International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Vehicle Density Detection Using Hybrid SSD-Yolo-V4 Model

[1]Mrs. M. Kavitha, *[2]Dr. G. T. Prabavathi
[1]Research Scholar, Department of Computer Science (PG) Gobi Arts and Science College
& Assistant Professor, Department of Software Systems, KG College of Arts and Science Mail: kavithalakshmi.m@gmail.com
[2]Associate Professor, Department of Computer Science, Gobi Arts and Science College, Gobichettipalayam. Mail: gtpraba@gmail.com
* - Corresponding author

Abstract— Vehicle traffic congestion is a serious problem in the present day. An intriguing field of study, traffic density estimate is utilized to regulate traffic light systems for more efficient traffic management. The low data resolution and extensive vehicle identification both make this endeavor a more challenging task. This article proposes a vehicle density detection technique using a hybrid deep learning model. The selected data frames from vehicle dataset are denoised using non-adaptive threshold approach. Denoised pictures are segmented using an upgraded Prewitt algorithm and then enhanced using the CLAHE (Contrast Limited Adaptive Histogram Equalization) approach. Then the important features are selected with the use of the Convolutional neural network (CNN) algorithm. The dataset is trained with Deep Q-Learning (DQN) with Improved Spatial Pyramid neural network (I-SPPNET) architecture. Finally, vehicle density identification is done using Improved Single Shot Detector (SSD) with You Only Look Once (YOLOv4) version 4 model and the obtained results proved to be significantly better than CNN, CNN with I-SPPNET and YOVOv3.

Keywords- Traffic Vehicle Density, DQN, CNN, SSD, YOLOV4

I. INTRODUCTION

The Intelligent Transport system aims to improve safety through proper coordination of transport infrastructure and vehicles. Traffic density detection is a research area ongoing in the past two decades and is still challenging due to factors such as slow detection rate of moving vehicles; low quality resolution in captured data due to shadows, lighting changes, etc. In particular, occlusions greatly increase the difficulty in traffic density detection. The exponential growth in highway traffic density has made traffic management a priority area of study. Typically, three metrics namely volume, velocity, and density are used to assess traffic conditions [9]. Many algorithms are able to recognize characteristics like volume and velocity [25][2][11], but density, although being a critical metric, is difficult to evaluate. Utilizing Traffic Density Estimation techniques may help alleviate congestion on commuter routes and flow prediction.

Denoising an image, or removing noise from a picture that has already been tainted by noise, is a basic challenge in image processing and computer vision[8]. The input video files are converted into *n* number of frames, denoised and the denoised images are taken as input image. Adaptive thresholding is used to calculate the smaller regions in the images using threshold value. The threshold value is adjusted and selected automatically according to the image pixels and layout for converting the image pixels to grayscale or a binary image[2].

Adjusting the grayscale such that the histogram of the generated picture is flat may increase the visual contrast of a digitized image[12]. Histogram Equalization (HE) increases contrast by expanding an image's intensity range. For better contrast in both natural and medical photos, Adaptive Histogram Equalization (AHE) is a great choice [17]. AHE determines the mapping for each pixel by analyzing the grayscale distribution in the surrounding pixels. The pixel's surrounding intensity values determine the contrast enhancement mapping that is applied to it. Limiting contrast enhancement in homogenous regions is one way to deal with the noise issue brought on by AHE. The photos are divided into contextual sections and the HE is applied to each one in an enhanced form of AHE called CLAHE [27]. The application of HE to contextual areas improves picture contrast by evening out the distribution of employed grey values, revealing previously concealed image details.

Many applications have found image segmentation beneficial, ranging from scene interpretation to medical image analysis to robotic perception to video surveillance to augmented reality to picture reduction. Various segmentation strategies have been proposed and investigated in the academic literature. These include thresholding, edge, region, watershed, clustering, and watershed segmentation, among others., have been established [10][18]. Prewitt is a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: 25 July 2023 Revised: 12 September 2023 Accepted: 30 September 2023

commonly used operator for edge education by gradient transform. It uses convolution to detect the edges with one horizontal filter mask and one vertical filter mask [19]. The similar idea underlies the operation of other gradient differential operators like Sobel's and Schar's. The maximum of the point's 8 gradient amplitude values is used by the original prewitt method. The influence of noise was minimized and edges were made smoother and more continuous thanks to the work of Yang, L., et al. [23]. Combining non-maximal suppression with an adaptive threshold value approach, Zhou, R.-G., et al.'s New upgraded Quantum Representation (NEQR) [24] is based on an upgraded Prewitt operator.

In the realm of object identification and processing, CNN is a popular network choice. Many studies [1, 13, 4, 22] have employed methods for detecting objects using CNNs. Machine learning's subfield known as "reinforcement learning" (RL) focuses on determining the best course of action to pursue in order to maximize reward. Since deep learning models have shown effective in several vision-related tasks, many studies have focused on creating deep learning-based picture segmentation methods [6, 22]. The Deep Q-Network (DQN) algorithm is an off-policy online approach of RL that doesn't need any kind of model to function. DQN agents are trained critics that use value-based reinforcement learning to predict the worth of future rewards.

The dimension of features from convolution layers may be reduced with the use of a pooling layer. Using a pooling approach called Spatial Pyramid Pooling (SPP-net), CNN can handle pictures of varying sizes without losing information or distorting the picture [26]. Any method of image classification that uses a Convolutional Neural Network (CNN) may benefit from SPP-net. Predictions and classifications benefit from optimization since it reduces error and increases precision. The model's precision is enhanced, and the possibility of mistake or loss is reduced. Mini-batch, Mini-batch Gradient Descent, and Stochastic Gradient Descent Stochastic gradient descent is only one of several optimization techniques used in many different areas. Adaptive Moment Estimation (AdaM) is a deep learning optimizer that may be used in place of stochastic gradient descent [14]. It does this faster and more accurately by modifying the learning rate for each weight in the neural network based on estimates of the first and second moments of the gradient. Predicting the presence and placement of objects in an input picture with a single pass is the goal of one-shot object detection. It does a single pass over an entire picture, which reduces computing overhead [15]. The YOLO architecture allows for fast, complete, and accurate training in real time [21][20][3].

The objective of this work is to use Non-Adaptive Threshold and Histogram Equalization using CLAHE for Denoising, detect the edges of the images, Extract features using CNN, Train and Test using hybrid approach (DQN with Improved SPPNet Architecture) and apply Adam Optimization. Then, vehicle density analysis is performed using SSD with YOLO-V4 Model. This project is broken down into the following sections: There are four parts to this study: The paper is divided into five sections: a short literature review (Section 2), the suggested model (Section 3), an evaluation of the outcomes of the implementation (Section 4), and a conclusion (Section 5).

1.1 Motivation of the paper

Addressing car traffic congestion inspired this study article. This paper presents a hybrid deep learning model for traffic density estimate, an important part of traffic management. The poor data resolution and broad vehicle identification provide challenges. The study proposes a multi-step solution. First, chosen vehicle dataset data frames are denoised. Next, improved prewitt and CLAHE segmentation are done. Important characteristics are identified using the CNN algorithm. Deep Q-Learning (DQN) with Improved Spatial Pyramid Neural Network (I-SPPNET) architecture trains the dataset. The Improved Single Shot Detector (SSD) with You Only Look Once (YOLOv4) version 4 model identifies vehicle density. The hybrid deep learning strategy outperforms classic CNN, CNN plus I-SPPNET, and YOLOv3 models in traffic congestion.

II. BACKGROUND STUDY

The low quality traffic signal images cannot find the density with high accuracy. A major limitation of Q-Learning is action spaces. With SPPNET, the network's maximum capacity is no longer a hard limit. In this study, we use an enhanced version of SPPNET (I-SPPNET) to tighten the constraints. As the edge based segmentation using prewitt has limitations, improved prewitt algorithm is used for segmentation. The object detection methods SSD and YOLO are gaining more importance nowadays. This research combines the SSD with YOLOv4 algorithm for finding the traffic vehicle density identification. In this section, papers related to the research CNN, SPPNET, SSD, YOLO are reviewed.

Agarwal. H et al. a new approach based on CNN for assessing highway traffic density was suggested. Adding max pooling layers to the model helped it perform better [2]. After 30 iterations of training using a normalized batch layer and a max pooling layer, the accuracy and loss were measured. Manchanda. C et al. proposed a HDNN that is a hybrid mix of the CNN and SVM model for the purpose of

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

traffic forecasting using photographs. The research conducted by the authors [16] relied on databases that documented traffic accidents in different regions of India. HDNN was trained on three CNN layers and then used to make predictions in four scenarios: heavy traffic, light traffic, an accident, and a vehicle fire. The program took into account both the time of day and the likelihood of road danger.

Large numbers of feature maps have to be generated for accurate object detection using CNN. Kim et. al,[13] a spatial pyramid pooling approach for detecting vehicles at several scales was presented. The authors used the popular Yolo-V3 model and added two more levels of item prediction logic to it. An additional layer was included between the first two levels (large-object detection, medium-object prediction), and another layer was inserted between the third and fourth layers (small-object prediction, fine-detection). The accuracy of detecting significant scale differences was further improved by adding 5 more SPP networks before each prediction layer. As a result, the authors were able to address the problem of early convolution layers not providing enough object information. This allowed for a respectable run time performance of 9-10 FPS (85.29% MAP).

For Counting Vehicles from a road image, many rule based algorithms insist which requires many predefined thresholds to detect and track vehicles. Instead of using featured engineering Chung, J., & Sohn, K. et al. [7] used deep convolutional neural networks (CNN) and a supervised learning approach to tally the number of cars in a video sequence. They counted cars using a collection counting method similar to that employed by humans. The number of vehicles in the photograph was manually tallied so that it could be labeled and used in the training process. Models that used CNN to count pedestrians scored worse than the suggested CNN, which outperformed the naïve background removal model and had a lower mean absolute error (MAE). One problem with the model was that it didn't take vehicle types into account.

Chen et al. [6] made a comparative study between 2 deep learning approaches (SSD & YOLOV3) to identify their performance in the traffic environment and found that YOLOV3 is better object distance estimation method. In order to examine how things like pedestrians and cars behave, the authors introduced an SSD-modified method. Objects were detected using the modified SSD, and then the location of the pedestrians who were in motion was approximated. The researchers' primary focus was on discovering how changes in the path of a moving vehicle or pedestrian may trigger an emergency 911 call. Prabavathi et al. reviewed the various algorithm in RL & DL for traffic signal control and intelligent transportation [18].

A. Problem definition

The problem addressed in this research paper is the challenge of vehicle traffic congestion, a prevalent issue in the modern world. Efficient traffic management is crucial, and traffic density estimation plays a key role in regulating traffic light systems. However, this task is made more difficult due to low data resolution and the need for extensive vehicle identification. Traditional methods struggle to handle these challenges effectively. This paper aims to tackle this problem by proposing a novel approach: a hybrid deep learning model for vehicle density detection. The study focuses on denoising low-resolution data frames, segmenting images, selecting important features, and employing advanced deep learning techniques to identify vehicle density accurately. The objective is to overcome the limitations posed by low data resolution and extensive vehicle identification, ultimately leading to more efficient traffic management solutions.

III. VEHICLE DENSITY DETECTION

A. Vehicle Density Detection Architecture

The proposed model aims to find the traffic vehicle density using a hybrid deep learning algorithm. The traffic video frames are selected for denoising using a non-adaptive threshold after which segmentation is performed. The best features are selected using CNN and the vehicle density has been detected using hybrid methods SSD and YOLO-V4. The dataset https://data.world/resear/vehcile containing more than 500 images is used for the training process. The proposed architecture is shown in figure 1.

First, the model is created using DQN with I-SPPNET architecture for denosing, then segmentation done using Prewitt algorithms, feature extracted with DL CNN then the testing and training will take place data from vehicle dataset using DQN Adam finally vehicle density will analyzed using SSD with YOLOv4 model.

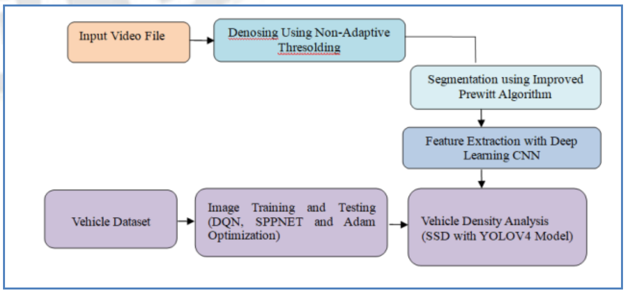

Figure 1: Vehicle Density Detection Architecture

B. Denoising Using Non-Adaptive Threshold

The adaptive threshold is calculated for very tiny regions and the threshold values have different regions. But the non-adaptive threshold has very large region space to detect the noises.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

In order to separate foreground elements from the remainder of a grayscale input picture (img), the Non-Adaptive Thresholding Algorithm with Denoising employs a two-stage approach. The threshold value (tr), a positive integer, and the maximum number of incorrect outcomes (n) are the three required inputs for the procedure.

Algorithm 1: Non-Adaptive Thresholding Algorithm with Denoising

Input:

```
img: Grayscale input image
tr: Threshold value (a positive integer)
n: Maximum number of erroneous outcomes allowed
 # Step 1: Non-Adaptive Thresholding
        binary_img = NonAdaptiveThresholding(img, tr)
# Step 2: Denoising (Correcting Erroneous Outcomes)
        erroneous_outcomes = CountErroneousOutcomes(binary_img, n)
      if erroneous_outcomes > n:
        denoised_binary_img = CorrectErroneousOutcomes(binary_img)
    else:
       denoised_binary_img = binary_img
        return denoised_binary_img

# Non-Adaptive Thresholding Function
Function NonAdaptiveThresholding(img, tr):
   height, width = img.dimensions()
   binary_img = new Image(height, width)  # Initialize binary output image with zeros
    if img(x, y) > tr:
        binary_img(x, y) = 1   # Set foreground value (1) for pixel above threshold
   else:
       binary_img(x, y) = 0   # Set background value (0) for pixel below or equal to threshold
       return binary_img

# Function to Count Erroneous Outcomes (Misclassifications)
Function CountErroneousOutcomes(binary_img, n):
   erroneous_outcomes = 0
   for each pixel in binary_img:
  if pixel == 1:   # Check for foreground pixels (1)
      # Check 8-neighbors of the pixel
      count_background = countNumberOfBackgroundNeighbors(binary_img, pixel)
  if count_background > n:
      erroneous_outcomes += 1
      return erroneous_outcomes
```

Output:

```
binary_img: Binary denoised image
Function NonAdaptiveThresholdingWithDenoising(img, tr, n):
```

C. Histogram Equalization using CLAHE

The poor qualities of image issues caused by AHE are reduced in CLAHE by restricting contrast augmentation in homogenous zones K. Honda et al. (2020). The CLAHE technique, which is used to increase an image's contrast by modifying the intensity values inside it, operates in tiny sections (called tiles) by employing bilinear interpolation to eliminate the region's boundaries, smoothing the small neighboring regions. CLAHE greatly decreases picture noise and avoids oversaturation of brightness when compared to standard histogram equalization. In this case, the exponential distribution will be employed, with the grayscale level distribution scattered more randomly across the pixels of the histogram.

D. Segmentation Using Improved Prewitt Algorithm

In the Prewitt edge detection operator, there are two possible values for each directional gradient component. Unfortunately, the gradient magnitude is commonly used as the pixel value of the edge image in conventional Prewitt edge detection methods, which leads to poor noise resistance and inaccurate edge identification D. R. D. Varma and R. Priyanka (2022). Therefore, choosing a suitable threshold for the gradient magnitude is of paramount importance. The edge will be broken and the associated data will be lost if the threshold is too high. If the threshold is too low, however, the edge will be excessively thick and false edges will emerge.

In the improved Prewitt approach, the point's gradient amplitude is determined by taking the largest of the eight possible values. The research averages these numbers to determine the size of the gradient at each position (because the absolute values of the gradient amplitudes for the two opposing directions are opposite numbers, the median value is the geometric mean of the four values). This method of using gradient values has the potential to reduce the impact of noise and provide a continuous, smoother edge. In this approach, the pixel's gradient threshold is calculated as k times the average of the grey values of its surrounding eight pixels. One alternative is to use a threshold on the magnitude of the gradient that is k times the *Otsu* automated threshold. Otsu is an automatic thresholding technique used to find an optimal threshold value for image segmentation.

For an image of N pixels in total and a grayscale of $[0,L-1]$, sort these pixels into two groups based on the grayscale values they contain. Both C_0 and C_1 stand for pixels with grayscale values between $[0, T-1]$ and $[T, L-1]$.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

There is a *100%* chance of C_0, and a *50%* chance of C_1, hence the probabilities for those outcomes are w_0 and w_1, respectively. The average grey value of all the pixels in the picture is u, whereas the average value of C_0 and C_1 are $u0$ and u_1 respectively. Assuming that 2 is the variance of the class, the formula for calculating it is:

$$\sigma^2 = w_0(u_o - u)^2 + w_1(u_1 - u)^2$$

$$\sigma^2 = w_0(u_o - u)^2 + w_1(u_1 - u)^2 \text{ ------- (1)}$$

$$= w_0 w_1 (u_0 - u_1)^2 = w_0 w_1 (u_0 - u_1)^2 \text{ ------- (2)}$$

When variance at its greatest, the optimum threshold is T. The aforementioned is the OTSU threshold concept in action. In order to produce cleaner edges, we double the OTSU threshold T by a constant k rather than using T alone as the gradient magnitude threshold. The optimum threshold, T_0, is represented by the equation:

$$T_o = T \times k T_o = T \times k \text{ ------ (3)}$$

The impact is maximized between the values of 0.25 and 0.45, as shown by the experimental data.

E. Feature extraction Convolutional Neural Network

After the image is segmented using an improved prewitt algorithm the features are extracted using CNN D. Pallavi and T. P. Anithaashri (2022). When it comes to processing data, CNNs are a kind of neural network that may mimic the processing capabilities of traditional networks. Weight sharing and local connection may be used to simplify the network. Images may be turned into multi-dimensional vector images before being fed into a neural network, simplifying data reconstruction and classification. This eliminates the need for time-consuming feature extraction.

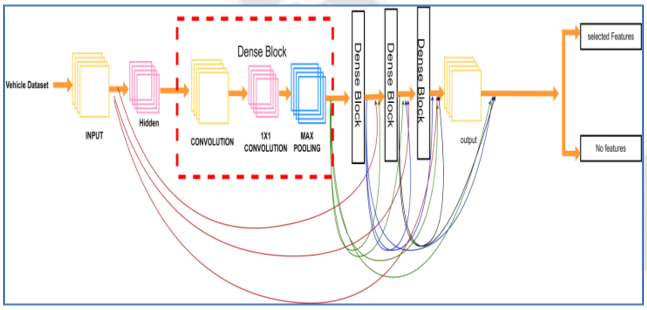

Figure 2: Feature extraction using CNN algorithm

CNN uses a pooling layer in between two convolution layers to minimize the number of parameters in the neural network and the size of the feature map. To improve the network's expressiveness and handle nonlinear output, the activation function may be utilised at the convolution layer. This is due to the fact that the neural network's connection layer as a whole performs the role of a classifier by transforming the space of the sample tags into the space of the distributed feature representation. An expression for this procedure is:

$$x_j^l = \sum_{i=1}^{n} x_i^{l-1} * w_{ij}^l + b_j^i x_j^l = \sum_{i=1}^{n} x_i^{l-1} * w_{ij}^l + b_j^i$$

----- (4)

$$y_i = f(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{n} e^{x_j}} y_i = f(x_i) = \frac{e^{x_i}}{\sum_{j=0}^{n} e^{x_j}} \text{ ------ (5)}$$

The target marker, x_i, in (5) is the actual marker in the neural network's input data, whereas y_i is the prediction marker. When training a convolutional neural network (CNN), the convolution layer is normalized by a BN (batch normalization) layer.

$$L = -\sum_{i-0}^{n} y_i \log(y_i) L = -\sum_{i-0}^{n} y_i \log(y_i) \text{ -------}$$

(6)

F. Training using hybrid Method

a. DQN

Image dataset has been trained with hybrid neural network DQN and improved SPPNet architecture. When compared to the Q-Learning algorithm used in conventional learning, the DQN algorithm performs better S. N. Aslan et al. (2020). Combining the strengths of Q-Learning and neural networks, it can learn from past events without first having to remember its surroundings. However, a model-free reinforcement learning approach used to solve the value function directly.

DQN algorithm development requires knowledge of the bonus value. For instance, let's consider the problem of traffic density prediction. In the traffic density prediction problem, a reward and punishment mechanism is used to assess the performance of a traffic management system. If the traffic density is low during peak hours, the system receives a +10 bonus for efficiently managing the traffic flow. Conversely, if the traffic congestion becomes severe, the system incurs a -10 penalty as it fails to effectively control the traffic. To achieve the optimal traffic flow in a smart city, the DQN algorithm is implemented in the traffic management system's virtual model. It learns and remembers the real-time traffic patterns and congestion levels, and then dynamically adjusts the traffic light timings and control strategies based on the real-time traffic conditions. By using the DQN algorithm for traffic density prediction and control, the traffic management system can continuously adapt to changing traffic patterns, reduce congestion, and improve overall traffic flow efficiency in the city.

b. Training using Improved Spatial Pyramid Pooling

In SPPNET it is difficult to handle large scale values and there is a lack of spatial information in deep layers. So, in this work, ISPPNET is used to fix the problem Z. Ke et al. (2020). When large images are reduced to a specific size for transmitting over a network it will result in

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

visual distortion and important target information may be lost reducing the accuracy of the model.

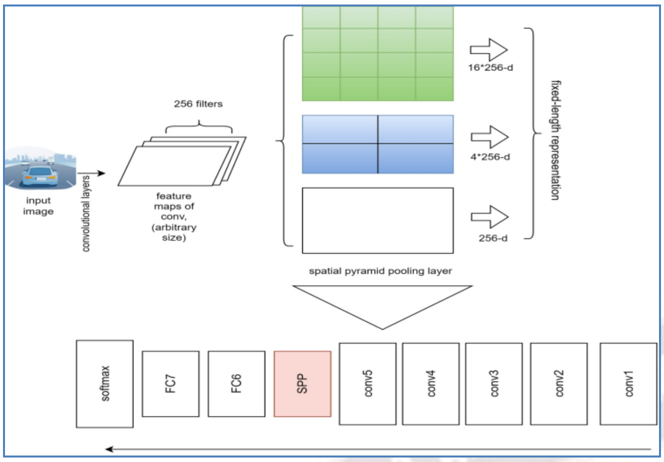

Figure 3: Improved Spatial Pyramid Pooling

Each of the green, blue and white panes in the picture has been allocated a unique feature map with sizes of 16, 4, and 1 dimension and 256 channels. The final product will be of a certain length. ISPPNET can address the problem when feature maps of a fixed size are rejected by the fully connected layer. The I-SPPNET can always provide the same-sized output regardless of the size of the input.

G. Optimization using Adam Algorithm

Adam's remarkable efficiency has led to its widespread adoption for usage in the training of deep learning models like CNN, DNN, and RNN. AdaGrad and RMSProp are combined in Adam. To be more specific, Adam use the BP method to calculate the objective function's gradients g_t with respect to the parameters, and then uses the moving average of gradients i.e,

$$m_t \leftarrow \beta_1 . m_{t-1} + (1 - \beta_1). gt$$

$$m_t \leftarrow \beta_1 . m_{t-1} + (1 - \beta_1). gt \quad \text{------ (7)}$$

$$v_t \leftarrow \beta_2 . v_{t-1} + (1 - \beta_2). gt^2$$

$$v_t \leftarrow \beta_2 . v_{t-1} + (1 - \beta_2). gt^2 \quad \text{------- (8)}$$

Where 1 and 2 are exponential decay rates for the instantaneous estimations. To account for the zero biases, the bias correction is then applied to the moments.

$$m_t \leftarrow \frac{m_t}{1-\beta_2^t} m_t \leftarrow \frac{m_t}{1-\beta_2^t} \quad \text{------- (9)}$$

$$v_t \leftarrow \frac{v_t}{1-\beta_2^t} v_t \leftarrow \frac{v_t}{1-\beta_2^t} \quad \text{------ (10)}$$

Adam uses both moments to update the parameters wt, i.e.,

$$w_t \leftarrow w_{t-1} - \alpha . \frac{m_t}{\sqrt{v_t + \epsilon}} w_t \leftarrow w_{t-1} - \alpha . \frac{m_t}{\sqrt{v_t + \epsilon}} \quad \text{--------}$$

(11)

Where the initial learning is rate and is a dividing-by-zero-avoidant smoothing term. For most training instances, the effective step is restricted by the step size hyper parameter, which stops the parameter update when such bias correction is used.

H. Traffic Vehicle Density analysis using Single Shot Detector with Yolov4

Data retrieval from the SSD model is a time-consuming and difficult task. The average number of automobiles on a given length of road is expressed as vehicles per mile or per kilometer. The rate at which automobiles pass one another on a road is referred to as flow. The primary detector model in YOLOv4 is the CSPDarknet53 neural network, consisting of 53 convolutional layers with 27.6M parameters and a receptive field of 3x3, 725x725. The addition of the I-SPP block to CSPDarknet53 effectively expands the receptive field and captures contextual features efficiently. It utilizes a novel data augmentation technique, including mosaic and adversarial training, Select the appropriate hyper parameters through existing algorithm and Improve current approaches for efficient training and detection.

Algorithm 2: Traffic Density Detection using SSD with YOLOv4

Input: Video stream or sequence of images containing traffic scenes.

Output: Estimated traffic density and flow rate.

1. **Load Pre-Trained Model:** Load the pre-trained YOLOv4 model with the CSPDarknet53 backbone and SSD architecture.
2. **Initialize Parameters:** Set the parameters for the SSD model, including confidence threshold for detection, non-maximum suppression threshold, and anchor box sizes.
3. **Capture Video Stream or Read Image Sequence:** Capture the video stream from a camera or read a sequence of images containing traffic scenes.
4. **Perform Object Detection:** Use the pre-trained SSD model to perform object detection on the preprocessed images. The model will detect vehicles in each frame of the video or image sequence.
5. **Filter Vehicle Detections:** Filter the detected objects to retain only the ones classified as vehicles. The class labels provided by the pre-trained model are used to identify vehicle detections.
6. **Calculate Traffic Density:** Based on the number of vehicle detections in each frame, calculate the traffic density (e.g., vehicles per mile or per kilometer).

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

7. **Estimate Traffic Flow Rate:** Analyze the vehicle detections over time to estimate the traffic flow rate—the rate at which vehicles pass each other on the road.
8. **Display Results:** Visualize the results by overlaying bounding boxes around the detected vehicles on the original frames or images. Display the estimated traffic density and flow rate.
9. **Repeat for Next Frame:** Repeat steps 4 to 8 for each subsequent frame in the video stream or image sequence to perform real-time or sequential traffic density detection.
10. **Post-Processing:** Optionally, perform post-processing steps such as smoothing or filtering the traffic density estimates to improve accuracy and reduce noise.

Display Final Results: Display the final traffic density and flow rate estimates along with any visualizations or plots to provide insights into the traffic patterns and congestion levels.

Object detection in YOLOv4 has demonstrated encouraging results, even though it has only been tested on publicly available datasets. In order to identify vehicle dataset, YOLOv4 needs certain improvements to the data sets it uses. The new method is built on the hybrid algorithm and can be adapted to fit a broad range of traffic circle tasks.

IV. RESULTS AND DISCUSSION

Classifications of vehicles were among those examined in the results. One versus many was used to divide the categorization data into two separate classification problems. Table 1 displays the results of training and testing the proposed model on the chosen vehicle dataset.

Epoch	Training Loss	Validation Loss	Training Accuracy	Testing Accuracy
1	00.1589	00.0594	00.9531	00.9800
2	00.0536	00.0491	00.9841	00.9837
3	00.0357	00.0442	00.9891	00.9853
4	00.0229	00.0459	00.9929	00.9860
5	00.0166	00.0403	00.9950	00.9866
6	00.0117	00.0466	00.9858	00.9963
7	00.0090	00.0573	00.9843	00.9970
8	00.0072	00.0526	00.9868	00.9978
9	00.0050	00.0533	00.9849	00.9984
10	00.0054	00.0455	00.9885	00.9983

Table 1: Training and testing accuracy and loss

Figure 4 shows the input image and denoising results are shown in Figure 5. It represents the non-adaptive thresholding and adaptive Gaussian thresholding with value as 127. Figure 6 represents the Grayscale Histogram Equalization using CLAHE. Image segmentation using Canny, Sobel, Prewitt and Improved Prewitt is shown in Figure 7.

Figure 4: Input image

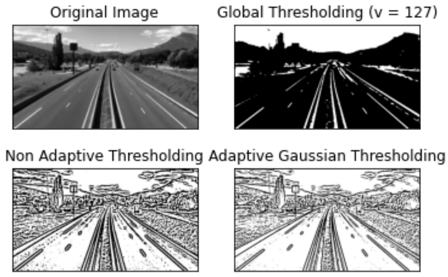

Figure 5: denoising image

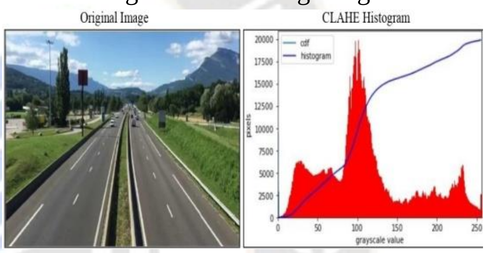

Figure 6: Grayscale HE using CLAHE

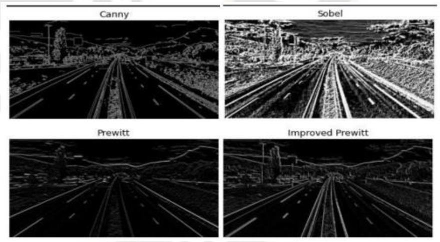

Figure 7: Image segmentation

The accuracy of the training and the tests is shown in Figure 8. Losses incurred during training and validation, RMSE value, Percentage of true positives and false positives for the DQN and Hybrid model is shown in Figure 9-11.

Figure: 8

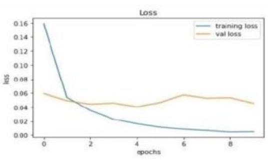

Figure: 9

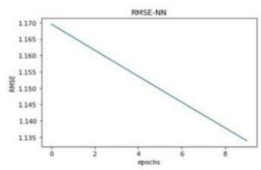

Figure : 10

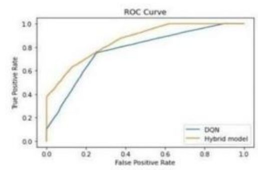

Figure : 11

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

After applying hybrid SSD with YOLOV4 model the vehicle densities are detected which is shown in Figure 12. Figure 13 shows the comparison of various models. Hybrid SSD with YOLOV4 model shows significant improvement in vehicle detection.

Figure 12: Traffic vehicle density detection

Table 2: Overall Performance metrics comparison

	Algorithm	Accuracy	Precision	Recall	F-measure
Existing authors	Abdul Qayyum, A. et al.	83.61	82.34	86.32	84.91
	Formosa, N. et al.	94.00	95.87	95.34	96.21
Existing methods	CNN	94.35	92.67	94.67	91.34
	SPPNet	93.24	93.62	94.25	93.71
	Yolo V4	96.32	95.32	95.31	95.17
	ISSD	97.35	96.31	95.89	95.21
Proposed methods	Yolo v4 with ISSD	99.31	99.24	99.14	99.37

Table 2 shows realm of object detection algorithms, several existing methods and authors have been evaluated based on their accuracy, precision, recall, and F-measure. Abdul Qayyum, A. et al. proposed a method with an accuracy of 83.61%, precision of 82.34%, recall of 86.32%, and F-measure of 84.91%. Formosa, N. et al. achieved higher results with an accuracy of 94.00%, precision of 95.87%, recall of 95.34%, and F-measure of 96.21%. Among existing methods, CNN demonstrated an accuracy of 94.35%, precision of 92.67%, recall of 94.67%, and F-measure of 91.34%, while SPPNet yielded 93.24% accuracy, 93.62% precision, 94.25% recall, and 93.71% F-measure. Yolo V4 outperformed others with an accuracy of 96.32%, precision of 95.32%, recall of 95.31%, and F-measure of 95.17%. ISSD achieved even higher results with an accuracy of 97.35%, precision of 96.31%, recall of 95.89%, and F-measure of 95.21%. In contrast, the proposed method, Yolo v4 with ISSD, surpassed all previous approaches, attaining an impressive accuracy of 99.31%, precision of 99.24%, recall of 99.14%, and F-measure of 99.37%. This suggests that the proposed combination of Yolo v4 with ISSD significantly enhances the object detection performance, achieving remarkable accuracy and precision while maintaining high recall and F-measure values, making it a promising advancement in the field of object detection algorithms.

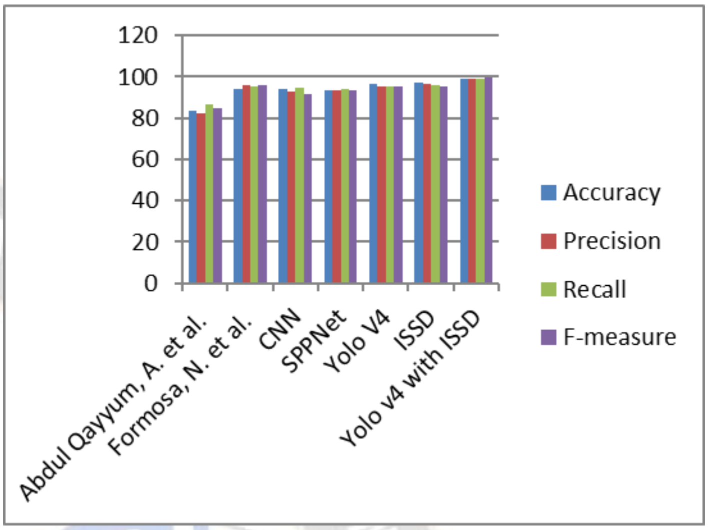

Figure 13: Overall performance merics comparison chart

The figure 13 shows overall performance metrics comparison chart the x axis shows methods and the y axis shows values.

V. Conclusion

This research work aims to detect traffic vehicle densities using hybrid deep neural network. This main idea of this research is to create a high-quality vehicle density by extracting characteristics from low-resolution feature maps and combining them at the network. The proposed network in the system is built to automatically learn the adaptive attention weights required to exploit and compose the spatial information contained in feature maps of various sizes. Sub pixel convolutes samples feature maps to recover the spatial information encoded in them in order to generate a high-quality density map. The proposed model has been executed with a hybrid neural network ISSD with YOLOV4 framework. The obtained results have been compared with CNN, CNN with SPPNET, YOLOV3, YOLOV4 and SSD. In this research hybrid neural network on real time video datasets are tested. The experimental results demonstrate that YOLOV4 with ISSD achieves better accuracy to detect vehicle density in various weather conditions.

REFERENCE

[1] Abdul Qayyum, A. B., Arefeen, A., & Shahnaz, C. (2019), "Convolutional Neural Network (CNN) Based Speech-Emotion Recognition". 2019 IEEE International Conference on Signal Processing, Information, Communication & Systems (SPICSCON).doi:10.1109/spicscon48833.2019.9065172

[2] Agarwal, H, Rana, V, Vats and Saraswat, M. (2020), "Efficient Traffic Density Estimation Using Convolutional Neural Network," 2020 6th International Conference on Signal Processing and Communication (ICSC), pp. 96-100, doi: 10.1109/ICSC48311.2020.9182718.

[3] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao (2020), "YOLOv4-object: an Efficient Model and Method for Object Discovery".

[4] Chauhan, N. K., & Singh, K. (2018), "A Review on Conventional Machine Learning vs Deep Learning". 2018 International Conference on Computing, Power and Communication Technologies (GUCON). doi:10.1109/gucon.2018.8675097

[5] Chen, H. –B, Fu, H.-L. (2009) "Non Adaptive algorithms for threshold group testing, Discrete Appl. Math". 157 (7) (2009) 1581–1585, https://doi.org/10.1016/j.dam. 2008.06.003

[6] Chen, Z., Khemmar, R., Decoux, B., Atahouet, A., & Ertaud, J.-Y. (2019), "Real Time Object Detection, Tracking, and Distance and Motion Estimation based on Deep Learning: Application to Smart Mobility". 2019 Eighth International Conference on Emerging Security Technologies (EST). doi:10.1109/est.2019.8806222

[7] Chung, J., & Sohn, K. (2018). "Image-Based Learning to Measure Traffic Density Using a Deep Convolutional Neural Network". IEEE Transactions on Intelligent Transportation Systems, 19(5), 1670–1675. doi:10.1109/tits.2017.2732029 doi: 10.1109/CVPR46437.2021.01283

[8] Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). "Brief review of image denoising techniques". Visual Computing for Industry, Biomedicine, and Art, 2(1). doi:10.1186/s42492-019-0016-7

[9] Formosa, N., Quddus, M., Ison, S., Abdel-Aty, M., & Yuan, J. (2020), "Predicting real-time traffic conflicts using deep learning. Accident Analysis & Prevention", 136, 105429. doi:10.1016/j.aap.2019.105429

[10] Haralick, R. M. and Shapiro, L. G. (1985), "Image segmentation techniques," Computer Vision Graphics Image Process., vol. 29, pp. 100–132, 1985

[11] Huang, Tian, Y., Liu, Z., Jin, X., Liu, Y., Zhao, S., & Tian, D. (2020), "A Traffic Density Estimation Model Based on Crowdsourcing Privacy Protection". ACM Transactions on Intelligent Systems and Technology, 11(4), 1–18. https://doi.org/10.1145/3391707

[12] Hummel, R. (1977), "Image Enhancement by Histogram Transformation," Computer Vision, Graphics and Image Processing, Volume 6, Issue 2, pp. 184–195.

[13] 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, 1–1. doi:10.1109/access.2019.2922479

[14] Kingma, D and Ba, J. (2014), "Adam: a method for stochastic optimization," Computer Science, 2014. https://arxiv.org/abs/1412.6980

[15] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., C.-Y. Fu, C. –Y., and Berg, A. C., (2016), ''SSD: Single shot multibox detector,'' in Proc. Eur. Conf. Comput. Vis. Springer, 2016, pp. 21–37.

[16] Manchanda, C., Rathi, R., & Sharma, N. (2019), "Traffic Density Investigation & Road Accident Analysis in India using Deep Learning", 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS). doi:10.1109/icccis48478.2019.8974528

[17] Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., Zuiderveld, K. (1987), "Adaptive histogram equalization and its variations". Computer Vision, Graphics, and Image Processing, 39(3), 355–368. doi:10.1016/s0734-189x(87)80186-x

[18] Prabavathi, G. T, Kavitha, M et al (2022), "Traffic Signal Analysis using Reinforcement Learning – A Review". Indian Journal of Natural Sciences Vol.13 Issue 74 ISSN: 0976 – 0997

[19] Prewitt J. M. S., B. Lipkin and A. Rosenfeld, Eds. (1970), "Object enhancement and extraction, Picture Processing and Psychopictorics" , New York: Academic Press, 1970, pp. 75-149.

[20] Redmon, J and Farhadi, A. (2018), ''YOLOv3: An incremental improvement,'' 2018, arXiv:1804.02767. [Online]. Available: https://arxiv.org/abs/1804. 02767

[21] Redmon, J, Divvala, S , Girshick, R and Farhadi, A. (2016) ''You only look once: Unified, real-time object detection,'' in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., June. 2016, pp. 779–788.

[22] Shi, Y., Feng, H., Geng, X., Tang, X., & Wang, Y. (2019), "A Survey of Hybrid Deep Learning Methods for Traffic Flow Prediction". Proceedings of the 2019 3rd International Conference on Advances in Image Processing. doi:10.1145/3373419.3373429

[23] Yang, L., Wu, X., Zhao, D., Li, H., & Zhai, J. (2011), "An improved Prewitt algorithm for edge detection based on noised image". 2011 4th International Congress on Image and Signal Processing. doi:10.1109/cisp.2011.6100495

[24] Zhou, R.-G., Yu, H., Cheng, Y., & Li, F.-X. (2019), "Quantum image edge extraction based on improved Prewitt operator", Quantum Information Processing, 18(9). doi:10.1007/s11128-019-2376-5

[25] Zhu, J, et al., (2018), "Urban Traffic Density Estimation Based on Ultrahigh-Resolution UAV Video and Deep Neural Network," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 11, no. 12, pp. 4968-4981, Dec. 2018, doi: 10.1109/JSTARS.2018.2879368.

[26] Zhu, R., Mao, X.-J., Zhu, Q.-H., Li, N., & Yang, Y.-B. (2016), "Text detection based on convolutional neural networks with spatial pyramid pooling". 2016 IEEE International Conference on Image Processing (ICIP). doi:10.1109/icip.2016.7532514

[27] Zuiderveld, K. (1994), "Contrast Limited Adaptive Histogram Equalization". Graphics Gems, 474–485. doi:10.1016/b978-0-12-336156-1.50061-6

[28] K. Honda, K. Wei, M. Arai and H. Amano, "CLAHE implementation on a low-end FPGA board by high-level synthesis," 2020 Eighth International Symposium on Computing and Networking Workshops (CANDARW), Naha, Japan, 2020, pp. 282-285, doi: 10.1109/CANDARW51189.2020.00061.

[29] D. R. D. Varma and R. Priyanka, "Performance Analysis of Novel Iris Monitoring System Based on Canny Detection Algorithm in comparison with Prewitt Algorithm," 2022 2nd International Conference on Innovative Practices in Technology and Management (ICIPTM), Gautam Buddha Nagar, India, 2022, pp. 523-527, doi: 10.1109/ICIPTM54933.2022.9754088.

[30] D. Pallavi and T. P. Anithaashri, "Novel Predictive Analyzer for the Intrusion Detection in Student Interactive Systems using Convolutional Neural Network algorithm over Artificial Neural Network Algorithm," 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 2022, pp. 638-641, doi: 10.1109/ICAC3N56670.2022.10074027.

[31] S. N. Aslan, A. Uçar and C. Güzeliş, "Development of Deep Learning Algorithm for Humanoid Robots to Walk to the Target Using Semantic Segmentation and Deep Q Network," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 2020, pp. 1-6, doi: 10.1109/ASYU50717.2020.9259888.

[32] Z. Ke, C. Le and Y. Yao, "A multivariate grey incidence model for different scale data based on spatial pyramid pooling," in Journal of Systems Engineering and Electronics, vol. 31, no. 4, pp. 770-779, Aug. 2020, doi: 10.23919/JSEE.2020.000052.