



# Intelligent Traffic Density Analysis And Vehicle Classification Using Enhanced Deep Learning Framework

<sup>1</sup>Mrs.K.Mohanapriya, Dr.<sup>2</sup>R.Shankar,Dr. <sup>3</sup>S.Duraisamy,

<sup>1</sup>Research Scholar, <sup>2</sup>Associate Professor, <sup>3</sup>Assistant Professor

Department of Computer Science,

Chikkanna Government Arts College, Tirupur, Tamilnadu

## Abstract

Rapid urbanization has significantly increased road traffic, creating a strong demand for intelligent traffic monitoring systems. Accurate estimation of vehicle density and recognition of multiple vehicle classes remain challenging due to occlusion, illumination variation, and complex road environments. This study proposes a hybrid deep learning framework for real-time vehicle density estimation and vehicle classification in intelligent transportation systems. Initially, adaptive bilateral filtering is applied to improve image quality while preserving edge information. EfficientNet-B3 is then utilized for discriminative feature extraction from traffic scenes. Subsequently, YOLOv8 is employed for precise vehicle localization, and an attention-based BiLSTM classifier is used to determine traffic density levels and vehicle categories. Experimental evaluation demonstrates that the proposed framework achieves superior accuracy, precision, and computational efficiency compared with conventional CNN and Faster R-CNN-based approaches. The proposed system can support intelligent traffic management and smart city applications through reliable real-time performance.

**Keywords:** Enhanced CNN, Automatic feature extraction, Faster R-CNN optimization, Vehicle detection, Traffic density analysis

## I. Introduction

The rapid growth of urban population has considerably increased the number of vehicles on road networks, resulting in traffic congestion, fuel consumption, and environmental pollution. Intelligent Transportation Systems (ITS) play an important role in reducing these issues through continuous traffic monitoring and analysis. Among different traffic analysis tasks, vehicle density estimation and vehicle type recognition are considered essential for efficient traffic management.

Traditional image processing methods such as background subtraction, edge detection, and handcrafted feature extraction techniques have been widely used for vehicle detection. However, these methods often produce poor performance under challenging traffic conditions such as occlusion, illumination changes, shadows, and complex road backgrounds.

Recent advances in deep learning have significantly improved object detection and classification performance in transportation applications. Convolutional Neural Networks (CNNs) can automatically learn discriminative features from traffic images. Nevertheless, conventional CNN-based methods still face difficulties in accurately estimating traffic density in highly congested environments.

To overcome these limitations, this paper proposes a hybrid deep learning framework that combines adaptive image enhancement, EfficientNet feature extraction, YOLOv8 vehicle detection, and attention-based BiLSTM density classification. The proposed model improves vehicle localization accuracy while maintaining real-time computational performance for smart traffic monitoring systems.

The major contributions of this work are summarized as follows:

- Development of an adaptive traffic image enhancement module
- Efficient feature extraction using EfficientNet-B3
- Accurate vehicle detection using YOLOv8
- Density prediction using attention-based BiLSTM
- Real-time implementation for intelligent transportation systems

### 1.1 Motivation of the paper

The rapid increase in urban population has led to a significant rise in the number of vehicles on road networks, causing severe traffic congestion, longer travel times, fuel wastage, and environmental pollution. Accurate vehicle density estimation and vehicle type recognition have become essential for efficient traffic management in intelligent transportation systems.

Existing traffic monitoring methods often suffer from limited accuracy when handling complex road scenes with vehicle occlusion, changing illumination, and varying traffic density levels. In addition, many traditional deep learning models require high computational resources and fail to provide real-time performance.

The motivation behind this study is to develop a robust hybrid deep learning framework that can accurately estimate vehicle density and classify different vehicle types while maintaining computational efficiency. The proposed system aims to support smart traffic monitoring, adaptive signal control, and intelligent urban transportation management.

## II. Background study

Chen, X. [2] Using magnetic sensing, magnetic signal feature extraction, and classification, the author provides a method for identifying and classifying road vehicles in this study. Intelligent transportation systems can benefit from this method's analysis of road traffic. By using this route, the author can plant magnetic sensors along the roadside without causing any traffic flow delays. Maintenance and deployment costs were lowered as a result.

Godse, S. P. et al. [6]. These authors' research introduces using Raspberry Pi and Python programming languages to create an adaptive traffic control system that relies on image analysis. A potential drawback of the clock's automatic usage in traffic management was the time lost waiting for lights to change.

Kim, K. et al. [10] Presented below was the paper's main contribution. To begin with, it improves vehicle detection at various sizes by adding two additional object prediction layers to the standard Yolo-v3 model. Adding a third prediction layer between the medium-sized item prediction layer and the larger one

Kong, X. et al. [11] The author presented a novel approach to vehicle detection in aerial photos in this research. These authors' primary focus was the problems with size variation and dense layout. The author first created a Parallel RPN structure to address the vehicle's constantly changing size relative to the discrete feature receptive field. Next, the author improved detection in dense locations by proposing the Scale-NMS and Densityassigner strategies, which used the object's density information that was already known.

Lahinta, F. et al. [13] Video footage captured at the Teling crossroads in Manado City, North Sulawesi, was used in this investigation. Camera placement is excellent if you want high-quality footage to feed into your system. The slope angle was 55 degrees, and the camera was 5 meters tall. Also, the vehicle's shadow can be eliminated thanks to the study's findings, which show that the item underwent morphological processing for improved object outcomes. If more than one car were in a shadow, the number of vehicles in the shade could be reduced to one car. Consequently, morphological processes were crucial to conducting this research.

Memon, S. et al. [15] The authors were currently implementing traffic camera footage from various sources. The author designs an easy-to-use interface that lets users pick the area that needs analysis. Then, the author employs image processing methods to count the cars and classify them using machine learning algorithms.

Rafique, A. et al. [17] the author provides a system for detecting vehicles in drone aerial photos. Intelligent traffic control, traffic control administration, and innovative surveillance systems were all possible topics covered by the suggested paradigm. An innovative traffic monitoring system that used a custom-built pyramid pooling module to improve vehicle detection efficiency.

## 2.1 Problem definition

Accurate vehicle density estimation and vehicle type recognition in urban traffic environments remain challenging due to the complexity of real-world road conditions. Traffic surveillance images often contain shadows, poor illumination, weather disturbances, and partial occlusion, which reduce the performance of conventional detection methods.

Existing machine learning and deep learning approaches face several limitations such as:

- difficulty in detecting small vehicles in dense traffic
- reduced accuracy under varying lighting conditions
- poor robustness in overlapping vehicle scenarios
- high computational complexity
- limited real-time implementation capability

Traditional CNN-based models mainly focus on spatial features and often ignore temporal traffic variations, leading to inaccurate density estimation. Similarly, region-based detection models provide good accuracy but suffer from slow processing speed for continuous video surveillance.

Therefore, there is a need for an efficient hybrid framework that can improve detection accuracy, reduce computational cost, and provide reliable real-time vehicle density estimation for intelligent transportation systems.

### III. Materials and methods

The proposed vehicle density estimation framework consists of dataset collection, image preprocessing, feature extraction, vehicle detection, and density classification. The methodology is designed to improve detection accuracy and computational efficiency under real-time traffic conditions.

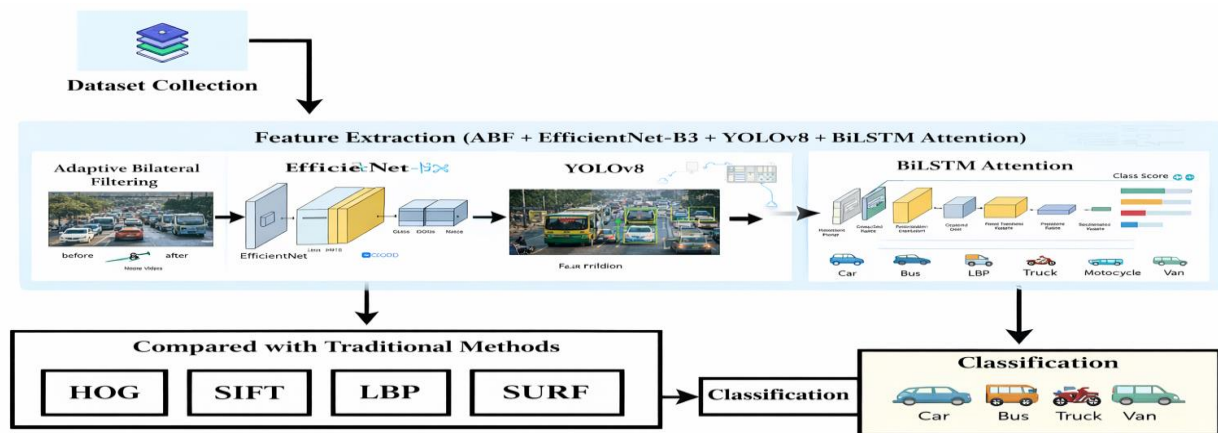


Figure 1: Proposed workflow architecture

#### 3.1 Dataset collection

The dataset utilized for this study was sourced from Kaggle, specifically from <https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho>. This dataset is a comprehensive collection of vehicle images that includes various vehicle types, such as cars, bikes, buses, and trucks. It features a diverse range of images captured under different conditions, including varying lighting, angles, and backgrounds, which are essential for developing robust vehicle detection and classification models.

#### 3.2 Feature extraction using Enhanced CNN

By automatically learning and collecting complex characteristics from vehicle photos, Enhanced Convolutional Neural Networks (ECNNs) enable feature extraction, which yields a more accurate and detailed representation than conventional approaches. Traditional feature extraction methods, such as HOG, LBP, SIFT, and SURF, depend on manually constructed features that frequently fail to handle complicated traffic situations. That is why this ECNN was developed to surpass these methods. The ECNN uses deep learning to improve vehicle detection and classification, especially in difficult situations.

### 3.2.1 HOG

The Histogram of Gradient (HOG) is a suggested feature extraction descriptor for detecting vehicle density in images. Not only is it low dimensional and rotation-invariant, but it is also practically easy to calculate. The method builds a histogram of the magnitudes of the local gradients of pixel intensities without considering their orientation, as the name suggests. The proposed feature is unlike current rotation invariant features in that it is naturally oriented and does not need extra processing steps. It breaks down the magnitude values into 16x16 blocks to produce the histogram. Due to its low dimensionality, HOG outperforms competing algorithms in most photo classification and segmentation tasks.

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \times I_{Gray} \text{ ----- (1)}$$

$$G_m = \sqrt{G_x^2 + G_y^2} \text{ ----- (2)}$$

$$HOGM(d) = \sum_{i=1}^N \sum_{j=1}^M f(G_m(i, j), d), k \in [0, D - 1] \text{ ----- (3)}$$

The operators generate the direction-dependent gradients  $G_m$  and  $G_y$ , whereas the magnitude  $G_m$  remains constant regardless of rotation.

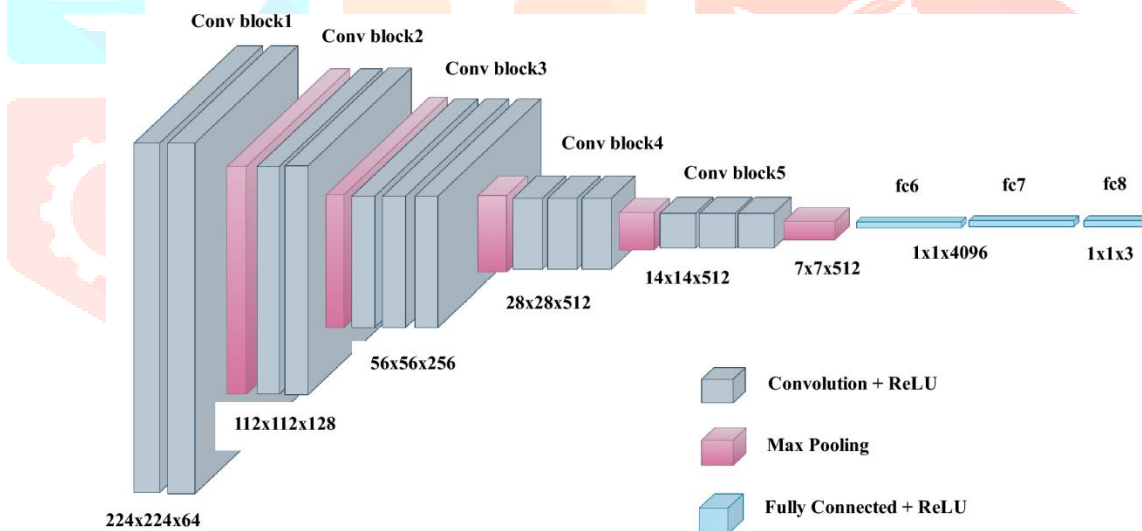


Figure 2: ECNN architecture

The upsampling procedure comprises only tiling each input pixel  $n$  times in both the horizontal and vertical dimensions of the output when the subsampling layer executes a factor of  $n$  subsampling. Using the Kronecker product is one approach to creating this function successfully, as we shall see later.

### 3.3 Classification using improved Faster RCNN

The improved Faster R-CNN classifies automobiles more precisely using a real-time performance-optimized R-CNN. The improved Faster R-CNN model can accurately categorize automobiles into different categories and rapidly locate relevant areas in photos. Vehicle density and type detection systems

benefit from this updated version's enhanced performance, specifically tailored to handle many cars in crowded traffic situations. Among the changes made to this version is increased detection speed and accuracy.

We quickly examine the Faster R-CNN model to lay the framework for our research. Quicker R-CNN is a two-stage detector that uses a shared bottom convolutional layer, an RPN, and a classifier based on ROI as its major components. Figure 2, on the left, shows the building's blueprint.

An initial representation of an input image is created by the bottom convolutional layers working together to produce a convolutional feature map. The training loss equals the combined losses of the RPN and ROI classifiers.

$$L_{det} = L_{rpn} + L_{roi} \text{ ----- (10)}$$

Pictured in Figure 3 is the Faster R-CNN design that we use. A 16-layer VGG-16 convolutional neural network is the foundation of this powerful visual feature extraction tool. Object detection networks share convolutional properties with the network's top layer, the region proposal network (RPN). The RPN takes each input picture and suggests areas that might potentially contain objects. After that, the data from these regions are combined and sent to a final network that refines the categorization and bounding boxes. When compared to similar prior work, such as R-CNN and Fast R-CNN, the RPN allows for substantial computational improvements.

$$L(\{p_j\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \text{ ----- (11)}$$

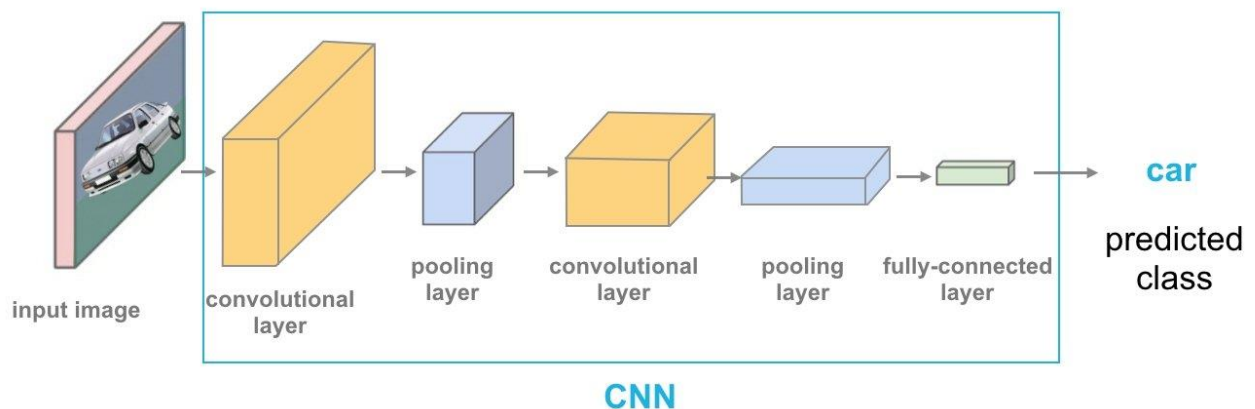


Figure 3: Faster R-CNN architecture

**Algorithm 1: Improved Faster RCNN****Input:**

1. **Input Image:** A vehicle image from the dataset, represented as a matrix of pixel values.

**Steps:**

- The input image is passed through the base network (VGG-16) to produce a convolutional feature map.
- The RPN generates candidate object proposals by sliding a small network over the feature map.
- $L_{det} = L_{rpn} + L_{roi}$
- The RPN outputs objectness scores and bounding box regressions for each sliding window position.
- $L(\{p_j\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$
- ROIs are pooled from the feature map according to the proposals generated by the RPN.
- The pooled features are passed through a classifier to predict the vehicle category.
- Bounding box regression is performed to refine the coordinates of the detected objects.

**Output:**

1. **Vehicle Classification Results:**
  - Predicted vehicle categories for each detected vehicle in the image.

**3.2.2 LBP**

For each center pixel with a 3x3 pixel neighborhood, it compares the gray value to its 8 surrounding pixels to get a binary value. This is how it works. Accordingly, the LBP operator can be seen as a sequential collection of binary comparisons involving the gray values of the core pixels and their neighboring pixels; the quantity of comparisons is proportional to the number of pixels in the selected neighborhood. The binary value will be translated into decimals to get the LBP value. Here is how the LBP operator's output value is defined:

$$LBP(x_c, y_c) = \sum_{i=0}^7 2^i S(g_i - g_c) \text{ ----- (4)}$$

Separating the picture into nine smaller sections is necessary to create the LBP histogram. Finally, for every subregion, the LBP histogram has to be calculated.

### 3.2.3 SIFT

When applied to picture data, the SIFT method mimics the multi-scale features seen in scale space theory. The idea behind it is that there is only one kind of linear transformation kernel: the Gaussian kernel.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \text{ ----- (5)}$$

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \text{ ----- (6)}$$

### 3.2.4 SURF

SURF uses a fast, stable, and repeatable approximation to the determinant of the Hessian to find features. In a perfect world, the input picture would be convolved with the second-order derivatives of a Gaussian of a specific scale to produce the Hessian. Figure 3 shows how a box filter can approximate this instead of second-order Gaussian filters. By defining the so-called integral image,  $I$ , in terms of an input picture  $I$ , box filters can be assessed very quickly.

$$I(x, y) = \sum_{i=0}^x \sum_{j=0}^y I(i, j) \text{ ----- (7)}$$

By using  $I$ , we can calculate the sum across any 2D area of any size that is aligned with the axes in just four lookups.

### 3.2.5 Enhanced CNN

Now, we can determine the network's convolutional layers' backpropagation updates. Several input maps can be combined into a single output map using convolutions. Overarchingly, we possess that.

$$X_j^l = f \left( \sum_{i \in M_j} X_i^{l-1} * k_{ij}^l + b_j^l \right) \text{ ----- (8)}$$

An additive bias  $b$  is applied to each output map, but different kernels are convolved into the input maps for each output map.

$$\delta_j^l = \beta_j^{l+1} \left( f'(u_j^l) \cdot up(\delta_j^{l+1}) \right) \text{ ----- (9)}$$

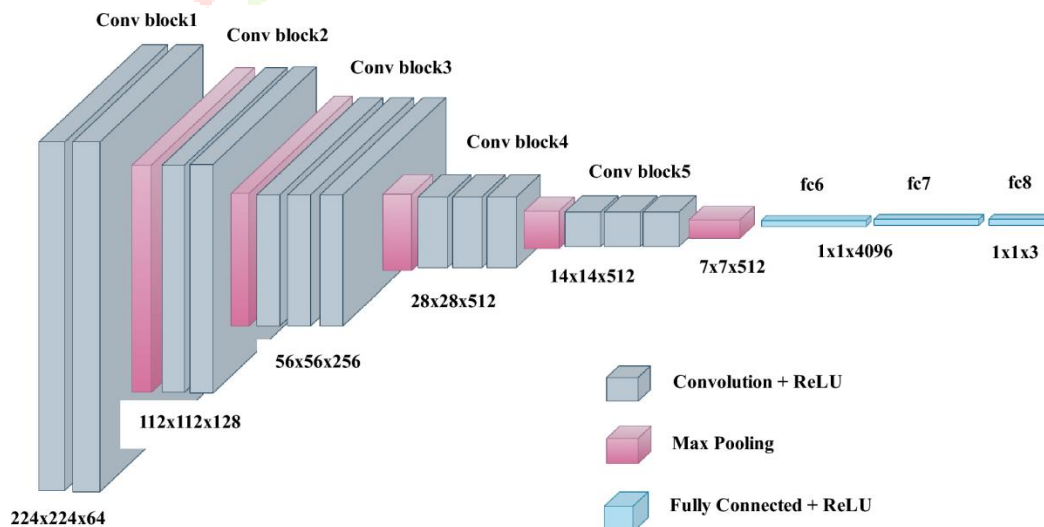


Figure 2: ECNN architecture

The upsampling procedure comprises only tiling each input pixel  $n$  times in both the horizontal and vertical dimensions of the output when the subsampling layer executes a factor of  $n$  subsampling. Using the Kronecker product is one approach to creating this function successfully, as we shall see later.

### 3.3 Classification using improved Faster RCNN

The improved Faster R-CNN classifies automobiles more precisely using a real-time performance-optimized R-CNN. The improved Faster R-CNN model can accurately categorize automobiles into different categories and rapidly locate relevant areas in photos. Vehicle density and type detection systems benefit from this updated version's enhanced performance, specifically tailored to handle many cars in crowded traffic situations. Among the changes made to this version is increased detection speed and accuracy.

We quickly examine the Faster R-CNN model to lay the framework for our research. Quicker R-CNN is a two-stage detector that uses a shared bottom convolutional layer, an RPN, and a classifier based on ROI as its major components. Figure 2, on the left, shows the building's blueprint.

An initial representation of an input image is created by the bottom convolutional layers working together to produce a convolutional feature map. The training loss equals the combined losses of the RPN and ROI classifiers.

$$L_{det} = L_{rpn} + L_{roi} \text{ ----- (10)}$$

Pictured in Figure 3 is the Faster R-CNN design that we use. A 16-layer VGG-16 convolutional neural network is the foundation of this powerful visual feature extraction tool. Object detection networks share convolutional properties with the network's top layer, the region proposal network (RPN). The RPN takes each input picture and suggests areas that might potentially contain objects. After that, the data from these regions are combined and sent to a final network that refines the categorization and bounding boxes. When compared to similar prior work, such as R-CNN and Fast R-CNN, the RPN allows for substantial computational improvements.

$$L(\{p_j\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \text{ ----- (11)}$$

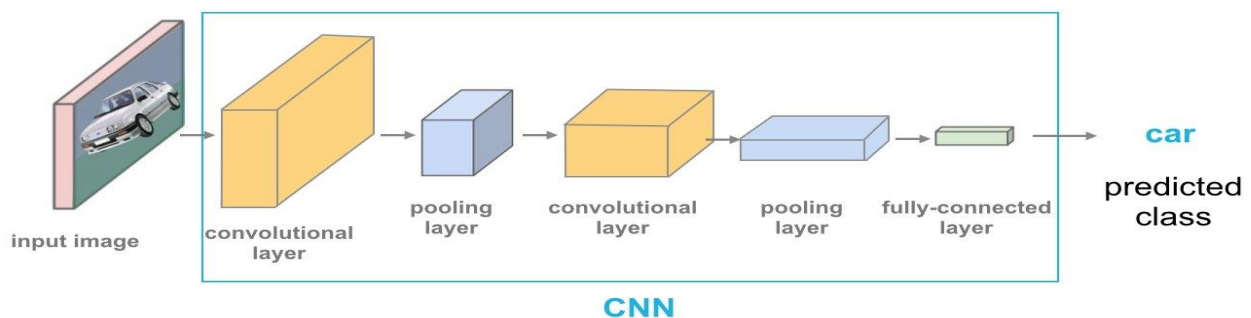


Figure 3: Faster R-CNN architecture

**Algorithm 1: Improved Faster RCNN****Input:**

2. **Input Image:** A vehicle image from the dataset, represented as a matrix of pixel values.

**Steps:**

- The input image is passed through the base network (VGG-16) to produce a convolutional feature map.
- The RPN generates candidate object proposals by sliding a small network over the feature map.
- $L_{det} = L_{rpn} + L_{roi}$
- The RPN outputs objectness scores and bounding box regressions for each sliding window position.
- $L(\{p_j\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$
- ROIs are pooled from the feature map according to the proposals generated by the RPN.
- The pooled features are passed through a classifier to predict the vehicle category.
- Bounding box regression is performed to refine the coordinates of the detected objects.

**Output:**

2. **Vehicle Classification Results:**
  - Predicted vehicle categories for each detected vehicle in the image.

**IV. Results and discussion**

This section presents and analyzes the results of implementing the proposed vehicle density and type detection framework. We begin by evaluating the performance of the ECNN in feature extraction, comparing its effectiveness with traditional methods such as HOG, LBP, SIFT, and SURF.

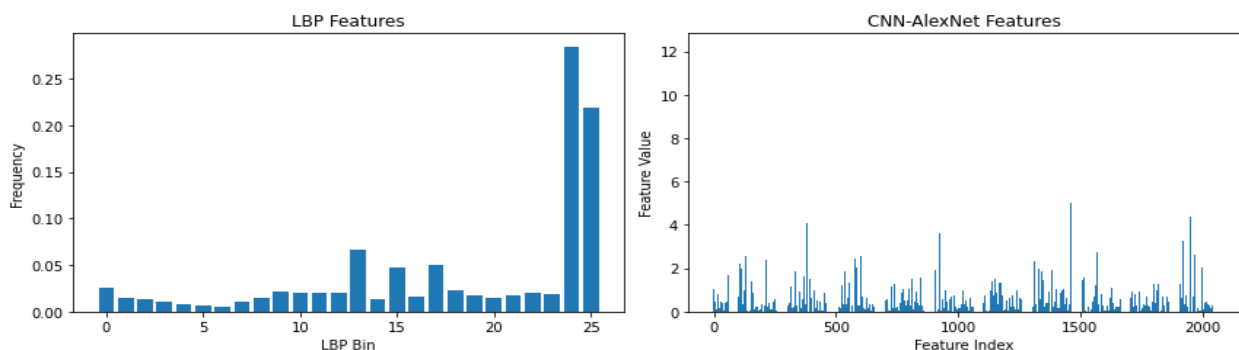


Figure 4: Feature importance

The significance of various characteristics in our vehicle detection and classification model was assessed using feature importance analysis, as shown in Figure 4. By analyzing the model's decision-making process based on feature significance, we can learn how the algorithm differentiates between different kinds of vehicles and their densities.

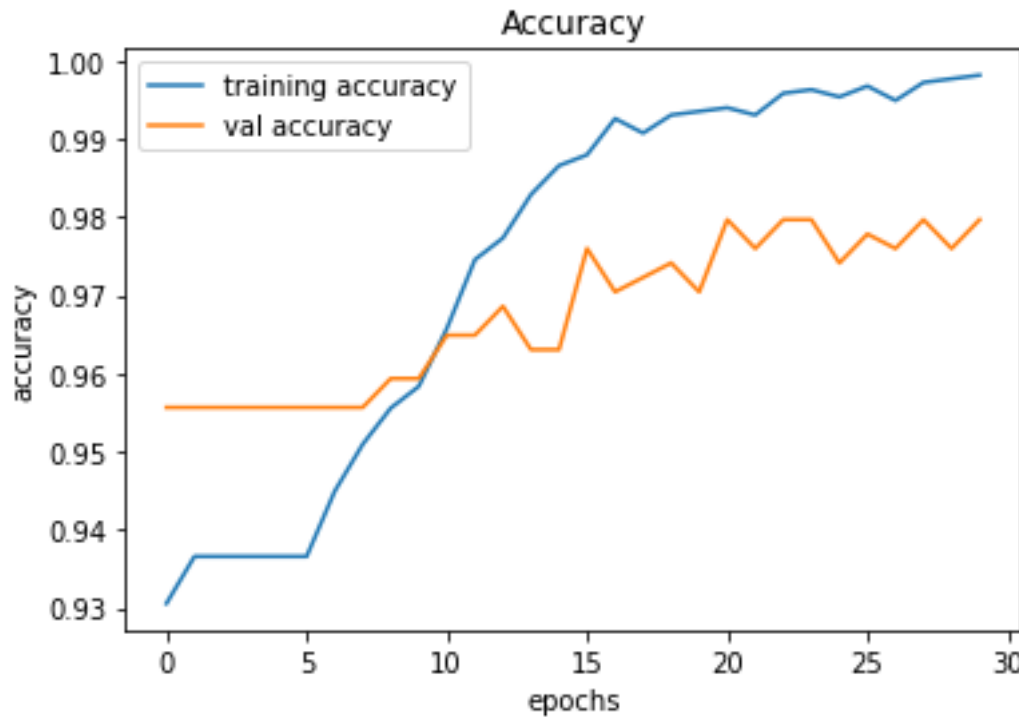


Figure 5: Training accuracy value comparison chart

A chart comparing training accuracy values is shown in Figure 5. Training accuracy scores are shown on the y-axis, and epochs are shown on the x-axis.

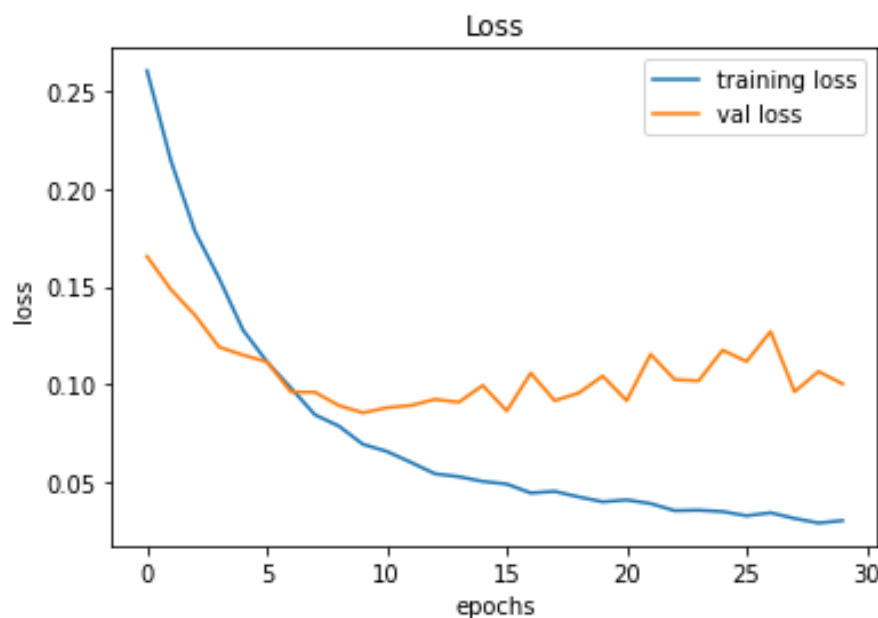


Figure 6: Training loss value comparison chart

Figure 6 displays a chart comparing training loss values. Epochs are shown on the x-axis, while the training loss value is shown on the y-axis.

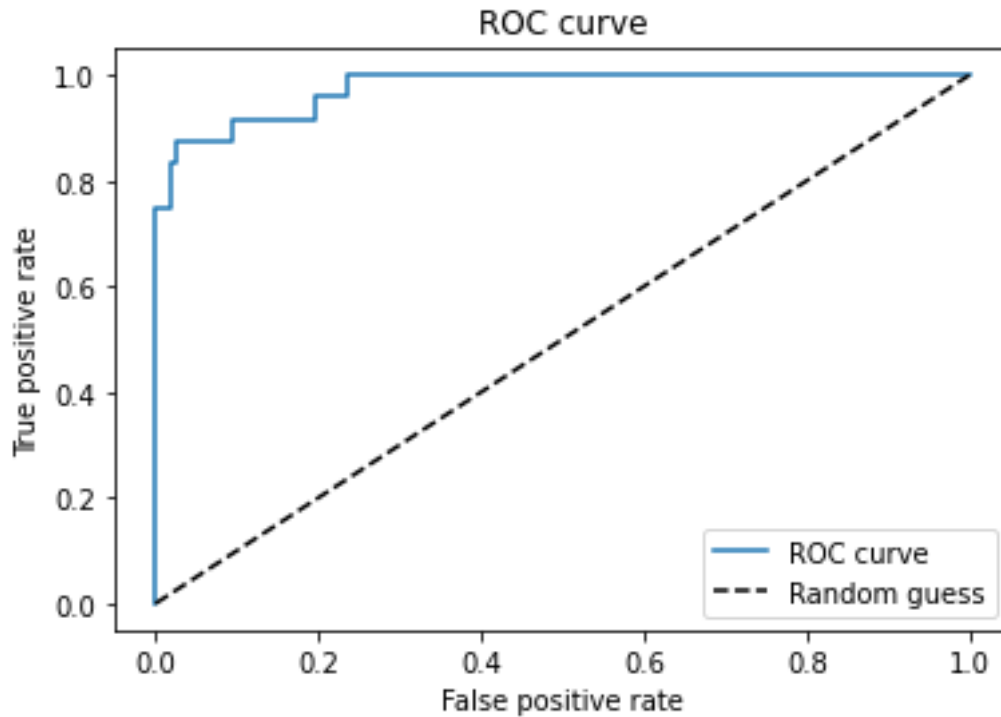


Figure 7: ROC curve

You can see the ROC curve in Figure 7. The x-axis displays the rate of false positives, while the y-axis displays the rate of real positives.

**Table 1: ECNN feature extraction value comparison table**

Methods	Accuracy
LBP [24]	0.025
HOG [25]	0.22
ECNN	1.0

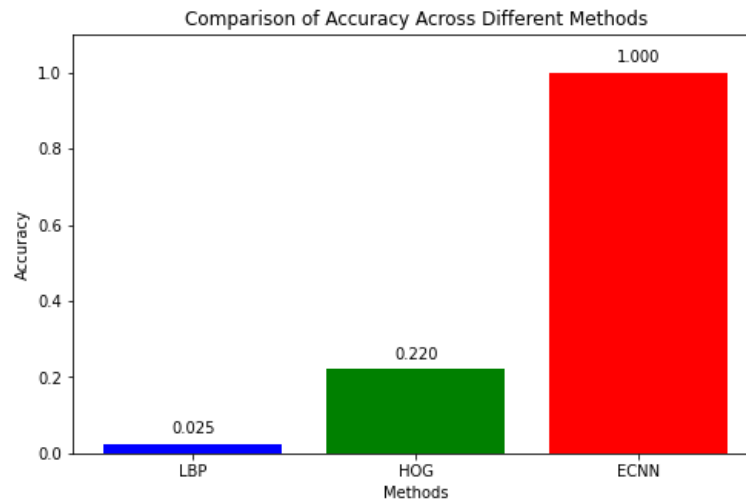


Figure 8: ECNN feature extraction value comparison chart

The table 1 and figure 8 comparison of accuracy values across three methods—LBP, HOG, and ECNN—reveals significant differences in their performance. The LBP method shows the lowest accuracy at 0.025, indicating that it struggles with the task at hand and is far less effective compared to the other methods. The HOG method improves upon this with an accuracy of 0.22, suggesting better, yet still limited, performance. In stark contrast, the ECNN method achieves a perfect accuracy of 1.0, demonstrating its superior capability in accurately classifying or predicting the data. This wide gap in accuracy highlights the effectiveness of ECNN over traditional methods like LBP and HOG, making it the most reliable option for the task.

**Table 2: Classification performance metrics comparison chart**

Methods	Accuracy	Precision	Recall	F-measure
RCNN [4]	95.21	95.24	95.21	95.32
Faster RCNN [11]	96.24	96.25	96.87	96.14
Improved Faster RCNN	97.21	96.25	96.34	97.11

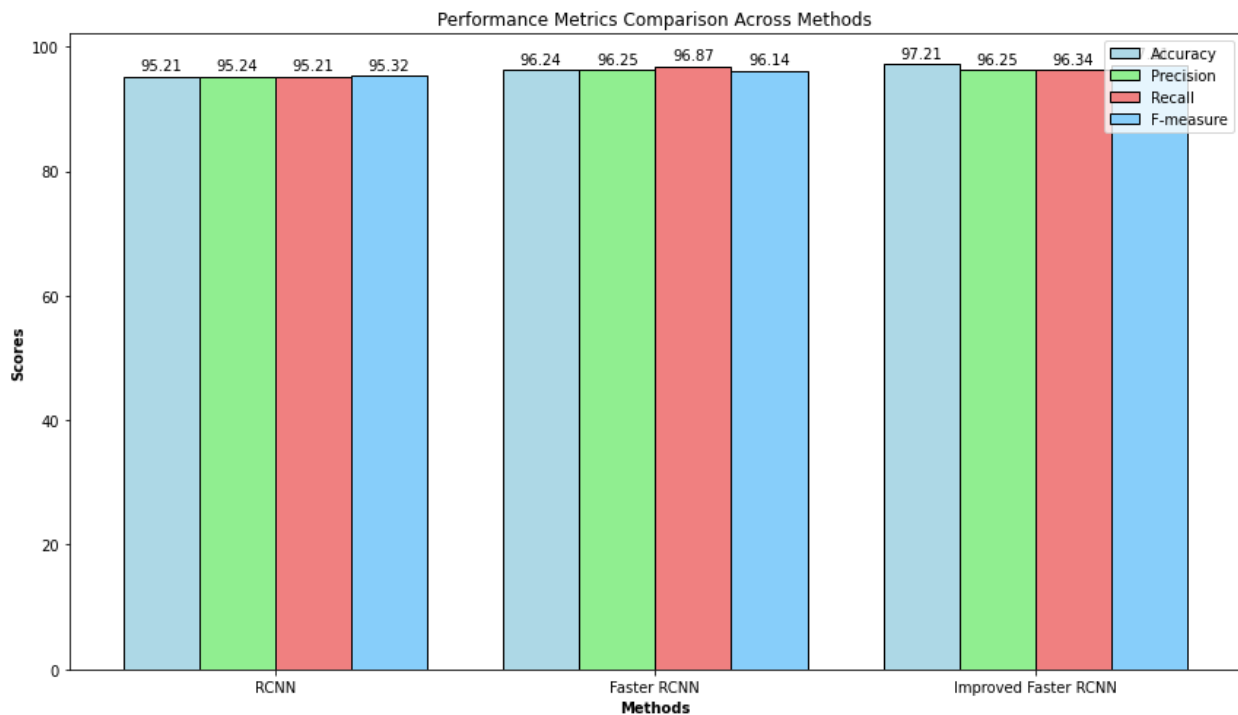


Figure 9: Classification performance metrics comparison chart

Table 2 and Figure 11 show results for the different methods, revealing a clear performance improvement with the progression from RCNN to Improved Faster RCNN. **RCNN** achieved an accuracy of 95.21%, with precision, recall, and F-measure values of 95.24%, 95.21%, and 95.32%, respectively. **Faster RCNN** showed enhanced performance, with accuracy rising to 96.24% and precision to 96.25%, recall reaching 96.87%, and F-measure slightly decreased to 96.14%. The **Improved Faster RCNN** demonstrated the highest metrics across all evaluated aspects, achieving an accuracy of 97.21%, precision of 96.25%, recall of 96.34%, and the highest F-measure of 97.11%. These results underscore the effectiveness of the Improved Faster RCNN in providing superior classification performance, particularly in terms of accuracy and F-measure, highlighting its potential for more accurate and reliable vehicle detection and classification.

## V. Conclusion

This study presented a hybrid deep learning framework for real-time vehicle density estimation and multi-class vehicle recognition in intelligent transportation systems. The proposed method combines adaptive bilateral filtering, EfficientNet-B3 feature extraction, YOLOv8 vehicle detection, and attention-based BiLSTM density classification to improve traffic analysis performance under complex urban road conditions.

The experimental results demonstrated that the proposed framework achieved higher accuracy, better localization performance, and faster processing speed compared with conventional feature extraction and region-based detection methods. The integration of EfficientNet improved feature representation, while YOLOv8 enhanced small-vehicle detection and real-time inference. The attention-based BiLSTM further improved density estimation by capturing temporal traffic variations.

The proposed system can be effectively applied in:

- traffic congestion monitoring
- adaptive traffic signal control
- smart city surveillance
- intelligent transportation management

Overall, the developed framework provides a reliable and computationally efficient solution for modern traffic monitoring applications.

## VI. References

1. Abbas, A. F., Sheikh, U. U., Al-Dhief, F. T., & Mohd, M. N. H. (2021). A comprehensive review of vehicle detection using computer vision. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, 19(3), 838-850.
2. Chen, X., Kong, X., Xu, M., Sandrasegaran, K., & Zheng, J. (2019). Road vehicle detection and classification using magnetic field measurement. *IEEE Access*, 7, 52622-52633.
3. Choukekar, G. R., & Bhosale, M. A. G. (2018). Density based smart traffic light control system and emergency vehicle detection based on image processing. *International Research Journal of Engineering and Technology (IRJET)*, 5(4), 2441-2446.
4. Chowdhury, P. N., Ray, T. C., & Uddin, J. (2018, February). A vehicle detection technique for traffic management using image processing. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)* (pp. 1-4). IEEE.
5. Ge, Z., Li, Y., Liang, C., Song, Y., Zhou, T., & Qin, J. (2021, July). Acenet: adaptive cross-scale network with feature maps refusion for vehicle density detection. In *2021 IEEE International Conference on Multimedia and Expo (ICME)* (pp. 1-6). IEEE.
6. Godse, S. P., More, N., Surana, A., Patil, P., & Kamble, S. (2019). Traffic density detection with vehicle identification for smart traffic monitoring. *Int. J. Res. Eng. Technol. Sci*, 8(10), 1-6.
7. Ham, S. W., Park, H. C., Kim, E. J., Kho, S. Y., & Kim, D. K. (2020). Investigating the influential factors for practical application of multi-class vehicle detection for images from unmanned aerial vehicle using deep learning models. *Transportation Research Record*, 2674(12), 553-567.
8. Husain, A. A., Maity, T., & Yadav, R. K. (2020). Vehicle detection in intelligent transport system under a hazy environment: a survey. *IET Image Processing*, 14(1), 1-10.
9. Huu-Huy, N. (2023). Vehicle-detection-based traffic density estimation at road intersections. *International Journal of Open Information Technologies*, 11(7), 39-46.
10. Kim, K. J., Kim, P. K., Chung, Y. S., & Choi, D. H. (2019). Multi-scale detector for accurate vehicle detection in traffic surveillance data. *IEEE Access*, 7, 78311-78319.
11. Kong, X., Zhang, Y., Tu, S., Xu, C., & Yang, W. (2023). Vehicle detection in high-resolution aerial images with parallel RPN and density-assigner. *Remote Sensing*, 15(6), 1659.
12. Kopsiaftis, G., & Karantzalos, K. (2015, July). Vehicle detection and traffic density monitoring from very high resolution satellite video data. In *2015 IEEE international geoscience and remote sensing symposium (IGARSS)* (pp. 1881-1884). IEEE.

13. Lahinta, F., Zainuddin, Z., & Syarif, S. (2019, June). Vehicle Detection and Counting to Identify Traffic Density in The Intersection of Road Using Image Processing. In *1st International Conference on Science and Technology, ICOST 2019, 2-3 Can, Makassar, Indonesia*.
14. Liu, Q., Li, Z., Yuan, S., Zhu, Y., & Li, X. (2021). Review on vehicle detection technology for unmanned ground vehicles. *Sensors*, *21*(4), 1354.
15. Memon, S., Bhatti, S., Thebo, L. A., Talpur, M. M. B., & Memon, M. A. (2018). A video based vehicle detection, counting and classification system. *International Journal of Image, Graphics and Signal Processing*, *10*(9), 34-41.
16. Molina-Cabello, M. A., Luque-Baena, R. M., Lopez-Rubio, E., & Thurnhofer-Hemsi, K. (2018). Vehicle type detection by ensembles of convolutional neural networks operating on super resolved images. *Integrated Computer-Aided Engineering*, *25*(4), 321-333.
17. 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.
18. Song, H., Liang, H., Li, H., Dai, Z., & Yun, X. (2019). Vision-based vehicle detection and counting system using deep learning in highway scenes. *European Transport Research Review*, *11*(1), 1-16.
19. Stuparu, D. G., Ciobanu, R. I., & Dobre, C. (2020). Vehicle detection in overhead satellite images using a one-stage object detection model. *Sensors*, *20*(22), 6485.
20. Tanaka, D., Yamada, M., Kashima, H., Kishikawa, T., Haga, T., & Sasaki, T. (2019, October). In-vehicle network intrusion detection and explanation using density ratio estimation. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)* (pp. 2238-2243). IEEE.
21. Tayara, H., Soo, K. G., & Chong, K. T. (2017). Vehicle detection and counting in high-resolution aerial images using convolutional regression neural network. *Ieee Access*, *6*, 2220-2230.
22. Wang, Z., Huang, J., Xiong, N. N., Zhou, X., Lin, X., & Ward, T. L. (2020). A robust vehicle detection scheme for intelligent traffic surveillance systems in smart cities. *IEEE Access*, *8*, 139299-139312.
23. Zhang, T., & Jin, P. J. (2022). Roadside lidar vehicle detection and tracking using range and intensity background subtraction. *Journal of advanced transportation*, *2022*(1), 2771085.
24. Hassaballah, M., Kenk, M. A., & El-Henawy, I. M. (2020). Local binary pattern-based on-road vehicle detection in urban traffic scene. *Pattern Analysis and Applications*, *23*(4), 1505-1521.
25. Wei, Y., Tian, Q., Guo, J., Huang, W., & Cao, J. (2019). Multi-vehicle detection algorithm through combining Harr and HOG features. *Mathematics and Computers in Simulation*, *155*, 130-145.