# TRAFFIC MONITORING SYSTEM FOR SMART CITY BASED ON TRAFFIC DENSITY ESTIMATION

Shraddha Bidwe

Department of Computer Engineering, SCTR's Pune Institute of Computer Technology,
Pune, Maharashtra, India
shraddhabidwe123@gmail.com

Dr. Geetanjali Kale

Department of Computer Engineering, SCTR's Pune Institute of Computer Technology,
Pune, Maharashtra, India
gvkale@pict.edu

Ranjeet Bidwe

Symbiosis Institute of Technology, Symbiosis International (Deemed University) (SIU), Lavale, Pune 412115
ranjeetbidwe@hotmail.com

**Abstract**

**Nowadays, traffic monitoring systems are at the frontline of smart city movement, and traffic density estimation is useful to a traffic monitoring system. The system of this work estimates traffic density using a five-layered CNN with a variety of input feature maps and filter sizes. There are 64, 64, 96, 96, and 96 feature maps for each pair of convolutions and max-pooling layers, and each pair's corresponding filter sizes are 5*5, 3*3, 5*5, 3*3, and 3*3. The proposed system divides the traffic into three categories: High, Medium, and Low traffic based on images that are taken from traffic videos that are recorded by traffic surveillance cameras. To test the system, we used the WSDT (Washington State Department of Traffic Transportation) Dataset of recorded video footage from the highway CCTVs in cities Seattle and Washington. The model is evaluated using parameters such as 0.99 precision, 0.99 recall, 0.99 f1- score, and model accuracy of 99.6 %. By examining the dataset, we have trained the model in such a manner to produce better results.**

***Keywords*: Traffic monitoring; Deep learning; Convolution Neural Network; Image processing; Computer vision.**

## 1. Introduction

The world's most serious problem right now is traffic on the roads. The primary traffic entities include vehicles, Pedestrians, lengths, routes, detours, traffic laws, and so on. Several causes contribute to road congestion or heavy traffic on the road. Increased vehicle numbers as a result of rapid urbanization are the primary cause of traffic congestion. Traffic congestion makes it even harder for a vehicle to move, and it is particularly difficult for emergency services such as ambulances and fire trucks. These issues primarily occurred in cities. The traffic monitoring system is useful in reducing these types of difficulties.

In recent years, there has been a growing interest in traffic monitoring systems in emerging countries in order to reduce the problems caused by traffic congestion. The major objective of this system is to do analysis of current traffic condition by extracting data on vehicle count, speed, vehicle type, and density. Ultrasonic detectors, radar sensors, loop detectors, microwave sensors, and video cameras are all useful in traffic monitoring systems. Video cameras are being utilized to collect and interpret traffic data due to recent ongoing advances in the domain of computer vision, deep learning and image processing. Due to the obviously increasing population, managing highway traffic flow is currently a difficult task.

One of the difficult issues in the traffic management system is traffic density assessment. This offers crucial information for planning, routing, and traffic monitoring for automotive networks. Based on data about traffic volume, drivers select routes. Therefore, supplying actual traffic density information with the least amount of delay is crucial in an intelligent transportation system. It would be beneficial if drivers were advised of traffic conditions with the aid of technologies that have learned to effectively estimate traffic density, as heavy traffic and blockages are important issues, particularly in India.

Estimating traffic density will be useful for traffic monitoring and management systems, which is the major goal of this work. The study's leftover few goals are

- To increase the elasticity of the traffic monitoring system so that it can anticipate and avert heavy traffic in advance and adjust the current system to the traffic scenario.
- To anticipate traffic jams in order to redirect traffic and stop any more emergencies from being caused by it.
- To contribute to the idea of a smart city by anticipating and minimizing traffic congestion.

## 2. Related Work

A method for traffic monitoring based on YOLO (You Only Look Once) and CFNN (Convolutional Fuzzy Neural Network) was presented by [Lin and Jhang, (2022)]. The suggested system operated in three phases: detection, counting, and classification of vehicles. For vehicle detection, a modified YOLOv4-tiny version was employed, while a Kalman Filter and a Hungarian algorithm were implemented for vehicle counting and tracking. For vehicle classification is done using efficient algorithms like CFNN and Vector-CFNN were utilized, and they provided a good level of classification accuracy.

The fourth version of YOLO (YOLOv4) was used by [Zuraimi and Zaman, (2021)]. to develop a system for vehicle identification and tracking. The Deep SORT method was combined with the TensorFlow machine learning Python framework to track and count the vehicle. When the YOLOv4 algorithm was compared to the prior model, it performed more accurately.

The technique in this work combines the best classifier with the object detection algorithm YOLO to improve the accuracy [Azimjonov and Ozmen, (2021)]. In this study, nine machine learning classifiers were utilized, although CNN-based classifiers perform better in classification accuracy. Bounding box-based and Kalman filter-based trackers were both employed to track the vehicle. Overall, the combined best algorithm (YOLO+CNN based classifier +Bounding box) achieved better accuracy.

A technique of counting vehicles and estimating traffic volume was proposed by [Yang et al. (2021)]. The assessment was done using the time-spatial imaging (TSI) approach and neural network. Although this system counts vehicles with good precision, it cannot manage to change traffic lanes.

A neural network for vehicle recognition and counting was presented in this paper [Liu and Juang, (2021)]. The traffic flow evaluation made use of the evaluated count. In this case, the author employed Deep SORT for tracking and YOLOv4 for vehicle detection. The vehicle and velocity counts were both done using virtual lines.

Real-time video footage was the foundation for the presented study [Singh et al., (2021)] using vehicle detection, monitoring, and counting. The YOLOv3 technique for object detection based on CNN and the OpenCV computer vision library were used to construct the system. And based on the number of vehicles, this information was used to manage traffic lights. On the COCO dataset, the author applied pre-trained model weights.

The traffic management system's two-step strategy was suggested [Dave et al. (2021)]. The YOLOv4 algorithm was used in the first stage to identify and count the number of vehicles, and the XGBoost method was used in the second to forecast when the traffic light would turn green. In this case, an MSCOCO dataset for the detection model and a Vadodara city crossroads dataset for the prediction model was employed.

To improve vehicle counting, [Bui and Cho (2020)] created a comprehensive vehicle counting model that uses the Deep SORT and YOLO for tracking and detection of objects. Monitoring the trajectory of a vehicle proved helpful under challenging circumstances.

The real-time traffic monitoring of video streams from street security cameras is the main subject of this study [Khazukov et al., (2020)]. Real-time data were used to gather details on average vehicle speed, travel directions, and traffic flow intensity. YOLOv3 neural network architecture was enhanced by including a mask branch, increasing the accuracy of vehicle recognition and object classification. The coordinate transformation approach was utilized to calculate the speed of the vehicle. This investigation's challenging tasks were the objects' overlap and the various viewing angles.

Using a restricted multi-target tracking approach, the system described by [Li et al. (2020)] consists of three parts: a pyramid-YOLO network for vehicle detection, a restricted multi-target tracking method for vehicle tracing, and vehicle counting after vehicle tracing with the use of tracking trajectory. These findings created a parameter evaluation model that considers traffic density, speed, and volume.

Using YOLOv4 and Deep SORT, [Doan and Truong (2020)] created a vehicle recognition and counting model. Deep SORT and YOLOv4 are used for tracking and counting, respectively. The most challenging aspect of this study was analyzing findings in different environmental situations, like at night and during a severe downpour.

A method for identifying and classifying vehicles was presented by [Pham et al. (2020)] proposed approach, which would divide vehicles into three categories based on their size: heavy, medium, and light. By computing the pixels on the vehicle, the size of the vehicle was determined.

With vehicle recognition and counting, [Huy and Duc (2020)] established a model for evaluating traffic flow. The Faster RCNN model was used to detect vehicles and categorize them into motorcycles, cars, trucks, and buses in the initial stage of the proposed model's operation. Additionally, the CSRT tracker was used to track automobiles. The system performed admirably under various lighting situations.

The vehicle detecting system was proposed by [Lou et al. (2019)]. YOLOv3 was utilized to detect moving cars, and the Kalman filter method was employed to follow a detected vehicle. Both day and night tests of the system were conducted.

A technique for detecting and counting vehicles using vision was proposed [Song et al., (2019)]. For this investigation, a dataset of highway vehicles was used. The vehicle was identified using the YOLOv3 technique, and the vehicle direction and count were obtained using the vehicle trajectories approach. The ORB feature extraction algorithm determined the vehicle's position.

Using vision-based traffic monitoring, [Bhuiyan et al. (2019)] published their results. For this study, the authors gathered data from several city roadways under various circumstances and obtained varying degrees of accuracy for different circumstances. For this investigation, the vehicle detection and counting approach was employed. Two Virtual Detection Lines (VDL) were employed to count the identified vehicles, and a Haar-like, feature-based AdaBoost classifier was used for detection. The writers categorized the traffic situation based on the count.

A method for counting automobiles and categorizing their density level was presented by [Lahinta et al. (2019)]. The video data for this investigation was gathered in Manado, Sulawesi. The GMM and MO (Gaussian Mixture Model and Morphological Operations) techniques in the form of Binary Large Objects (BLOB) were used to identify the vehicles in the video. The morphological operation erased the shadow cast by the car. The number of cars was determined using the feature extraction approach, and the count was divided into densities.

[Asha and Narasimhadhan (2018)] presented an intelligent transportation system. For this investigation, handheld cameras were used to gather the video data. The three sections of this investigation were object detection, tracking, and vehicle counting. YOLO was utilized for object detection. Vehicle tracking was done using a correlation filter, and vehicle counting was accomplished using bounding boxes produced by the YOLO framework.

[Zhu et al. (2018)] presented a system for evaluating traffic density using deep learning methods. Unmanned aerial vehicle (UAV) video data was gathered over a busy intersection in a megacity. Vehicle detection was carried out using the Enhanced-SSD (Single Shot Multi Box Detector), and the proposed Deep Vehicle Counting Framework (DVCF) was employed to count various vehicle kinds. This system's only flaw was that it took much processing time.

A framework for vehicle counting was provided by [Xiang et al. (2018)]. Unmanned aerial vehicles (UAVs) were employed to acquire the aerial footage, which was then used as data for this investigation. The framework handled the two situations, including static background and shifting backdrop. The static backdrop was detected using the foreground detector, and the moving background was seen using image registration for the moving background. The multi-threading technique evaluated the state of the tracked vehicle. Instead of a moving background, the proposed approach provided more accurate input for a static background video.

| Ref. No. | Dataset | Algorithm/Method | Features /Limitations | Performance /Remarks |
|---|---|---|---|---|
| [Lin and Jhang, (2022)] | Public dataset GRAM-RTM by Beijing Institute of Technology | YOLO and CFNN | The CFNN and VCFNN model gives better accuracy for vehicle classification. | 90.45% |
| [Zuraimi and Zaman, (2021)] | Customized Image dataset (7319 images) | YOLOv4 for detection and Deep SORT for tracking and counting purposes | YOLOv4 model gives better accuracy for detection rather than the other models of YOLO | 82.08% |
| [Azimjonov and Ozmen, (2021)] | They prepared their dataset of 7216 images from recorded videos of highways. | YOLO for detection, Classification using ML and CNN-based classifiers, Kalman filter-based tracker, and Bounding box-based tracker for tracking purposes. | Combining a CNN- and bounding box tracker with YOLO achieves better accuracy. | 95.45% |
| [Yang et al., (2021)] | UA-DETRAC dataset containing 10 hours of recorded footage at 24 distinct places | Time-spatial image (TSI) method and the neural network. | This method cannot handle the situation when traffic changes lanes. | 97.96% |

| | | | | |
|---|---|---|---|---|
| | throughout Beijing and Tianjin. | | | |
| [Liu and Juang, (2021)] | Mp4 video taken from the national highway. | YOLOv4 and Deep SORT | The system does not count detected vehicles properly | 11.65% is an average error between actual and detected vehicles. |
| [Singh et al., (2021)] | Pretrained COCO dataset | YOLO, CNN, OpenCV | It required a high specification system for running the model. | When the traffic congestion was less, the system gave better accuracy, and the detection was run at 0.50,0.19,0.35 frames per second. |
| [Dave et al., (2021)] | MS COCO dataset and Vadodara city crossroads dataset | YOLOv4 and XGBoost technique | This study predicted the green light time for traffic management. | Considering normal traffic volume, the recommended method minimizes waiting time by 32.3 %. |
| [Bui and Cho, (2020)] | CVPR AI City Challenge 2020 dataset. | For detection and tracking purposes, use YOLO and Deep SORT, respectively. | This method had an issue detecting big vehicles like tractor-trailers and 18-wheeler vehicles. | 85% |
| [Khazukov et al., (2020)] | Image dataset with 6000 images. | YOLOv3 for detection and coordinate transformation method for determining the vehicle speed. | Different viewing angles and overlapping objects are the challenges of this study | 92% |
| [Li et al., (2020)] | The UA-DETRAC public urban traffic data set (With frame rate of 25 and a resolution of 960 x 540) | Pyramid-YOLO | This system is not very useful for complex traffic scenes. | 96.10% |
| [Doan and Truong, (2020)] | COCO dataset (47,202 images), Open Image (147,699 images), and their traffic data collected from a video camera (9,253 images) | YOLOv4, Deep Sort | Evaluating results in complex conditions like night-time and heavy rain. | 79% |
| [Pham et al., (2020)] | Traffic data from a single pole-mounted camera in Ho Chi Minh City, Vietnam. | Proposed a strong vehicle detection, classification, and counting approach. | Not applicable for different situations like nighttime and rush hours. | 95.3% |
| [Huy and Duc, (2020)] | Collected data from surveillance cameras (4000 images) | R-CNN and CSRT model | It gives good accuracy in different light conditions | 86% |
| [Lou et al., (2019)] | Used collected traffic flow videos as experimental data | YOLO, Kalman filter algorithm | In this study on the vehicle, the detection was done that information not used for traffic management system | 92.11% |
| [Song et al., (2019)] | Stanford Car Dataset, Tsinghua-Tencent Traffic-Sign Dataset (100,000 images) | YOLOv3 and ORB algorithms were used for the trajectories of the vehicle | The method of this paper performs well for Counting than detection. | 92.3% for detection and 93.2% for counting the vehicles. |
| [Bhuiyan et al., (2019)] | Authors collected data from different locations | Adaboost classifier and virtual detection lines | The performance of the system was decreased in high traffic conditions. | 90% to 94% |
| [Lahinta et al., (2019)] | Data collected from the Teling intersection of Manado City, Indonesia | Gaussian Mixture Model (GMM), Morphological Operation (MO) | By using a morphological process, the shadow of the vehicle is removed. This helps in counting the vehicle | 90.9% |
| [Asha and Narasimhadhan, (2018)] | Prepared the video dataset using a mobile device with a 13MP camera. | Correlation filter with YOLO. | High detection processing times and only a single lane was considered for the vehicle counting. | 97% |
| [Zhu et al., (2018)] | UAV City Traffic Video Dataset | Enhanced-SSD | The processing time was relatively high. | 93.7% |
| [Xiang et al., (2018)] | Aerial videos collected from UAV | Proposed a framework based on foreground detector and image registration for static and moving background. | It gives less accuracy for moving background input. | 90% and 85% for Static and Moving background, respectively. |

Table 1. Literature Survey Table.

## 3. General Framework

Fig. 1. depicts the basic flow of the traffic monitoring system. In a traffic monitoring system, data is collected from security cameras in the form of videos or images. The following actions are taken after the data has been obtained.

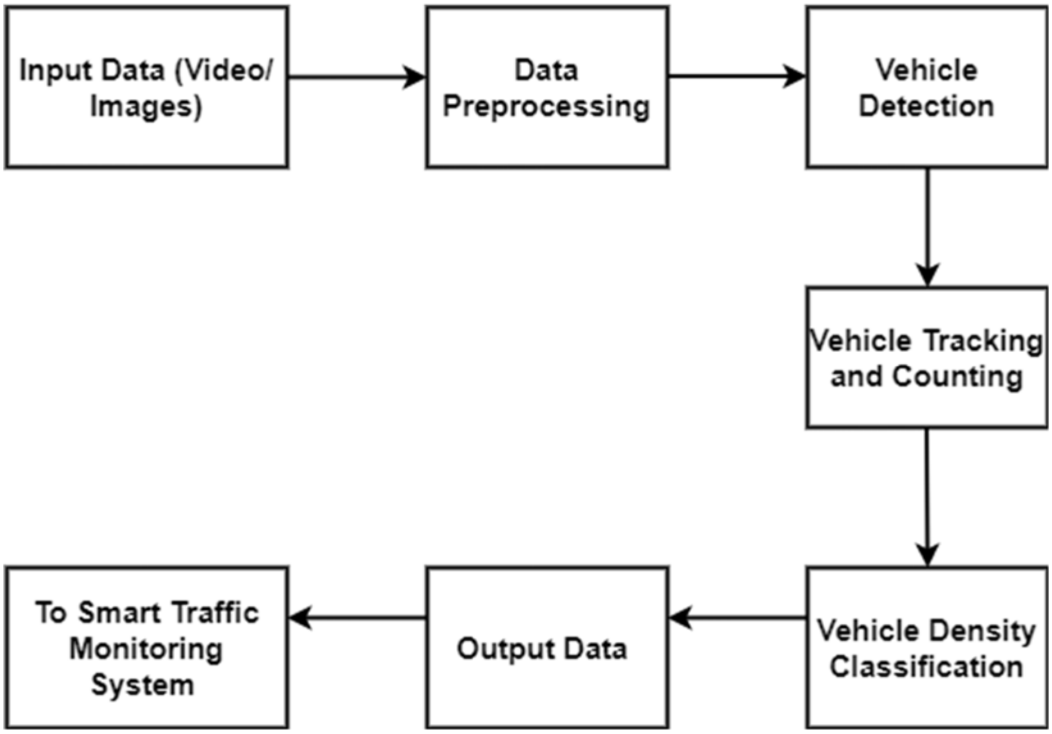


Fig. 1. General Flow of the System.

### 3.1. *Data Preprocessing*

At the Data Pre-processing stage, it converts raw data correctly so that computers and machine learning can comprehend and assess. Data cleansing and data integration are instances of data pre-processing techniques. The process of changing data to make sure that it is free of unnecessary and inaccurate information is known as data cleaning. At this phase, the noise is brought down from the video. The procedure of merging information from different sources into a single, centralized view for better data management is known as data integration. It is the data extraction, transformation, and loading (ETL) process. The frames from the videos are retrieved and transformed from RGB to Grayscale.

### 3.2. *Vehicle Detection*

The most significant phase of this model is vehicle detection, which is done using different algorithms or approaches. YOLO [Lin and Jhang, (2022)],[Zuraimi and Zaman, (2021)],[Azimjonov and Ozmen, (2021)],[Liu and Juang, (2021)],[Singh et al., (2021)],[Dave et al., (2021)],[Bui and Cho, (2020)],[Khazukov et al., (2020)],[Doan and Truong, (2020)],[Lou et al., (2019)],[Song et al., (2019)],[Asha and Narasimhadhan, (2018)], Pyramid-YOLO network [Li et al., (2020)], AdaBoost classifier [Bhuiyan et al., (2019)] and Enhanced SSD model [Zhu et al., (2018)], were used for vehicle detection. The researchers employed different variants of the YOLO method in most of the articles, which offered a high percentage of accuracy for vehicle detection.

### 3.3. *Vehicle Tracking and Counting*

Vehicle tracking is the next most important stage in a traffic monitoring system. The vehicle is tracked after it is detected to count. Regarding vehicle tracking, earlier studies used the ORB algorithm [Song et al., (2019)], CSRT tracking model [Huy and Duc, (2020)], Deep SORT [Zuraimi and Zaman, (2021)],[Liu and Juang, (2021)],[Bui and Cho, (2020)],[Doan and Truong, (2020)], Hungarian algorithm [Lin and Jhang, (2022)], Kalman filter algorithm [Lou et al., (2019)]. For vehicle counting, the trajectory tracking method, TSI method [Yang et al., (2021)], and Virtual Detection Line [Bhuiyan et al., (2019)] were used. Vehicle tracking helps to count the vehicle.

### 3.4. *Vehicle Density Classification*

During the classification process, vehicles are classified into numerous groups. Previously, the categorization was carried out using the Faster RCNN [Huy and Duc, (2020)] model, Convolutional Fuzzy Neural Network (CFNN) [Lin and Jhang, (2022)], and various machine learning-based algorithms [Azimjonov and Ozmens, (2021)], etc. At this stage, the identified cars were classified into three categories: high, medium, and low traffic density.

## 4. Proposed Methodology for Density Estimation

In the model of CNN architecture, the traffic density images are recognized and classified in a way that produces superior outcomes. For various convolutional layers, a variable kernel size, such as 5*5 or 3*3, has been used. The presented method may classify the current traffic block after using footage taken by installed cameras to monitor traffic. The goal is to use this as a model to give expert systems data on traffic density from various locations and make other significant decisions regarding traffic control. To achieve more accuracy, sophisticated Deep Neural Networks are implemented using NVIDIA GPU (Graphics Processing Unit). Fig. 2. depicts the overall framework design for traffic density classification.

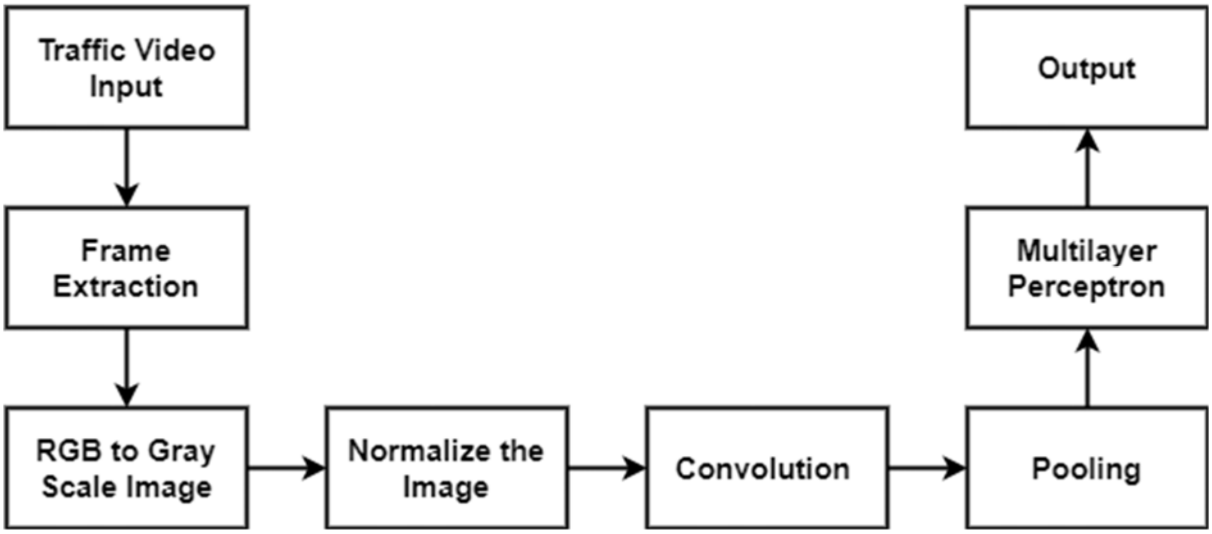


Fig. 2. General Flow Architecture of Density Estimation.

A fully connected multi-layered perceptron is joined to a CNN's convolutional layer, which is subsequently followed by sub-sampling or pooling layers. It takes advantage of the image's 2D structure. CNN's primary goal is to extract stratified characteristics from images automatically. Convolution is performed using randomly initialized filters to lower the image size without losing any features, achieved by the number of feature maps.
Generally, pooling is performed after convolution. With max pooling, the highest possible pixel value is identified and rendered as a single pixel value. The characteristic leads to the most significant number of attributes with equivalent values. The multi-layered perceptron (MLP), which is fully connected, receives the parts that were ultimately retrieved to classify the test input.
The CNN classifier is employed in the presented method to determine traffic volume. We adjusted the various tuning parameters to give testing results with greater precision than the previous ones. The Traffic images are resized to transmit to various CNN layers after collecting the data. To extraction of features, every input is sent to the convolutional layer and then to max-pooling layer. A CNN is a very famous and widely used Deep Learning technique that takes image as an input, and by using learnable biases and weights, prioritize different characteristics and objects and differentiate between them. CNN gives better accuracy for image classification.

### 4.1. *CNN Model*

Fundamental mathematics at the back of all the above-discussed processes is covered in this part.

#### *4.1.1. Convolution Operation*

The outcome of the preceding layer, $\vec{\alpha}^{(\overrightarrow{l-1})}$ with height and width $(\vec{h}^{(\overrightarrow{l-1})}, \vec{w}^{(\overrightarrow{l-1})})$ provides as the input for the convolution layer, $\vec{\alpha}^{(0)}$ acting as the input image. Afterward, employing kernel/filter F, the convolution operator $\odot$ is applied to this input.

$$(\vec{\alpha}^{(\overrightarrow{l-1})} \odot F)_{\vec{p},\vec{q}} = \sum_{1}^{\vec{h}^{(\overrightarrow{l-1})}} \sum_{1}^{\vec{w}^{(\overrightarrow{l-1})}} F_{i,j} * \vec{\alpha}^{(\overrightarrow{l-1})}{}_{(\vec{p}+i-1,\vec{q}+j-1)} \quad (1)$$

*4.1.2. Activation Function*

After the convolution operation, the activation function ($\psi$) is utilized to introduce nonlinearity into the transformation. As a result, the layer's outcome will be:

$$\psi\left((\vec{\alpha}^{(\overrightarrow{l-1})} \odot F)_{\vec{p},\vec{q}}\right) = \psi(\textstyle\sum_1^{\vec{h}^{(\overrightarrow{l-1})}} \sum_1^{\vec{w}^{(\overrightarrow{l-1})}} F_{i,j} * \vec{\alpha}^{(\overrightarrow{l-1})}{}_{(\vec{p}+i-1,\vec{q}+j-1)}) \qquad (2)$$

Sigmoid, tanh, relu, etc. are the most widely used options for activation functions ($\psi$).The Relu function is employed as an activation function whereas other functions have issues with vanishing gradients. In this instance, the result will be:

$$Max(0, \psi(\textstyle\sum_1^{\vec{h}^{(\overrightarrow{l-1})}} \sum_1^{\vec{w}^{(\overrightarrow{l-1})}} F_{i,j} * \vec{\alpha}^{(\overrightarrow{l-1})}{}_{(\vec{p}+i-1,\vec{q}+j-1)})) \qquad (3)$$

*4.1.3. Pooling Layer*

It is not essential where the characteristic is located in the image exactly. But occasionally, this spatial data is also documented. Downsampling is one technique to get around this. The pooling layer aids in the achievement of this. In this method, the max-pooling strategy is employed. A matrix A of size (n1, n2) and a pooling filter of size f will be used by the pool operator to determine the most significant value in a $fxf$ block.

*4.1.4. Fully Connected Layer*

Due to the single-channel available, the outcome of the convolution layer or pooling layer will be a 2-D matrix. We need to flatten the 2-D (3-D in general) array and construct the vector to feed this to a fully connected layer. A fully connected layer is then given this vector.
Forward propagation: In the ith layer, the output vector for the forward pass is determined as:

$$\vec{X}^{(i)} = \psi^{(i)}(\vec{Y}^{(i)}) = \psi^{(i)}(\vec{W}^{(i)^T}\vec{X}^{(i-1)} + \vec{b}^{(i)}) \qquad (4)$$

$the\ the\ \vec{W}^{(i)}$ is the weight vthe ector of the ith layer and bias term for ith layer is $\vec{b}^{(i)}$.
$\psi^{(i)}$ is an activation functions for the ith layer, it is clear from this phase that there can be several activation functions for various layers. Relu is employed in all but one layer of the system, while Sigmoid is employed in the top layer.
Backpropagation: During this stage, a prediction error is inserted into the network to refresh the characteristics appropriately.

## 5. Dataset

The system's performance is evaluated using the WSDT Dataset. A total of 13326, 96*96 images and their corresponding classes [0, 1, or 2] are included in the WSDT dataset. Among those, 80 % can be used for training, and 20% have been used for testing. The information was stored after being transformed to the appropriate Gray-scale pixels for that image. As input for the training, CSV files were made available. The WSDT provided the real-time dataset, which included footage of Seattle, Washington, traffic. These CCTV videos were used to extract the frames. The images were initially 320x 240 pixels in size, and they were resized to 96*96 pixels before being converted to.csv files. On a dataset of 13326 images, using the described CNN architecture, we were able to achieve a 99.6 % accuracy rate. Furthermore, the CNN was trained on GPUs for 100 iterations.

## 6. Experimentation and Results

The Washington State Department of Transportation provided a real-time dataset of footage captured by highway CCTVs in Seattle, WA, which was employed to evaluate how well the proposed model performed. Based on current traffic, experiment findings categorize traffic density as high, low, and medium and accurately anticipate it up to 99.6 %. We've settled on a five-layer design with various input feature maps and filter sizes shown in Fig. 3. There are 64, 64, 96, 96, and 96 feature maps for each pair of convolution and max-pooling layers. Each pair's corresponding filter sizes are 5*5, 3*3, 5*5, 3*3, and 3*3.

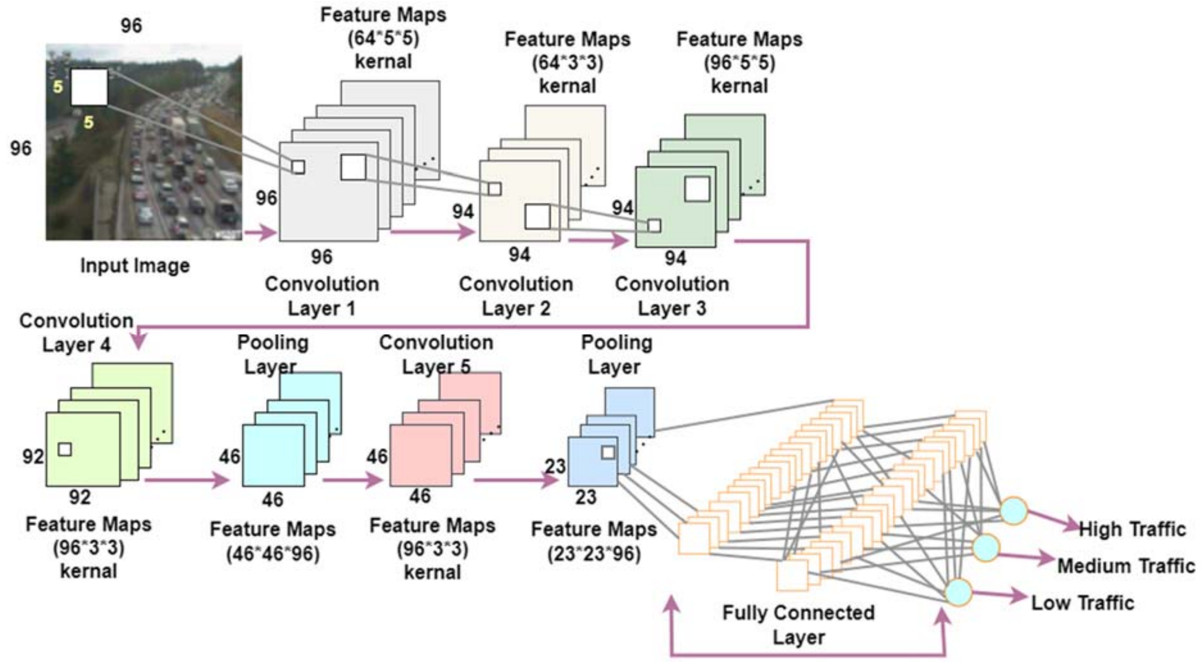


Fig. 3. Five-layered CNN Architecture

The final layer, a fully connected multi-layered perceptron with two fully connected layers, is used to classify the image into its last three categories: 0 (low traffic), 1 (medium traffic), and 2 (high traffic). To automatically extract the features from the image, the Convolution and Max Pooling procedures were applied five times and twice, respectively. The activation function used is Rectified Linear Unit (ReLU) because it is known to produce superior results for complicated situations, and 0.5 was used as the dropout for fully connected MLPs to exclude specific neurons from the training phase. In our example, there are three groups of output neurons for the traffic density.


Fig. 4. Sample Output.

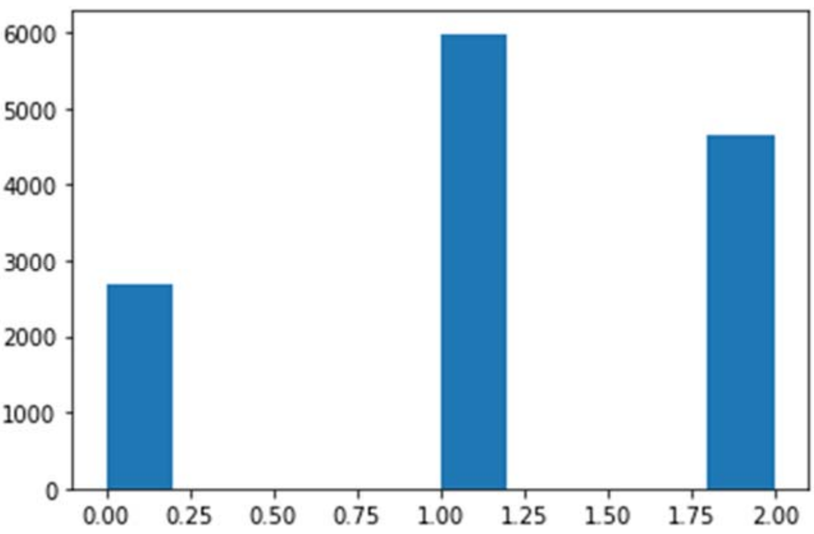

Fig. 5. Category-wise WSDT dataset.

Fig. 4. shows the sample output, and the WSDT dataset's overall sample count for each classification is shown in Fig. 5. Data for the class low contains 2691 images (Label 0), the class medium has 5982 images (Label 1), and the class high has 4653 images (Label 2).

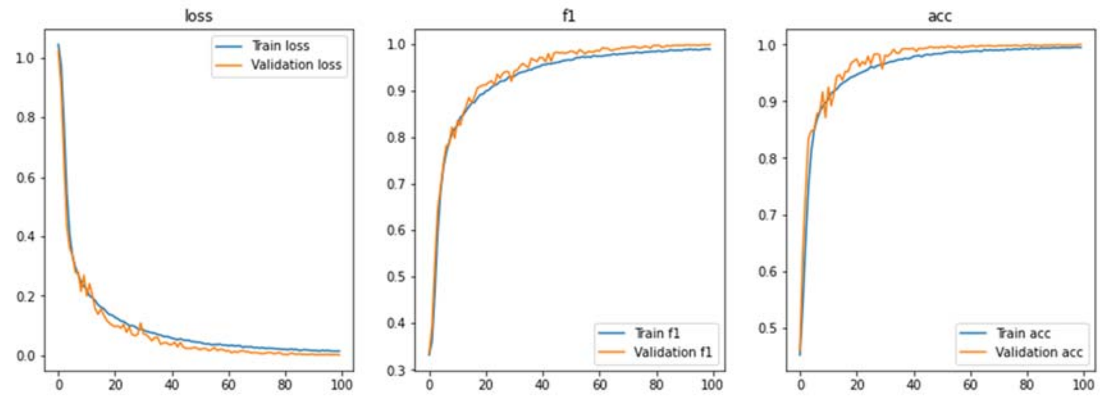


Fig. 6. Training vs. Validation loss for WSDT dataset.

Fig. 6. shows loss graphs at the time of training of the WSDT dataset and Fig. 7. shows the CNN sequential model for the WSDT dataset.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 96, 96, 64)        4864

 conv2d_1 (Conv2D)           (None, 94, 94, 64)        36928

 conv2d_2 (Conv2D)           (None, 94, 94, 96)        153696

 conv2d_3 (Conv2D)           (None, 92, 92, 96)        83040

 max_pooling2d (MaxPooling2D  (None, 46, 46, 96)       0
 )

 dropout (Dropout)           (None, 46, 46, 96)        0

 conv2d_4 (Conv2D)           (None, 46, 46, 96)        83040

 max_pooling2d_1 (MaxPooling  (None, 23, 23, 96)       0
 2D)

 dropout_1 (Dropout)         (None, 23, 23, 96)        0

 flatten (Flatten)           (None, 50784)             0

 dense (Dense)               (None, 512)               26001920

 dropout_2 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 3)                 1539

=================================================================
Total params: 26,365,027
Trainable params: 26,365,027
Non-trainable params: 0
```

Fig. 7. CNN sequential model for WSDT dataset.

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| class 0(NO DR) | 0.99 | 0.99 | 0.99 | 530 |
| class 1(MILD DR) | 1.00 | 1.00 | 1.00 | 1192 |
| class 2(MODERATE DR) | 1.00 | 1.00 | 1.00 | 944 |
| accuracy | | | 1.00 | 2666 |
| macro avg | 1.00 | 1.00 | 1.00 | 2666 |
| weighted avg | 1.00 | 1.00 | 1.00 | 2666 |

Fig. 8. WSDT Classification Report.

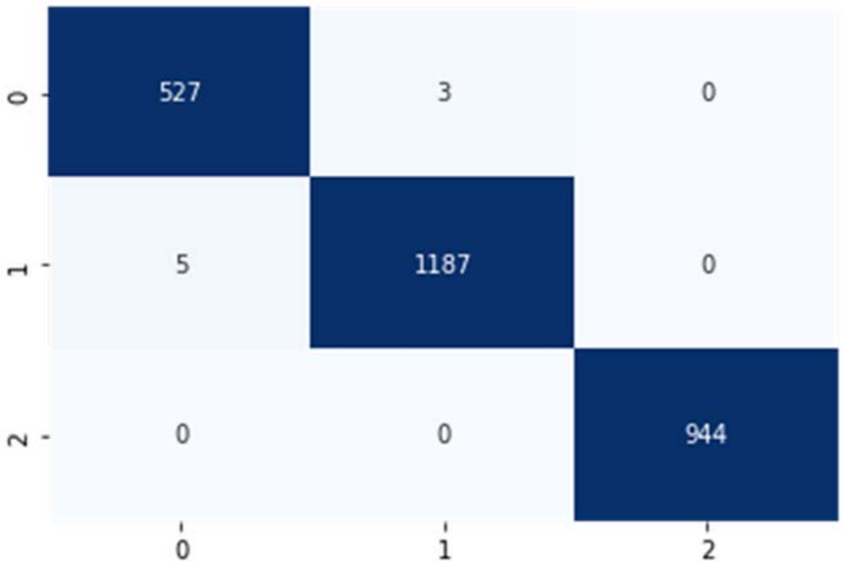


Fig. 9. Confusion matrix of WSDT dataset.

Fig. 8. depicts classification report After determining True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The confusion matrix evaluates the proposed classifier based on the collected values. The Recall is a measurement of the proper categorization of an object across all positive objects; the Precision value describes the appropriate categorization of objects by the classifier. The harmonic mean of Precision and Recall is the F1 score. Support is a measure of the overall sample size. The model's performance is determined by creating a confusion matrix and counting the number of accurate forecasts the model has produced out of all potential predictions.

Fig. 9. illustrates the Confusion Matrix to show how actual vs. expected images were predicted.

## 7. Discussion and Conclusion

Deep learning techniques are widely used for a variety of purposes. It has its advantages, disadvantages, issues, and challenges [Bidwe et al., (2022)]. Also, a massive amount of data is available for trying deep learning solutions on them. The intelligent traffic monitoring system is one of the widely explored possible application for deep learning. It has been tried and implemented in various cities. The paper [Mane et al., (2022)] has published the results of recorded videos from Pune city. This paper's proposed traffic monitoring system is trying to improve image classification on traffic images and categorize the images into three categories (high, medium, and low) using the traffic multiclass vehicle condition using the CNN Model. The WSDT dataset, which contains footage captured by highway CCTVs in Seattle, Washington, is used to evaluate the accuracy of the system. According to experimental findings, traffic is categorized into three classes: Low, Medium, and High, denoted by the numbers 0, 1, and 2, respectively. To assess the trained model's performance as a model, we have used parameters like accuracy, precision, f1-score, and support. On the WSDT dataset, the proposed system provides an accuracy of 99.6 %. The Traffic Monitoring System is beneficial for various applications like traffic signal time management, vehicle speed monitoring, path identification of the vehicle, traffic controlling or management, and the best route finding. The traffic density estimation data is beneficial for all these applications.

## References

[1] Lin, C. J., & Jhang, J. Y. (2022). Intelligent Traffic-Monitoring System Based on YOLO and Convolutional Fuzzy Neural Networks. IEEE Access, 10, 14120-14133.
[2] Zuraimi, M. A.B., & Zaman, F. H. K. (2021, April). Vehicle Detection and Tracking using YOLO and DeepSORT. In 2021 IEEE 11th IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE) (pp. 23-29). IEEE.
[3] Azimjonov, J., & Özmen, A. (2021). Real-time vehicle detection and novel vehicle tracking systems for estimating and monitoring highway traffic flow. Advanced Engineering Informatics, 50, 101393.
[4] Yang, H., Zhang, Y., Zhang, Y., Meng, H., Li, S., & Dai, X. (2021). A Fast Vehicle Counting and Traffic Volume Estimation Method Based on Convolutional Neural Network. IEEE Access, 9, 150522-150531.
[5] Liu, C.M., & Juang, J.C. (2021). Estimation of Lane-Level Traffic Flow Using a Deep Learning Technique. Applied Sciences, 11(12), 5619.
[6] Singh, M. K., Mishra, K. D., & Sahana, S. (2021). An Intelligent Real-time Traffic Control Based on Vehicle Density.
[7] Dave, P., Chandarana, A., Goel, P., & Ganatra, A. (2021). An amalgamation of YOLOv4 and XGBoost for the next-gen smart traffic management system. PeerJ Computer Science, 7, e586.
[8] Bui, N., Yi, H., & Cho, J. (2020). A vehicle counts by the class framework using distinguished regions tracking at multiple intersections. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 578-579).

[9] Khazukov, K., Shepelev, V., Karpeta, T., Shabiev, S., Slobodin, I., Charbadze, I., & Alferova, I. (2020). Real-time monitoring of traffic parameters. Journal of Big Data, 7(1), 1-20.
[10] Li, S., Chang, F., Liu, C., & Li, N. (2020). Vehicle counting and traffic flow parameter estimation for dense traffic scenes. IET Intelligent Transport Systems, 14(12), 1517-1523.
[11] Doan, T. N., & Truong, M. T. (2020, November). Real-time vehicle detection and counting based on YOLO and DeepSORT. In 2020 12th International Conference on Knowledge and Systems Engineering (KSE) (pp. 67-72). IEEE.
[12] Pham, L. H., Phan, H. N., Chung, N.M., Vu, T. A., & Ha, S. V. U. (2020, October). A robust multiclass vehicle detection and classification algorithm for a traffic surveillance system. In 2020 RIVF International Conference on Computing and Communication Technologies (RIVF) (pp. 1-6). IEEE.
[13] Huy, T. N., & Duc, B. H. (2020, November). Traffic Flow Estimation Using Deep Learning. In 2020 5th International Conference on Green Technology and Sustainable Development (GTSD) (pp. 180-184). IEEE.
[14] Lou, L., Zhang, Q., Liu, C., Sheng, M., Zheng, Y., & Liu, X. (2019, May). Vehicle detection of traffic flow video using deep learning. In 2019 IEEE 8th Data-Driven Control and Learning Systems Conference (DDCLS) (pp. 1012-1017). IEEE.
[15] 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.
[16] Bhuiyan, T. A. U. H., Das, M., & Sajib, M. S. R. (2019). Computer vision-based traffic monitoring and analyzing from on-road videos. Global Journal of Computer Science and Technology.
[17] Lahinta, F.C., Zainuddin, Z., & Syarif, S. (2019). Vehicle Detection and Counting to Identify Traffic Density in The Intersection of Road Using Image Processing.
[18] Asha, C. S., & Narasimhadhan, A. V. (2018, March). Vehicle counting for traffic management system using YOLO and correlation filter. In 2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT) (pp. 1-6). IEEE.
[19] Zhu, J., Sun, K., Jia, S., Li, Q., Hou, X., Lin, W., ... & Qiu, G. (2018). Urban traffic density estimation based on ultrahigh-resolution UAV video and deep neural network. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 11(12), 4968-4981.
[20] Xiang, X., Zhai, M., Lv, N., & El Saddik, A. (2018). Vehicle counting based on vehicle detection and tracking from aerial videos. Sensors, 18(8), 2560.
[21] Bidwe, R.V.; Mishra, S.; Patil, S.; Shaw, K.; Vora, D.R.; Kotecha, K.; Zope, B. Deep Learning Approaches for Video Compression: A Bibliometric Analysis. Big Data Cogn. Comput. 2022, 6, 44. https://doi.org/10.3390/bdcc6020044
[22] Mane, D., Bidwe, R., Zope, B., & Ranjan, N. (2022). Traffic Density Classification for Multiclass Vehicles Using Customized Convolutional Neural Network for Smart City. In Communication and Intelligent Systems (pp. 1015-1030). Springer, Singapore.

## Authors Profile

**Shraddha Prakash Bidwe** has born in Latur, Maharashtra, India. She has completed a master's degree in 'Computer Engineering (Data Science)' from 'The Pune Institute of Computer Technology, Pune.' She has completed a bachelor's in Electronics Engineering (BE EC) from M. S. Bidve College of Engineering, Latur, India. Her research interests lie in Computer Vision, Machine Learning, and Deep Learning.

**Dr. Geetanjali V. Kale** completed her B.E. in 2001 and M.E. in 2004 from PICT and Ph.D. from Savitribai Phule Pune University in 2017. Dr. Geetanjali is currently working as an Associate Professor and Head of the Department of Computer Engineering at SCTR's PICT. She has completed funded projects and consultancy projects as a principal investigator. She has one patent and has published more than 40 research papers in international journals and conferences. She has also contributed as the author of book chapters and as a reviewer and TPC for various journals & conferences. She has guided 48+ UG and 18+ PG projects in multiple domains and is an SPPU-recognized Ph.D. Guide. She has received the honor of west zone representative for the first Mission 10X India meet at Bangalore, and the best performer award in the first Maharashtra meet for her contributions to the innovative teaching-learning process. Currently, she is contributing as a member of the Board of Studies Computer Engineering, SPPU, Member of the board of studies Vocational Courses, and SPPU. Vice Chair of ACM-W Pune Professional Chapter, Secretary of ACM Pune Professional Chapter, Working as a core team member of AnitaBorg Pune Chapter. She is also a Chapter Councillor for PICT ACM Student Chapter, and under her guidance and leadership chapter has bagged ACM India Best Chapter Award four times in a row. Her interests are Computer Vision, Machine Learning, Data Science, and innovation in education.

**Ranjeet Vasant Bidwe** is currently working as an Assistant Professor at Symbiosis institute of technology, Pune. He has completed a Master's Degree in 'Computer Engineering' from 'The Pune Institute of Computer Technology, Pune.' He has completed a bachelor's in Computer Science & Engineering from 'M. S. Bidve College of Engineering, Latur, India. His research interests lie in Computer Vision, Deep Learning. He has authored 9 Peer-reviewed articles in publishing houses, including IEEE, Elsevier, Springer, MDPI, Scopus, etc. He has presented two papers at national and four at international conferences. He has nine years of experience in teaching and research and served as a TPC member and reviewer for several reputed journals and conferences by IEEE and Springer. He is currently pursuing his Ph.D. from Symbiosis International Deemed University.