS

In [76] a method, 2D-DBM, introduces automatic vehicle classification using color images. It utilizes the power of a Deep Boltzmann Machine (DBM) and a bilinear projection technique. The proposed approach was compared with a Convolutional Neural Network (CNN) and a traditional DBM model that did not use bilinear projection. Experimental results highlighted the method's strengths, demonstrating a significant reduction in network size and processing time. However, the system has some limitations, only one dataset is used which limits the generalization. In contrast our system uses two datasets which increase the generalization of our system. In [77] the authors proposed Multi-Task Cascaded Convolutional Neural Network (MC-CNN) which addresses vehicle attribute recognition. It is a two-stage model that integrates an improved Faster R-CNN network (IFR-CNN) and an enhanced CNN network (ICNN). In contrast our system outperforms MC-CNN by offering a more efficient and streamlined architecture, maintaining high accuracy. It is robust against illumination changes and occlusions, making it highly effective in real-world conditions where MC-CNN may struggle. In [78] the paper presents a vehicle type classification approach using a semi supervised convolutional neural network (CNN) trained on frontal-view vehicle images. The system integrates sparse Laplacian filter learning with the CNN to automatically learn informative features from a large set of unlabeled data. By using this unsupervised feature learning, the network captures rich, discriminative vehicle characteristics. The system uses limited data due to which the reliance on sparse Laplacian filter learning may not always ensure optimal performance in highly complex or cluttered scenes where the learned features may not generalize well. In [81] ScaleDet is a scalable multi-dataset detector designed to generalize across heterogeneous datasets without the need for manual relabeling or complex optimizations. It achieves this by deriving a unified semantic label space through visual-textual alignment, which allows the model to map labels from various datasets based on their semantic similarities. However, its performance is dependent on the quality of training datasets, and challenges remain in semantic label alignment, class imbalance, and scalability, particularly as the number of datasets grows. In [82] the proposed system for vehicle detection and classification uses static pole-mounted roadside cameras to analyze traffic on busy streets. It classifies vehicles into three categories: light (motorbikes, bikes, tricycles), medium (cars, sedans, SUVs), and heavy (trucks, buses). A novel tracking algorithm, combined with majority voting, addresses challenges like motorbikes' sudden direction changes.

Compared to traditional methods, which often fail under real-world conditions due to high computational demands and sensitivity to external factors, the proposed hybrid approach demonstrates superior performance:

- Traditional R-CNN models struggle to segment vehicles in cluttered or low-contrast scenes, whereas the

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3524453

integration of advanced segmentation techniques with YOLOv8 achieves high accuracy even in complex environments.

- While earlier methods rely on extensive datasets and exhibit poor generalization to unseen conditions, the proposed transfer learning-based architecture achieves high accuracy with moderate-sized datasets.

Tables 7 and 8 provide a comparative summary of performance metrics between the proposed method and state-of-the-art approaches under different conditions, showcasing the effectiveness of our approach.

TABLE 7. ACCURACY RECOGNITION COMPARISON BETWEEN PROPOSED METHODS AND OTHER STATE OF ARTS METHODS

Author/Method	Mean Accuracy %	Classifier
D. F. S. Santos, et al. [76]	80.62	2D-DBM
H. Gong, et al. [77]	86.8	Cascaded Convolutional Neural Network (MC-CNN)
Z. Dong, et al.[78]	88.11	Semisupervised CNN
W. Sun, et al. [79]	90.1	Two stage classification strategy
M. N. Roecker, et al. [80]	93.90	CNN
Proposed Model	**94.0**	**R-CNN**

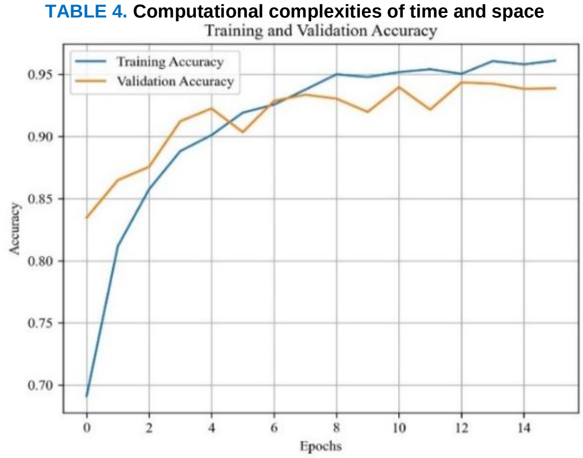
TABLE 8. COMPARISON OF RECOGNITION ACCURACY BETWEEN PROPOSED METHODS AND OTHER STATE OF ARTS TECHNIQUES

Author/Method	Mean Accuracy %	Classifier
Y. Chen, et al.[81]	76.2	Scalable multi-dataset detector (ScaleDet)
L. H. Pham, et al.[82]	95.3	GMM
A. S. Azim, et al.[83]	91	GentleBoost boosting
B. R. Chughtai, et al. [84]	96	CNN
Proposed Model	**98.0**	**R-CNN**

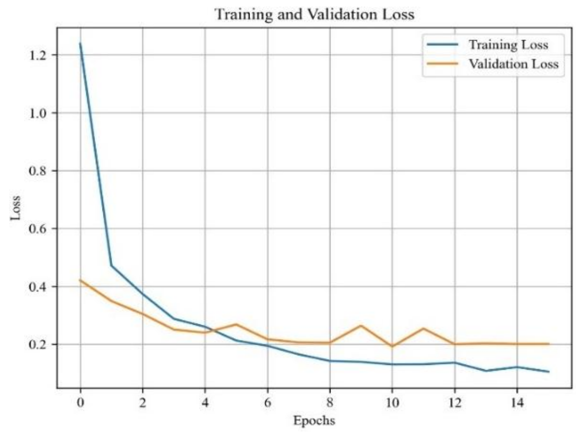
H) ANALYSIS OF TRAINING AND VALIDATION ACCURACY

The model's accuracy changes during training and validation, as depicted in the graph. The accuracy percentage is displayed on the y-axis, and the number of training epochs is displayed on the x-axis. By comparing the model's predictions with the actual values, training loss evaluates the model's performance on the training set. It aids in the model's comprehension of its level of improvement. The blue indicator of training accuracy indicates how effectively the model learns from the training set. The orange indicator of validation accuracy indicates how effectively the model generalizes to fresh, untested data. These metrics are shown graphically in Figure 13.

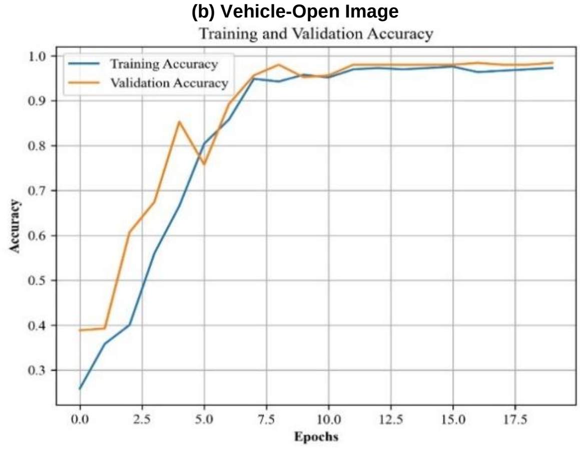
TABLE 4. Computational complexities of time and space



(a) BITVehicle



(b) Vehicle-Open Image



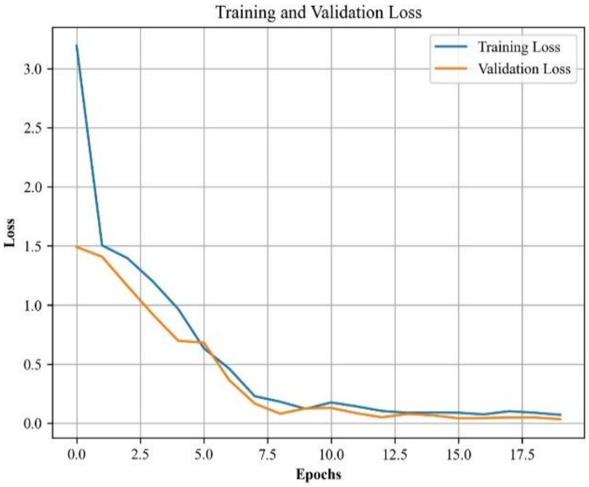


FIGURE 13. Data loss curves during training / testing

V. RESEARCH LIMITATIONS AND FUTURE WORK

In our proposed system, we achieved promising results in vehicle detection and classification using the BIT Vehicle and Vehicle-OpenImage datasets, our research does have some limitations. Although we effectively tackled illumination challenges such as images with high intensity during the day and low light at night our work does not yet cover extreme weather conditions, which remain unaddressed. Additionally, while we optimized our approach to handle various lighting conditions and moderate occlusions, the datasets primarily represent typical traffic scenarios, limiting our ability to fully validate robustness under severe environmental variations. The model's complexity also poses challenges for real-time deployment on devices with limited resources. Moreover, its performance is optimized for specific vehicle types and traffic conditions, raising concerns about its adaptability to diverse categories and regions. To overcome these limitations, future work will involve using or simulating datasets with harsh weather conditions, exploring lightweight architectures for better efficiency, and applying ensemble learning to improve robustness. We will also investigate domain adaptation techniques to enhance the model's ability to generalize across different real-world environments.

VI. CONCLUSION

This paper presents a method for detecting objects in complex images. The process eliminates the noise in the images then detects vehicles, and then extracts various features using machine learning techniques. R-CNN is then used for object recognition. Combining features is crucial for improving recognition rates compared to standard benchmarks. Our method outperforms other recognition systems in accuracy. The conclusion of the paper outlines the main areas for future development, indicating a commitment to further enhancing the vehicle's performance and capabilities in autonomous driving. We are committed to further developing our research with more feature extraction, and feature fusion for both general scene recognition and aerial applications. This will help make our system more reliable and effective in real-world scenarios.

REFERENCES

[1] M. Chowdhury and A. Sadek, Fundamentals of Intelligent Transportation Systems Planning, Norwood, MA: Artech House, 2003.
[2] Y. Wang, "Vehicle Detection and Classification for ITS: A Review," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 2, pp. 568-580, Feb. 2019.
[3] A. Fernandes, P. Silva, M. Torres, and L. Duarte, "The Role of Vehicle Classification in ITS," Journal of Transportation Technologies, vol. 8, no. 4, pp. 123-134, 2020.
[4] J. Smith and L. Brown, "Challenges in Traditional Vehicle Detection Systems," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 1234-1245, Mar. 2020.
[5] M. Johnson, "Limitations of Classical Computer Vision in Traffic Systems," Computer Vision and Image Understanding, vol. 189, pp. 102805, Jan. 2019.
[6] X. Wang, X. Chen, Z. Zhang, and H. He, "Vehicle type classification in intelligent transportation systems using deep learning," J. Intell. Fuzzy Syst., vol. (Preprint), pp. 1-12, 2024.
[7] D. Kleyko, R. Hostettler, W. Birk, and E. Osipov, "Comparison of Machine Learning Techniques for Vehicle Classification Using Road Side Sensors," in Proc. 2015 IEEE 18th Int. Conf. Intell. Transp. Syst., Gran Canaria, Spain, 2015, pp. 572-577, doi: 10.1109/ITSC.2015.100.
[8] C. Sun, C. Liu, X. Li, X. Zhao and J. Xu, "Embedded front vehicle detection system based on fusion of machine learning and traditional methods," in Proc. 2022 3rd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), Xi'an, China, 2022, pp. 719-725, doi: 10.1109/ICBAIE56435.2022.9985918.
[9] J. R. Kumar, B. Sujatha, and N. Leelavathi, "Automatic vehicle number plate recognition system using machine learning," IOP Conference Series: Materials Science and Engineering, vol. 1074, no. 1, p. 012012, Feb. 2021, IOP Publishing.
[10] K. Dimililer, Y. K. Ever, and S. M. Mustafa, "Vehicle detection and tracking using machine learning techniques," in Proc. International Conference on Theory and Application of Soft Computing, Computing with Words and Perceptions, Cham: Springer International Publishing, Aug. 2019, pp. 373-381.
[11] P. K. Bhaskar and S.-P. Yong, "Image processing-based vehicle detection and tracking method," in Proc. 2014 International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 2014, pp. 1-5, doi: 10.1109/ICCOINS.2014.6868357.
[12] M. Anandhalli, V. P. Baligar, P. Baligar, P. Deepsir, and M. Iti, "Vehicle detection and tracking for traffic management," IAES International Journal of Artificial Intelligence, vol. 10, no. 1, pp. 66, 2021.
[13] N. Chhadikar, P. Bhamare, K. Patil, and S. Kumari, "Image processing-based tracking and counting vehicles," in 2019 3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 335-339, IEEE.
[14] B. Cui, T. Xue, and K. Yang, "Vehicle recognition based on support vector machine," in 2008 International Symposium on Intelligent Information Technology Application Workshops, Shanghai, China, 2008, pp. 443-446, doi: 10.1109/IITA.Workshops.2008.23.
[15] A. Ahmad et al., "Vehicle recognition using multi-layer perceptron and SMOTE technique," in 2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH), Riyadh, Saudi Arabia, 2022, pp. 190-193, doi: 10.1109/SMARTTECH54121.2022.00049.
[16] D. T. Munroe and M. G. Madden, "Multi-class and single-class classification approaches to vehicle model recognition from images," in Proc. AICS, 2005, pp. 1-11.
[17] M. A. Al-Shaher, "Detection and classification of vehicle types using machine learning technology," Periodicals of Engineering and Natural

Sciences, vol. 7, no. 3, pp. 985-815, 2019.

[18] D. Kleyko, R. Hostettler, W. Birk, and E. Osipov, "Comparison of Machine Learning Techniques for Vehicle Classification Using Road Side Sensors," in 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Gran Canaria, Spain, 2015, pp. 572-577, doi: 10.1109/ITSC.2015.100.

[19] K. Ying, A. Ameri, A. Trivedi, D. Ravindra, D. Patel, and M. Mozumdar, "Decision tree-based machine learning algorithm for in-node vehicle classification," in 2015 IEEE Green Energy and Systems Conference (IGESC), Long Beach, CA, USA, 2015, pp. 71-76, doi: 10.1109/IGESC.2015.7359454.

[20] L.-W. Wang, X.-F. Yang, and W.-C. Siu, "Learning approach with random forests on vehicle detection," in 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), Shanghai, China, 2018, pp. 1-5, doi: 10.1109/ICDSP.2018.8631871.

[21] Z. Chen, N. Pears, M. Freeman, and J. Austin, "Road vehicle classification using support vector machines," in *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, vol. 4, Nov. 2009, pp. 214-218, IEEE.

[22] K. Saripan and C. Nuthong, "Tree-based vehicle classification system," in 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, Thailand, 2017, pp. 439-442, doi: 10.1109/ECTICon.2017.8096268.

[23] N. Rao, K. Rajendra Prasad, S. Siva Rao, R. Kumar Tammineni, M. Annavarapu, and U. Pyla, "An efficient object detection by autonomous vehicle using deep learning," Int. J. Electrical and Computer Engineering, vol. 14, no. 4, pp. 4287-4295, 2024, doi: 10.11591/ijece.v14i4.pp4287-4295.

[24] Z. Xiao and C. Huang, "A deep learning based vehicle detection method for road scenes," in *2024 IEEE 6th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, Chongqing, China, 2024, pp. 784-787, doi: 10.1109/IMCEC59810.2024.10575384.

[25] P. Ankireddy, S. Ceronmani, and D. V. Reddy, "Improving the efficiency of image processing with deep learning for vehicle detection and tracking," vol. 205, pp. 227, 2024, doi: 10.58532/v3bdio4p2ch4.

[26] N. A. Pethiyagoda, M. W. P. Maduranga, D. M. R. Kulasekara, and T. L. Weerawardane, "Deep Learning-Based Vehicle Type Detection and Classification," vol. 3, pp. 18-26, 2023, doi: 10.37394/232028.2023.3.3.

[27] W. Maungmai and C. Nuthong, "Vehicle Classification with Deep Learning," in 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS), Singapore, 2019, pp. 294-298, doi: 10.1109/CCOMS.2019.8821689.

[28] V. Vijayaraghavan and L. Mohan, "Vehicle Classification and Detection using Deep Learning," International Journal of Engineering and Advanced Technology, vol. 9, no. 1S5, pp. 24-28, 2019, doi: 10.35940/ijeat.a1006.1291s519.

[29] J. Chen, "Vehicle Detection and Tracking in Intelligent Transportation Systems Based on Deep Learning," Applied and Computational Engineering, 2024, doi: 10.54254/2755-2721/33/20230227.

[30] B. Mouna and O. Mohamed, "A Vehicle Detection Approach Using Deep Learning Network," in 2019 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), Tunis, Tunisia, 2019, pp. 41-45, doi: 10.1109/IINTEC48298.2019.9112137.

[31] K. Vishal, "A Novel Study for Identification of Vehicles Using Deep Learning Techniques," Vol. 10, no. 11, Nov. 2022, pp. 2320-2882.

[32] V. Muruganm V.Vijaykumar, and A. Nidhila, "A Deep Learning RCNN Approach for Vehicle Recognition in Traffic Surveillance System," in 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0157-0160, doi: 10.1109/ICCSP.2019.8698018.

[33] M. C. E. Orozco and C. B. Rebong, "Vehicular Detection and Classification for Intelligent Transportation System: A Deep Learning Approach Using Faster R-CNN Model," Platero, vol. 180, no. 1, p. 36551, 2019.

[34] M. Sheng, C. Liu, Q. Zhang, L. Lou, and Y. Zheng, "Vehicle Detection and Classification Using Convolutional Neural Networks," in 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS), Enshi, China, 2018, pp. 581-587, doi: 10.1109/DDCLS.2018.8516099.

[35] M. Pavan Kumar, Dr. K. Manivel, and N. Jayanthi, "Classification & Detection of Vehicles Using Deep Learning," International Journal of Trend in Scientific Research and Development (IJTSRD), vol. 4, no. 3, pp. 283-291, Apr. 2020, ISSN: 2456-6470.

[36] X. Wang, X. Chen, Z. Zhang, and H. He, "Vehicle type classification in intelligent transportation systems using deep learning," Journal of Intelligent & Fuzzy Systems, vol. (Preprint), pp. 1-12, 2024.

[37] S. U. Hassan, M. Z. Khan, H. U. Draz, and M. U. Ghani Khan, "Detection and Classification of Vehicles Based on Deep Semantic Features of Faster R-CNN," in 2019 International Conference on Innovative Computing (ICIC), Lahore, Pakistan, 2019, pp. 1-6, doi: 10.1109/ICIC48496.2019.8966733.

[38] A. Naseer, N. A. Almujally, S. S. Alotaibi, A. Alazeb, and J. Park, "Efficient Object Segmentation and Recognition Using Multi-Layer Perceptron Networks," CMC-COMPUTERS MATERIALS & CONTINUA, vol. 78, no. 1, pp. 1381-1398, 2024.

[39] P. Goyal and V. Chaurasia, "Application of median filter in removal of random valued impulse noise from natural images," in 2017 International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2017, pp. 125-128, doi: 10.1109/ICECA.2017.8203657.

[40] P. K. Garg, P. Verma, and A. Bhardwaz, "A survey paper on various median filtering techniques for noise removal from digital images," American International Journal of Research in Formal, Applied & Natural Sciences, vol. 7, no. 1, 2014.

[41] D. Anveshini, K. Gayathri, C. Kalyani, N. Bhuvaneswari, and M. S. Sana, "Comparative Analysis on Median Filter Categories for Accurate Noise Removal," in 2023 IEEE 12th International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 2023, pp. 384-390, doi: 10.1109/CSNT57126.2023.10134646.

[42] W. Jia, H. Zhang, and X. He, "Mean shift for accurate number plate detection," in Third International Conference on Information Technology and Applications (ICITA'05), Sydney, NSW, Australia, 2005, vol. 1, pp. 732-737, doi: 10.1109/ICITA.2005.176.

[43] S. Lefebvre and S. Ambellouis, "Vehicle detection and tracking using Mean Shift segmentation on semi-dense disparity maps," in 2012 IEEE Intelligent Vehicles Symposium, Madrid, Spain, 2012, pp. 855-860, doi: 10.1109/IVS.2012.6232280.

[44] H. Cho, S.-J. Kang, and Y. H. Kim, "Image segmentation using linked mean-shift vectors and global/local attributes," IEEE Transactions on Circuits and Systems for Video Technology, vol. 27, no. 10, pp. 2132-2140, Oct. 2017, doi: 10.1109/TCSVT.2016.2576918.

[45] X. Chen, "Vehicle object detection algorithm based on improved YOLOv8," in 2024 5th International Conference on Computer Vision, Image and Deep Learning (CVIDL), Zhuhai, China, 2024, pp. 1513-1516, doi: 10.1109/CVIDL62147.2024.10603727.

[46] N. Al Mudawi et al., "Vehicle detection and classification via YOLOv8 and deep belief network over aerial image sequences," Sustainability, vol. 15, no. 19, pp. 14597, 2023.

[47] R. Varghese and S. M., "YOLOv8: A novel object detection algorithm with enhanced performance and robustness," in 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 2024, pp. 1-6, doi: 10.1109/ADICS58448.2024.10533619.

[48] C. Jia, D. Wang, J. Liu, and W. Deng, "Performance optimization and application research of YOLOv8 model in object detection," Academic Journal of Science and Technology, vol. 10, no. 1, pp. 325-329, 2024.

[49] M. Safaldin, N. Zaghden, and M. Mejdoub, "An improved YOLOv8 to detect moving objects," IEEE Access, vol. 12, pp. 59782-59806, 2024, doi: 10.1109/ACCESS.2024.3393835.

[50] M. Talib, A. H. Al-Noori, and J. Suad, "YOLOv8-CAB: Improved YOLOv8 for real-time object detection," Karbala International Journal of Modern Science, vol. 10, no. 1, pp. 5, 2024.

[51] A. Akula, R. Ghosh, S. Kumar, and H. K. Sardana, "WignerMSER: Pseudo-Wigner distribution enriched MSER feature detector for object recognition in thermal infrared images," IEEE Sensors Journal, vol. 19, no. 11, pp. 4221-4228, Jun. 2019, doi: 10.1109/JSEN.2019.2900268.

[52] S. Karim, I. A. Halepoto, A. Manzoor, N. H. Phulpoto, and A. A. Laghari, "Vehicle detection in satellite imagery using maximally stable extremal regions," IJCSNS, vol. 18, no. 4, pp. 75, 2018.
[53] J. B. Kim, "MSER and SVM-based vehicle license plate detection and recognition system," in International Conference on Hybrid Information Technology, Berlin, Heidelberg: Springer Berlin Heidelberg, Aug. 2012, pp. 529-535.
[54] A. A. 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," *Multimedia Tools and Applications*, vol. 82, no. 9, pp. 13401-13430, 2023.
[55] K. Ji, X. Xing, H. Zou, and R. Li, "3-D geometric feature extraction of vehicle target from 2-D high resolution SAR imagery based on shadow information," in 2011 IEEE Conference on Radar, Communication and Electronics, 2011, pp. 1582-1585, doi: 10.1109/CIE-RADAR.2011.6159866.
[56] M. M. Hasan, N. A. Ibraheem, and N. M. Abdulhadi, "2D geometric object shapes detection and classification," Webology, vol. 19, no. 1, pp. 1689-1702, 2022.
[57] F. Lafarge, G. Gimel'farb, and X. Descombes, "Geometric feature extraction by a multimarked point process," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1597-1609, Sept. 2010, doi: 10.1109/TPAMI.2009.152.
[58] A. Naseer, N. Al Mudawi, M. Abdelhaq, M. Alonazi, A. Alazaib, A. Algarni, and A. Jalal, "CNN-based object detection via segmentation capabilities in outdoor natural scenes," IEEE Access, vol. xx, no. xx, pp. xx-xx, 2024.
[59] A. Jakubović and J. Velagić, "Image feature matching and object detection using brute-force matchers," in 2018 International Symposium ELMAR, Zadar, Croatia, 2018, pp. 83-86, doi: 10.23919/ELMAR.2018.8534641.
[60] A. Canclini, M. Cesana, A. Redondi, M. Tagliasacchi, J. Ascenso, and R. Cilla, "Evaluation of low-complexity visual feature detectors and descriptors," in 2013 18th International Conference on Digital Signal Processing (DSP), 2013, pp. 1-7, doi: 10.1109/DSP.2013.6624006.
[61] M. Ravikumar, S. Sampathkumar, M. C. Prashanth, and B. J. Shivaprasad, "Object recognition using SBMHF features," in International Conference on Intelligent Computing and Smart Communication 2019: Proceedings of ICSC 2019, Springer Singapore, 2020, pp. 973-983.
[62] Y. Liu, H. Zhang, H. Guo, and N. N. Xiong, "A fast-BRISK feature detector with depth information," Sensors, vol. 18, no. 11, pp. 3908, 2018.
[63] Ş. Işık, "A comparative evaluation of well-known feature detectors and descriptors," International Journal of Applied Mathematics Electronics and Computers, vol. 3, no. 1, pp. 1-6, 2014.
[64] M. Kashif, T. M. Deserno, D. Haak, and S. Jonas, "Feature description with SIFT, SURF, BRIEF, BRISK, or FREAK? A general question answered for bone age assessment," Computers in Biology and Medicine, vol. 68, pp. 67-75, 2016.
[65] M. Karpushin, G. Valenzise, and F. Dufaux, "Improving distinctiveness of BRISK features using depth maps," in 2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC, Canada, 2015, pp. 2399-2403, doi: 10.1109/ICIP.2015.7351232.
[66] V. Murugan, V. R. Vijaykumar, and A. Nidhila, "Vehicle logo recognition using RCNN for intelligent transportation systems," in 2019 International Conference on Wireless Communications Signal Processing and Networking (WiSPNET), Chennai, India, 2019, pp. 107-111, doi: 10.1109/WiSPNET45539.2019.9032733.
[67] M. Yahya, R. A. Reddy, K. Al-Attabi, I. H. Mohammed, and B. Arunasree, "Object detection and recognition in autonomous vehicles using fast region-convolutional neural network," in 2023 International Conference on Integrated Intelligence and Communication Systems (ICIICS), Kalaburagi, India, 2023, pp. 1-5, doi: 10.1109/ICIICS59993.2023.10421358.
[68] M. W. Nafi'i, E. M. Yuniarno, and A. Affandi, "Vehicle brands and types detection using Mask R-CNN," in 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 2019, pp. 422-427, doi: 10.1109/ISITIA.2019.8937278.
[69] M. V., V. V. R., and N. A., "A deep learning RCNN approach for vehicle recognition in traffic surveillance system," in 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0157-0160, doi: 10.1109/ICCSP.2019.8698018.
[70] C. Tu and S. Du, "A hierarchical RCNN for vehicle and vehicle license plate detection and recognition," International Journal of Electrical and Computer Engineering, vol. 12, no. 1, pp. 731, 2022.
[71] M. I. Lakhal, S. Escalera, and H. Cevikalp, "CRN: End-to-end Convolutional Recurrent Network Structure Applied to Vehicle Classification," in VISIGRAPP (5: VISAPP), 2018, pp. 137-144.
[72] T. K. Cheang, Y. S. Chong, and Y. H. Tay, "Segmentation-free Vehicle License Plate Recognition Using ConvNet-RNN," arXiv preprint arXiv:1701.06439, 2017.
[73] Y. Ren and S. Lan, "Vehicle Make and Model Recognition Based on Convolutional Neural Networks," in 2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2016, pp. 692-695, doi: 10.1109/ICSESS.2016.7883162.
[74] G. Monfardini, "Recursive Neural Networks for the Classification of Vehicles in Image Sequences," in Biological and Artificial Intelligence Environments: 15th Italian Workshop on Neural Nets, WIRN VIETRI 2004, Springer Netherlands, 2005, pp. 359-366.
[75] M. Chate and V. Gohokar, "Vehicle Detection Using Faster Recurrent Convolution Neural Network," in ICDSMLA 2019: Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications, Springer Singapore, 2020, pp. 1190-1195.
[76] D. F. Silva Santos, G. Botelho De Souza, and A. N. Marana, "A 2D Deep Boltzmann Machine for Robust and Fast Vehicle Classification," in 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Niteroi, Brazil, 2017, pp. 155-162, doi: 10.1109/SIBGRAPI.2017.27.
[77] H. Gong, Y. Zhang, F. Liu, and K. Xu, "Vehicle recognition using multi-task cascaded network," in Fifth Symposium on Novel Optoelectronic Detection Technology and Application, vol. 11023, pp. 1259-1265, March 2019, doi: 10.1117/12.2524123.
[78] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle Type Classification Using a Semisupervised Convolutional Neural Network," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 4, pp. 2247-2256, Aug. 2015, doi: 10.1109/TITS.2015.2402438.
[79] W. Sun, X. Zhang, S. Shi, J. He, and Y. Jin, "Vehicle Type Recognition Combining Global and Local Features via Two- Stage Classification," Mathematical Problems in Engineering, vol. 2017, Article ID 5019592, 2017, doi: 10.1155/2017/5019592.
[80] M. N. Roecker, Y. M. G. Costa, J. L. R. Almeida, and G. H. G. Matsushita, "Automatic Vehicle Type Classification with Convolutional Neural Networks," in 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP), Maribor, Slovenia, 2018, pp. 1-5, doi: 10.1109/IWSSIP.2018.8439406.
[81] Y. Chen *et al*., "ScaleDet: A Scalable Multi-Dataset Object Detector," *(CVPR)*, pp. 7288-7297, 2023.
[82] L. H. Pham et al. "A Robust Multiclass Vehicle Detection and Classification Algorithm for Traffic Surveillance System," pp. 1-6, 2020.
[83] A. S. Azim, A. Jafri and A. Alkhairy, "Multiclass Vehicle Classification Across Different Environments," (IJACSA), pp. 12(3), 2021.
[84] B. R. Chughtai and A. Jalal, "Traffic Surveillance System: Robust Multiclass Vehicle Detection and Classification," *2024 5th International Conference on Advancements in Computational Sciences (ICACS)*, Lahore, Pakistan, 2024, pp. 1-8, doi: 10.1109/ICACS60934.2024.10473304.

BISMA RIAZ CHUGHTAI received an M.S. Computer Science degree from Air University, Islamabad. He is currently a Research Assistant at the Intelligent Media Centre. His research interests include machine learning, deep learning, camera and sensor-based gesture recognition, and virtual reality.

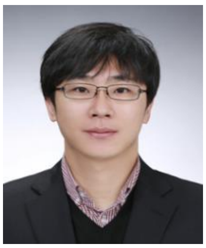
Haifa F. Alhasson (Member, IEEE) received a BSc in Computer Science from Qassim University, an MSc in Computer Science from King Saud University, Saudi Arabia, and her PhD in Computer Science from Durham University, UK. She is currently working as Assistant Professor at Computer College, Qassim University, Saudi Arabia. Her interdisciplinary

This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2024.3524453

research focuses on image processing and machine learning. In particular, her research aimed to understand better machine learning in object detection and recognition required for variable tasks.

HANAN ALJUAID received the B.S. degree from KAU University, and the M.S. and Ph.D. degrees in computer science from UTM University, in 2014. She is currently with the Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah bint Abdul Rahman University (PNU), Saudi Arabia. She has published numerous articles in pattern recognition, the IoT, and data science. Her research interests span both computer vision and NLP. Much of her work has been on improving the understanding, design, and performance of pattern recognition, mainly through the application of data mining and machine learning. She has given numerous invited talks and tutorials.

AHMAD JALAL is currently an Associate Professor from Department of Computer Science and Engineering, Air University, Pakistan. He received his Ph.D. degree in the Department of Biomedical Engineering at Kyung Hee University, Republic of Korea. Now, he was working as Post-doctoral Research fellowship at POSTECH. His research interest includes multimedia contents, artificial intelligence.



JEONGMIN PARK is an Associate Professor with the Department of Computer Engineering at Tech University of Korea in South Korea. Before joining Tech University of Korea in 2014, he was a senior researcher at Electronics and Telecommunications Research Institute (ETRI) and a research professor at Sungkyunkwan University, in South Korea. He received his Ph.D. degree in College of Information and Communication Engineering from Sungkyunkwan University in 2009. His research interests focus on high-reliable autonomic computing mechanism and human-oriented interaction system.