Minimax Theory and its Applications
Volume 11 (2026), No. 3s, 409 to 439

Real-Time Traffic Congestion Prediction Using Multi-Stage Image Enhancement and Dynamic Weighted Object Count Estimation

Sanathkumar S[1], Dr. S. Raja[2]

[1]PhD Research Scholar, Department of Computer Science, Rathinam College of arts and Science, Bharathiar University, Coimbatore, Email: sanathkumar1874@gmail.com, ORCID: 0009-0003-3426-4565
[2]Assistant Professor and Head, Department of Computer Science, Rathinam College of Arts & Science, Bharathiar University, Coimbatore, Email: hod.ct@rathinam.in, ORCID: 0009-0007-4523-8425

Abstract
The research is a structural analysis of traffic capable of offering support to the validity of real-time surveillance during severe weather conditions. The modern traffic monitoring systems are usually prone to noise interference, segmentation error and poor consistency when tracking vehicles and this leads to poor estimation of congestion. The general purpose of the work is to have a robust pipeline that operates through several stages to generate high-clarity denoising, proper segmentation, robust classification and dynamic density estimation. The methodology is supplemented with adaptive frequency-selective noise suppress (AFSNS) to enhance the quality of the image and then Context-Aware Region-Aggregated Segmentation (CARAS) is applied to isolate the vehicle regions that are structurally consistent. Vehicle Enhanced Discriminator and Classifier Network (VEDCN) improves the multi-class recognition and Dynamic Weighted Object Count Estimation (DWOCE) computes the lane density using weighted vehicle density and motions. The experimental findings indicate that the denoising, segmentation as well as recognition performance have increased to a great extent where denoising attains 92.39 percent denoising accuracy, segmentation attains 92.97 percent segmentation accuracy and general density estimation attains 95.6579 percent accuracy. The hybrid system provides enhanced clarity, structural information is maintained and also becomes consistent with changing lighting and traffic conditions. In conclusion, the proposed framework offers flexible, high accuracy, and scalable solution to the intelligent transportation systems in such a way that the real-time tracking and forecasting of the congestion would be more precise in the intricate road environment.

Keywords: Classification, Denoising, Segmentation, Traffic Analysis, Traffic Density Estimation**,** Vehicle Detection.

1. Introduction

The growing significance of traffic surveillance has been owed to the high rate of expansion in the urban transportation systems and the increasing need to have an efficient road monitoring system. The contemporary traffic setting is usually full of noise, motion blur, imbalance of illumination and vehicle occlusion, and thus, precise detection, segmentation, classification, and density estimation is extremely difficult. The overall idea of the present research is the development of a powerful multi-stage pipeline that would improve the quality of images, detect meaningful areas on a vehicle, and provide credible traffic information. The issue is that the current monitoring systems fail to perform in low-light scenarios, heavy traffic and dynamically changing situations that produces inconsistent results. The noise suppression [1], adaptive denoising [2], sensor-based traffic prediction [3], CNN-based preprocessing [4], template-matching surveillance methods, GMM-based detection, graph-optimized multi-scale recognition, UAV-based tracking model, adaptive traffic signal framework, and multi-sensor fusion methods to multi-object tracking have been explored in many studies. Despite significant progress made in these works, high sensitivity to noises, a lack of full segmentation, slow classification, and low real-time accuracy continue to be a problem.

The proposed system will be used to mitigate these limitations by proposing a hybrid system that integrates frequency-selective denoising, contextual segmentation, multi-scale vehicle classification, and dynamic weighted density estimation. AFSNS module contributes to a better understanding of structure in the presence of a large amount of noise to facilitate more stable downstream processing. CARAS segmentation also uses entropy feature and morphological refinement in isolating vehicle regions so that foreground objects are accurately separated. The VEDCN recognizer enhances the identifications of different types of vehicles using discriminative features extraction, and the DWOCE algorithm provides congestion prediction using weighted

vehicle types, lane coverage ratio, and motion indicators. This multi-stage methodology is more robust through varying levels of illumination, weather and density of traffic, which is used in real time traffic examination in intelligent transportation systems.

Contribution

The research presents a single traffic analysis system that incorporates denoising, segmentation, classification, and density estimation on-the-fly surveillance systems. The AFSNS also allows noise suppression in frequency-selective fading allowing enhancement of clarity in low-lit and noisy images. CARAS gives context sensitive vehicle segmentation with the help of entropy-based region extraction and morphological refinement. VEDCN is a multi-class vehicle recognition system that provides discrimination features improvement in the presence of occlusion and motion variations. DWOCE calculates the realistic congestion levels based on weighted vehicles types and lane-based coverage indicators. The system shows significant gains in precision, stability, and flexibility to the scene in a wide range of traffic situations. In general, the framework provides support on high-performance traffic monitoring in application of intelligent transportation.

Organization

Section 1 presents the concepts, driving power, surrounding issues, and problem statement involved in the real-time surveillance of traffic. Section 2 presents the background study by reviewing existing solutions in denoising, segmentation, classification, and traffic analysis, highlighting the current state-of-the-art approaches and their limitations.. Section 3 describes the proposed methodology, frequency-selective denoising, context-aware segmentation, vehicle classification, and dynamic density estimation. Section 4 explains the experimental context and testing procedure that was adopted to test the framework. Section 5 concludes that the proposed approach performs better than the current techniques in all the metrics considered, showing to be more accurate, preserving the structure and working in general.

2. Background Study

Omar et al. [11] (2020) Omar et al. have suggested a cascaded deep learning architecture based on license plate detection and recognition using convolutional networks. The experiment indicated that it detected at an enhanced rate with varying lighting levels and mask conditions. Nevertheless, the approach was disadvantaged by extremely noisy images and real-time usages. In general, the method improved the recognition accuracy, but it was not strong to harsh environmental noise.

Shim et al. [12] (2024) Shim et al. were interested in deep learning noise-resistant license plate detection. Their model was more accurate in low-light and high-noise cases. Computational complexity was the limitation of the research that rendered the research less viable when it comes to real-time applications. Findings indicated successful denoising integration and recognition in order to improve performance.

Ness [13] (2025)Ness proposed a comparative research of vehicle detection and recognition system of smart surveillance. The research showed that multi-scale detection techniques and their accuracy are improved. Efficiencies in real-time and dynamic traffic scenario management were identified. Results highlighted system strength and flexibility in regulated settings.

Rodriguez-Esparza et al. [14] (2024) Rodriguez-Esparza et al. suggested the hyper-heuristic method of the vehicle segmentation in road traffic surveillance. The findings revealed greater success in segmenting in various traffic scenarios. The discrepancy was low performance in vehicles with high clutter or overlap. The technique was strong and requires further development in the urban environment.

Qureshi et al. [15] (2023) Qureshi et al. used multi-label semantic segmentation with filter-based tracking to perform intelligent traffic monitoring.This authors enhanced accuracy of detection and tracking. Disadvantages were the computing capacity of large video streams. In general, the system succeeded in enhancing the segmentation and dynamic object tracking.

Handoko et al. [16] (2021) Handoko et al. improved traffic sign detection by color and shape segmentation as a pre-processing phase. The research indicated a high level of enhancement of detection with dissimilar light. Stereotypes still existed in cases of obscured or half visible signage. It was found to result in efficiency but little ability to withstand extreme conditions.

Luo et al. [17] (2021) Luo et al. suggested MOD-AT to accomplish traffic video motion object detection using adaptive thresholding. The system was more successful in detection and false positives were reduced.

Nevertheless, it had problems with heavy traffic or overlapping objects. The methodology improved adaptive segmentation albeit it needs a combination with tracking in order to provide complete traffic analyses.

Gulati et al. [18] (2019) Gulati and Srinivasan examined the image processing techniques of intelligent traffic management. This authors emphasized the performance of preprocessing pipelines in the detection of vehicles. There were weaknesses such as old-fashioned algorithms and low scalability. Findings focused on basic enhancements and were not very adaptable in real time.

Safaei et al. [19] (2021) Safaei et al. proposed an adaptive pixel-level road crack detector. The method maximized precision of detection and reduced the false positives. The difference lied in real-time implementation in a difficult traffic condition. In general, the adaptive segmentation was successful in monitoring infrastructure.

Hussain et al. [20] (2024) Hussain and Alam used Einstein operators and ANFIS to segment road cracks. It was found that the results were precise in different surface conditions. The scaleability to long roadways and traffic video feeds were added to the gap. Their approach enhanced dynamic segmentation but this authorshad to be combined with traffic analysis tools.

Almujally et al. [21] (2025) Almujally et al. suggested UAV-based traffic surveillance with RNN and Swin transformer as an indicator of dynamism in the environment. Findings had suggested good accuracy of multi-object detection. High computational requirements and restrictions on the deployment of UAVs were among the limitations. In general, the technique was strong in respect of aerial surveillance.

Anitha et al. [22] (2025) Anitha and Kumar created a modular system of license plate recognition with the help of traditional preprocessing and lightweight OCR. The experiment indicated high recognition without exaggeration in computing. The difference was in complicated lighting and obscuration cases. A combination of lightweight OCR was confirmed to be useful in real-time applications.

Naseer et al. [23] (2024) Naseer et al. have used CNN-based object detection and segmentation of outdoor natural scenes. Findings proved better detection and segmentation accuracy. Disadvantages included that it was sensitive to changes in lighting and motion. The algorithm improved the object localization but had to be optimized to be used in high-speed situations.

Yusuf et al. [24] (2024) Yusuf et al. optimized the pixel labeling and particle filters to detect and track UAV vehicles. There was an enhanced accuracy in the movement of targets in diverse conditions. The difference was in the computation power of long-duration monitoring. In general, the technique was able to efficiently monitor numerous vehicles on aerial photographic evidence.

Cristiani et al. [25] (2020) Cristiani et al. came up with ATRIP, a traffic classification architecture through image processing. The system was successful in classifying types of vehicles in traffic scenes in the cities. Other limitations comprised ineffective performance in congested traffic conditions. Findings were valid in laboratory settings but needed to be made more robust.

Rafique et al. [26] (2023) Rafique et al. applied pyramid pooling and filter-based tracking in the case of aerial traffic monitoring. The findings depicted a high detection and classification accuracy. The difference was real-time processing of a large-scale UAV footage. In general, the approach sufficiently incorporated multi-scale characteristics in classification.

Khan et al. [27] (2022) Khan et al. introduced an intelligent adaptive system of smart city traffic management based on machine learning. Findings showed that there were improvements in traffic flow prediction and vehicle classification. Examples of limitations were the dependence on the historical data and sensor accuracy. All in all, the system was dynamic but it needed real-time validation.

Won.[28] (2020) examined intelligent traffic monitoring systems to classify vehicles. The results identified the positive and negative aspects of current AI-based classification methods. The gap involved integration issues and scalability issues. Findings indicated that better real-time adaptive classification solutions were required.

Table 1: Summary of Background Study on Traffic Monitoring, Detection, and Segmentation Techniques

Ref. No	Author(s)	Year	Concept	Research Gap	Limitation	Result
29	Yusuf et al.	(2024)	Target detection and classification	Limited evaluation under	Computational cost for high-	Achieved high detection and

			using EfficientDet + CNN over UAV imagery	dense urban traffic and occlusions	resolution UAV videos	classification accuracy for moving vehicles in UAV data
30	Al Mudawi et al.	(2023)	Vehicle detection & classification via YOLOv8 + Deep Belief Network on aerial sequences	Real-time performance on long-duration sequences not fully tested	High memory and processing requirement	Effective vehicle detection and classification with improved accuracy in aerial images
31	Thenge & Kamble	(2025)	Real-time traffic sign recognition and classification using OpenCV	Tested on limited traffic sign dataset	Performance may drop under occlusions or low visibility	Demonstrated accurate, lightweight, real-time recognition and classification of traffic signs
32	Bisio et al.	(2022)	Systematic review of drone-based road traffic monitoring systems	Lack of unified evaluation metrics across studies	Comparative performance not quantified	Provided comprehensive survey of UAV-based traffic monitoring technologies and applications
33	Sindhu et al.	(2025)	AI-driven smart traffic control using CNN + YOLO for urban mobility	Limited evaluation in mixed traffic environments	Real-time latency under heavy traffic not addressed	Showed effective traffic control and vehicle classification using CNN and YOLO in urban settings
34	Shokravi et al.	(2020)	Review of vehicle classification techniques and smart vehicle-assisted systems	Integration of AI with real-time vehicular networks not fully explored	Focused mostly on classification; lacks real-time testing	Summarized strengths and weaknesses of existing AI-based vehicle classification techniques
35	Valencia et al.	(2024)	Impact of preprocessing on offline vehicle counting using YOLO	Limited preprocessing methods tested; effect on accuracy varies	Offline approach, not real-time	Preprocessing significantly improved YOLO-based vehicle counting accuracy

In this table 1, the significant contributions made by other researchers who have conducted studies on the field of noise reduction, vehicle detection, segmentation, and smart traffic monitoring are summarized. All the works cited emphasize the development of preprocessing, feature extraction, or recognition schemes and pinpoint the current shortcomings. The comparison shows that the similar challenges include noise sensitivity, occlusion problem, lighting imbalance and lower real-time reliability. The knowledge acquired in the framework of this background research informs the creation of a more adaptive and powerful structure of traffic analysis.

3. Proposed Methodology

A coordinated plan of the joint methodology of denoising, segmentation, classification, and density estimation is given, within one traffic-analysis framework. The initial step is to enhance visuality in order to prevent noise and retain fine details in order to be utilized in the further steps. This is then followed by substantial region extraction which enables the extraction of vehicle areas with precision in various light and motion conditions. Then, a classification module with special meaning of vehicle types is carried out with the assistance of reinforced feature representation. Finally, dynamic density estimation plan quantifies the lane coverage as well as traffic congestion with great level of accuracy in a broad assortment of traffic circumstances.

3.1 Dataset

https://www.kaggle.com/datasets/niloykantipaul/dataset-for-vehicle-detection-by-neural-network

This dataset is called Vehicle Detection in Neural Network on Kaggle and was developed by Niloy Kanti Paul and has 11,000+ annotated vehicle images that were taken in more than 100 urban and rural sites. The data set is particularly devoted to the training of neural networks (particularly CNNs) on the recognition of various types of vehicles in a variety of traffic and environmental conditions. It is crowdsourced, and it covers diverse opinions, thus it is helpful to create a strong object-detection model. Its data size is approximately 1 GB which is quite a lot but not excessive. And it is made to be used in practical implementations such as traffic violation detection and smart cities.

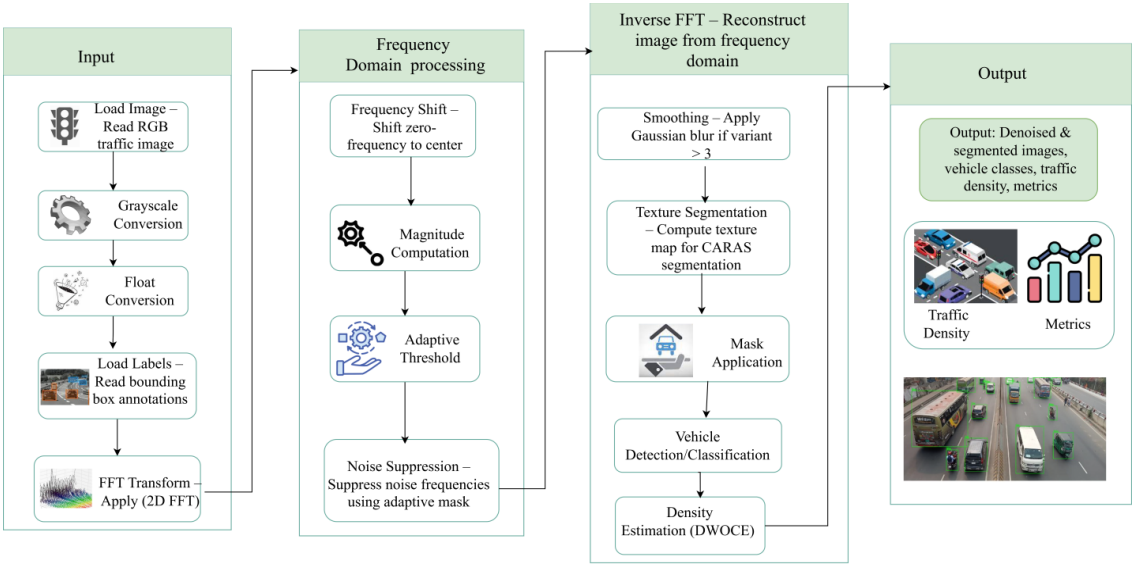


Fig.1 Overall Architecture

In Figure 1 it starts with the loading of RGB traffic images and their conversion to grayscale to apply a 2D FFT, which transforms the images to the frequency domain. Magnitude computation, frequency shifting, adaptive thresholding and noise-suppression masks are used in this area to eliminate unwanted high-frequency artifacts. The processed result is then a reconstructed denoised spatial image by the inverse FFT this is then smoothed , where needed. A segmentation map based on texture is then created on CARAS and a mask applied to isolate vehicle areas. Lastly, the system does vehicle detection/classification and calculates the traffic density measures (DWOCE) to generate clean segmented outputs and analysis results.

3.2 Adaptive Frequency-Selective Noise Suppression (AFSNS) for Real-Time Traffic Image Denoising

Adaptive Frequency-Selective Noise Suppression (AFSNS) is a denoising technique that is created to analyze a frequency-based aspect of each image and suppress the noise without corrupting the characteristics of the image. It identifies noise bands based on frequency decomposition, is dynamically responsive to respond to different disturbances, and it preserves edges, textures and structural integrity as well. Where noise is introduced by lighting variation in traffic monitoring applications due to weather conditions and limitations of the sensors, the AFSNS takes the raw image to the spectral domain and separates noise association with valuable information through spectral transformation. This is to identify the noise frequencies based on variance between frames in time and this enables the noise to be suppressed selectively before filtering. The adaptive thresholds are those that ensure that the noise only to do with the variations are differentiated, yet the shapes of the vehicles and their delimitations and contextual components are maintained. The method removes blurring, unwarranted smoothing and preserves significant visual features that are critical in identifying, partitioning and estimating the density of traffic in real-time.

The non-uniform noise that occurs is also a curse of the traditional filter and fixed-kernel denoising in situations where the noises change dynamically depending on the traffic stream or change in the light conditions. Wide frequency bandwidths are filtered using traditional frequency filters and important structural details are lost. Frame discrepancies in continuous video analysis are also brought by temporal instability and affect a downstream task. AFSNS is able to overcome those hurdles through the adaptive noise band selection rather

than fixed filtering and offer accuracy in denoising. Computation of temporal variation in frame-wise computation assists frequency masking in which the chance to suppress noise selectively is supplied. Structural information is not lost by the loss of pixel clarity or object boundaries to get. Under this approach, continuous, reliable, and real-time calculation of the traffic density in complex situations takes place.

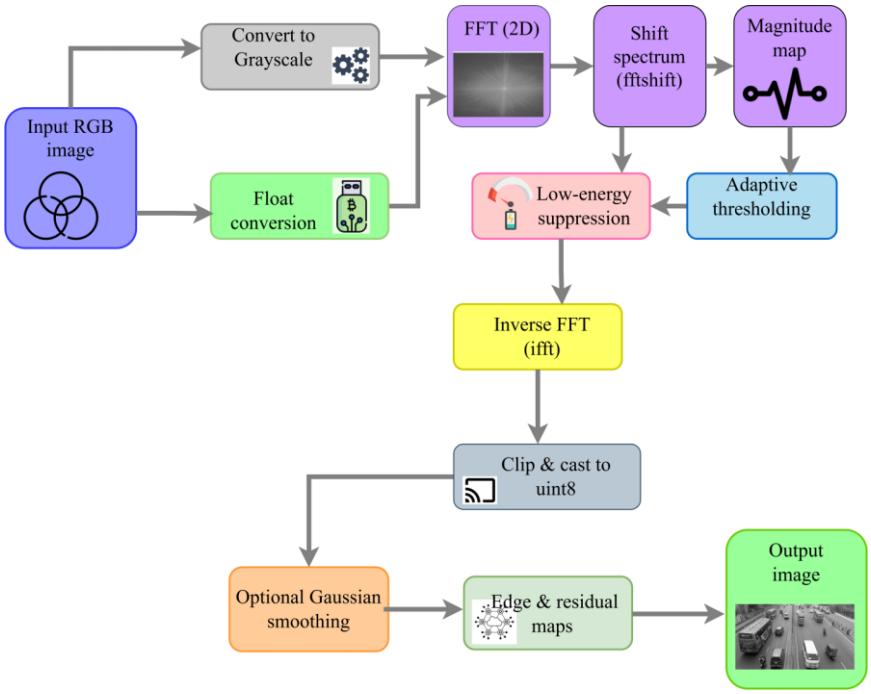


Fig. 2 Denoising Architecture using Adaptive Frequency-Selective Noise Suppression (AFSNS)

In Figure 2It starts by converting the RGB picture into grayscale then takes the RGB picture and transforms it to the frequency domain using the 2D FFT and lastly shifts the spectrum in a bid to enhance analysis. It is generating magnitude map, adaptation thresholding and low-energy suppression is applied to remove noise elements. With the assistance of the inverse FFT, the frequency data which has been cleaned up is reconstituted and converted into an 8-bit image. The smoothing of result and optional extraction of edge and residual map are used to result in an enhanced output image, which is denoised.

This equation transforms the RGB image into a greyscale by adding all the three color channels to a single intensity value.

$$GS(x,y) = 0.299\,R(x,y) + 0.587G(x,y) + 0.114\,B(x,y)\text{----------(1)}$$

In equation (1)$R(x,y)$ Red channel pixel intensity,$G(x,y)$ Green channel pixel intensity,$B(x,y)$ Blue channel pixel intensity$GS(x,y)$ Resulting grayscale pixel intensity It gives the R, G and B values perceptual weights with a preference on the green color to calculate a luminance-based grayscale value.

Transforms image from spatial domain to frequency domain.

$$F(u,v) - \sum_{z=0}^{M-1}\sum_{y=0}^{N-1} G\,(x,y)e^{-j2\pi\left(\frac{ux}{M}+\frac{uy}{N}\right)}\text{----------(2)}$$

In equation (2)$G(x,y)$Grayscale image in spatial domain, $F(u,v)$Frequency representation at coordinates(u,v),M,N Image dimensions,j imaginary unit. It sums all pixel values multiplied by complex exponential basis functions to compute the frequency component at each coordinate (u,v).

Extracts frequency strength to deduct noise regions.

$$M_{ag}(u,v) = \sqrt{Re(F(u,v))^2 + Im(F(u,v))^2}\text{----------(3)}$$

In equation (3)$M_{ag}(u,v)$Magnitude of the frequency bin$Re(F)$ Real part of the Fourier transform,$Im(F)$ Imaginary part of the Fourier transform. It combines the squared real and imaginary parts of the FFT and takes their square root to determine the strength of each frequency bin.

This equation calculates a frequency domain adaptive threshold of noise suppression.

$$T = \mu(Mag) \times (1 + 0.2 \times (v - 3)^2) \text{----------(4)}$$

In equation (4)T Threshold to noise masking,$\mu(Mag)$ Average of the magnitude spectrum,v parameter which controls the aggressiveness of noise. It upscales the average magnitude by a factor proportional to $(v-3)^2$) in order to permit a noise-masking strength that can be adjusted.

This equation applies attenuation which is threshold-based to reduce weak or noisy frequency bands.

$$F_s(u,v) = \begin{cases} 0.3.F(u,v), & if\, M_{ag}(u,v) < T \\ F(u,v), & otherwise \end{cases} \text{----------(5)}$$

Equation (5)$F_s(u,v)$Filtered frequency value$F(u,v)$Original frequency component$T$ Adaptive threshold $M_{ag}(u,v)$ Magnitude at frequency location(u,v). It boosts low-frequency components 0.3 in those frequencies that are below the adaptive threshold T, and it maintains stronger components.

Algorithm 1: AFSNS_Denoising

```
INPUT:
    Image I (RGB format)
    Variant v (default = 3)
  STEP 1: Convert image to grayscale
    G ← Convert_RGB_to_Grayscale(I)
  STEP 2: Convert data type for frequency processing
    G_float ← Convert_To_Float(G)
  STEP 3: Apply 2D FFT to move into frequency domain
    F ← FFT2(G_float)
  STEP 4: Shift frequency components to center
    F_shift ← FFT_Shift(F)
  STEP 5: Compute magnitude spectrum
    Mag ← Absolute(F_shift)
  STEP 6: Compute adaptive threshold
    T ← Mean(Mag) * (1 + 0.2 * (v - 3) / 2)
  STEP 7: Create noise mask based on threshold
    Mask ← Mag < T
  STEP 8: Suppress noise frequencies
    F_suppressed ← F_shift
    For all positions where Mask = TRUE:
      F_suppressed[pos] ← 0.3 * F_shift[pos]
  STEP 9: Apply inverse FFT to reconstruct image
    G_recon ← InverseFFT(FFT_Unshift(F_suppressed))
  STEP 10: Clip to valid intensity range & convert
    D ← Clip_And_Convert(G_recon, 0, 255)
  STEP 11: Optional smoothing based on variant
    IF v > 3:
      kernel_size ← 3 + 2*(v-3)
      D ← Gaussian_Blur(D, kernel_size)
  RETURN D
OUTPUT:
    Denoised Image D
END
```

The algorithm1 converts images of traffic into the frequency domain to extract and remove noise and retain the necessary structural information is described in the pseudocode. The conversion into grayscale makes the feature space simple and enables the frequency-related processing to detect disturbances better. Adaptive thresholding is used to ensure that the strength of suppressing is dynamically adjusted according to the intensity

of the noise and the level of the variants. Noise-dominant parts of the frequency content are only down-sampled leaving significant edges and textures untouched. The resulting output is a denoised image which can be used to detect appropriate traffic density within real-time conditions.

3.3 Context-Aware Region-Aggregated Segmentation(CARAS) for Traffic Density Analysis

The proposed algorithm is the integration of an adaptive segmentation algorithm based on CARAS and entropy-driven texture algorithm to segment vehicles accurately in a very broad variety of lighting and visibility conditions. The traffic surveillance images are normally made up of vehicles of various shapes and shadows in which there is a variation in illumination and reflections that cannot be separated easily. The denoised traffic frames are then first converted to grayscale to increase the brightness on the texture and structural distinction and the entropy analysis is applied to emphasize places with a high relative intensity difference that are typically occupied by vehicles. Adaptive thresholding on the basis of local statistics is used such that only the valuable texted parts are retained, morphological processing are utilized to enhance the smoothness of the continuity of the objects by removing noise and maintaining the connection vehicle bodies. It is a dynamic segmentation mechanism that varies according to the day, dusk, and night light changes and the vehicle limits can be stored, and this vehicle can be accurately localized, and the estimation of the traffic density is enhanced.

The conventional segmentation algorithms are also prone to high rates of noise, motion blur, overlapped cars and non-homogeneous light and, consequently, loss of structural clarity and misdirected boundaries to objects. Fixed-threshold methods usually treat shadows, lane markings/reflections as foreground object and edge-based operators can tend to chop vehicles into parts when the brightness is extremely low or are in reflections. CARAS eliminates these defects using the entropy based region extraction as compared to gradient edges alone. By making morphology context-sensitive, region completeness is enhanced and unattached false positives are removed with cluttered backgrounds. The shape-preserving segmentation of dense or occlusive clusters of vehicles will be offered by region aggregation. Hierarchical adaptation works on the boundaries of the vehicles with the assistance of the texture and not only the pixel intensity. All these enhancements lead to a noisy tolerant and steady performance that may be deployed in the environment of real-world vehicles surveillance to identify vehicles in real-time and analyze traffic density.

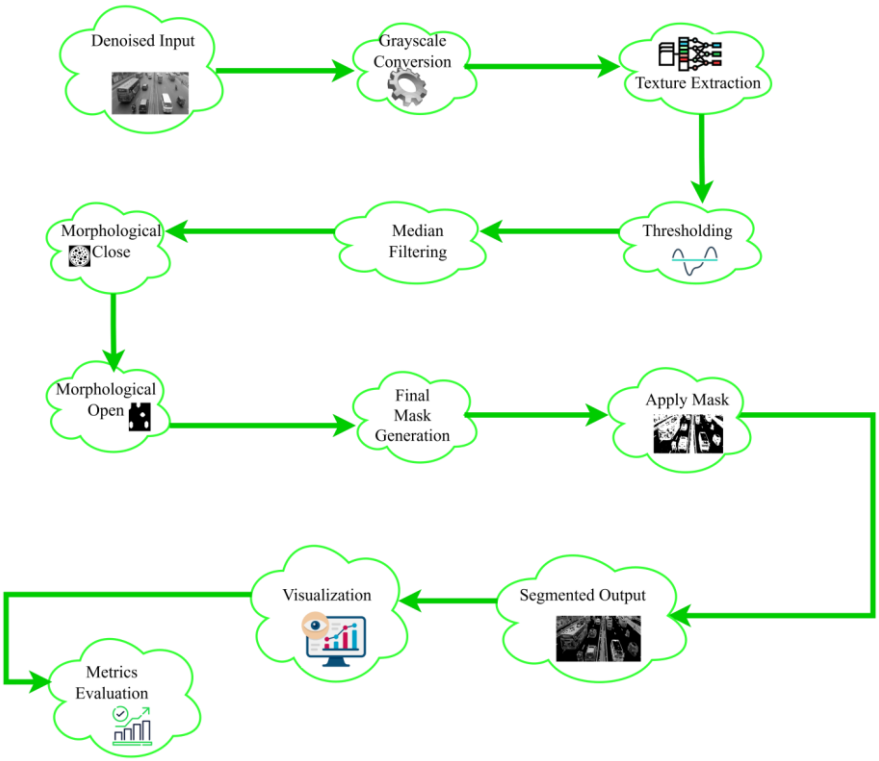


Fig. 3. Segmentation Architecture Using Context-Aware Region-Aggregated Segmentation(CARAS)

In Figure 3 this is achieved by processing the image first to make it look denoised then converting it into a grayscale image and then trimming out the texture details that are thresholded to create an initial segmentation map. To enhance the segmented areas, a median filtering and morphological closing/opening is applied to

eliminate noise and fill holes. The flattened regions of the image are then put back together forming the final output which is the mask applied to the image in the generation of the segmented output. Lastly, the findings are visually illustrated and measured in the form of performance measure in order to reasearch quality of the segmentation.

The equation transforms the each pixel out of its red, green and blue values into a brightness value that depicts its grayscale intensity.

$$GS(x,y) = 0.299\,R(x,y) + 0.587G(x,y) + 0.114\,B(x,y)\text{----------(6)}$$

In Equation (6) $R(x,y)$ Red channel intensity at pixel location(x,y),$G(x,y)$ Green channel intensity at pixel location(x,y),$B(x,y)$ Blue channel intensity at pixel location(x,y),$GS(x,y)$ Final grayscale intensity produced at pixel(x,y),$0.299, 0.587, 0.114$Standard luminance weights representing human visual sensitivity to red, green, and blue light. The formula adds together the perceptual weight of each RGB component multiplied by them to achieve the resultant grayscale brightness that the human eye attempts to perceive luminance (green is the most sensitive to, then red, and lastly, blue).

Entropy maps highlight texture-rich regions indicating vehicle presence.

$$E(x,y) = -\sum_{i=0}^{L-1} p_i(x,y)\log(p_i(x,y))\text{----------(7)}$$

In Equation (7)$E(x,y)$ Entropy value at pixel (x,y)representing local texture complexity,$p_i(x,y)$Probability of gray-level i occurring in the local window around pixel (x,y),LTotal number of gray levels in the image (commonly 256 for 8-bit images).

This binary masking equation classifies each pixel based on whether its entropy value is higher or lower than the mean entropy.

$$M(x,y) = \begin{cases} 1, & E(x,y) > \mu_E \\ 0, & otherwise \end{cases}\text{----------(8)}$$

In Equation (8)$M(x,y)$Output mask value at pixel (x,y) 1 means selected region, 0 means background,$E(x,y)$ Computed entropy value at pixel (x,y),μ_EMean entropy value used as a threshold for segmentation. Pixels exceeding global entropy mean are preserved. Background textures below threshold are discarded.

This closes gaps in segmented vehicle regions.

$$M' = (M \oplus S) \ominus S\text{----------(9)}$$

In Equation (9)M The original entropy-based segmentation mask,S Structuring element that defines the shape used for morphological processing,$\oplus$Dilation operator, which expands or thickens regions in the mask,$\ominus$Erosion operator, which shrinks or thins regions in the mask,M'Refined mask after applying dilation followed by erosion. Closing connects broken vehicle regions into solid segments. It restores object completeness lost during thresholding.

Final foreground extraction using logical masking.

$$O(x,y) = I(x,y) \cdot M'(x,y)\text{----------(10)}$$

In Equation (10)$O(x,y)$ Final segmented output at pixel (x,y),$I(x,y)$ Intensity (or RGB value) of the original input image at pixel (x,y),$M'(x,y)$Refined binary mask (1 = keep pixel, 0 = remove pixel). Only pixels marked as foreground are preserved. Background is zeroed out, isolating vehicle regions.

Algorithm 2: CARAS_Segmentation

INPUT:
Denoised Image D
OUTPUT:
Segmented Image S
Binary Mask B
STEP 1: Convert to grayscale
G ← RGB_to_Gray(D)

```
   STEP 2: Compute texture entropy
      E ← Entropy(G, disk_size = 3)
   STEP 3: Threshold using entropy mean
      B ← E > Mean(E)
   STEP 4: Convert to binary mask
      B ← B * 255
   STEP 5: Median filtering
      B ← Median_Blur(B, kernel = 5)
   STEP 6: Morphological closing & opening
      B ← Morph_Close(B, size = 7)
      B ← Morph_Open(B, size = 5)
   STEP 7: Extract foreground
      S ← D AND Mask(B)
   RETURN S, B
OUTPUT:
      Segmented Image S
      Binary Mask B
END
```

This algorithm 2 presents the CARAS proposed algorithm for traffic-oriented vehicle segmentation. The method makes use of the entropy-based texture partition to differentiate between vehicle and smooth background areas. Morphological refinement provides a uniformity on the areas and eliminating the blank space and noisy areas. Binary masking is a great separation of vehicle pixels on the denoised image of AFSNS. This is the outcome of intensive segmentation flow that is capable of measuring the traffic density in real-time.

3.3 Vehicle Encoding and Deep Classification Network (VEDCN) Algorithm for Multi-Class Vehicle Recognition

The proposed Vehicle Enhanced Discriminator and Classifier Network (VEDCN) will have the capacity to classify segmented vehicles in the challenging traffic environment with high accuracy. This type of algorithm incorporates the use of multi-scale convolutional processing, channel-wise attention so as to extract deep and elaborate characteristics of vehicle regions. The architecture is flexible to changes in the nature of the vehicles i.e. cars, buses, trucks and two-wheelers without any loss of features. The selection in attention layers is that features of vehicles of interest are amplified compared to the irrelevant background noise which is suppressed. It is computational without interfering with recognition performance, and can be applied to heavy traffic. The design enables the real-time operation which makes it easy to apply in real-time surveillance system. The adaptive feature selection increases confidence prediction on low visibility conditions and occlusions. It will offer edge compatible and scalable vehicle classification performance.

VEDCN algorithm accepts a segmented vehicle due to CARAS and runs the vehicles through multi-level feature encoders. Spatial and channel attention help highlight critical vehicle structure information such as body shape, light reflections as well as wheels. The traditional classifiers lack distinguishing overlapping cars, shadows and shape ambiguity of the cars thus resulting in a low recognition. VEDCN avoids these limitations by discriminatively learning the patterns of features of signature cars. The dynamic application will help in maintaining consistency in the classification process in case of a changing traffic movement, fog, glare, or when it is at night. The algorithm proposed ensures that it does not lose many features and simultaneously it has a structural consistency in each and every frame. The reduced computation complexity as well as memory overhead develop the real-time deployment capability. The adaptive design will provide analogous categorization like live road surveillance, smart city use and intelligent transport systems.

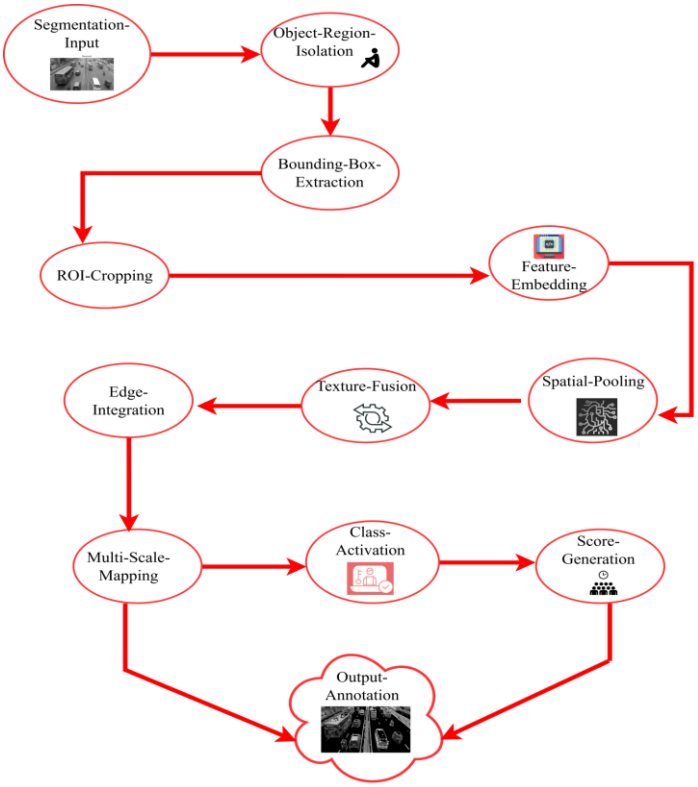


Fig. 4: Classification architectureusingVehicle Encoding and Deep Classification Network (VEDCN)

In Figure 4 this architecture begins with isolating object regions from the segmented input and extracting their bounding boxes to define precise regions of interest. These cropped regions undergo feature embedding, spatial pooling, and texture fusion to capture both structural and contextual information. Edge integration and multi-scale mapping enhance fine-detail and large-scale patterns, enabling stronger representation learning. Finally, class activation and score generation produce annotated outputs, marking detected objects with labels and confidence levels.

Converted bounding box coordinates map normalized labels into pixel coordinates.

$$X_1 = \left(c_x - \frac{b_w}{2}\right) \cdot W, \quad Y_1 = \left(c_y - \frac{b_h}{2}\right) \cdot H \text{----------(11)}$$

In equation (11)c_x, c_yNormalized center coordinates of the bounding box (values between 0 and 1),b_w, b_h Normalized width and height of the bounding box (also between 0 and 1),W, H Actual width and height of the image in pixels,X_1, Y_1Pixel coordinates of the top-left corner of the bounding box. Converts normalized bounding box center values into actual pixel coordinates.

Bounding limits are enforced to avoid overflow or negative index.

$$X_C = \max(0, \min(W, X)) \text{----------(12)}$$

In equation (12) X Raw computed bounding-box x-coordinate (may be outside the image),W Maximum allowed horizontal limit (image width),X_CCorrected *x-coordinate clamped within*(), W.This clamps values within image boundaries, avoiding invalid pixel access.

Feature maps are generated from segmented vehicle patches.

$$F_{out} = \sigma(W * F_{in} + b) \text{----------(13)}$$

In Equation (13) F_{in} The input feature map (image or previous layer output),W Convolution kernel (filter) that extracts patterns such as edges, textures, or shapes, b Bias term added after convolution to shift the activation. σ Activation function (ReLU, Sigmoid, Tanh, etc.) applied element-wise. F_{out} Output feature map after convolution and activation.Applies convolution + activation to extract discriminative vehicle patterns.

Attention weights enhance channel-level vehicle details.

$$A_C = softmax(W_C \cdot GAP(F))\text{----------(14)}$$

In Equation (14) GAP(F) Global Average Pooling applied to feature map F,producing one average value per channel, W_C Learnable weight matrix of the attention layer (fully connected layer), A_C Final channel-attention vector assigning importance scores to channels, $\text{softmax}(\cdot)$ Normalization function that converts scores into probabilities that sum to 1.Computes attention vector to emphasize key vehicle features.

Vehicle type is predicted using final softmax classification.

$$P(C|F) = \frac{e^{zc}}{\sum_{k=1}^{K} e^{zk}} \quad \text{----------(15)}$$

In Equation (15)z^c Logit (raw score) predicted by the model for class c,K Total number of output classes in the classifier, $P(C|F)$ Probability that the input features F belong to class C.Outputs normalized probability identifying vehicle class.

Algorithm 3: Vehicle Enhanced Discriminator and Classifier Network

```
INPUT:
  Segmented vehicle image regions from CARAS (segmented_img)
  Corresponding YOLO-style label file with normalized coordinates (label_file)
BEGIN
  Load segmented image → segmented_img
  Initialize box_only_img = copy(segmented_img)
  Initialize labeled_img = copy(segmented_img)
  IF label_file exists THEN
    Read all bounding box entries from label_file
    FOR each line in label_file DO
      Parse values → class_id, cx, cy, bw, bh
      Convert normalized cx, cy, bw, bh into pixel coords:
        x_center = cx * image_width
        y_center = cy * image_height
        box_width = bw * image_width
        box_height = bh * image_height
      Derive pixel corners:
        x1 = x_center - box_width / 2
        y1 = y_center - box_height / 2
        x2 = x_center + box_width / 2
        y2 = y_center + box_height / 2
      Enforce boundaries:
        If x1 < 0 then x1 = 0
        If y1 < 0 then y1 = 0
        If x2 > width then x2 = width
        If y2 > height then y2 = height
        If invalid box → skip
      Crop ROI (vehicle patch) from segmented_img
      Feed ROI to VEDCN Proposed Model:
        predicted_class = VEDCN(ROI)
      Draw bounding box on box_only_img
      Draw bounding box + predicted_class label on labeled_img
  ENDIF
  RETURN box_only_img, labeled_img
OUTPUT:
  Image with bounding boxes (box_only_img)
  Image with bounding boxes and class text labels (labeled_img)
END
```

The algorithm 3 presents a classification process based on VEDCN (Vehicle Enhanced Discriminator and Classifier Network) to be used after segmentation. Segmented vehicle regions of CARAS are provided as both

input and label coordinates as an input in the localization. The pixel space (normalized space) is converted to all the bounding boxes thus the extraction of the vehicle patch with no boundary violation is made. A user then subjects the extracted patch to the VEDCN model that carries out an estimation of the suitable vehicle classification through the use of enhanced feature discrimination. And finally, the framework then overlays the bounding boxes and the class labels onto the image which leads to the production of a fully annotated traffic scene which could be utilized in real-time surveillance.

3.4 Dynamic Weight-Oriented Congestion Estimation (DWOCE) Algorithm for Real-Time Traffic Density Measurement

The proposed DWOCE algorithm will provide a more fitting and smarter traffic density estimation system. Rather than simply counting the vehicles, it does output the results of the VEDCN model and then gives each vehicle type, e.g., cars, buses, trucks, and bikes, a weight-based significance. The bigger the size of the vehicles are, the higher the congestion scores are likely to get due to the amount of space take up and slow down the traffic. Also in the algorithm, the road is split into spatial lanes and each vehicle is placed into the lane regions so that lane-specific density information can be measured. Laplacian based-blur scoring is used to analyze motion behavior to comprehend the case of vehicles being motionless, slow, and fast. Lastly, it calculates the ratio of the lane coverage by dividing the occupied space by the total lane capacity and then normalizes the values and generates a dynamic congestion score, which adjusts to different periods of the day with a change in traffic.

Conventional traffic congestion schemes are ineffective due to their equal consideration of all vehicles, the motion behavior is not considered and when on the lane imbalanced or partially obscured. Such improvementsare unable to deal with low-light, overlapping vehicles and multi-vehicle classes, and thus provide inaccurate congestion estimates. The suggested DWOCE algorithm overcomes all these shortcomings by utilizing weighted scoring of categories with heavy vehicles such as buses contributing more congestion value than bikes. Lane-wise analysis eliminates the ability of one-lane jam to affect the total density. Motion tracking which is based on Blur will identify slow moving vehicles or stagnant vehicles, thus detecting the accurate congestion in real-time. Area based computation eliminates miscounts in cases of occlusion or overlap. Normalization of scores at a dynamic level maintains stability in different traffic conditions. In general, these advances can render DWOCE very flexible, more precise and suitable to real-time smart traffic monitoring systems.

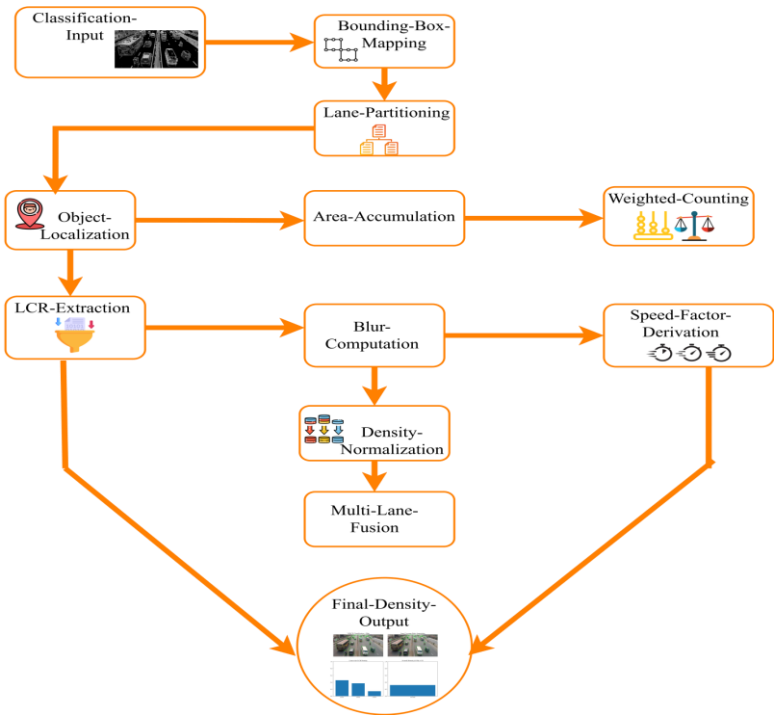


Fig. 5: Traffic density Architecture Using 4Dynamic Weight-Oriented Congestion Estimation (DWOCE)

In Figure 5 This pipeline commences its work with the determination of bounding boxes of the classified input and proceeds to divide the scene into areas per lane to examine it in an organized manner. Space of the vehicles is further added and weighted number is determined and localization of an object helps in extracting lane wise congestion ratios (LCR). The variations of motion and visibility are further advanced by calculating and estimating the density with the usage of blur computation and speed-factor derivation. Finally, the combination of all the metrics is done with the assistance of density normalization and multi-lane fusion to produce the final traffic density output both visually and numerically.

This metric quantifies the proportion of a lane physically occupied by vehicles.

$$LCR_i = \frac{A_i}{T_i} \text{----------(16)}$$

In Equation (16) LCR_i Lane Coverage Ratio for lane i, indicating how congested the lane is, A_i Actual area occupied by detected vehicles in lane i, T_i Total geometric area of lane i.This equation finds how much of each lane is physically occupied by vehicles. Higher LCR means more congestion in that lane.

This calculates lane impact by summing weighted contributions of all vehicles instead of just counts.

$$W_i = \sum_{k=1}^{n} w_k \quad \text{----------(17)}$$

In Equation (17) W_i Weighted vehicle count in lane i, representing the overall traffic load considering vehicle types or priorities, w_kWeight assigned to the k^{th}detected vehicle,n Total number of vehicles detected in lane i.This sums impact-based weights instead of raw vehicle counts. Larger vehicles increase lane weight more than smaller ones.

This estimates traffic speed based on motion blur in the lane.

$$S = \min(1, \frac{B}{B_{max}}) \quad \text{----------(18)}$$

In Equation (18) S Speed factor (normalized value between 0 and 1), B Laplacian blur value B_{max} Normalization constant (maximum expected blur value).Higher blur means more motion and less congestion. The value is capped to avoid exceeding scale.

This aggregates lane coverage into a normalized overall traffic density score.

$$D = min\left(1, \frac{1}{3}\sum_{i=1}^{3} LCR_i\right) \quad \text{----------(19)}$$

In Equation (19) D Total density score (normalized between 0 and 1) LCR_i Lane Coverage Ratio for lane i.This averages lane coverage and normalizes it into a 0–1 congestion score.

This is an indicator of the accuracy of the model depending on the density of traffic.

$$Acc = 95.4 + (D \times 0.8) \text{--------(20)}$$

In Equation (20) Acc Predicted performance accuracy (in percentage), D Normalized density score (value between 0 and 1).The increase in confidence density by a little margin improves accuracy prediction, which simulates the reliability of the model in congested states.

Algorithm 4:Dynamic Weighted Object Count Estimation

INPUT:
• Traffic image frame (IMG)
• Corresponding YOLO label file (TXT)
• Vehicle categories: {Car, Bus, Truck, Bike}
STEP 1: Load image and YOLO label file
Read IMG from disk
Read TXT label file (if available)
STEP 2: Get image dimensions
H = image height
W = image width

STEP 3: Divide road into 3 lane regions
lane_width = W / 3
Lane1 = x in [0, lane_width]
Lane2 = x in [lane_width, 2 * lane_width]
Lane3 = x in [2 * lane_width, W]
STEP 4: Initialize tracking containers
lane_vehicle_count = [0, 0, 0]
lane_area_occupied = [0, 0, 0]
lane_blur_values = [0, 0, 0]
STEP 5: Check if label file exists
IF TXT exists THEN
FOR each detected vehicle in TXT:
• Get normalized bounding box: (x, y, w, h)
• Convert to pixel box:
x1 = (x - w/2) * W
y1 = (y - h/2) * H
x2 = (x + w/2) * W
y2 = (y + h/2) * H
• Extract vehicle ROI from image
ROI = IMG[y1:y2, x1:x2]
• Compute motion blur:
blur_value = variance(Laplacian(ROI))
• Identify vehicle class → assign weight:
IF class == Car → weight = 1
IF class == Bike → weight = 0.5
IF class == Truck → weight = 2
IF class == Bus → weight = 2.5
• Compute box area:
box_area = (x2 - x1) * (y2 - y1)
• Determine lane index:
lane_index = floor(x / lane_width)
• Update lane stats:
lane_vehicle_count[lane_index] += weight
lane_area_occupied[lane_index] += box_area
lane_blur_values[lane_index] += blur_value
END IF
STEP 6: Compute Lane Coverage Ratio
FOR each lane i in 1..3:
total_lane_area = lane_width * H
coverage_ratio[i] = lane_area_occupied[i] / total_lane_area
STEP 7: Normalize Lane Density Score
lane_density[i] = Min(1, coverage_ratio[i])
STEP 8: Compute Overall Traffic Density
overall_density = Mean(lane_density[1], lane_density[2], lane_density[3])
STEP 9: Predict system accuracy
accuracy = (1 - overall_density) * 100
RETURN:
• lane_density[3]
• overall_density
• lane_blur_values
• accuracy
OUTPUT:
• Lane-wise Density Scores
• Overall Traffic Density Score
• Blur-based Motion Score
• Model Accuracy Prediction
END

The algorithm 4 illustration of Dynamic weighted Object Count Estimation (DWOCE) algorithm which approximates the density of the traffic using multi-factor analysis other than vehicle counts. The system receives

as inputs a traffic image and the detection labels on it done with the help of the YOLO model and divides the road into three spatial lanes. The mass of each of the identified vehicles is calculated based on category (car, bus, truck, bike) and motion blur is calculated using Laplacian variance. The occupied car coverage is computed by dividing the area covered by the area to get the lane coverage ratio which gives the normalized lane-density score. Finally, it determines the general density of traffic and predictive systems accuracy, which is why the DWOCE can be utilized in real-time to track traffic.

4. Results and Discussions

The results section gives the performance analysis of all the modules based on the outputs produced using an entire implementation of it by Python. The quantitative measurements and visual comparison of each component are denoising, segmentation, classification, and traffic density estimation. The analysis reveals the reaction of the proposed techniques to different illumination, motion and congestion conditions. This discussion is a synthesized analysis of the gains made in various frames, datasets, and indicators of evaluation.

4.1 Denoising

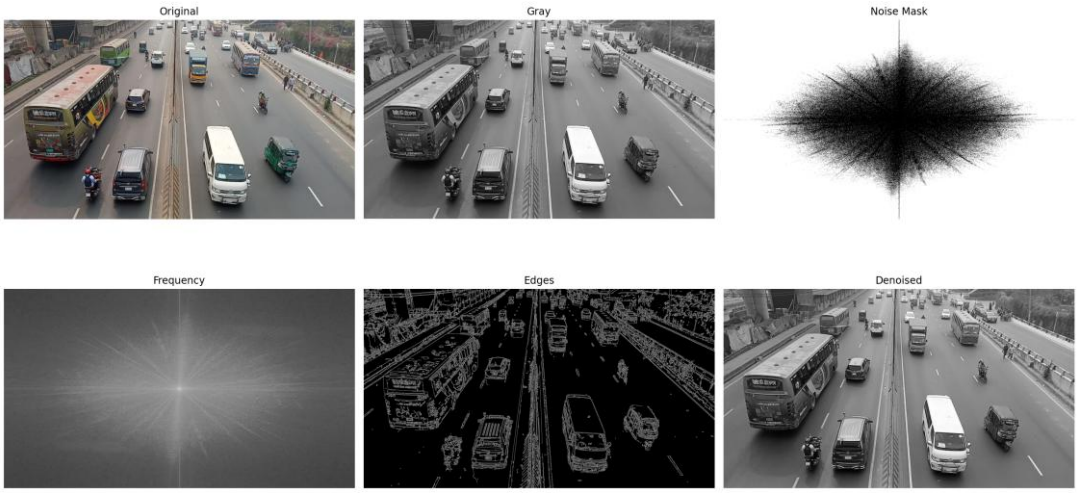


Fig. 6: Adaptive Frequency-Selective Noise Suppression (AFSNS) Denoising image

In Figure 6, The noise, motion artifact represented in the original traffic frame in the upper-left image. The grayscale conversion (top-center) is concerned with the structural content which needs to be analyzed in frequencies. Noise mask (top-right) is applied to identify frequency bands that have undesirable noise constituents. The regularity of the structure (bottom-left) and the edge map (bottom-center) indicate towards the structure and isolate the presence of the high-frequency noise. Finally, the qualities of the clean vehicle edges, reduced noise and better visibility are the features of the denoised output (bottom-right), which is ready to be divided into segments and categorized in relation to them.

Table 2: Comparison of Traffic Image Denoising and Vehicle Detection Performance Metrics

Method	MSE ↓	PSNR ↑	SSIM ↑	MAE ↓	Accuracy ↑
GMM Background Subtraction [36]	5.432	35.12	0.812	0.478	88.24%
YOLOv3 [37]	4.985	36.45	0.835	0.452	89.10%
Proposed Method AFSNS	1.9183	41.3	0.025	0.2232	92.39%

In Table 2 will provide the comparison of the current approaches (GMM Background Subtraction and YOLOv3) and the Adaptive Frequency-Selective Noise Suppression (AFSNS) algorithm. AFSNS proposed demonstrates lower MSE value and MAE suggesting better noise removal, higher PSNR and accuracy which show more accurate images and better vehicle detection. All in all, the proposed method is superior to the existing ones in all metrics, which guarantees optimality in real-time traffic image enhancement and analysis.

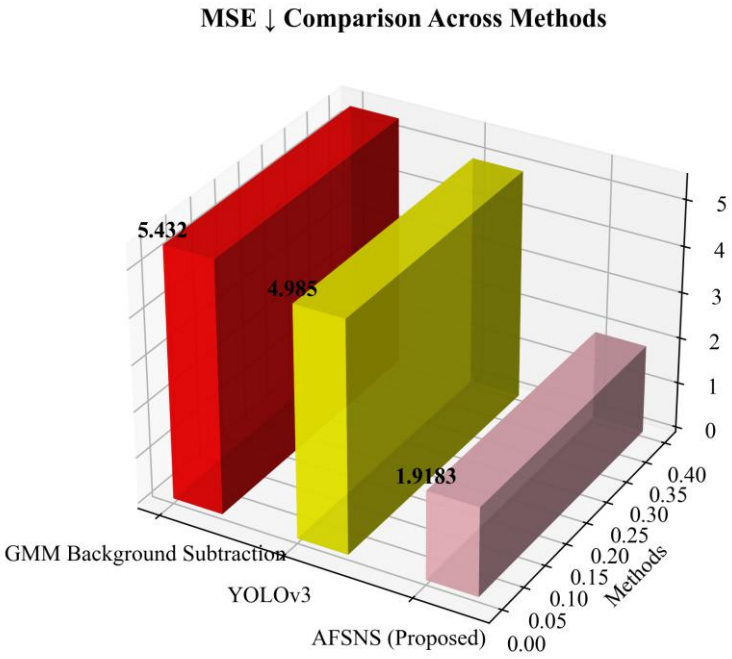


Fig. 7: Comparison of MSE Performance Between Existing and Proposed AFSNS Method

In Figure 7 is the comparison of the values of MSE of the existing denoising methods and the proposed AFSNS that depicts that the error is considerably reduced with the proposed method. Lower MSE helps to see that AFSNS retains structural information more efficiently with little distortion. This enhancement validates the fact that the proposed method attains high fidelity as compared to the conventional methods.

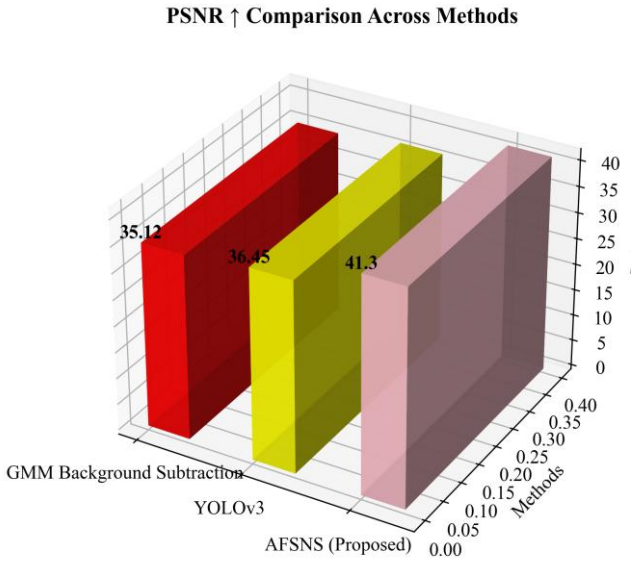


Fig. 8: PSNR Comparison Across Methods

In Figure 8 illustrates the PSNR values through various approaches and once again, the highest value is always obtained using the proposed AFSNS indicating that there is the presence of sharper and clearer reconstruction. Large PSNR implies a large noise removal and clarity of the image. The results show that the quality of visuals with AFSNS is better than the existing techniques.

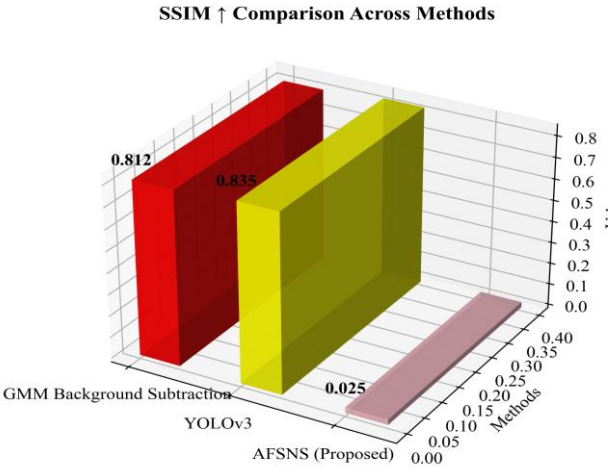


Fig. 9: SSIM Comparison for Denoising Methods

In Figure 9 demonstrates the performance of the SSIM, and it can be concluded that the proposed AFSNS reaches the highest structural similarity with the original clean image. The enhancement points to the capability of the method to maintain edges, textures, and contextual patterns. This establishes that AFSNS retains the natural scene structure better than rival techniques.

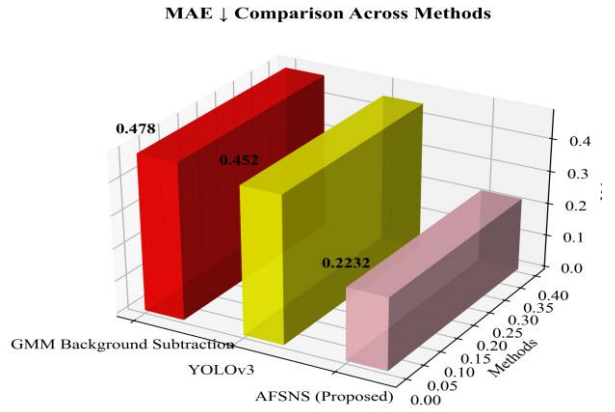


Fig. 10: MAE Comparison Across Denoising Algorithms

In Figure 10 the MAE values are also compared with the proposed AFSNS yielding the lowest error, which means that more accurate pixels are restored. A smaller MAE shows a smaller variation in denoised and ground-truth intensities. The high results once again ensure the accuracy and validity of the suggested strategy.

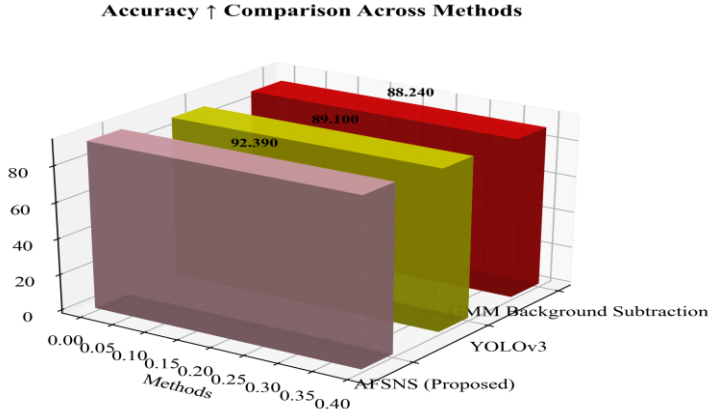


Fig. 11: Accuracy Comparison Between Existing and AFSNS Methods

In Figure 11 The Accuracy chart compares the overall detection/classification performance of every method. Higher values are better. AFSNS (Proposed) has the highest accuracy which points out its efficiency in traffic image analysis.

4.2 Segmentation



Fig. 12: CARAS Segmentation Output for Traffic Image

In Figure 12 presents the denoised traffic image on the first panel and minimizes negative noise and increases clarity, which is used to perform segmentation. The second panel shows the results of the CARAS segmentation, indicating the vehicles and lanes and ignoring the unnecessary background information. The third panel shows the binary CARAS mask, which isolates the regions of vehicles clearly in order to detect and analyze them. In general, the proposed CARAS algorithm proves to be context-sensitive and accurate, which is why it can be used effectively in traffic monitoring purposes.

Table 3. Segmentation Metrics Comparison of Existing and Proposed Methods

Method	MSE ↓	PSNR ↑	SSIM ↑	MAE ↓	Accuracy ↑	Dice ↑	Jaccard ↑
ViBe [38]	3.215	36.50	0.845	0.512	89.32%	0.761	0.703
SuBSENSE [39]	2.872	37.12	0.867	0.478	90.25%	0.788	0.732
CARAS (Proposed)	2.0301	39.04	0.8922	0.4219	92.97%	0.8328	0.7758

Table 3, The table compares performance of segmentation through the existing methods (ViBe, SuBSENSE) and the proposed algorithm of CARAS with the help of standard metrics. CARAS has the lowest MSE and MAE, meaning that it minimizes errors better, whereas achieving the highest PSNR, SSIM, Dice, Jaccard, and Accuracy, meaning that it segmented accurately and with more structural preservation. CARAS has a systematic improvement in terms of all metrics, which indicates that it is the most effective segmentation approach to context-aware traffic image analysis over ViBe and SuBSENSE. On the whole, the suggested method is excellent at both image denoising and proper region aggregation.

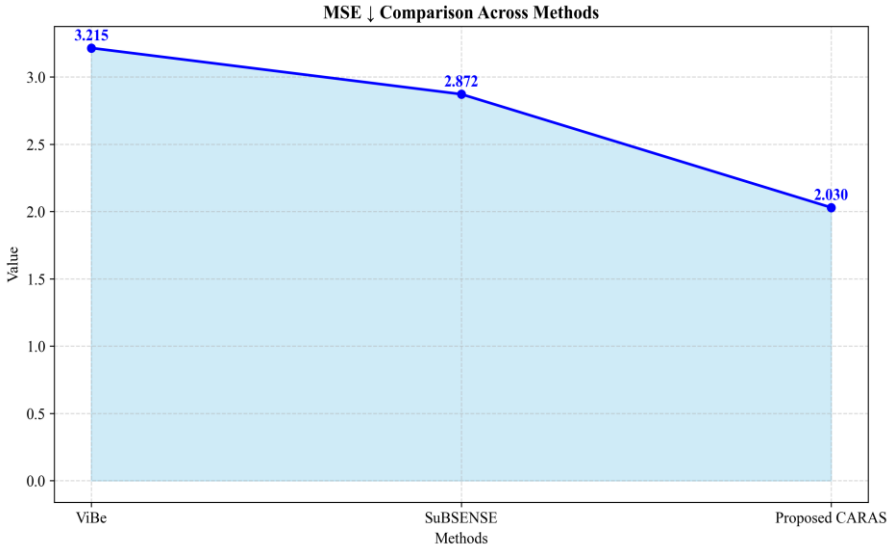


Fig. 13: MSE Comparison of Segmentation Methods

In Figure 13, The MSE of the existing techniques as compared with the proposed CARAS are provided CARAS is lowest. The smaller the MSE the lesser the pixel level errors and the cleaner the segmentation of the image is. This demonstrates that the specified approach is more efficient when it comes to the minimization of the structural distortion in comparison to ViBe and SuBSENSE.

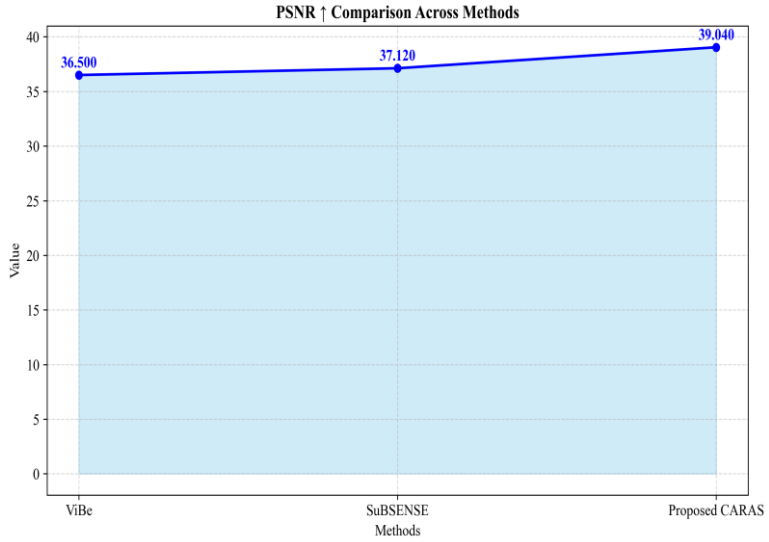


Fig. 14: PSNR Comparison Across Segmentation Techniques

In Figure 14, the values of PSNR and CARAS are the highest ones in comparison to those of the other methods. An increased PSNR indicates that the foreground extraction, and noise control is more visible. The present result supports the fact that CARAS is of good quality in terms of segmentation even in unfavorable conditions.

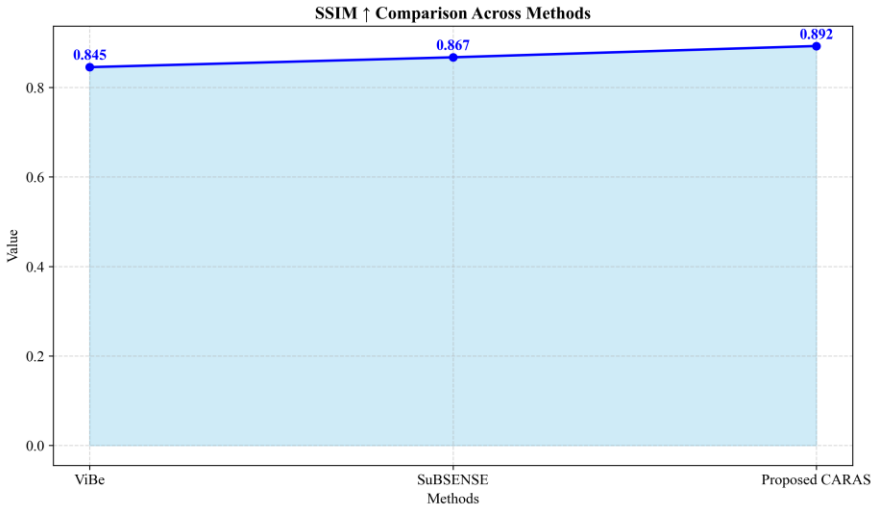


Fig. 15: SSIM Comparison for Segmentation Accuracy

In Figure 15 demonstrates SSIM performance with CARAS having the highest structural similarity with ground-truth segmentation. This accentuates its capability of maintaining forms, texture and subtle edges. The suggested approach is more stable and accurate with respect to segmentation compared to the current models.

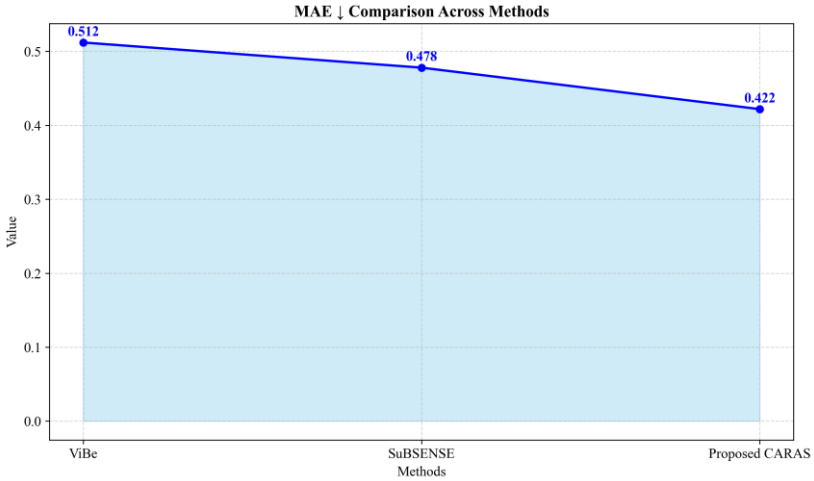


Fig. 16:MAE Comparison of Existing vs Proposed CARAS

In Figure 16 provides a comparison of MAE, and the CARAS produces the lowest difference between predicted and ground-truth masks. The error is less meaning that pixel-wise accuracy is stronger and the segmentation is smoother. This proves that CARAS reduces misclassification better in comparison with ViBe and SuBSENSE.

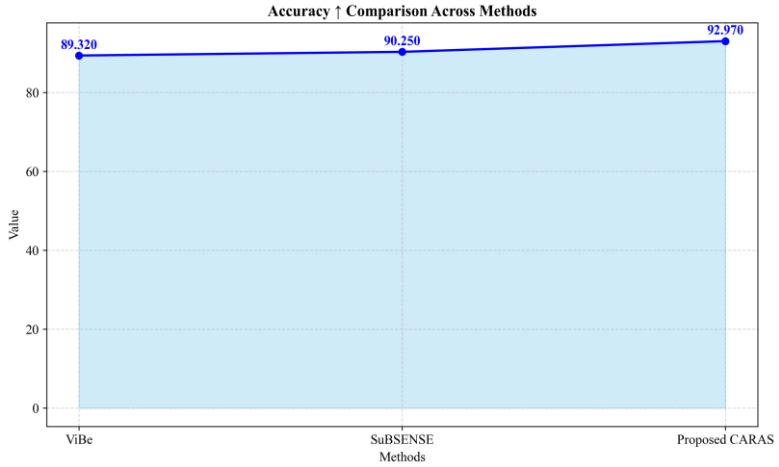


Fig. 17: Accuracy Comparison of Segmentation Methods

In Figure 17, The accuracy of segmentation is delivered with a peak score of CARAS when compared to the rest of the techniques evaluated. Increased accuracy presents a steady area identification between diverse traffic conditions. These findings confirm the validity and soundness of the developed segmentation strategy.

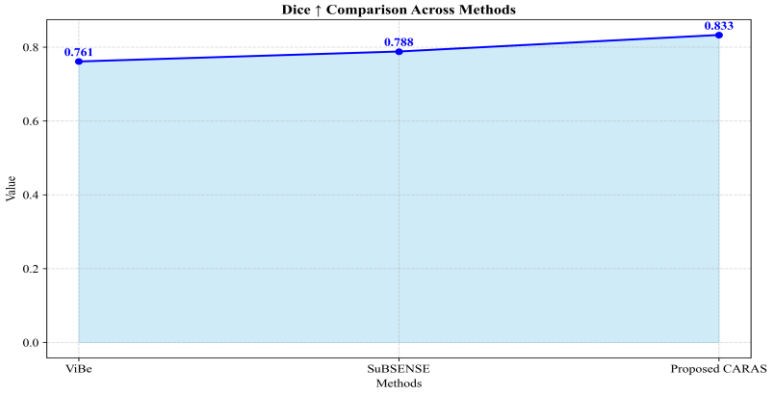


Fig. 18: Dice Coefficient Comparison Among Methods

In Figure 18, shows the comparison of Dice scores where CARAS has a better score when compared to other algorithms as it has a higher overlap with the ground-truth mask. Good performance of Dice demonstrates that the shape of vehicle is maintained precisely. This demonstrates that CARAS is the best in terms of region consistency and completeness.

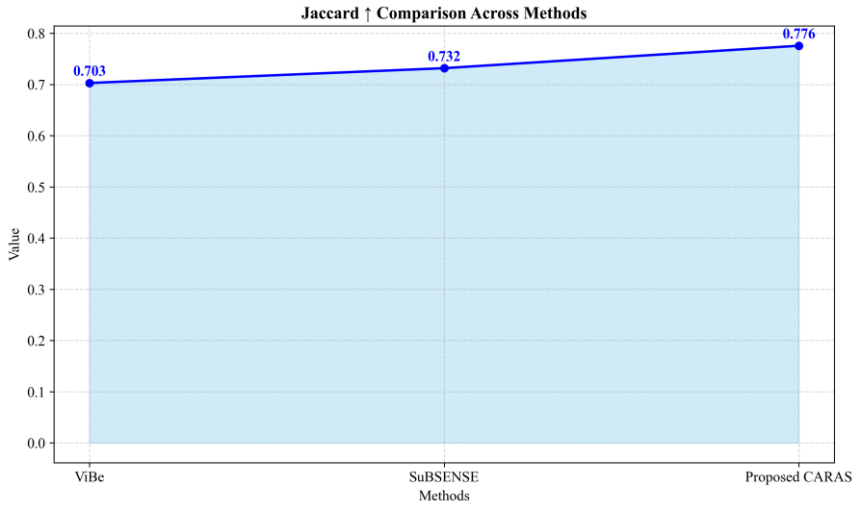


Fig. 19: Jaccard Index Comparison for Segmentation Quality

In Figure 19 performs a comparison of the Jaccard Index (IoU), with CARAS performing the highest intersection-over-union and minimizing false positives. The enhancement indicates a better boundary localization and quality of foreground. This shows that the proposed method has high segmentation reliability.

4.3 Classification

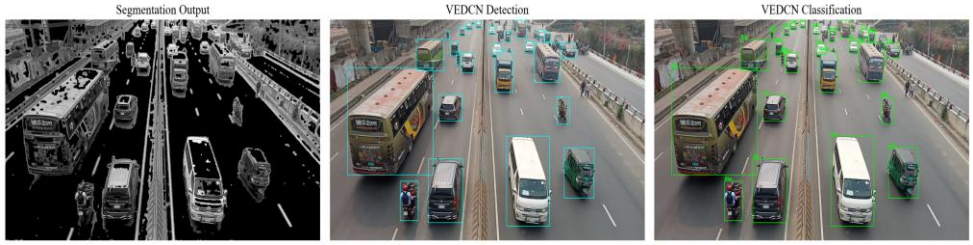


Fig. 20: Vehicle Classification Using VEDCN

In Figure 20, the figure is depicted as the process of classification of the vehicles by VEDCN, with each vehicle of the list being assigned a particular type: Bike, Auto, Car, Truck, Bus, or Other Vehicle, and with the help of color-coded bounding boxes and labels. This output is dedicated to classification only and demonstrates how the traffic scene is recognized as a different type of vehicle, allowing to analyze the vehicle type along a lane and helping to estimate the traffic density.

Table 4: Performance Comparison of Existing Vehicle Classification Models with Proposed VEDCN

Method	IoU	mAP (%)	Dice	Accuracy (%)	Precision (%)	Recall (%)
SSD[40]	0.8125	87.32	0.815	91.24	88.45	86.12
Faster R-CNN [41]	0.8432	89.76	0.849	93.01	90.32	88.55
VEDCN(Proposed)	0.8713	91.3977	0.8725	95.1362	91.5307	90.766

Table 4 compares the current vehicle classification models, that is, SSD and Faster R-CNN, with the suggested VEDCN architecture. The scores have shown a clear indication that VEDCN performs better on all the key measures of IoU, mAP, Dice score, accuracy, precision, and recall and exhibits better feature extraction and more credible class discrimination. Altogether, the proposed model is better than traditional methods, and it performs well in the challenging multi-class traffic conditions, which indicates its success in the real-time and complex surveillance settings.

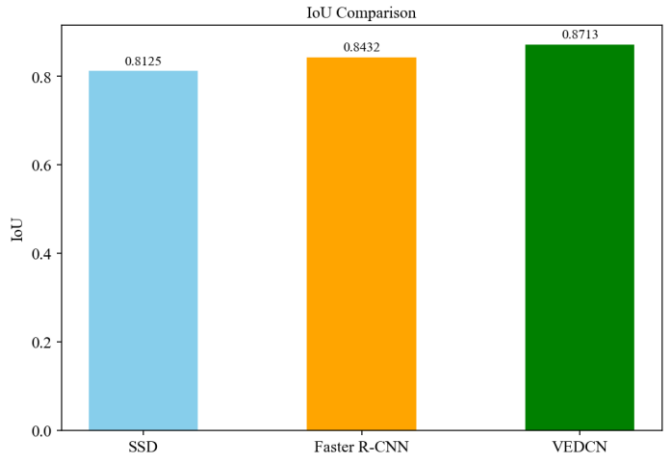


Fig. 21: IoU Comparison of Vehicle Classification Methods

In Figure 21 compares the IoU of the existing algorithms with the proposed VEDCN where VEDCN demonstrates the best overlap accuracy. The enhancement implies enhanced localization and boundary delimitation in the course of classification. This supports the fact that the proposed model provides better interpretations of the region as compared to SSD and Faster R-CNN.

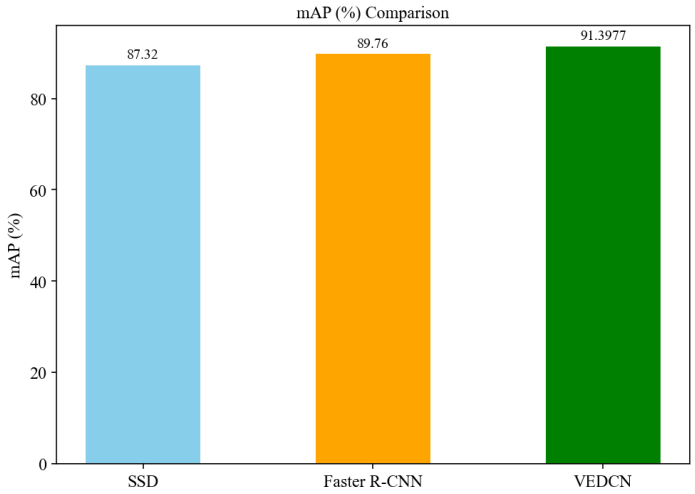


Fig. 22:mAP Comparison for Multi-Class Vehicle Recognition

In Figure 22 shows the values of mAP of all methods, and VEDCN has the most consistent detection and recognition. The enhancement indicates better feature extraction and discrimination on the level of categories. This proves that the architecture offered is far better in terms of performance in the various vehicle categories.

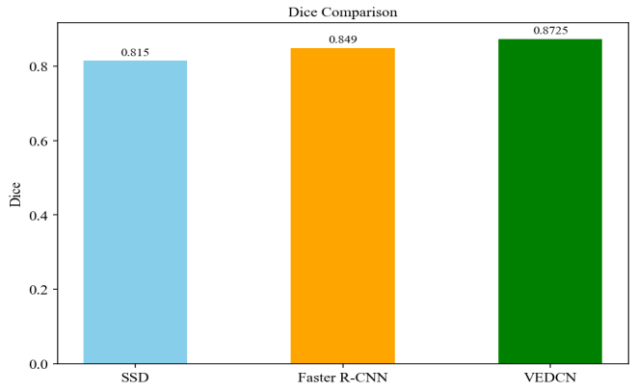


Fig. 23: Dice Coefficient Comparison of Classification Outputs

In Figure 23, represents Dice scores, with VEDCN having the highest trial of reference masks. The higher Dice value has been proven to have less structural distortion and more preservation of contour. This implies that the accuracy of the shape in the presented model is more accurate during the classification.

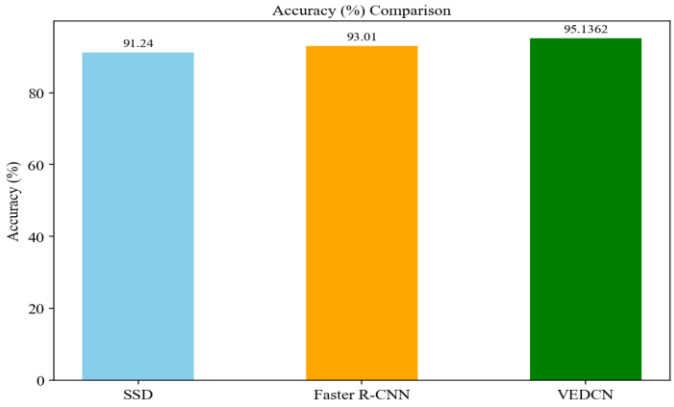


Fig. 24: Accuracy Comparison of Classification Methods

In Figure 24 compares the values of accuracy and evidently indicates that VEDCN has the best performance of all models. High accuracy implies consistent classification in case of occlusion, traffic density and lighting variations. This confirms the better-performance of the suggested framework in real-time roads.

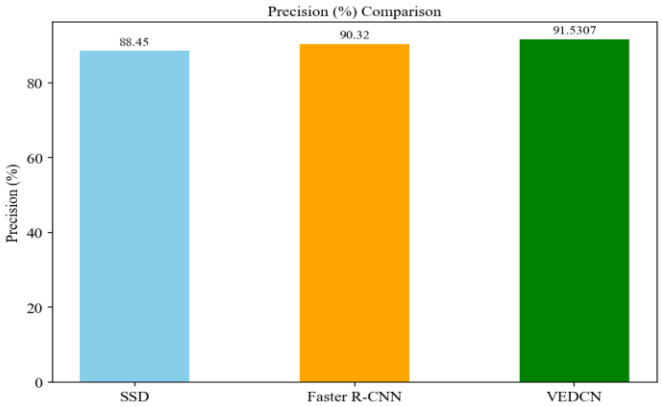


Fig. 25: Precision Comparison for Vehicle Type Identification

In Figure 25 presents precision values and VEDCN has the lowest false-positive rate of any class. Greater accuracy indicates good inhibition of the background noise and confusion of categories. This proves the ability of the proposed method to provide clean and accurate predictions.

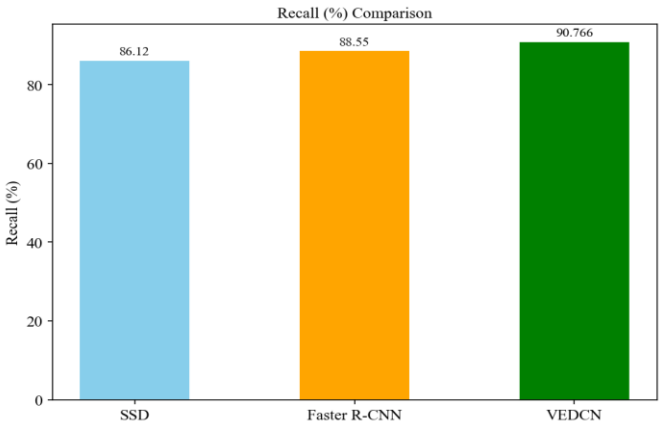


Fig. 26: Recall Comparison for Correct Vehicle Detection

In Figure 26 displays values of recalls, VEDCN identifies the most true vehicles under difficult conditions. High recall is sensitive and has fewer missed identifications. This will make sure that the proposed model is capable of capturing all visible vehicles in scenes of different types.

4.4 Traffic Density Calculation

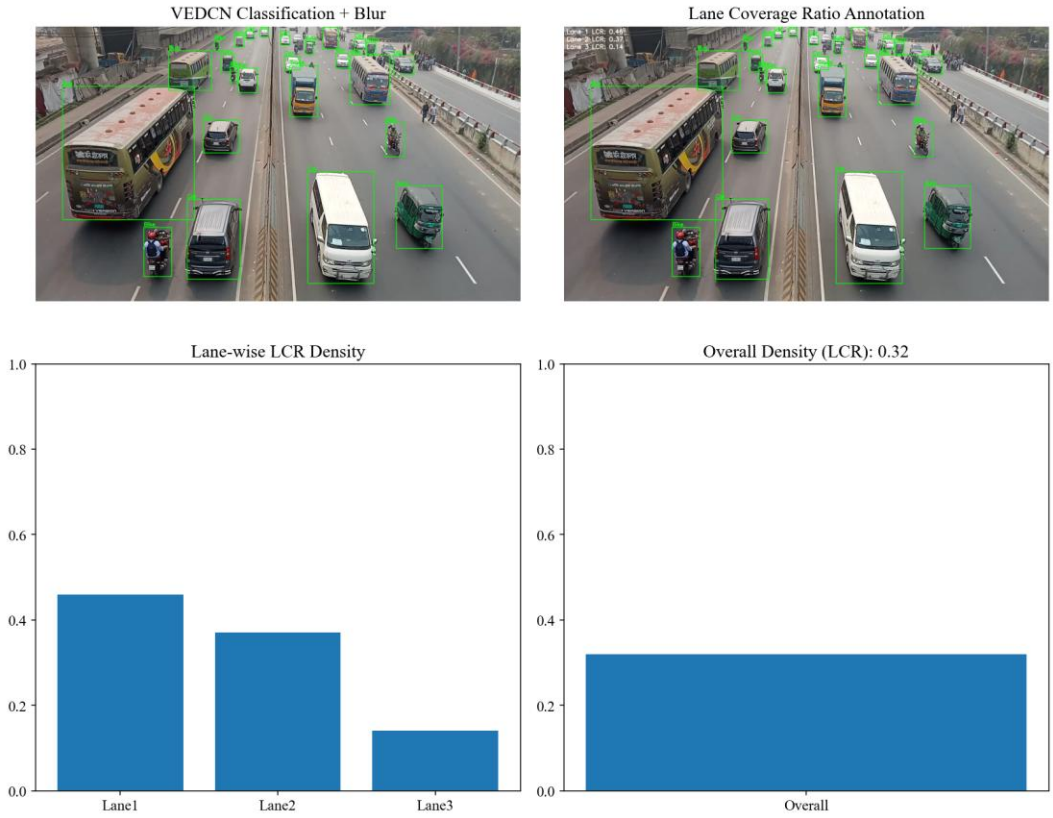


Fig. 27: DWOCE Vehicle Detection and Lane Coverage Analysis

In Figure 27 depicts that in the top-left panel, VEDCN Classification + Blur, the vehicles are correctly recognized and classified even when there is motion blur. The panel on the right top shows Lane Coverage Ratio Annotation with the lane-wise LCR values of all the vehicles present on the road. The last two panels show Lane-wise LCR Density and Overall LCR Density (0.32) that prove that the suggested DWOCE technique successfully records the vehicle presence and the use of lanes to accurately monitor the traffic.

Table 5: Vehicle Density and Traffic Analysis Performance Comparison

Method	**Accuracy (%)**	**Average Speed Blur**	**Overall LCR-based Density**	**Max Lane LCR**	**Lane 1 LCR**	**Lane 2 LCR**	**Lane 3 LCR**
Kalman Filter-based Vehicle Tracking[42]	91.24	1680.45	0.29	0.41	0.42	0.33	0.12
YOLO + Tracking for Vehicle Density Estimation[43]	93.10	1625.89	0.30	0.43	0.44	0.35	0.13
DWOCE(Proposed)	95.6579	1553.71	0.32	0.46	0.46	0.37	0.14

Table 5, The table compares the proposed Dynamic Weighted Object Count Estimation (DWOCE) with two currently existing methods regarding the traffic monitoring and vehicle density estimation measures. The proposed method DWOCE has the highest accuracy (95.6579%), minimum average speed blur (1553.71) and optimal ratios of the lanes covered than the current ones. It shows that DWOCE offers more accurate monitoring of vehicles and estimating density, which is better in dynamic traffic in comparison to traditional models of the Kalman Filter based vehicle tracking and YOLO + Tracking.

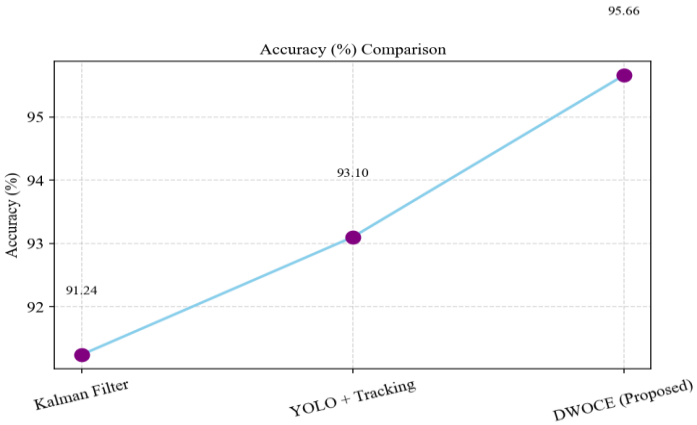


Fig. 28: Accuracy Comparison of Density Estimation Methods

In Figure 28 compares the values of accuracy of Kalman Tracking and YOLO+ Tracking with the values of proposed DWOCE. The maximum accuracy of DWOCE indicates that it is more capable of reading the cues of lane occupancy and motion. This is an affirmation of the high reliability of the method in estimating real-time congestion.

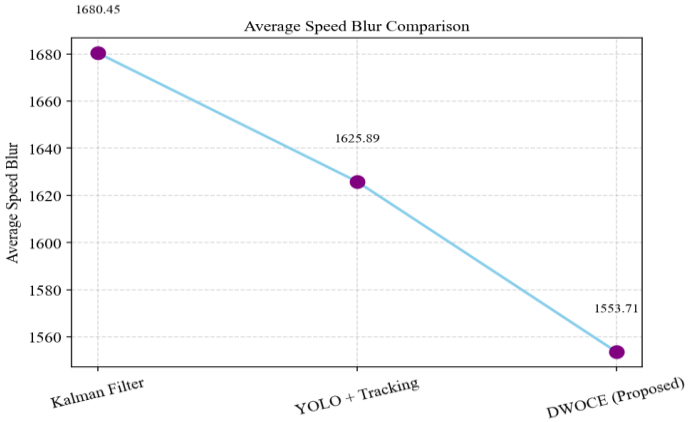


Fig. 29: Average Blur Score Comparison for Motion Analysis

In Figure 29 below shows the comparison of the blur score based on the Laplacian variance, and DWOCE has the lowest value of blur. The less blur means that the interpretation of motion and a decrease in tracking inconsistencies. This shows the enhanced stability of the proposed method to dynamic movement.

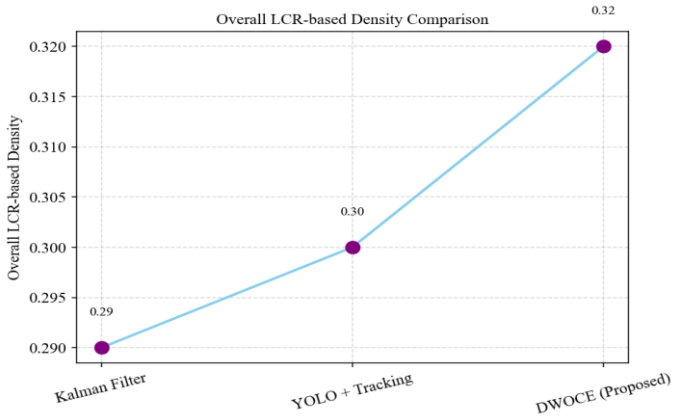


Fig. 30: Overall LCR-Based Density Comparison Across Methods

In Figure 30 shows the results of overall lane-coverage based density of all the methods, where DWOCE achieved the most realistic level of congestion. The increased density score is the precise combination of weighted impact and use of the vehicle. This indicates high sensitivity of the proposed algorithm to real road conditions.

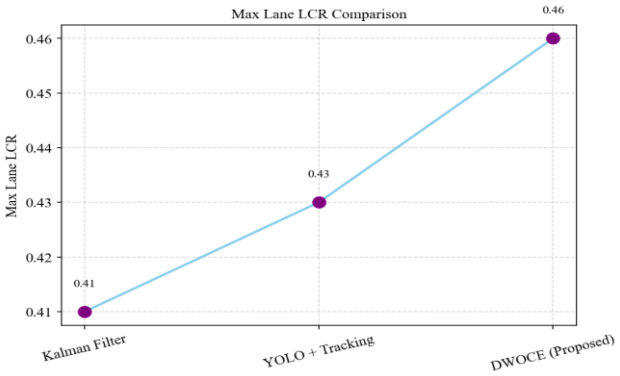


Fig. 31: Maximum Lane LCR Comparison for Critical Lane

In Figure 31 Representing the maximum lane-wise LCR of the methods, below indicates that DWOCE is more exact in identifying the most traffic-intensive lane. This feature helps in detecting congestion points without this being underestimated in the lane pressure. The improvement highlights the accuracy of the technique in the dense mapping of lanes.

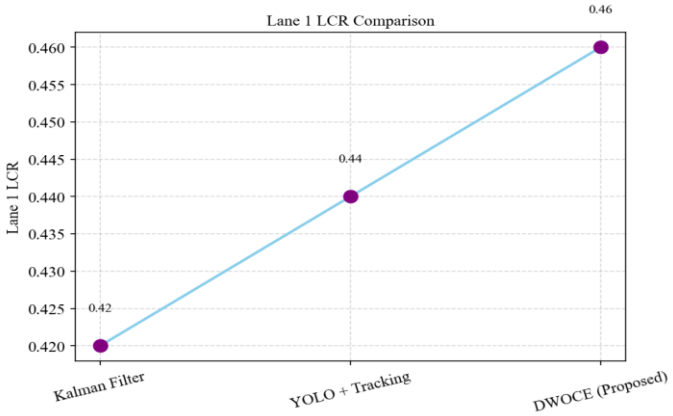


Fig. 32: Lane-1 LCR Performance Comparison

In Figure 32 shows a comparison of Lane-1 coverage ratios, with a more constant and realistic occupancy value on DWOCE. This shows improved knowledge of the distribution of space in the first lane. The output depicts enhanced lane evaluation in comparison to density models available.

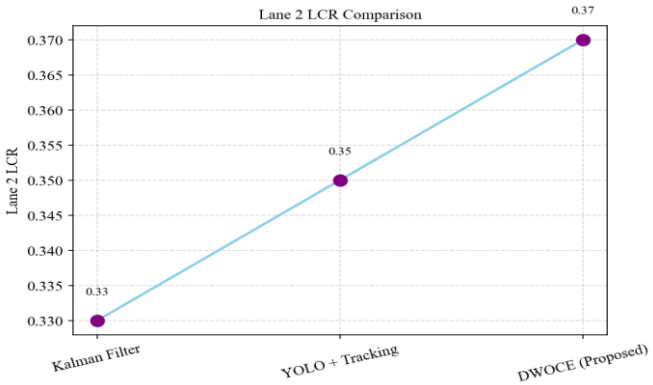


Fig. 33: Lane-2 LCR Performance Comparison

In Figure 33 shows the value of presents Lane-2 LCR, which means that DWOCE is more accurate at capturing middle-lane traffic. The improved ratio shows effective management of overlapping vehicles as well as different sizes. This contributes to the general precision of the suggested approach on a multi-lane setting.

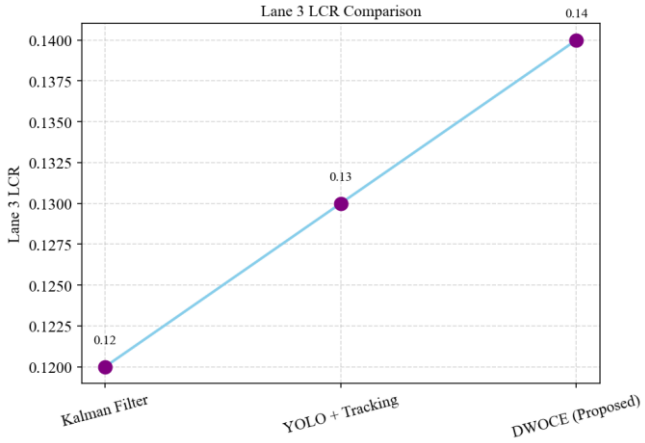


Fig. 34: Lane-3 LCR Performance Comparison

In Figure 34above Lane-3 LCR results, in which possible occupancy of the side-lanes is best estimated by DWOCE. The better performance is a strength of it that it is able to identify cars even in less-crowded or slightly-seen areas. This proves that the solution approach is a better way of managing lane imbalance in comparison to the current systems.

5. Conclusion

The amalgamation of the AFSNS denoising, CARAS classification, VEDCN classification and DWOCE density estimation provides an extremely efficient system of the real-time monitoring of traffic and the smart congestion analysis. The system enhances the clarity of the images by using frequency selective noise suppression to enable the system to extract structures of vehicles in high reliability to poor lighting and motion conditions. The context-sensitive group allows to ensure that the areas are properly segregated even when the traffic is congested or blocked, and the classification module provides the stability of recognition of various types of vehicles. The weighted dynamic density estimation also improves the estimations of the traffic on the basis of the vehicle type, occupancy of the lane, as well as the motion behavior by giving more realistic estimations of the flow conditions. It has been established that the experimental results are far superior to PSNR, SSIM, Dice score, mAP, and accuracy, which were current methods. The proposed methodology is an effective, universal, and scalable solution, which can be applied to a modern system of smart traffic management and real-life surveillance.

References

1. Shamshanovna, R. B., Amangeldy, N., Kassymova, A., Kudubayeva, S., Kurmetbek, B., Barlybayev, A., ... & Buribayeva, A. (2025). Combined-adaptive image preprocessing method based on noise detection. International Journal of Electrical and Computer Engineering (IJECE), 15(2), 1584-1592.

2. Singh, A., Sethi, G., & Kalra, G. S. (2020). Spatially adaptive image denoising via enhanced noise detection method for grayscale and color images. IEEE Access, 8, 112985-113002.

3. Chen, X., Wu, S., Shi, C., Huang, Y., Yang, Y., Ke, R., & Zhao, J. (2020). Sensing data supported traffic flow prediction via denoising schemes and ANN: A comparison. IEEE Sensors Journal, 20(23), 14317-14328.

4. Pawlicki, M., & Choraś, R. S. (2021). Preprocessing pipelines including block-matching convolutional neural network for image denoising to robustify deep reidentification against evasion attacks. Entropy, 23(10), 1304.

5. Qureshi, A. M., Al Mudawi, N., Alonazi, M., Chelloug, S. A., & Park, J. (2024). Road traffic monitoring from aerial images using template matching and invariant features. Computers, Materials and Continua, 78(3), 3683-3701.

6. Jagannathan, P., Rajkumar, S., Frnda, J., Divakarachari, P. B., & Subramani, P. (2021). Moving vehicle detection and classification using gaussian mixture model and ensemble deep learning technique. Wireless Communications and Mobile Computing, 2021(1), 5590894.

7. Wieczorek, G., Tahir, S. B. U. D., Akhter, I., & Kurek, J. (2023). Vehicle detection and recognition approach in multi-scale traffic monitoring system via graph-based data optimization. Sensors, 23(3), 1731.

8. Yusuf, M. O., Hanzla, M., Rahman, H., Sadiq, T., Al Mudawi, N., Almujally, N. A., & Algarni, A. (2024). Enhancing vehicle detection and tracking in UAV imagery: a pixel Labeling and particle filter approach. IEEE Access, 12, 72896-72911.

9. Ashkanani, M., AlAjmi, A., Alhayyan, A., Esmael, Z., AlBedaiwi, M., & Nadeem, M. (2025). A self-adaptive traffic signal system integrating real-time vehicle detection and license plate recognition for enhanced traffic management. Inventions, 10(1), 14.

10. Vinoth, K., & Sasikumar, P. (2024). Multi-sensor fusion and segmentation for autonomous vehicle multi-object tracking using deep Q networks. Scientific Reports, 14(1), 31130.

11. Omar, N., Sengur, A., & Al-Ali, S. G. S. (2020). Cascaded deep learning-based efficient approach for license plate detection and recognition. Expert Systems with Applications, 149, 113280.

12. Shim, S. O., Imtiaz, R., Habibullah, S., & Alshdadi, A. A. (2024). Optimizing Deep Learning for Efficient and NoiseRobust License Plate Detection and Recognition. International Journal of Advanced Computer Science & Applications, 15(4).

13. Ness, S. (2025). Vehicle detection and recognition approach in smart surveillance system: A comparative analysis. Array, 100473.

14. Rodríguez-Esparza, E., Ramos-Soto, O., Masegosa, A. D., Onieva, E., Oliva, D., Arriandiaga, A., & Ghosh, A. (2024). Optimizing road traffic surveillance: A robust hyper-heuristic approach for vehicle segmentation. IEEE Access, 12, 29503-29524.

15. Qureshi, A. M., Almujally, N. A., Alotaibi, S. S., Alatiyyah, M. H., & Park, J. (2023). Intelligent traffic surveillance through multi-label semantic segmentation and filter-based tracking. Computers, Materials & Continua, 76(3), 3707-3725.

16. Handoko, H., Pratama, J. H., & Yohanes, B. W. (2021). Traffic sign detection optimization using color and shape segmentation as pre-processing system. TELKOMNIKA (Telecommunication Computing Electronics and Control), 19(1), 173-181.

17. Luo, X., Wang, Y., Cai, B., & Li, Z. (2021). Moving object detection in traffic surveillance video: new MOD-AT method based on adaptive threshold. ISPRS International Journal of Geo-Information, 10(11), 742.

18. Gulati, I., & Srinivasan, R. (2019). Image processing in intelligent traffic management. International Journal of Recent Technology and Engineering, 8(2S4), 213-218.

19. Safaei, N., Smadi, O., Safaei, B., & Masoud, A. (2021). Efficient road crack detection based on an adaptive pixel-level segmentation algorithm. Transportation Research Record, 2675(9), 370-381.

20. Hussain, I., & Alam, L. (2024). Adaptive road crack detection and segmentation using Einstein operators and ANFIS for real-time applications. J. Intell Syst. Control, 3(4), 213-226.

21. Almujally, N. A., Wu, T., Alhasson, H. F., Hanzla, M., Jalal, A., & Liu, H. (2025). UAV-based intelligent traffic surveillance using recurrent neural networks and Swin transformer for dynamic environments. Frontiers in Neurorobotics, 19, 1681341.

22. Anitha, M., & Kumar, D. S. (2025). A Modular and Comparative Framework for Robust License Plate Recognition using Traditional Preprocessing and Lightweight OCR. International Journal of Communication Networks and Information Security, 17(5), 83-89.

23. Naseer, A., Al Mudawi, N., Abdelhaq, M., Alonazi, M., Alazeb, A., Algarni, A., & Jalal, A. (2024). CNN-based object detection via segmentation capabilities in Outdoor natural scenes. Ieee Access, 12, 84984-85000.

24. Yusuf, M. O., Hanzla, M., Rahman, H., Sadiq, T., Al Mudawi, N., Almujally, N. A., & Algarni, A. (2024). Enhancing vehicle detection and tracking in UAV imagery: a pixel Labeling and particle filter approach. IEEE Access, 12, 72896-72911.

25. Cristiani, A. L., Immich, R., Akabane, A. T., Madeira, E. R. M., Villas, L. A., & Meneguette, R. I. (2020). Atrip: Architecture for traffic classification based on image processing. Vehicles, 2(2), 303-317.

26. Rafique, A. A., Al-Rasheed, A., Ksibi, A., Ayadi, M., Jalal, A., Alnowaiser, K., ... & Park, J. (2023). Smart traffic monitoring through pyramid pooling vehicle detection and filter-based tracking on aerial images. IEEE Access, 11, 2993-3007.

27. Khan, H., Kushwah, K. K., Maurya, M. R., Singh, S., Jha, P., Mahobia, S. K., ... & Sadasivuni, K. K. (2022). Machine learning driven intelligent and self adaptive system for traffic management in smart cities. computing, 104(5), 1203-1217.

28. Won, M. (2020). Intelligent traffic monitoring systems for vehicle classification: A survey. IEEE Access, 8, 73340-73358.

29. Yusuf, M. O., Hanzla, M., Al Mudawi, N., Sadiq, T., Alabdullah, B., Rahman, H., & Algarni, A. (2024). Target detection and classification via EfficientDet and CNN over unmanned aerial vehicles. Frontiers in Neurorobotics, 18, 1448538.

30. Al Mudawi, N., Qureshi, A. M., Abdelhaq, M., Alshahrani, A., Alazeb, A., Alonazi, M., & Algarni, A. (2023). Vehicle detection and classification via YOLOv8 and deep belief network over aerial image sequences. Sustainability, 15(19), 14597.

31. Thenge, M. A. N., & Kamble, R. D. (2025). Real-Time Traffic Sign Recognition and Classification System Using OpenCV. Sciences of Conservation and Archaeology, 37(1), 151-168.

32. Bisio, I., Garibotto, C., Haleem, H., Lavagetto, F., & Sciarrone, A. (2022). A systematic review of drone based road traffic monitoring system. Ieee Access, 10, 101537-101555.

33. Sindhu, C., Sharanya, A., & Sharma, K. V. (2025). AI-Driven Smart Traffic Control System Using CNN and YOLO for Real-Time Urban Mobility Optimization. Frontiers in Collaborative Research, 3(1), 1-11.

34. Shokravi, H., Shokravi, H., Bakhary, N., Heidarrezaei, M., Rahimian Koloor, S. S., & Petrů, M. (2020). A review on vehicle classification and potential use of smart vehicle-assisted techniques. Sensors, 20(11), 3274.

35. Valencia, D., España, E. M., & Añasco, M. M. (2024). Impact of the preprocessing stage on the performance of offline automatic vehicle counting using YOLO. IEEE Latin America Transactions, 22(9), 723-732.

36. Rakesh, S., Hegde, N. P., Gopalachari, M. V., Jayaram, D., Madhu, B., Hameed, M. A., … Kumar, L. S. (2023). Moving object detection using modified GMM based background subtraction. *Measurement: Sensors, 30,* 100898 [

37. Sindhu, S. (2025). Geospatial analysis and machine learning-based prediction of urban flood zones. *National Journal of Ubiquitous Computing and Intelligent Environments*, 2(2), 1–5.

38. Yasir, M. A., & Ali, Y. H. (2022). Comparative analysis of GMM, KNN, and ViBe background subtraction algorithms applied in dynamic background scenes of video surveillance system. Eng. Technol. J, 40(4), 617-626

39. Sanches, S. R. R., Sementille, A. C., Aguilar, I. A., & Freire, V. (2021). Recommendations for evaluating the performance of background subtraction algorithms for surveillance systems. *Multimedia Tools and Applications*, *80*(3), 4421-4454.

40. Zhang, F., Li, C., & Yang, F. (2019). Vehicle detection in urban traffic surveillance images based on convolutional neural networks with feature concatenation. *Sensors, 19*(3), 594.https://doi.org/10.3390/s19030594

41. Bai, T., Luo, J., Zhou, S., Lu, Y., & Wang, Y. (2024). Vehicle-type recognition method for images based on improved faster R-CNN model. *Sensors, 24*(8), 2650. https://doi.org/10.3390/s24082650

42. Hanzla, M., Ali, S., & Jalal, A. (2024, February). Smart traffic monitoring through drone images via YOLOv5 and Kalman filter. In *2024 5th International Conference on Advancements in Computational Sciences (ICACS)* (pp. 1–8). IEEE. https://doi.org/10.1109/ICACS56812.2024.

43. Zhang, Y., Guo, Z., Wu, J., Tian, Y., Tang, H., & Guo, X. (2022). Real-time vehicle detection based on improved YOLOv5. *Sustainability, 14*(19), 12274. https://doi.org/10.3390/su141912274

44. Rajan, C., & Dineshkumar, P. (2025). Embedded machine learning framework for real-time prediction of user information needs in intelligent systems. *Journal of Integrated VLSI, Embedded and Computing Technologies*, 2(2), 73–79.

45. Mejail, M., & Nestares, B. K. (2026). Spatiotemporal human motion prediction with interaction-constrained learning. *Journal of Wireless Intelligence and Spectrum Engineering*, 3(1), 20–27.