Categorical Vehicle Classification and Tracking using Deep Neural Networks

Deependra Sharma, Zainul Abdin Jaffery
Department of Electrical Engineering
Jamia Millia Islamia, New Delhi, India

***Abstract*—The classification and tracking of vehicles is a crucial component of modern transportation infrastructure. Transport authorities make significant investments in it since it is one of the most critical transportation facilities for collecting and analyzing traffic data to optimize route utilization, increase transportation safety, and build future transportation plans. Numerous novel traffic evaluation and monitoring systems have been developed as a result of recent improvements in fast computing technologies. However, still the camera-based systems lag in accuracy as mostly the systems are constructed using limited traffic datasets that do not adequately account for weather conditions, camera viewpoints, and highway layouts, forcing the system to make trade-offs in terms of the number of actual detections. This research offers a categorical vehicle classification and tracking system based on deep neural networks to overcome these difficulties. The capabilities of generative adversarial networks framework to compensate for weather variability, Gaussian models to look for roadway configurations, single shot multibox detector for categorical vehicle detections with high precision and boosted efficient binary local image descriptor for tracking multiple vehicle objects are all incorporated into the research. The study also includes the publication of a high-quality traffic dataset with four different perspectives in various environments. The proposed approach has been applied on the published dataset and the performance has been evaluated. The results verify that using the proposed flow of approach one can attain higher detection and tracking accuracy.**

Keywords—Vehicle classification; generative adversarial networks; single shot multibox detector; vehicle tracking; deep neural networks

I. Introduction

With a rising count of vehicles on road, and those in a huge variety, resulting in traffic congestion and a slew of related difficulties, it is necessary to address these issues [1]. It motivates us to consider an intelligent and smart traffic monitoring system that could assist traffic agencies in addressing issues such as routing traffic based on the density of vehicle movement on the road, collecting traffic data like count of vehicles, vehicle type, and vehicle motion parameters, and managing roadside assistance in the event of an accident or other anomalous incident. It conducts traffic analysis using the acquired data to optimize the use of highway networks, forecast future transportation demands, and enhance transportation safety [2]. The primary functions of an intelligent and intelligent traffic monitoring system are vehicle categorization and tracking on a category basis. Due to the substantial technological problems associated with the same, several research topics have been studied, resulting in the creation of numerous vehicle categorization, and tracking systems. Classifying vehicles and maintaining their trajectories properly in a variety of environmental circumstances is critical for efficient traffic operation and transportation planning.

The scientific advancements have resulted in the development of several novel vehicle categorization systems. Three types of categorical vehicle classification systems may be found in use today: in-road, over-road, and side-road. Each category of vehicle classification is further divided into subcategories depending on the sensors utilized, the techniques used to utilize the sensors, and the processes used to classify cars [3]. While both in-road and side-road approaches are capable of accurate categorical vehicle classification, they differ significantly in terms of sensor types, hardware configurations, configuration process, parameterization, operational requirements, and even expenses, making it even more difficult to determine the most suitable solution for a given vehicle in the first instance. These techniques have limitations when more than one vehicle is in the same location at the same time [4]. So, these techniques can’t be utilized for tracking the vehicles.

To circumvent the restrictions, over-the-road-based methods for category vehicle classification and tracking are used. Camera-based systems are the most popular technology for over-road-based systems [5] [6]. The cameras are mounted at a height sufficient to cover the road's wide field of vision and can span several lanes. There are two primary obstacles to attaining our aim that are linked with camera-based systems. To begin, their performance is significantly impacted by weather and lighting conditions, resulting in blurred, hazy, and rainy observations in collected pictures. The same findings are made in captured pictures when automobiles are travelling at high speeds on the road. Second, a higher viewing angle allows for consideration of more distant road surfaces, however, the vehicle's object size changes significantly, and the accuracy of detection of tiny objects located distant from the road suffers because of the shift. We focus on above two difficulties in this work to provide a feasible solution, and we demonstrate how to adapt the category vehicle recognition findings to multiple object tracking.

II. Related Work

A. *Image Restoration*

Images restoration problems such as image deblurring, dehazing and deraining being all focused at creating an accurate representation of a clear final picture out of an insufficiently clear input image. Numerous studies have been conducted in this area. A multi-layer perceptron technique for deblurring that eliminates noise and artefacts [7]. To cope with outliers, a CNN based on the single value dissemination is used [8]. Certain techniques [9], [10] begin by estimating blur kernels with convolutional neural networks and subsequently deblur images using traditional restoration methods. Many edge adaptive neural networks have been developed for the purpose of recovering clear images instantly [11], [12]. Recent deep learning-based approaches for image dehazing [13], [14] estimate transmission maps first and subsequently restore clear images using conventional methodologies [15]. Typically, traditional methods for image deraining are created using the statistical characteristics of rainy streaks [16-19]. The author in [20] built neural network for removing rain and/or dirt from pictures. Having been developed with the aid of the ResNet [21], [22] built deep network for image deraining. The author in [23] introduced Generative Adversarial Network (GAN) architecture for generating realistic pictures from random noise. Numerous techniques for visual tasks have been developed because of this framework [24–27]. The authors in [28-31] have also utilized the GAN framework to low-level vision issues. We chose to apply the capabilities of the GAN framework physics model [32] for picture restoration jobs due to the positive findings.

B. *Detection of Vehicles*

Now, vehicle detection can be accomplished using both standard machine vision techniques and sophisticated deep learning techniques. Traditionally, machine vision techniques employ a vehicle's motion to distinguish it from a fixed backdrop picture. This approach may be classified into three categories [33] as background subtraction [34], frame subtraction on a continual basis [35], and optical flow [36]. Variance is determined by applying the frame subtraction technique, which compares pixel data of two or three successive frames. Additionally, threshold separates the shifting foreground region [35]. By employing this technique and reducing noise, the vehicle's halt may also be recognized [37]. When the video's backdrop picture being stationary, background data is used to build the model [37]. Following that, it is possible to segment the moving object as well as the frame pictures by comparing each frame image to the backdrop model. Optical flow approach being exploited to detect a motion area in frames. The resulting optical flow field encodes the direction of motion and speed of each pixel [36]. While the classic machine vision approach detects the vehicle more quickly, it does not perform well in case the image brightness varies, there being a continuous motion in backdrop, or there are vehicles moving with low speed or some complicated sceneries. Vehicle identification using deep convolutional neural networks [52] may be classified into two broad groups. The two-stage technique begins by generating a candidate box for the item using multiple methods and then classifying it using a CNN. Second, a single-stage technique could not produce candidate box but instead turns object bounding box placement problem straight transform it into a regression problem that can be processed. Region-CNN (R-CNN) [38] employs a two-stage technique that utilizes selective search of region [39] in image. CNN image input must be fixed size, and the network's deeper structure needs a lengthy training period and uses a significant amount of storage capacity. SPP NET [40], which is based on concept of spatial pyramid matching, enables the network to accept pictures of varying sizes and provide fixed outputs. Among the one-stage techniques, the Single Shot Multibox Detector (SSMD) [41] and You Only Look Once (YOLO) [42] frameworks are most important. For many categories, SSD for single shot detectors (YOLO) that is significantly faster than the preceding state-of-the-art and as accurate as slower techniques that undertake explicit area recommendations and pooling, such as the Faster R-CNN [43]. SSMD's central idea is to forecast category scores and box offsets for a specific set of default bounding boxes by applying tiny convolutional filters on feature maps. We chose to use the SSD framework [43] for categorical vehicle identification and classification tasks due to the positive findings.

C. *Tracking of Vehicles*

Aspects of the functioning of an intelligent traffic system that need advanced vehicle object identification applications, such as multiple object tracking, are also crucial [44]. DBT (Detection-Based Tracking) and Detection-Free Tracking (DFT) are the two most common methods of initializing objects in multi-object tracking systems (DFT). To detect moving objects in video frames, the DBT method first uses background modelling to detect them before tracking them. However, the DFT technique is only capable of initializing the tracking object and cannot deal with the addition of new objects or the removal of current ones. Multi-object tracking algorithms must consider the similarity of items inside a frame, as well the associated problem of objects across frames, when developing their algorithms. The normalized cross-correlation function may be used to determine the similarity of objects inside a frame. As shown in [45], the Bhattacharyya distance is being used to calculate the distance between two objects based on the colour histograms of their respective images. When connecting inter-frame items, it is critical to specify that each item may appear on no more than one track at a time and that each track may include no more than one object. It is now possible to fix this issue by using either detection-level exclusion or trajectory-level exclusion. SIFT and ORB feature points were used for object tracking to overcome the difficulties caused by size and illumination changes in moving objects in [46] and [47], however this approach is slow and requires many feature points. The feature point detection technique Boosted Efficient Binary Local Image Descriptor (BEBLID) is proposed for use in this study [48]. BEBLID is considerably faster than SIFT and ORB in extracting feature points.

D. *Our Contributions Comprise the following Items*

- On the foundation of this work, a large-scale dataset of vehicle movement on roads has been developed, which may offer many distinct category vehicle objects that have been thoroughly annotated under diverse situations taken by high-mounted cameras. It is possible to utilize the dataset to test the performance of a variety of vehicle detection methods.

- For recovering blurred, hazy, or rainy images recorded in road scenes, a method based on the GAN framework for image restoration has been developed. This approach is utilized to increase the accuracy of vehicle detection in road scenes.

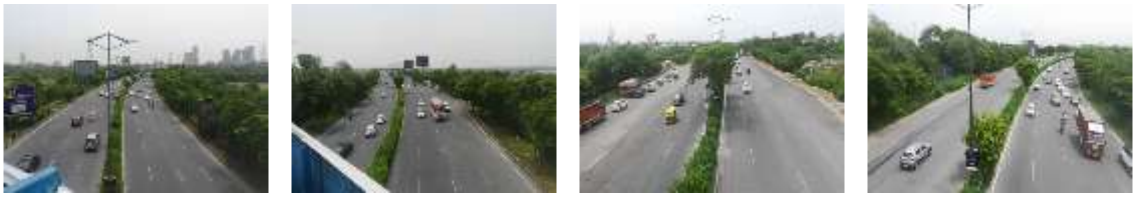
- A technique based on convolutional neural networks, i.e., SSMD, is implemented for category vehicle detection.
- A system for tracking and analyzing several vehicles is presented for road situations. The BEBLID method extracts and matches the detected object's feature points.

Findings of this investigation will be discussed in further detail in the following sections. Section 3 introduces the vehicle dataset that will be utilized in this work. During Section 4, you'll learn about the general procedure of the suggested system. Section 5 shows the results of the experiments as well as the relevant analyses. Section 6 provides a comprehensive summary of the complete method.

III. Vehicle Dataset

Because of concerns about copyright, privacy, and security, traffic dataset is rarely made public owing to the widespread use of traffic surveillance cameras on highways across the world. With images of highway sceneries and typical road scenes, the KITTI benchmark dataset [31] aids in the solution of issues such as 3D object identification and tracking, which are commonly encountered in automated vehicle driving applications. The Tsinghua-Tencent Traffic-Sign Dataset [32] contains pictures captured by automobile cameras in a variety of lighting and weather situations, however there are no cars identified. The Stanford Car Dataset [33] and the Comprehensive Cars Dataset [34] are vehicle datasets captured by non-monitoring cameras and featuring a bright car look; they are used in research and development. The datasets are captured by security cameras; one such dataset is BV Dataset [35], which is an example. Even though this dataset categorizes vehicles into 6 categories, shooting angle being positive, and the vehicle object is too tiny for each image, making the generalization impossible for CNN training. A dataset called Traffic and Congestions [36] comprises photos of cars on roads collected by security cameras, however most of the images have some degree of occlusion in them. This dataset has a small number of images and contains no information on the vehicle's classification, making it less helpful. As a result, only a few datasets have pertinent annotations, and there are only a few images of traffic scenes available. This section provides an overview of the vehicle dataset from the standpoint of road surveillance footage that we created. Dataset available on link: https://drive.google.com/drive/folders/1vYwLPkZZ2OX1cIIP QZA4SgB3dum7vPwV?usp=sharing. The video in the dataset is taken from the DND road in Delhi, India as shown in Fig. 1. The road monitoring camera was put on the side of the road and built at a height of 10 meters with a fixed angle of view. The photos taken from this vantage point span a large portion of the road in the distance and include cars of all types. The pictures in the dataset were taken from four surveillance cameras at different times of day and under varied lighting situations to provide a diverse range of photographs. The vehicles in this dataset are divided into three categories: two-wheelers, Light Motor Vehicles (LMV), which include three-wheelers, automobiles, minivans, and other similar vehicles, and Heavy Motor Vehicles (HMV), which include buses, trucks, and other similar vehicles (Fig. 2). The Table I details out the information about the dataset published.

An initial training set and a second test set are included in this dataset, which is separated into two sections. Two-wheelers accounted for 28.45 percent of all vehicles in our dataset, while light motor vehicles (LMV) accounted for 61.34 percent and heavy motor vehicles (HMV) accounted for 10.21 percent. On average, each image has 3.64 instances of annotated instances. Comparing our dataset to the current vehicle datasets, ours has a greater number of categorized vehicle pictures, adequate lighting conditions, and comprehensive annotations.



(a) View 1 (b) View 2 (c)View 3 (d) View 4

Fig. 1. Different views of the Dataset Collected.

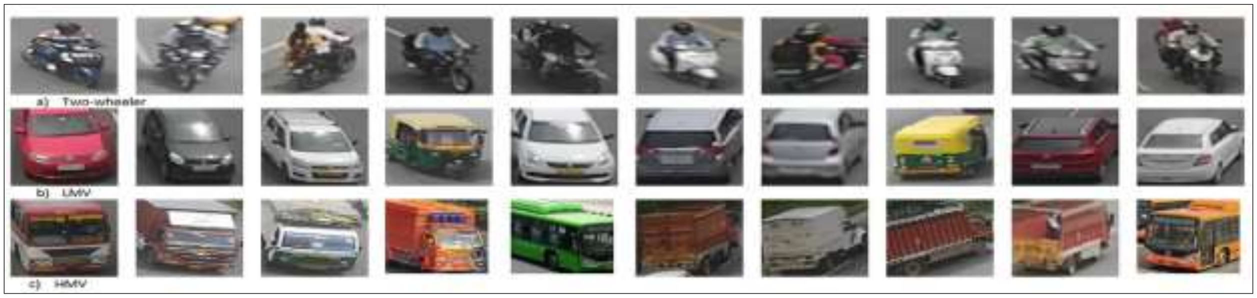


Fig. 2. Dataset with Three Categories of Vehicles. (a) Two-Wheeler, (b) LMV and (c) HMV.

TABLE I. VEHICLE DATASET DETAIL

Image format	Size	Total number of images	Total number of annotated instances	Average annotated instances per image[$]
RGB	1280X720	10502	38228	3.64

[$] Total number of instances/Total number of images

IV. METHODOLOGY

The technique of the categorical vehicle classification and tracking system is described in detail in this section. First, the video data from the road traffic scenario is imported into the system. Second, the GAN framework is used to recover the pictures that have been captured. After that, the road area is excavated. The SSMD deep learning object detection technique is being used to recognize presence of vehicles belonging to three different categories in a road traffic environment. Finally, BEBLID feature extraction is carried out on the identified vehicle box to complete the tracking of numerous vehicle objects. In the proposed technique, the essential components of picture restoration, vehicle detection, propagating object states into future frames, linking current detections with existing objects, and controlling the lifespan of tracked objects are all discussed in detail. Diagram of the methodology's building blocks is depicted in Fig. 3.

A. Image Restoration

As previously stated, weather and lighting circumstances have a significant impact on the performance of camera-based systems, resulting in blurring, hazing, and precipitation observations in the captured pictures. High-speed vehicle movement on the road is observed in captured images, and the same or similar observations can be deduced from those images. The former scenario is caused by environmental changes and is thus less likely to occur, but the latter situation occurs almost without fail, necessitating the need for restoration. To achieve precise vehicle detection, it is necessary to repair the images to eliminate the issues that have arisen. Following a study of the literature on picture restoration approaches, we were encouraged by the positive results to apply the capabilities of the GAN framework physics model [32] to image restoration problems in our own research.

1) Image Restoration with GAN: An image restoration task is to predict a clear picture x from an input image y that as been provided. Fundamentally, the estimated x should be compatible with the input y under the picture creation paradigm, which is as follows:

$$y = H(x) \tag{1}$$

The operator H is used to transfer the unknown outcome x to the seen picture y. Depending on the situation, the blur, haze, or rain operation may be used. It is required to apply extra constraints on x to regularize it since the estimation of x from y is not well-posed. In the maximum a posteriori (MAP) paradigm, one frequently used method is predicated on the assumption that x may be solved by,

$$x^* = arg \max_x p(x|y) = arg \max_x p(y|x)p(x) \tag{2}$$

In the above equation, $p(y|x)$ and $p(x)$ are probability density functions, which are referred to as the likelihood term and image prior in the scientific literature, respectively. The mapping functions between *x* and *y* are directly learned using mathematical approaches,

$$x^* = G(y) \tag{3}$$

G is the mapping function in this case. In the case of the function *G*, it can be considered an inverse operator of *H*. If the mapping function can be predicted accurately, *G(y)* should be near to the ground truth, theoretically speaking.

The adversarial learning method used by the GAN algorithm is used to learn a generative model. It trains a generative network and a discriminative network at the same time by optimizing, among other things.

$$\min_G \max_D E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{4}$$

in which *z* represents random noise, *x* represents a genuine picture, and *D* represents a discriminative network are used. For the sake of convenience, we will also refer to a generative network as *G*. As part of the training process, the generator generates samples (*G(z)*) that may be used to deceive the discriminator, while the discriminator learns to discriminate between actual data and samples generated by the generator. A binary classifier is used as the discriminator. If the observed image y serves as the input to the generator, then the adversarial loss is,

$$\max_D E_{x\sim p_{data}(x)}[\log D(x)] + E_{y\sim data(z)}[\log(1 - D(G(y)))] \tag{5}$$

The value of (5) is near to zero if the distribution of the produced picture *G(y)* differs considerably from the distribution of the clear image, and it is greater if the distribution differs significantly from the clear image. It is possible to address the image restoration difficulty by doing the negative log procedure,

$$x^* = arg \min_x \rho(x, y) + \varphi(x) \tag{6}$$

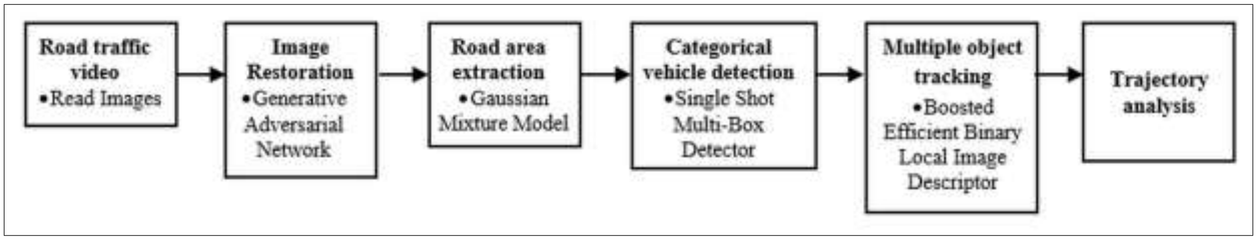

Road traffic video
•Read Images
Image Restoration
•Generative Adversarial Network
Road area extraction
•Gaussian Mixture Model
Categorical vehicle detection
•Single Shot Multi-Box Detector
Multiple object tracking
•Boosted Efficient Binary Local Image Descriptor
Trajectory analysis

Fig. 3. Block Diagram of the Proposed System Methodology.

If we consider the data term to ensure that the recovered image x and the input image y are consistent under the appropriate image degradation model, then we get $\rho(x, y)$. The regularisation of the recovered image x is denoted by $\varphi(x)$ and models the characteristics of the recovered image, respectively. In vision tasks, the function $\varphi(x)$ functions as a discriminator, with the value of the function being considerably smaller if x is clear and much bigger otherwise. In other words, maximizing the goal function as Eq. 3 will result in a decrease in the value of x. As a result, the predicted intermediate picture will be significantly more detailed. Accordingly, in order to regularize the solution space of picture restoration, adversarial loss can be employed as a precursor to the restoration. Fig. 4 depicts the major components of the GAN method, which include two discriminative networks, one generative network, and one picture degradation model [32], as well as their interactions.

Let x_i and y_i indicate the clear and blurred images, respectively. The generative network derives the mapping function G from the input y_i and creates the intermediate restored image $G(y_i)$. The physics model for regenerating the image $\widetilde{y_i}$ for various operations is as follows: for image deblurring,

$$\widetilde{y_i} = k_i \otimes G(y_i) \tag{7}$$

where k_i being the kernel for blur, and $\otimes$ represents convolution operator. For image dehazing and deraining,

$$\widetilde{y_i} = G(x_i)t_i + A_i(1 - t_i) \tag{8}$$

where A_i representing an atmospheric factor and t_i being the transmission map. The discriminative network D_g is used to determine if the distributions of the generator G outputs are comparable to those of the ground truth images. It is required to categorize using the discriminative network D_h whether the regenerated result $\widetilde{y_i}$ is consistent with the observed image y_i. All the networks are taught in a collaborative manner from beginning to end.

During training, we rely on an Adam optimizer, which starts with a learning rate of 0.0002, with the method outlined in [24] being used. To get our results, we choose a batch size of one and a slope of 0.2 for the Leaky-ReLU. We use the same weight initialization strategy [24] uses. We must first get the generator G to create $G(y_i)$ and y_i. We may utilise the relevant physics model parameters to employ the generator, as we know the training data as well as the physics model parameters $\widetilde{y_i}$. The discriminators D_g and D_h accept input data sets $\{x_i, G(y_i)\}$ and $\{y_i, \widetilde{y_i}\}$ respectively. We update the discriminators using a history of produced pictures (rather than the most recent generative networks' images) according to the methods discussed in [24]. The generator and the discriminators have a one-to-one update ratio set between them.

B. Excavation of the Road Area

The next section covers the procedure for removing the road surface. We developed it using an image processing approach called the Gaussian mixture model, which results in superior vehicle detection results when combined with the deep learning object detection method, as shown in Fig. 2. The video picture of traffic on the road has a wide field of vision. In this investigation, the cars are the primary centre of attention, and the road area is the zone of interest in the resulting image. Meanwhile, depending on the camera's view angle, road area being focused for certain range of the image's horizontal and vertical planes. We were able to extract the road segments from the video using this function. In a traffic scenario, a perfect background is not always accessible and may always be modified in crucial circumstances by the introduction or removal of items from the picture, as well as the presence of objects that are either slow moving or immobile. The Gaussian mixture model (GMM) was used to account for all these factors correctly. According to the method, background is visible more frequently than foreground and model variance is small [49].

The recent history of the intensity values of each pixel X_1, ..., X_t is modeled by a mixture of K Gaussian distribution. The probability of observing the current pixel value is given by the formula:

$$P(X_t) = \sum_{k=1}^{K} w_{k,t} * \eta(X_t, \mu_{k,t}, \Sigma_{k,t}) \tag{9}$$

where K gives the number of Gaussian distributions, $w_{k,t}$ is the weight of the k^{th} Gaussian in the mixture at time t having mean $\mu_{k,t}$ and covariance matrix $\Sigma_{k,t}$ and η is a Gaussian probability density function which is given by

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}} exp^{-\frac{1}{2}(X_t-\mu)^T\Sigma^{-1}(X_t-\mu)} \tag{10}$$

where n is the dimension of the colour space and is the number of colours in the colour space. As soon as the parameters have been initialized, the K Gaussians are sorted in the order of the ratio 1/(k). Due to the fact that backgrounds are more prevalent in scenes than moving objects, as well as the fact that their values are almost constant, it follows that a backdrop pixel equates to a high weight with low variation. The first B Gaussian distributions that surpass a specific threshold T_1 are kept for use as a background distribution. For example,

$$B = \arg\min_b\left(\sum_{k=1}^{b} w_k > T_1\right) \tag{11}$$

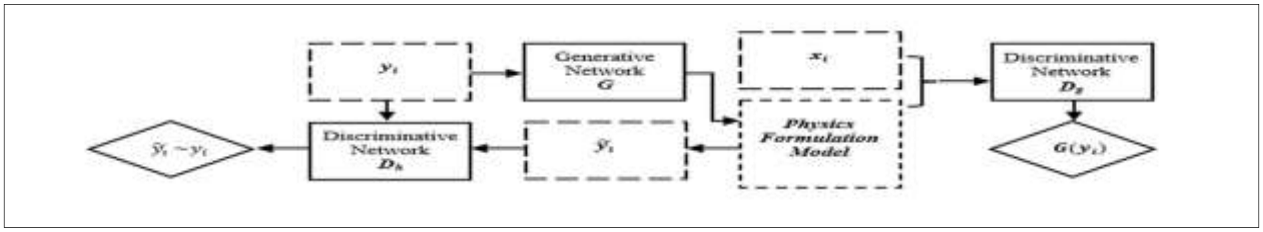
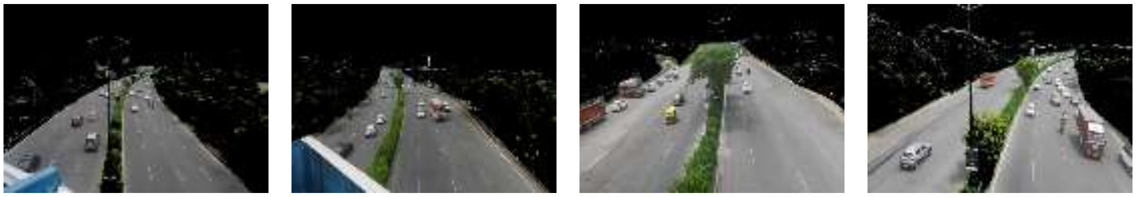

y_i
Generative Network G
x_i
Discriminative Network D_g
Physics Formulation Model
$\widetilde{y_i} \sim y_i$
Discriminative Network D_h
$\widetilde{y_i}$
$G(y_i)$

Fig. 4. Major Components of the GAN Framework.



(a) View 1 (b) View 2 (c) View 3 (d) View 4

Fig. 5. Road Area Extracted for all Four Views.

Distributed data that is part of the foreground represents other distributions. Until a match is found, the process repeats as the system computes and compares every new X_t value to the K Gaussian distributions. A pixel's value follows a Gaussian distribution if it is 2.5 standard deviations away from that distribution's mean. The background image is smoothed using a Gaussian filter once the road section has been extracted as the background picture. The MeanShift method smoothes the input image's colour. The final step is to finish filling the holes and carrying out morphological procedures in order to get most of the road surface. To extract the road regions, we made use of a variety of landscapes and have the results in Fig. 5.

C. Categorical Vehicle Detection using SSMD

Here is a description of the object detection approach that was employed in this study. The SSMD network was utilised in the development of the categorical vehicle detection framework and its deployment. The SSD approach's final detections are created by feeding bounding-boxes and scores of object class occurrences into a fixed-size feed-forward convolutional network followed by a non-maximum suppression phase. Addition of an auxiliary structure to the base network, such as the VGG-16, results in detections that have the following important characteristics:

1) Maps of multi-scale feature for identifying anomalies: At end of the truncated base network, convolutional feature layers are added to complete the network. These layers get smaller and smaller as time goes on, and they allow for predictions of detections at various sizes.

2) Convolutional neural network prediction techniques: A sequence of convolutional filters is associated with each feature layer, and it creates a discrete set of detection results. The three-dimensional tiny kernels provide either a score for each category or an offset in the shape relative to the default box coordinates and are the essential element used for the prediction of parameters in a feature layer of size mxn with channels. For each kernel location, it generates a number as an output. When it comes to figuring out the bounding box offset output values, it is crucial to first understand the differences between measurements made on various feature maps.

3) Box and aspect ratio defaults: In the design of feature map cells, each is equipped with default bounding boxes, even if many feature maps are employed above the cell. Due to the tiling of the feature map's boxes, with the position of each box in relation to its associated cell fixed, the boxes' arrangements in the feature map are fixed. We predict the offsets, class scores, and the box shapes for each feature map cell. From there, we calculate the class scores and four offsets to get the final bounding box, as seen in the illustration. The (c + 4)k filters being applied around each spot in the feature map amount to (c + 4)kmn outputs for a m x n feature map.

a) Training: For SSD training to be effective, the ground truth information must be allotted to certain detector outputs in the fixed set of detector outputs. Once a decision has been made on this assignment, it is applied completely to the loss function and back propagation. Additionally, you must pick the set of default boxes and scales that you will use for the data augmentation and the hard negative mining and methods.

i) Training method for matching: For training, we need to discover the ground truth boxes and train the network according to that discovery. For each ground truth box we create, we are using a preset box that is predefined with a variety of attributes, such as box size, box aspect ratio, and box placement. For every ground truth box, we compare it to the best-overlapping default box. Any boxes that meet the requirements are then matched to ground truth in which the jaccard overlap is over a certain level (0.5). By contrast, the learning challenge is made easier since the network may make predictions about a large number of default boxes that overlap, instead of needing to select one single box as the biggest overlapper.

ii) Loss function: The training aim is to be able to deal with a variety of vehicle types. We'll define an indication of matching a box in the i-th category to a box in the j-th category as $x_{ij}^{p} = \{1,0\}$. $\sum_i x_{ij}^{p} \geq 1$ holds under the matching strategy shown above. The weighted sum of the localization loss (loc) and the confidence loss (conf) are the overall objective loss function:

$$L(x,c,l,g) = \frac{1}{N}\left(L_{conf}(x,c) + \alpha L_{loc}(x,l,g)\right) \qquad (12)$$

where N is the number of matching default boxes, and the weight term has been adjusted to one via cross validation. If N equals 0, the loss is set to zero. In a localization test, the localization loss is the difference between the expected box (l) parameters and the ground truth box (g) values.

iii) Scales and aspect ratios for default boxes: To manage diverse object scales, feature maps from many distinct layers in a single network are used for prediction, with parameters shared across all object scales. This allows the network to handle several object scales at the same time. In addition, it has been depicted that feature maps from the lower layers could help to enhance the quality of semantic segmentation since the lower layers capture finer features of the input objects. For detection, we make use of both the bottom and higher feature maps. With the tiling of default boxes, we may train individual feature maps to be sensitive to objects of different sizes and shapes over time. Assume that we wish to make predictions using m feature maps. The following formula is used to determine scale of default boxes for every feature map:

$$s_k = s_{min} + \frac{s_{max} - s_{min}}{m-1}(k-1), k \in [1, m] \quad (13)$$

Where s_{min} equals 0.2 and s_{max} equals 0.9, the lowest layer has a scale of 0.2, the topmost layer has a scale of 0.9, and all levels in between are evenly spaced. We impose various aspect ratios on the default boxes, denoted by the variables $a_r \in \{1,2,3,1/2,1/3\}$. We can determine the width $w_k^a = s_k\sqrt{a_r}$ and height $h_k^a = s_k\sqrt{a_r}$ of each default box. The centre of each default box is set to $\left(\frac{i+0.5}{|f_k|}, \frac{j+0.5}{|f_k|}\right)$, where $|f_k|$ denotes the size of the k-th square feature map, $i, j \in [0, |f_k|]$.

iv) Hard negative mining: we rank the default boxes according to their largest confidence loss and choose just those at the top of the list, ensuring that the ratio of negatives to positives is no more than 3:1. This resulted in a speedier optimization process and more uniform training.

v) Enhancement of data: To make the model more robust to a broad range of input object sizes and shapes, each training image is randomly chosen using one of the following methods:

- Utilize the whole original input image.
- Sample a patch with values of 0.1, 0.3, 0.5, 0.7, or 0.9 to obtain the least feasible jaccard overlap with the objects.
- Take a sample of a patch at random.

Each sampled patch is between [0.1 and 1] of the original image's size, with an aspect ratio of between 1/2 and 2. Following the preceding sampling step, each sampled patch is given a fixed size, and the patches are then horizontally flipped with a probability of 50%.

D. Multiple Vehicle Object Tracking

This section describes how numerous vehicle objects are tracked using the object box discovered in the preceding section. During this stage, the BEBLID algorithm was employed to extract vehicle characteristics, and good results were achieved. The BEBLID method surpasses the competition by a considerable margin in terms of computing performance and matching costs. This algorithm is a superior alternative to other image description algorithms that have been previously described in the literature. Feature computations for the BEBLID algorithm are based on differences in grey values between a pair of box image regions, with the integral image serving as a basis for computations for the BEBLID algorithm features based on differences in grey values between a pair of box image regions. The technique takes use of AdaBoost to train a descriptor on an imbalanced data set to handle the challenge of highly asymmetric image matching. Binarization in a descriptor is achieved by minimizing the amount of new similarity loss in which all weak learners share a common weight. The coordinate system must be established by assuming the feature point to be at the centre of a circle and using the centroid of the point region to represent the coordinate system's x-axis. Thus, when the image is rotated, the coordinate system may be adjusted to match the image's rotation, resulting in rotation consistency in the feature point descriptor. When viewed from a different angle, a consistent point can be made. After getting the binarization, the feature points are matched using the XOR operation, which improves the overall efficiency of the matching process.

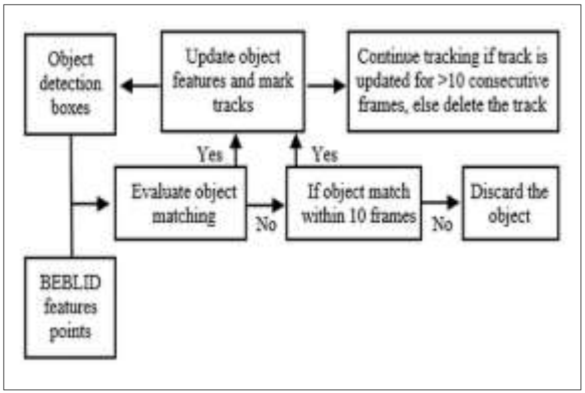

Object detection boxes
Update object features and mark tracks
Continue tracking if track is updated for >10 consecutive frames, else delete the track
Yes
Yes
Evaluate object matching
No
If object match within 10 frames
No
Discard the object
BEBLID features points

Fig. 6. Multiple Vehicle Object Tracking Method.

Fig. 6 illustrates the tracking method. When the number of matching points collected reaches a predefined threshold, the point is regarded successfully matched, and the object's matching box is painted around it. The following information relates to the source of the prediction box: Purification of feature points is performed using the Maximum Likelihood Estimator Sample Consensus (MLESAC) algorithm, which can exclude incorrect noise points caused by matching errors, and estimation of the homography matrix is performed using the MLESAC algorithm, which is capable of excluding incorrect noise points caused by matching errors. The estimated homography matrix and the location of the original object detection box are transformed into a perspective to get a matching prediction box for the original object detection box. Both the prediction box in the first frame and the detection box in the second frame must fulfil the centre point's criterion for the smallest distance between them to match the same item effectively. To be more specific, we define a threshold T equal to the greatest pixel change between the observed centre point of the vehicle object box and the vehicle object box's centre point when it moves between two subsequent video frames. The difference between two successive frames of the same vehicle in terms of positional movement is less than the threshold T. When the centre point of the vehicle object box crosses T in two subsequent frames, the vehicles in those two frames become unrelated, and the data connection fails. The threshold T value is proportional to the size of the vehicle object box, taking scale shift into vehicle. The thresholds for each vehicle object box are set to a variety of values. This definition is sufficiently flexible to accommodate vehicle mobility and a variety of different video input sizes. When T = box height/0.25 is used, the height of the vehicle object box is utilized as the input parameter for the calculation. We discard any trajectory that has not been updated in ten consecutive frames, which is suitable for a camera scene with a wide-angle image collection along the route under investigation. If the prediction box does not match the item in future frames, it is determined that the object is absent from the video scene and the prediction box is removed. The method outlined above results in the collection of global object identification and tracking trajectories from the viewpoint of the whole road surveillance video.

E. Analysis of Trajectories

This section discusses both the analysis of moving objects' trajectories and the gathering of data on numerous items in a traffic flow. The majority of roadways are split into two lanes, separated by isolation barriers. We identify the vehicle's orientation in the world coordinate system based on its tracking trajectory and mark it as approaching or fleeing the camera. A straight line is drawn across the traffic scene image to serve as a detection line for the purpose of calculating vehicle classification data. The detection line must be centred on the 1/2 point of the traffic image's high side. Concurrently, the road's traffic flow in both directions is counted. The object's memory is accessed when the object's trajectory crosses the detection line. The number of objects in different orientations and categories over a certain time may be calculated at the end of the operation.

V. Simulation and Results

Many measures have been developed in the past for evaluating the systems performance quantitatively. The proper one depends heavily on the application, and the search for a single, universal evaluation criterion is currently underway. On one side, it being ideal to condense results into a single number that can be compared directly. On the other side, one could not want to lose knowledge about the algorithms' specific faults and present a large number of performance estimations, which makes a clear voting impossible. So, we would be evaluating the performances with more than one parameter.

A. For Image Restoration

1) Peak signal to noise ratio(PSNR): Considering a reference image f and a test image g, which have a resolution of MxN, the PSNR score among f and g being calculated as:

$$PSNR(f,g) = 10log_{10}((\max \text{pixel value})^2/MSE(f,g) \quad (14)$$

$$where, MSE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_{ij}-g_{ij})^2 \quad (15)$$

The PSNR score increases as the mean squared error (MSE) decreases; this indicates that a greater PSNR value results in a higher image quality.

2) Structural similarity index (SSIM): The SSIM being a well-known quality statistic that is used to compare two images. It is thought to be connected to the human visual system's perception of quality. The SSIM score being calculated as:

$$SSIM(f,g) = l(f,g)c(f,g)s(f,g) \quad (16)$$

$$where, l(f,g) = \frac{2\mu_f\mu_g+C_1}{\mu_f^2+\mu_g^2+C_1} \quad (17)$$

$$c(f,g) = \frac{2\sigma_f\sigma_g+C_2}{\sigma_f^2+\sigma_g^2+C_2} \quad (18)$$

$$s(f,g) = \frac{\sigma_{fg}+C_3}{\sigma_f\sigma_g+C_3} \quad (19)$$

l: luminance, ***c***: contrast and ***s***: structural comparison function Few results of GAN framework for image restoration are shown in Fig. 7.

The images are randomly selected, and their performance is quantified in terms of PSNR and SSIM. The average of the two parameters' scores, is shown in Table II.

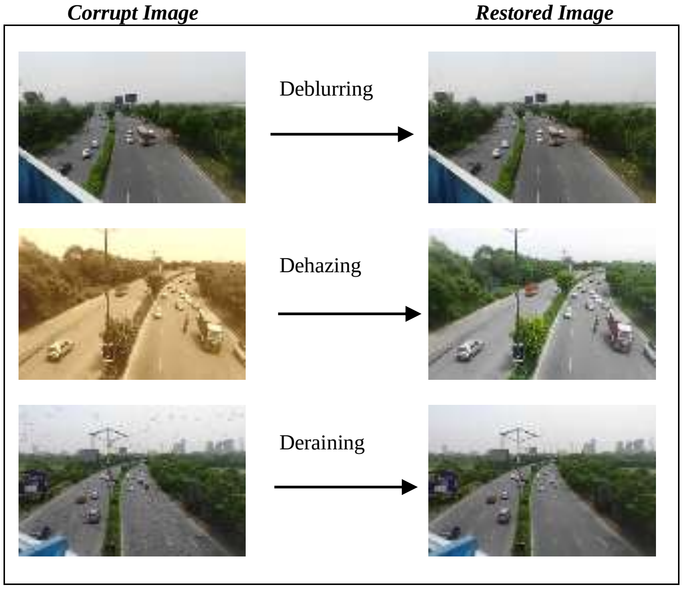


Fig. 7. Few Results of GAN Framework for Image Restoration.

TABLE II. PERFORMANCE EVALUATION OF IMAGE RESTORATION METHOD

Parameter [50]	**Input**	**Average Scores**		
		Deblurring	***Dehazing***	***Deraining***
PSNR	20.42	27.44	25.61	24.86
SSIM	0.5691	0.8811	0.9187	0.8367

B. For Vehicle Detection

It was necessary to use the test set to compute the mean average precision (mAP); mAP is an acronym for Average Precision (AP), which is defined as calculating the area under the precision-recall curve for a given total number of object class instances [43]. The experiment is divided into three classes, which include two-wheelers, light motor vehicles, and heavy motor vehicles. The mean of 11 points for each potential threshold in the category's precision/recall curve is described for each category by AP. We utilized a series of criteria [0, 0.1, 0.2,..., 1] to measure our results. For recall values larger than each threshold (in this experiment, the barrier is 0.25), there will be a matching maximum precision value, denoted by *pmax(recall)*. The precisions listed above are computed, and AP is the average of these 11 maximum precisions (recall). This number was used to describe the overall quality of our model.

$$AP = \frac{1}{11}\sum_{recall=0}^{1} p_{max}(recall), recall \in [0, 0.1, \dots, 1] \quad (20)$$

$$mAP = \frac{\sum AP}{class\ number} \quad (21)$$

The calculation of precision, recall and IoU (Intersection over union) is as follows:

$$Precision = \frac{TP}{TP+FP} \quad (22)$$

$$Recall = \frac{TP}{TP+FN} \quad (23)$$

$$IoU = \frac{area\ of\ overlap}{area\ of\ union} \quad (24)$$

in which TP, FN, and FP denote the number of true positives, false negatives, and false positives, respectively We used the following formulas to compute the parameter scores for both categories:

1) When the dataset was sent directly into the object detection algorithm, that is, when no image restoration procedure was used to restore the image.

2) When a picture is restored using the GAN framework, a dataset is fed into the object detection algorithm.

Tables III and IV provide the results of the parameters for each of the two categories. There is a 13.7 percent difference in the two-category results for the metric mAP when comparing them. This improvement figure clearly demonstrates that restoring the pictures has a significant influence on the quality of object identification and, indirectly, on the accuracy of tracking while tracking objects.

Few results of SSMD approach for categorical vehicle detection id depicted in Fig. 8.

C. Multiple Vehicle Object Tracking

The performance evaluation for multiple vehicle object tracking is done through following parameters [51]:

1) Multiple Object Tracking Accuracy (MOTA): This parameter takes into account three different types of errors: false positives, missed targets, and identity changes. For improved tracking accuracy, a high MOTA value is preferred. It is calculated as:

$$MOTA = 1 - \frac{\sum_t (\boldsymbol{FN_t + FP_t + IDSW_t})}{\sum_t \boldsymbol{GT_t}} \quad (25)$$

The frame index is ***t***, and the count of ground truth objects is GT. MOTA could be negative if count of mistakes produced by tracker is more than total object count in the scene. MOTA score being solid indicator of tracking system's overall performance.

2) Multiple Object Tracking Precision (MOTP): Refers to average difference between all true positives and their ground truth objectives. For improved tracking, a high MOTP value is preferred. Average dissimilarity among all true positives and their matching ground truth targets is Multiple Object Tracking Precision. This being calculated as, for bounding box overlap:

$$MOTP = \frac{\sum_{t,i} \boldsymbol{d_{t,i}}}{\sum_t \boldsymbol{c_t}} \quad (26)$$

$\boldsymbol{d_{t,i}}$ is the bounding box overlap of target $\boldsymbol{i}$ with its assigned ground truth object, and $\boldsymbol{c_t}$ is count of matches in frame $\boldsymbol{t}$. Average overlap among all properly matched hypotheses and their corresponding objects being given by MOTP, which spans among $\boldsymbol{t_d}$: 50% and 100%.

3) False Alarms per Frame (FAF): It reflects per-frame amount of false alarms. A lower value of FAF is desirable for better tracking.

4) Mostly Tracked (MT): It indicates the number of paths that have been mainly tracked. i.e. the target has had the same label for at least 80% of its existence. A high value of MT parameter is desirable for better tracking.

5) Mostly Lost (ML): It indicates the amount of trajectories that have been lost for the most part. i.e. the target being not monitored for at least 20% of the time it is alive. A lower value of ML parameter is desirable for better tracking.

6) False Positive (FP): It reflects number of false detections. A lower value of FP parameter is desirable for better tracking.

7) False Negative (FN): It reflects number of missed detections. A lower value of FN parameter is desirable for better tracking.

8) IDsw: The amount of times an ID changes to a formerly tracked object. A lower value of IDsw parameter is desirable for better tracking.

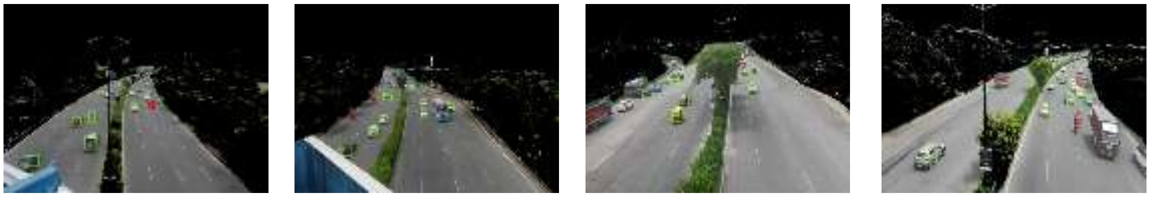
9) Frag: The amount of times a track is fragmented due to a miss detection. A lower value of Frag parameter is desirable for better tracking.

TABLE III. PERFORMANCE EVALUATION OF VEHICLE DETECTION METHOD[1]

Para-meter	AP(%)			Precision	Recall	Average IoU (%)	mAP (%)
	Two-wheeler	***LMV***	***HMV***				
Scores	68.4	72.6	71.1	0.66	0.71	62.41	70.7

TABLE IV. PERFORMANCE EVALUATION OF VEHICLE DETECTION METHOD[2]

Para-meter	AP(%)			Precision	Recall	Average IoU (%)	mAP (%)
	Two-wheeler	***LMV***	***HMV***				
Scores	84.7	87.5	87.1	0.86	0.88	73.64	84.4

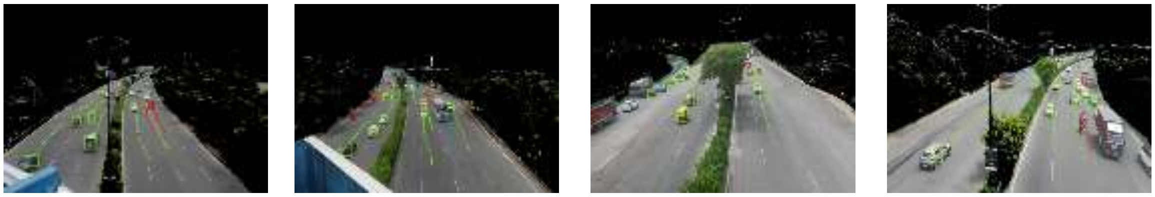


(a) View 1 (b) View 2 (c)View 3 (d) View 4

Fig. 8. Few Results of SSMD Approach for Categorical Vehicle Detection.

TABLE V. PERFORMANCE EVALUATION OF MULTIPLE VEHICLE OBJECT TRACKING METHOD

Parameter	**MOTA(↑)**	**MOTP(↑)**	**FAF(↓)**	**MT(↑)**	**ML(↓)**	**FP(↓)**	**FN(↓)**	**IDsw(↓)**	**Frag(↓)**
Scores	36.3	72.9	1.4%	13.4%	33.4%	140	304	35	28



(a) View 1 (b) View 2 (c)View 3 (d) View 4

Fig. 9. Few Results of Trajectory Estimation for Multiple Vehicle Object Tracking.

The score of the various tracking parameters is depicted in Table V. Trajectory estimation done on the dataset is depicted in Fig. 9. It summarizes the movement of vehicles with direction information and maps the future state predictions.

VI. CONCLUSION

This research developed from the standpoint of surveillance cameras, a dataset of vehicle objects and presented a technique for image restoration, object detection, and tracking for road traffic video scenes. The use of the GAN framework for picture restoration, as well as the GMM for road area extraction, resulted in a more effective detection system. The annotated road vehicle object dataset was used to train the SSMD object identification algorithm, which resulted in the development of an end-to-end vehicle detection model. The location of the object in the image being evaluated by the BEBLID feature extraction method based on results of the object detection technique and image data. The trajectory of the vehicle might thus be determined by tracking the binary characteristics of many objects. Lastly, the vehicle trajectories were examined to obtain information on the road traffic scene, such as driving direction as well as vehicle category and traffic density. Testing findings confirmed that suggested vehicle identification and tracking approach for road traffic scene has good performance and is practicable, as demonstrated by the outcomes of the experiments. The method described in this paper being low in cost and high in stability when compared to the traditional method of monitoring vehicle traffic by hardware. It also requires no large-scale construction or installation work on existing monitoring equipment, which is a significant advantage over the traditional method.

REFERENCES

[1] M. Won, T. Park, and S. H. Son, “Toward mitigating phantom jam using vehicle-to-vehicle communication,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 5, 2016, pp. 1313–1324.

[2] M. Won, S. Sahu, and K.-J. Park, “DeepWiTraffic: Low cost WiFi-based traffic monitoring system using deep learning,” arXiv:1812.08208 preprint, 2018.

[3] Myounggyu Won, “Intelligent Traffic Monitoring Systems for Vehicle Classification: A Survey,” IEEE Access, vol. 8, 2020, pp. 73340-73358.

[4] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, “Survey of unmanned aerial vehicles (UAVs) for traffic monitoring,” Handbook of unmanned aerial vehicles, pp. 2643–2666, 2015.

[5] Z. Chen, T. Ellis, and S. A. Velastin, “Vehicle detection, tracking and classification in urban traffic,” in International IEEE Conference on Intelligent Transportation Systems, 2012, pp. 951–956.

[6] C. M. Bautista, C. A. Dy, M. I. Mañalac, R. A. Orbe, and M. Cordel, “Convolutional neural network for vehicle detection in low resolution traffic videos,” in IEEE Region 10 Symposium (TENSYMP), 2016, pp. 277–281.

[7] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Sch¨olkopf, “A machine learning approach for non-blind image deconvolution,” in CVPR, 2013, pp. 1067–1074.

[8] L. Xu, J. S. J. Ren, C. Liu, and J. Jia, “Deep convolutional neural network for image deconvolution,” in NIPS, 2014, pp. 1790– 1798.

[9] J. Sun, W. Cao, Z. Xu, and J. Ponce, “Learning a convolutional neural network for non-uniform motion blur removal,” in CVPR, 2015, pp. 769–777.

[10] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Sch¨olkopf, “Learning to deblur,” IEEE TPAMI, vol. 38, no. 7, 2016, pp. 1439–1451.

[11] M. Hradis, J. Kotera, P. Zemc´ık, and F. Sroubek, “Convolutional neural networks for direct text deblurring,” in BMVC, 2015, pp. 6.1–6.13.

[12] S. Nah, T. Hyun Kim, and K. Mu Lee, “Deep multi-scale convolutional neural network for dynamic scene deblurring,” in CVPR, 2017, pp. 3883–3891.

[13] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, “Single image dehazing via multi-scale convolutional neural networks,” in ECCV, 2016, pp. 154–169.

[14] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, “Dehazenet: An end-to-end system for single image haze removal,” IEEE TIP, vol. 25, no. 11, 2016, pp. 5187–5198.

[15] K. He, J. Sun, and X. Tang, “Single image haze removal using dark channel prior,” in CVPR, 2009, pp. 1956–1963.

[16] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, “Automatic single-imagebased rain streaks removal via image decomposition,” IEEE TIP, vol. 21, no. 4, 2012, pp. 1742–1755.

[17] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” in ICCV, 2015, pp. 3397–3405.

[18] Y.-L. Chen and C.-T. Hsu, “A generalized low-rank appearance model for spatio-temporally correlated rain streaks,” in ICCV, 2013, pp. 1968–1975.

[19] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, “Rain streak removal using layer priors,” in CVPR, 2016, pp. 2736–2744.

[20] D. Eigen, D. Krishnan, and R. Fergus, “Restoring an image taken through a window covered with dirt or rain,” in ICCV, 2013, pp. 633–640.

[21] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in CVPR, 2016, pp. 770–778.

[22] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” in CVPR, 2017, pp. 3855–3863.

[23] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, “Generative adversarial nets,” in NIPS, 2014, pp. 2672–2680.

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired imageto-image translation using cycle-consistent adversarial networks,” in ICCV, 2017, pp. 2223–2232.

[25] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” in ICML, 2017, pp. 1857–1865.

[26] Z. Yi, H. Zhang, P. Tan, and M. Gong, “Dualgan: Unsupervised dual learning for image-to-image translation,” in CoRR, 2017, pp. 2849–2857.

[27] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang, “Multi-class generative adversarial networks with the L2 loss function,” CoRR, vol. abs/1611.04076, 2016.

[28] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, “Photo-realistic single image super-resolution using a generative adversarial network,” in CVPR, 2017, pp. 4681–4690.

[29] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, “Deblurgan: Blind motion deblurring using conditional adversarial networks,” in CVPR, 2018, pp. 8183–8192.

[30] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Husz´ar, “Amortised MAP inference for image super-resolution,” in ICLR, 2017.

[31] X. Xu, D. Sun, J. Pan, Y. Zhang, H. Pfister, and M.-H. Yang, “Learning to super-resolve blurry face and text images,” in ICCV, 2017, pp. 251–260.

[32] Jinshan Pan, Jiangxin Dong, Yang Liu, Jiawei Zhang, Jimmy Ren, Jinhui Tang, Yu-Wing Tai, and Ming-Hsuan Yang, “Physics-Based Generative Adversarial Models for Image Restoration and Beyond,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020.

[33] Al-Smadi M., Abdulrahim K., and Salam R.A., “Traffic surveillance: A review of vision based vehicle detection, recognition and tracking,” International Journal of Applied Engineering Research, vol. 11, no. 1, 2016, pp. 713–726.

[34] Radhakrishnan M., “Video object extraction by using background subtraction techniques for sports applications,” Digital Image Processing, vol. 5, no. 9, 2013, pp. 91–97.

[35] Qiu-Lin L.I., and Jia-Feng H.E, “Vehicles detection based on three-frame-difference method and cross-entropy threshold method,” Computer Engineering, vol. 37, no. 4, 2011, 172–174.

[36] Liu Y., Yao L., Shi Q., and Ding J., “Optical flow based urban road vehicle tracking,” IEEE Conference on Computational Intelligence and Security, 2013.

[37] Park K., Lee D., and Park Y., “Video-based detection of street-parking violation,” IEEE Conference on Image Processing,” vol. 1, 2007, pp. 152–156.

[38] Girshick R., Donahue J., Darrell T., and Malik J., “Rich feature hierarchies for accurate object detection and semantic segmentation,” IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[39] Uijlings J.R.R., Van de Sande K.E.A., Gevers T., and Smeulders A.W.M., “Selective search for object recognition,” International Journal of Computer Vision, vol. 104, no. 2, 2013, pp. 154–171.

[40] Kaiming H., Xiangyu Z., Shaoqing R., and Jian S., “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 37, no. 9, 2014, pp. 1904–16.

[41] Liu W., Anguelov D., Erhan D., Szegedy C., Reed S., Fu C.Y., and Berg A.C., “SSD: Single shot multibox detector,” European conference on computer vision, 2016, pp. 21–37.

[42] Redmon J., Divvala S., Girshick R., and Farhadi A., “You Only Look Once: Unified, real-time object detection,” IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[43] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg, “SSD: Single Shot MultiBox Detector,” 2016.

[44] Luo W., Xing J., Milan A., Zhang X., Liu W., Zhao X., and Kim T.K., “Multiple object tracking: A literature review,” arXiv:1409.7618 preprint, 2014.

[45] Xing J., Ai H., and Lao S., “Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses,” IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1200–1207.

[46] Zhou H., Yuan Y., and Shi C., “Object tracking using sift features and mean shift,” Computer Vision & Image Understanding, vol. 113, no. 3, 2009, pp. 345–352.

[47] Rublee E., Rabaud V., Konolige K., and Bradski G.R., “ORB: An Efficient Alternative to SIFT or SURF,” International Conference on Computer Vision, 2011.

[48] Iago Su´areza, Ghesn Sfeira, Jos´e M. Buenaposadac, and Luis Baumela, “BEBLID: Boosted Efficient Binary Local Image Descriptor,” Pattern Recognition Letters, 2020.

[49] Zezhi Chen, Tim Ellis, and Sergio A Velastin, “Vehicle Detection, Tracking and Classification in Urban Traffic,” IEEE Conference on Intelligent Transportation Systems, Alaska, USA, 2012.

[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity”, IEEE Transactions on Image Processing, vol. 13, no. 4, 2004, pp. 600-612.

[51] K. Bernardin and R. Stiefel hagen, “Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics,” Image and Video Processing, 2008.

[52] D. Sharma, Z. A. Jaffery and N. Ahmad, "Categorical vehicle classification using Deep Neural Networks," International Conference on Power Electronics, Control and Automation, 2019, pp. 1-6.