Journal of Information Systems and Telecommunication
Vol.10, No.3, July-September 2022, 201-210

http://jist.acecr.org
ISSN 2322-1437 / EISSN:2345-2773

Edge Detection and Identification using Deep Learning to Identify Vehicles

Zohreh Dorrani[1], Hassan Farsi[1], Sajad Mohammadzadeh[1 *]

[1]. Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran

Received: 23 June 2021/ Revised: 20 Nov 2021/ Accepted: 11 Dec 2021

Abstract
A deep convolution neural network (CNN) is used to detect the edge. First, the initial features are extracted using VGG-16, which consists of 5 convolutions, each step is connected to a pooling layer. For edge detection of the image, it is necessary to extract information of different levels from each layer to the pixel space of the edge, and then re-extract the feature, and perform sampling. The attributes are mapped to the pixel space of the edge and a threshold extractor of the edges. It is then compared with a background model. Using background subtraction, foreground objects are detected. The Gaussian mixture model is used to detect the vehicle. This method is performed on three videos, and compared with other methods; the results show higher accuracy. Therefore, the proposed method is stable against sharpness, light, and traffic. Moreover, to improve the detection accuracy of the vehicle, shadow removal conducted, which uses a combination of color and contour features to identify the shadow. For this purpose, the moving target is extracted, and the connected domain is marked to be compared with the background. The moving target contour is extracted, and the direction of the shadow is checked according to the contour trend to obtain shadow points and remove these points. The results show that the proposed method is very resistant to changes in light, high-traffic environments, and the presence of shadows, and has the best performance compared to the current methods.

Keywords: Deep Convolution Neural Network; Edge Detection; Gaussian Mixing Method; Vehicle Detection.

1- Introduction

Better monitoring of traffic flows is essential to reduce accidents. Therefore, the primary purpose of this article is to provide better traffic monitoring. For the traffic monitoring program, fixed cameras are generally used to the static background (e.g., fixed surveillance cameras) and a common background subtraction approach is employed to obtain an initial estimate of moving objects. The reported solutions for traffic focus on optimization because real-time and convenient data collection methods are essential and promise better traffic monitoring that provides more detail to understand traffic flow patterns. The primary purpose of this article is to create an algorithm that can count vehicles to better monitor traffic. Various tasks such as edge detection, background subtraction, and threshold techniques are performed to provide appropriate video-based surveillance techniques. Edges are one of the simplest and most important parts of image processing. If the edges of an image are correctly specified, the location of all objects in the image is specified, and some of their properties such as surface and environment are specified, and can be easily measured [1, 2]. Therefore, edge detection can be an essential tool in vehicle detection. In an image, edges define the boundary between an object and the background or the boundary between overlapping objects. There are many methods for the edge extraction and detection that differ in finding the right edges [3]. These methods intermittently assess various factors such as light intensity, camera type, lens, motion, temperature, atmospheric effects, and humidity in detecting edges, and then detect vehicles [4,5]. In this paper, the deep learning method is used to distinguish the edges. This algorithm can detect strong and smooth edge information, which increases the accuracy of vehicle detection compared to methods that do not use edge detection. The task of vehicle detection is based on the edge detection theory in image processing, which offers a new algorithm for monitoring traffic using vehicle counting. Vehicle counting is carried out by subtracting the background. Initially, a reference framework is used to extract background information. Hence when a new object

🖂 **Sajad Mohammadzadeh**
s.mohamadzadeh@birjand.ac.ir

enters the frame, it is detected by subtracting the background. Background subtraction is performed when the image is part of a video stream and shows the difference in pixel locations that have changed in two frames. Background information is identified using the reference frame as the background model. Video stream are recorded and tested with the proposed algorithm. Some conditions make vehicle detection more difficult. One of these challenges is environmental conditions, such as the presence of clouds of dust. In addition, the presence of shadow, which is part of the vehicle and makes it difficult to detect is another critical problem that reduces the detection accuracy.

The incomplete appearance of a moving vehicle or even a considerable distance between the vehicle cameras, with very low-resolution images, are also among the detection problems. Therefore, to show the stability of the proposed method and its accuracy, various videos that have these conditions have been used for the simulation, and the final results are very satisfactory. Using the deep learning method in edge detection and then using the edges to detect the vehicle increases the accuracy and makes the algorithm resistant to the problems expressed. The proposed algorithm uses several steps to increase accuracy. Thus, the deep CNN is described. In the next part, this network is used to extract the edges, and after that, it is explained how the vehicle is detected and counted. The results are given in the next section, and finally, the conclusion is presented.

2- Related Works

At present, vehicle detection is performed using the traditional machine vision method and the complex deep learning method. Traditional methods of machine vision use the movement of a vehicle to separate it from a still image. Vehicles can be identified using features such as color, pixels, and edges, along with some machine learning algorithms. Some detectors can place and classify objects in video frames using visual features. Among the various methods proposed in this field are the Haar Cascade method [6], the You Only Look Once (YOLO) method [7], the Single Shot MultiBox Detector (SSD) method [8], and the Mask R-CNN [9]. Haar Cascade, initially proposed by Viola and Jones [10], refers to set visual features that can take advantage of rectangular areas in a particular location, pixel intensities and differences between regions. This method is a robust classifier that moves the search window across the image to cover all areas and identify objects of different sizes and has been used to detect vehicle traffic.

The use of deep convolutional networks has led to great success in vehicle detection. These networks have a high ability to learn image features, and perform several related tasks such as classification and finite box regression, as shown in Figure 1.

The YOLO detector consists of a CNN that has 24 layers of convolution for feature extraction and two dense layers for prediction.

In this research, a CNN was proposed based on vehicle detection, and the work is novel in the following three ways, compared to prior studies.

1- This is the first time edge extraction is performed with VGG-16, and used for vehicle detection. One of the features of VGG-16 is that it has a very good architecture for measuring a specific task.

2. Another aspect is the use of edge detection in the algorithm, which leads to increased speed and reduced computations.

3. In addition, removing shadows to increase accuracy can be another aspect of this article's innovation.

Major headings are to be column centered in a bold font without underline. They need be numbered. "2. Headings and Footnotes" at the top of this paragraph is a major heading.

3- CNN

CNN is a particular type of deep learning method. This type of network calculates the output used for subsequent units using the input data. Today, artificial intelligence has grown significantly to increase accuracy and comfort. In this regard, various algorithms and networks have been proposed and utilized. One of the most famous networks developed in the field of deep learning is CNN. The main strength of CNNs is providing an efficient compact network that can predict or identify. A huge data set applied to CNNs, and larger data is thought to require more accuracy.

CNN has the power to detect distinctive features of images alone, without real human intervention. The purpose of using CNNs is to predict and categorize various databases without human intervention. A CNN consists of neural layers with weight and bias, which are capable of learning. Artificial neurons are processing units that can receive inputs, an internal multiplication is done between the neurons' weights and the inputs, and the result is biased. Finally, a nonlinear function (the same as the activation function) is passed [19].

The convolution layer is the core of the CNN in which most calculations are performed. Each convolution layer contains filter sets, each filter has a specific pattern, and the output of the convolution layer is a set of different patterns, so the number of filters indicates the same number of detected features. The output is generated from the convolution of filters and the input layer.

The convolution operator is one of the most essential components that make CNN resistant to spatial variation.

By layering or padding, the input size is increased, and thus, the output matrix is equal to the input matrix in size. A simple, and common way to layer is to add zero rows, and columns symmetrically around the input matrix. In this case, the convolution filter has more space for stepping and scanning, leading to a larger output.

The size of the stride is also defined in the step convolution layer. The meaning of the step is that the filter, after calculation in an input window, has to move a few doors or houses to perform the calculations again. Similar to other neural networks, CNN [20] uses the excitation function after the convolution layer. Using a nonlinear function creates a nonlinear property in the neural network, which is very important. Defining a nonlinear function separately from the canonization layer provides more flexibility. Of all the nonlinear functions, the ReLU function is the most popular. There are other members of the ReLU family, such as PreLU, and Leaky-ReLU, among others. The polishing layer is another important layer in the CNN. The purpose of the polishing layer is to reduce the spatial size of the feature map obtained by using the convolution layer. The polishing layer has no teachable parameter and performs a simple, and effective sampling. Pulling has a convolution-like function, and a window moves on the image. The most common examples of pooling are max pooling and average pooling. Max polishing is usually used in the middle layers, and at the end of the grid, the average polishing layer is typically used. Usually, the last layers of a CNN for classification are the Fully Connect layers. These layers are the same layers that exist in the MLP neural network. One of the main applications of the fully connected layer in the convolutional network is to be used as a classifier. That is, the set of features extracted using convolutional layers are eventually transformed into a vector. Finally, this attribute vector is passed to a fully connected classifier to identify the correct class.

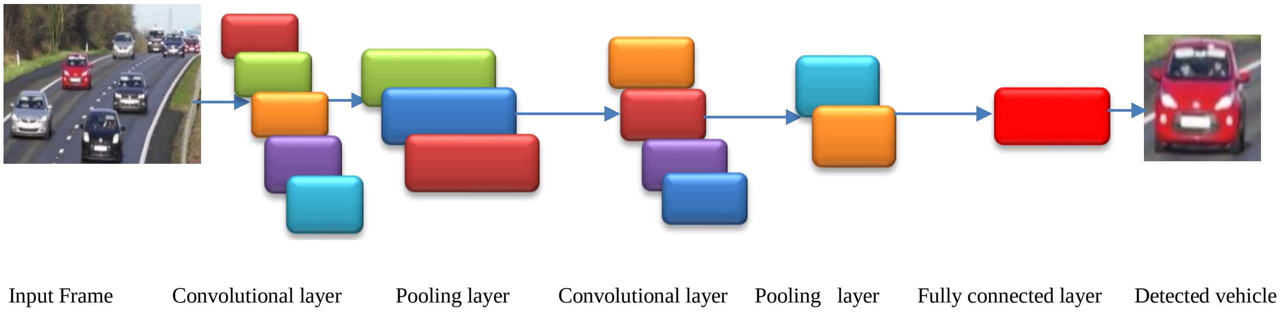


Fig. 1. Demonstration of algorithm and performance of convolution neural network in vehicle detection.

4- The proposed Method

4-1- Edge Detection using a Deep Convolutional Network

Initially, the extraction of basic features is provided, for which VGG-16 [21], VGG-19 [22], ResNet-50 [23], ResNet-101 [24], etc. can be used. Although the accuracy of the VGG-19 and ResNet series is higher, we use VGG-16 due to the need for much more network parameters. The main VGG16 consists of 5 convolutions, each step is connected to a pooling layer with stride = 2, and each step contains a different level of information. To find the edges of the image, it is necessary to extract information of different levels from each step layer to the pixel space of the edge. The next block handles feature retrieval and sampling. Thus, features can be mapped to the edge pixel space. The conv1_2 layer is a 1 × 1 × 1 convolution layer that reduce the size and integration of features. The re-extraction block of the three convolution layers is 1 × 1 × 32, 3 × 3 × 32, 1 × 1 × 28, and so on. At the end of the grid, a 1 × 1 kernel convolution layer is used to generate the final edge recognition image.

In an artificial neural network [25-27], each neuron is connected to all the neurons in the adjacent layer. The output y of each latent neuron is obtained from the sum of each of the weighted inputs w · x plus a bias b from the following equation [25]:

$$y = \mathrm{w} \cdot \mathrm{x} + \mathrm{b} \qquad (1)$$

Neural network training is mainly based on the diffusion cost function to set the parameters. The purpose of reducing the cost of the parameters is mainly: the weight of the connection between the nerve cells and the bias of each neuron. Setting is to use the gradient descent algorithm to adjust the size of the parameters along the slope direction. For this purpose, the entropy reduction, and weighting functions are used, which are added according to the following equation [1].

$$C = -[tln(a) + (1 - t)\ln(1 - a)] \qquad (2)$$

$$a = sigmoid(y) = \frac{1}{1 + e^{-y}}$$

C is the reciprocal of the loss of a pixel, and t is the target value; also, a is calculated by the sigmoid equation function. The output y of each latent neuron is calculated from the sum of each of the weighted inputs w · x plus a bias b from Equation 1.

Threshold step: By introducing a threshold and comparing it with matrix pixels, the edge points can be obtained. All pixels are divided into edge and non-edge pixels. The initial matrix T is equal to the mean value of the matrix. The points of the pheromone matrix are then categorized into two groups based on whether their value is less or more than T. The new threshold is obtained by averaging the two mean values of each group. The above steps continue until the threshold value changes too much [2]

$$T = \frac{\sum_{k=1:M_1} p_{k.l}^{(N)} \sum_{l=1:M_2} p_{k,l}^{(N)}}{M_1 M_2} \qquad (3)$$

P is the probability. $M_1$and M_2 are the dimensions of the image. The final matrix is divided into two groups; the first group contains all the pixels less than T. The second group contains all the pixels greater than T. A binary decision-maker is then applied to each the pixel to determine ifthe point is on edge ($E_{k,l}$=1) or not ($E_{k,l}$= 0) [2].

$$E_{k.l} = \begin{cases} 1 & p_{k,l}^{(N)} \geq T \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

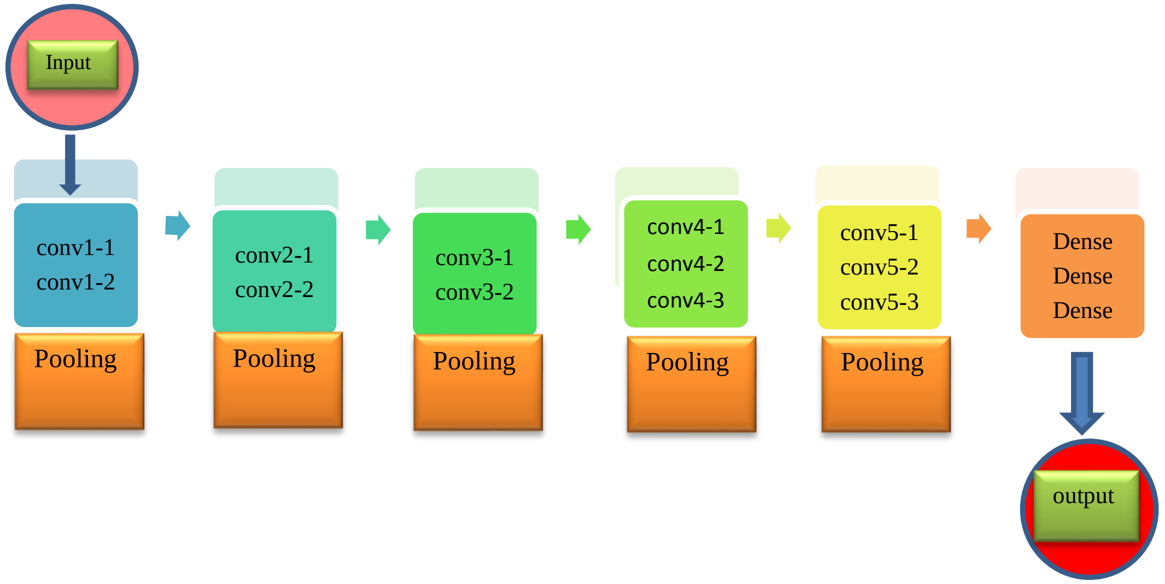


Fig. 2. VGG 16.

4-2- Vehicle Detection and Counting

To provide a monitoring and traffic system with reasonable accuracy and speed after edge detection, the field can be used to extract vehicles, according to Figure 3. In this case, it is compared to a background model to determine if the individual pixels are part of the background or foreground. Then a foreground mask is calculated. Using background subtraction, foreground objects are detected.

Shadow removal is also required to improve vehicle detection accuracy. The method of combining color and contour features is suitable for identifying shadows. For this purpose, a moving target is extracted. The connected domain is marked to be compared with the entire foreground area. The moving target contour is extracted, and the shadow direction is assessed according to the contour trend. Hence, the grain shadow points are obtained

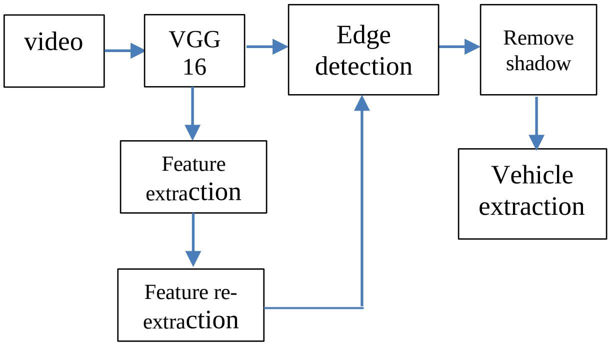


Fig. 3. The overall flow of the method

based on the contour direction, and the found points are removed.

The Gaussian mixture model is used to detect the vehicle. The model needs to be updated for fast and continuous detection. The system is highly resistant to changes in light, high or low speeds, high-traffic environments, and the entry or removal of objects from the scene. In this model, the distribution B is used for the covariance of the function resulting from the density function of the observation probability of the pixel value for the image.

$$B = argmin_b(\sum_{k=1}^{b} W_k > T) \qquad (5)$$

T is the threshold value and W is the Gaussian weight. Weight values are updated. Structural features are extracted, and used for identification. The pseudocode of the proposed vehicle detection is presented.

The pseudocode for the vehicle detection:

```
START
STEP 1: Read Video
STEP 2: Extract a video frame
STEP 3: Convert into gray level image
STEP 4: the extraction of basic features with VGG-16
    the input training data set is set to $\mathbf{s} = \{(\mathbf{x_n}.\mathbf{z_n})\}$,

STEP 5: Threshold step:
Old T
calculate threshold $\mathbf{T} = \frac{\sum_{\mathbf{k=1:M_1}} \mathbf{p_{k.l}^{(N)}} \sum_{l=1:M_2} p_{k,l}^{(N)}}{\mathbf{M_1 M_2}}$,
stopping Condition( $|\mathbf{T} - \mathbf{oldT} < \boldsymbol{\varepsilon}|$)
All pixels are divided into edge and non-edge pixels with
$\mathbf{E_{k.l}} = \begin{cases} \mathbf{1} & \mathbf{p_{k,l}^{(N)}} \geq \mathbf{T} \\ \mathbf{0} & \mathbf{otherwise} \end{cases}$
STEP 6: Shadow removal
STEP 7: detect the vehicle with The Gaussian mixing
    method
END
```

5- Results

5-1- Dataset

1) To find the effectiveness of the proposed method, the simulations have been conducted and evaluated on images from Berkeley Segmentation Dataset.

2)The first dataset is Road-Tracing Monitoring [28]. This dataset detects vehicles on a single route and consists of three videos in separate frames. The "M-30" movie, contains 7520 frames recorded on a sunny day with a resolution of 640 × 480 pixels. The second movie "M-30-HD", contains 9390 frames recorded in the same place as the previous movie but on a cloudy day and, with a higher resolution of 1280 × 720. The third set of "Urban1" contains 23,435 frames with a low resolution of 480 × 320. This dataset enables the evaluation of vehicle detectors.

3) To evaluate the vehicle detection performance, the CDnet2014 benchmark dataset proposed in [29] was used. The CDnet2014 dataset includes 11 video categories with 4-6 movie sequences per group. We use the baseline video that shows a highway.

5-2- Evaluation Criteria

1) Entropy: One of the criteria for evaluating the output performance is the Shannon entropy function [30]:

$$H(I) = -\sum_{i=0}^{L} q_i \, log q_i \qquad (6)$$

where I and q_i are the desired image and the frequency of the pixel i, respectively.The purpose of this criterion is to measure information. Higher entropy values are due to greater randomness and less information. This criterion can describe the complexity of a part of the image, and the difference between the pixel and the entropy value is positively correlated. The amount of dynamic background entropy or edge of the object is high.

2) The correct metric detections: One of the criteria for evaluating the correct detector is in the extraction of the vehicle; for this purpose, the identified vehicles can be compared with the real vehicle. The result of this evaluation has a significant impact on all the subsequent stages of processing. The criterion for correctly identifying vehicles is calculated for each frame. The evaluator is dependent on the parameters P and G and is expressed by the following relation [31]:

$$J(G.P) = \frac{|G \cap \mathrm{P}|}{|G \cup \mathrm{P}|}. \qquad (7)$$

Parameters P and G are vehicle detected, and real, respectively.

3) Calculation time: Another evaluation criterion is the calculation time which is the time required to process one frame per replication, and can be used for evaluation in online applications.

4) Accuracy: The accuracy criterion is obtained by the following equation [31]:

$$accuracy = \frac{(TP\ +\ TN)}{(TP\ +\ TN\ +\ FP\ +\ FN)}. \qquad (8)$$

P is true positive samples, TN is true negative samples, FP is false-positive samples and FN is false-negative samples.

4) Average precision: Using the formula provided by Equation 6, average precision is calculated:

$$\gamma = \frac{TP}{TP\ + TN +\ FP}. \tag{9}$$

5) F-measure: This criterion is a popular metric used to quantify the vehicle detection performance and is determined using the following relation [32]:

$$\eta = \frac{TN}{TN\ +\ FP}. \tag{10}$$

$$\phi = \frac{TP}{TP +\ FP}.$$

$$F\text{-}measure = \frac{2\eta\phi}{\eta +\ \phi}.$$

5-3- Deep Learning Edge Detection (DLED)

Initialization of the model is performed using a pre-trained VGG-16 model. To run and view the results, the algorithm is first implemented on the Berkley segmentation dataset.

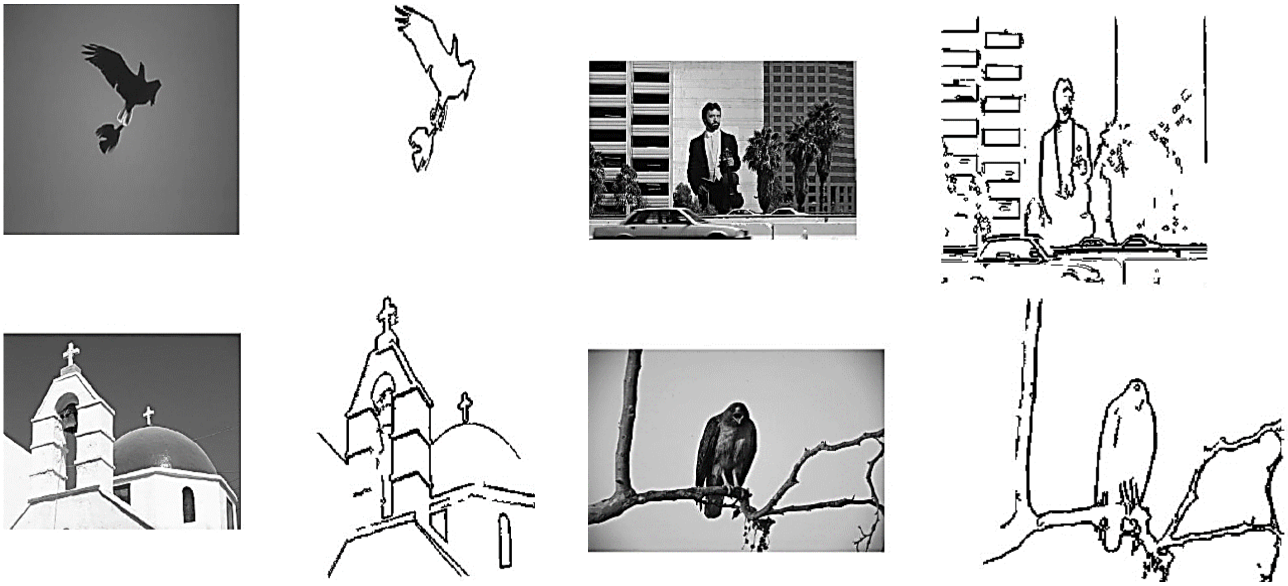


Fig. 4. Results of images edge detection on Berkley segmentation dataset.

The continuous, accurate, and precise edges in the output show the excellent performance of the proposed method.

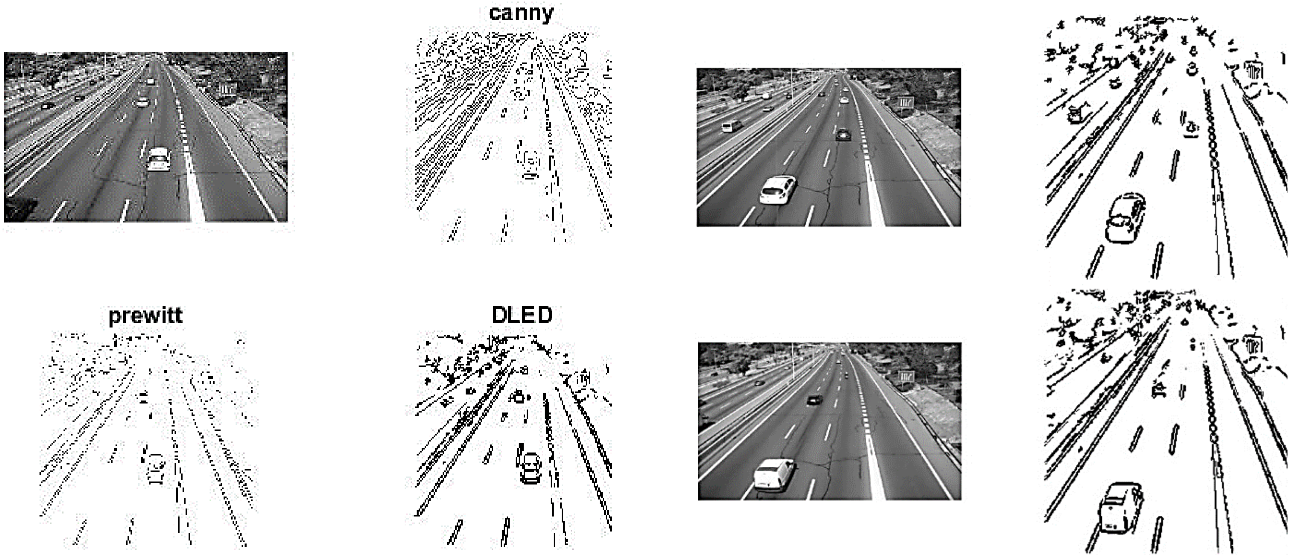


Fig. 5. Edge detection with canny, Prewitt, DLED.

Then, the algorithm is implemented on the Road-Traffic Monitoring dataset [25]. To assess edge detection efficiency, the output of the detected edges is compared with that of robust edge detectors such as CANNY and Prewitt, the results of which are shown in Figure 5.

Table 1: Entropy values for the GRAM images.

Image	**Canny**	**Prewitt**	**DLED**
M-30	0.5385	0.4183	**0.3996**
M-30-HD	0.2815	0.2784	**0.2612**
Urban1	0.4916	0.4625	**0.4501**

Table 1 indicates the entropy values for the images in the dataset studied dataset.

Higher entropy values are due to greater randomness and less information. The proposed method has the lowest value and can detect essential edges.

Table 2 shows the mean values of the edge accuracy on several images in the Berkley segmentation dataset, which are compared with several methods for image edge detection.

Table 2. γ values for the outputs of some of the edge detectors on Berkley segmentation dataset.

Image	**GA[11]**	**PSO[12]**	**ACO [13]**	**Deep Learning[14]**	**BFA[15]**	**Fuzzy+BFO[16]**	**Neuro-Fuzzy[17]**	**PSO for noisy[18]**	**Proposed method**
35010	0.15	0.24	0.07	0.32	0.26	0.30	0.06	0.23	**0.35**
42049	0.11	0.28	0.09	0.31	0.12	0.29	0.08	0.30	**0.33**
118035	0.19	0.14	0.22	0.07	0.25	0.06	0.24	**0.27**	0.12
135069	0.21	0.12	0.21	0.06	0.26	0.07	0.22	**0.33**	0.15
119082	0.22	0.15	0.10	0.27	0.18	0.30	0.09	0.23	**0.31**

For the accuracy to be high, the number of false-positive samples must be reduced, thus decreasing the level of positive sensitivity. High accuracy is required in identifying objects. According to the values in this table, the average accuracy calculated in three images has the highest value. However, in two images, the method mentioned in reference [18] is more accurate.

According to the results in Table 2, the proposed method has achieved the highest value of γ for three images of the Berkley segmentation dataset and more than other methods. Table 3 displays the entropy values for the outputs of some of the edge detectors. The proposed method has the minimum value. A higher value of entropy relates to more uncertainty and less information.

Table 3. The entropy values for the outputs of some of the edge detectors on Berkley segmentation dataset.

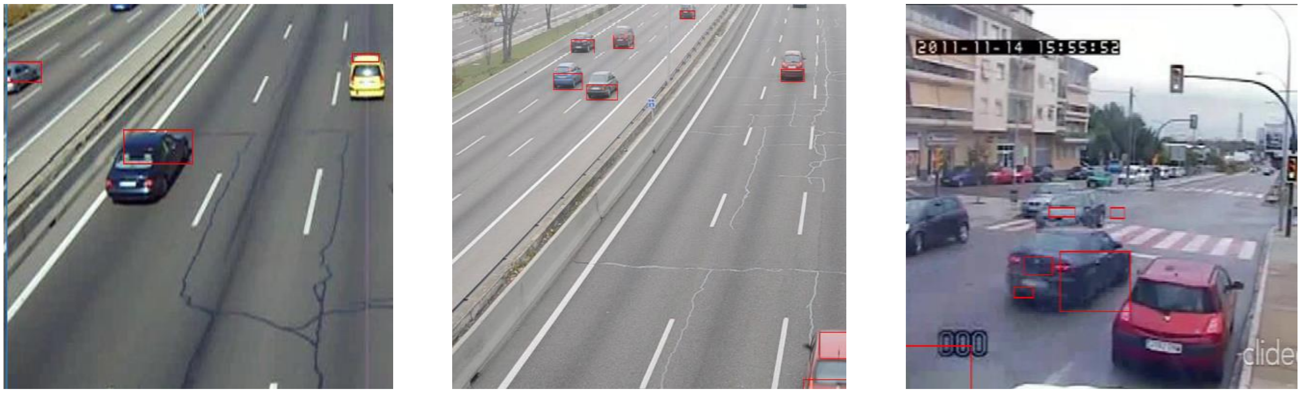
Image	**GA[11]**	**PSO[12]**	**ACO [13]**	**Deep Learning[14]**	**BFA[15]**	**Fuzzy+BFO[16]**	**Neuro-Fuzzy[17]**	**PSO for noisy[18]**	**Proposed method**
35010	0.8211	0.7213	0.7715	0.6882	1.6124	0.6331	0.6110	0.6995	**0.5125**
42049	0.8322	0.6811	0.7722	0.6243	1.5216	0.5999	0.6561	0.6778	**0.3798**
118035	0.8836	0.6992	0.7765	0.6836	1.4245	0.5876	0.5788	0.6476	**0.3819**
135069	0.9214	0.8112	0.8833	0.7210	1.2124	0.6675	0.6689	0.7889	**0.4089**
119082	0.9914	0.8365	0.8987	0.7991	1.4642	0.7999	0.7989	0.8999	**0.5592**

Table 3 shows that the lowest entropy value is obtained in the proposed method, which is valid for all the images in this dataset. This method can find meaningful edges.

It is observed that the best score of this value is achieved by the proposed method, followed by the Neuro-Fuzzy method [17] however, the proposed method has a better performance with the lowest entropy.

According to the values given in this table, the highest entropy value is related to the Bacterial Foraging Algorithm (BFA).

Deep learning has significant advantages compared to traditional edge detection algorithms. The proposed method shows higher performance compared to those methods.



M-30 M-30-HD Urban1

Fig. 6. vehicle detection with the proposed method on Road-Traccing Monitoring.

Table 4 reports the vehicle diagnostic results in the Road-Traffic Monitoring GRAM dataset.

Table 4. Comparison of processing time performance and vehicle detection accuracy in object detection in the GRAM dataset.

		M-30	**M-30-HD**	**Urban1**
Haar Cascade[6]	Time [s]	**0.08–0.13**	0.3–0.44	0.02–0.06
	Accuracy	43%	75%	40%
SSD[8]	Time [s]	4–7	11–14	2.6–5.6
	Accuracy	22%	70%	**69%**
YOLO v3[7]	Time [s]	1.0–1.8	1.0–1.8	1.0–1.8
	Accuracy	82%	86%	91%
Mask R-CNN[9]	Time [s]	2.4–3.0	2.4–3.0	2.4–3.0
	Accuracy	89%	91%	46%
Proposed method	Time [s]	2.3-2.8	2.3-2.8	2.3-2.8
	Accuracy	**91%**	**93%**	54%

The results show that the fastest detector is Haar Cascade, but it offers a maximum accuracy of 75%. The proposed method is the best in terms of accuracy.

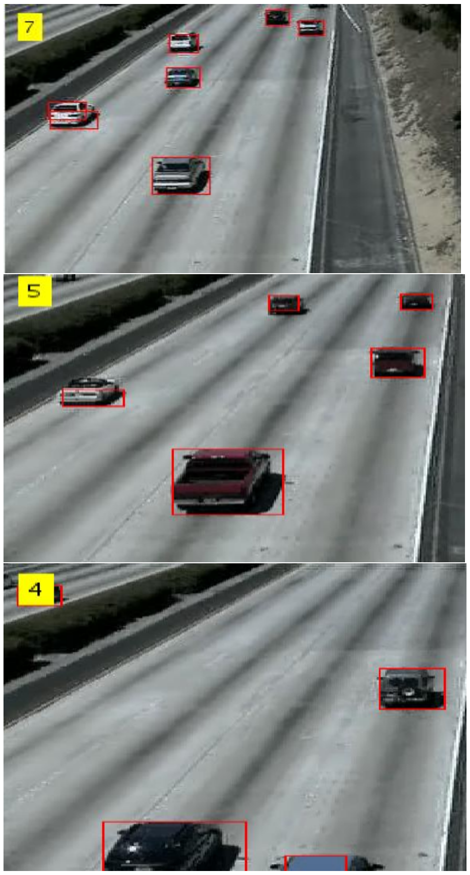


Fig. 7. Execute the proposed method on the images of the shaded vehicle.

In traffic image analysis, shadows are sometimes identified as the background. Therefore, shadows also play a significant role, and during sunny days, it may be difficult to successfully identify the vehicle. To evaluate the success of vehicle identification in this situation, we use the baseline video, which shows the highway and vehicles in this video have shadows. The results of the diagnosis are shown in Figure 7. The findings show that this method is also resistant to shadows and can yield excellent results.

Finally, two criteria on the CDnet 2014 dataset were compared with another method, which is presented in Figure 8.

The results obtained using the proposed method demonstrate that the values of the evaluated criteria have improved compared to the other seven methods, especially in terms of "F-measure". Auspicious results promise higher accuracy and higher performance.

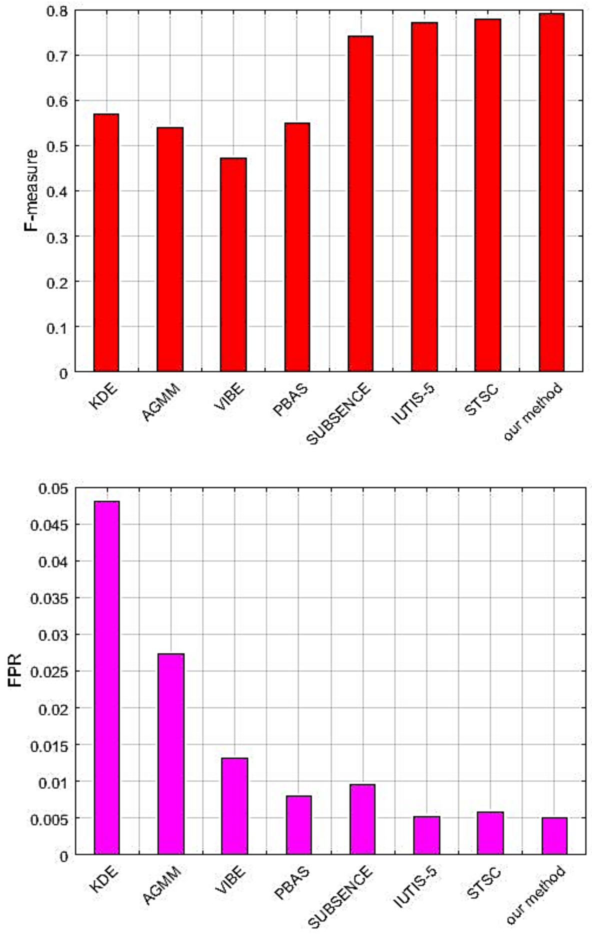


Fig. 8. Evaluation Metrics of Seven Methods on CDnet 2014 Dataset.

6- Conclusions

In this paper, for vehicle detection, first, the deep edge CNN detection is performed and then a comparison with a background model is conducted. Using background subtraction, foreground objects are detected. The Gaussian mixed method is employed to detect the vehicle, which must be updated for rapid and continuous detection. Three different videos are selected in terms, of weather conditions, traffic load, and resolution. The first video is on a sunny day. The second video shows the same place but on a cloudy day and with a higher resolution. The third video is in low resolution and displays the same street. The results are compared with those of several other methods, which shows the higher accuracy of the results of the proposed method. This method is very resistant to entering or removing objects from the scene.

In future research, other neural network architectures can be used to increase accuracy. On the other hand, the proposed method can be utilized to add other purposes such as vehicle classification, traffic classification, and speed detection.

References

[1] Z. Qu, SY. Wang, L. Liu, and DY. Zhou, "Visual Cross-Image Fusion Using Deep Neural Networks for Image Edge Detection", IEEE Access., Vol. 7, No. 1, 2019, pp. 57604-57615.

[2] Z. Dorrani, and M. S. Mahmoodi, "Noisy images edge detection: Ant colony optimization algorithm", Journal of AI and Data Mining, Vol. 4, No. 1, 2016, pp. 77-83.

[3] SM. Ismail, LA. Said, AG. Radwan, and AH Madian. "A novel image encryption system merging fractional-order edge detection and generalized chaotic maps, Signal Processing, Vol. 167, No.1, 2020, pp. 107280.

[4] N. Balamuralidhar, S. Tilon, and Francesco Nex. "MultEYE: Monitoring System for Real-Time Vehicle Detection, Tracking and Speed Estimation from UAV Imagery on Edge-Computing Platforms", Remote Sensing Vol. 13, No. 4, 2021, pp. 573.

[5] B. Wang, LL. Chen, ZY. Zhang, "A novel method on the edge detection of infrared image", Optik, Vol. 180, No.1, 2019, pp. 610-614.

[6] S. Choudhury, S.P. Chattopadhyay, and T.K. Hazra, "Vehicle detection and counting using haar feature based classifier", In receding's of the 8th Annual Industrial Automation and Electromechanical Engineering Conference, 2017, Vol. 8, pp. 106–109.

[7] J. Sang, Z. Wu, P. Guo, H. Hu, H. Xiang, Q. Zhang, Cai, B. "An Improved YOLOv2 for Vehicle Detection", Sensors, Vol.18, No. 12, 2018, pp. 4272.

[8] W.U. Qiong, and L. Sheng-bin, "Single Shot MultiBox Detector for Vehicles and Pedestrians Detection and Classification", in 2nd International Seminar on Applied Physics, Optoelectronics and Photonics Inc. Lancaster, 2017, Vol. 2, pp. 1-7.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal

networks", IEEE Trans. Pattern Anal, Vol. 39, No.6, 2017, pp. 1137–1149.

[10] A. Dubey, and S. Rane, "Implementation of an intelligent traffic control system and real time traffic statistics broadcasting", In Proceedings of the International conference of Electronics, Communication and Aerospace Technology, 2017, Vol.1, pp. 33–37.

[11] SM. Bhandarkar, Y. Zhang, and WD. Potter, "An edge detection technique using genetic algorithm-based optimization", Pattern Recognit, Vol. 27, No. 9, 1994, pp.1159–1180.

[12] R. Chaudhary, A. Patel, S. Kumar, and S. Tomar, "Edge detection using particle swarm optimization technique", International Conference on Computing, Communication and Automation IEEE, 2017, Vol.1, pp.363–367.

[13] DS. Lu, CC. Chen, "Edge detection improvement by ant colony optimization", Pattern Recognit Lett. Vol. 9, No. 4, 2008, pp. 416–425.

[14] S. Xie, and Z. Tu. "Holistically-nested edge detection", In Proceedings of the IEEE international conference on computer vision, 2015, Vol. 125, pp. 3–18.

[15] S. Rajaraman, and A. Chokkalinga, "Chromosomal edge detection using modified bacterial foraging algorithm", Int J Bio-Science Bio-Technology, Vol. 6, No. 1, 2014, pp. 111–122.

[16] OP. Verma, M. Hanmandlu, AK. Sultania, and AS. Parihar, "A novel fuzzy system for edge detection in noisy image using bacterial foraging", Multidimens Syst Signal Process, Vol. 24, No. 1, 2013, pp.181–198.

[17] ME. Yüksel, "Edge detection in noisy images by neuro-fuzzy processing", Int J Electroni Commun, Vol. 61, No. 2, 2007, pp. 82–89.

[18] M. Setayesh, M. Zhang, and M. Johnston, "Edge detection using constrained discrete particle swarm optimization in noisy images", IEEE Congress of Evolutionary Computation (CEC), 2011, pp. 246–253.

[19] Y. Wang, Y. Li, Y. Song, and X. Rong, "The influence of the activation function in a convolution neural network model of facial expression recognition", Applied Sciences, Vol. 10. No. 5, 2020, pp. 1897.

[20] A. Sezavar, H. Farsi, and S. Mohamadzadeh. "Content-based image retrieval by combining convolutional neural networks and sparse representation", Multimedia Tools and Applications, Vol. 78, No. 15, 2019, pp. 20895-20912.

[21] Z. Song, L. Fu, J. Wu, Z. Liu, R. Li, and Y. Cui, "Kiwifruit detection in field images using Faster R-CNN with VGG16", IFAC-Papers On Line, Vol. 52, No. 30, 2019, pp. 76-81.

[22] S. Manali, and M. Pawar, "Transfer learning for image classification", Second International Conference on Electronics, Communication and Aerospace Technology, 2018, Vol. 2, pp. 656-660.

[23] W. Long, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on ResNet-50", Neural Computing and Applications, Vol. 32, No. 1, 2019, pp. 1-14.

[24] P. Ghosal, L. Nandanwar, and S. Kanchan, "Brain tumor classification using ResNet-101 based squeeze and excitation deep neural network", Second International Conference on Advanced Computational and Communication Paradigms (ICACCP). IEEE, 2019, Vol.2, pp. 1-6.

[25] A. Sezavar, H. Farsi, and S. Mohamadzadeh. "A modified grasshopper optimization algorithm combined with CNN for content based image retrieval", International Journal of Engineering 32.7, 2019, 924-930.

[26] R. Nasiripour, H. Farsi, S. Mohamadzadeh. "Visual saliency object detection using sparse learning", IET Image Processing, Vol. 13, No. 13, 2019, pp. 2436-2447.

[27] H. Zamanian, H. Farsi, and S. Mohamadzadeh. "Improvement in Accuracy and Speed of Image Semantic Segmentation via Convolution Neural Network Encoder-Decoder", Information Systems & Telecommunication, Vol. 6, No. 3, 2018, pp. 128-135.

[28] R. Guerrero-Gomez-Olmedo, and RJ Lopez-Sastre, "Vehicle tracking by simultaneous detection and viewpoint estimation. "International Work-Conference on the Interplay Between Natural and Artificial Computation", Springer, 2013, pp. 306-316.

[29] Y. Wang, P.M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An expanded change detection benchmark dataset", inProc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2014, pp. 393–400.

[30] Z. Dorrani, H. Farsi, and S. Mohamadzadeh. "Image Edge Detection with Fuzzy Ant Colony Optimization Algorithm", International Journal of Engineering, Vol. 33, No. 12, 2020, pp. 2464-2470.

[31] D. Impedovo, F. Balducci, V. Dentamaro, and G. Pirlo, "Vehicular traffic congestion classification by visual features and deep learning approaches: a comparison", Sensors, Vol. 19, No. 23, 2019, pp. 5213-5225.

[32] C. Wen, P. Liu, W. Ma, Z. Jian, C. Lv, and J. Hong, "Edge detection with feature re-extraction deep convolutional neural network", Journal of Visual Communication and Image Representation, Vol. 57, No. 1, 2018, pp. 84-90.