JJEE
Jordan Journal of Electrical Engineering

Volume 8, Number 2, June 2022
Pages 165-178
ISSN (Print): 2409-9600, ISSN (Online): 2409-9619

Deep Learning in Vehicle Detection Using ResUNet-a Architecture

Zohreh Dorrani[1], Hassan Farsi[2*], Sajad Mohamadzadeh[3]

[1, 2, 3] Department of Electrical and Computer Engineering, University of Birjand, Birjand, Iran
E-mail: hfarsi@birjand.ac.ir

Received: December 07, 2021 *Revised: January 23, 2022* *Accepted: January 31, 2022*

Abstract— Vehicle detection is still a challenge in object detection. Although there are many related research achievements, there is still a room for improvement. In this context, this paper presents a method that utilizes the ResUNet-a architecture – that is characterized by its high accuracy - to extract features for improved vehicle detection performance. Edge detection is used on these features to reduce the number of calculations. The removal of shadows by combining color and contour features - for increased detection accuracy - is one of the advantages of the proposed method and it is a critical step in improving vehicle detection. The obtained results show that the proposed method can detect vehicles with an accuracy of 92.3%. This - in addition to the obtained F-measure and η values of 0.9264 and 0.8854, respectively - clearly state that the proposed method - which is based on deep learning and edge detection - creates a reasonable balance between speed and accuracy.

Keywords— Vehicle detection; ResUNet-a; Convolutional neural network; Deep learning.

1. INTRODUCTION

One of the critical components of an intelligent city is automated traffic management. Vehicle detection and counting in video surveillance scenes are important for intelligent traffic management and control, leading to better monitoring of traffic flow to reduce accidents [1, 2]. A vehicle detection system can quickly and simultaneously perform several valuable tasks such as: i) detecting the number of vehicles at intersections, roads, streets and other locations; ii) the amount of traffic during the nights/ days; iii) hours of traffic generation and iv) type of vehicles and similar issues [3].

If such information is available, many other problems can be solved such as optimizing traffic and distributing it through another street, thinking about parking space and numbers, understanding and studying traffic patterns and implementing the most efficient traffic management method. In addition, innovative technology will be applied to other systems such as toll collection and parking access control [4].

Human can quickly and instantly distinguish objects from complex scenes; but translating this through process using a machine requires learning the art of recognizing objects using computer vision algorithms. The main goal is to provide a system for identifying and detecting vehicles efficiently and accurately so that vehicles can be counted by creating an algorithm for better traffic monitoring [5].

The use of deep learning and its flagship, i.e., the deep convolution neural network (CNN), is widely used in various fields [6-9]. CNN provide the best results in diagnosing hand position [9]. These grids have improved the accuracy and speed of semantic image segmentation [8]. These applications have also been used in edge detection [10, 11].

[*] Corresponding author

Article's DOI: 10.5455/jjee.204-1638861465

Better monitoring of traffic flows is essential to reduce accidents. To achieve this, we must improve vehicle detection methods. This is even more important for detecting unusual conditions such as clouds, dust mites, light changes, shadows, poor camera quality and poor vehicle appearance. There are many diagnostic methods in this field that can be used for different purposes.

This paper uses the ResUNet-a [12] architecture to extract feature maps. The edges of the image are detected using these features. This algorithm can detect solid and smooth edge information, which increases the accuracy of vehicle detection compared to methods that do not use edge detection [13]. Also, shadow removal is done in the next step, which can play an essential role in improving the performance of the presentation method. The following describes the beginning of the ResUNet-a architecture and then the steps. After describing how to identify the vehicle, the results, and finally the conclusion are drawn.

2. RELATED WORKS

Deep learning is one of the methods that can be useful in solving many problems including vehicle detection.

In the SSD-VGG16 method [14] - the feature maps from the shallow layer with more location information, feature maps from the intermediate layer and feature maps from the deep layer with rich semantic information - were interconnected. This led to better detection performance in small objects.

You only look once (YOLO) method for vehicle detection was reported in literature. For this purpose, an improvement of the YOLOv3 model was presented in [15]. The number of network layers was reduced and used to detect low-resolution vehicles in infrared aerial images. To improve vehicle tracking, the YOLOv4 detector [16, 17] was introduced along with some existing methods such as prediction optimization. Also, using YOLOv4+HNEM as a detection model using secondary transfer learning from visible datasets, a reasonable degree of accuracy was achieved [18].

Haar cascade refers to a set of visual features that can use rectangular areas in a specific location. Therefore, pixel intensity and differences between areas were used to detect vehicle traffic [19]. CNN used in the single shot multibox detector (SSD), was successfully implemented for vehicle detection [20]. The R-CNN mask - which is an evolved R-CNN that uses a selective search method - resulted in very few calculations [21].

Other algorithms - consisting of vehicle detection through combined applications of road marking features – were designed for single goal detection [22]. Since there is no common interaction between the objects, it was difficult to identify an object relatively simply.

Chong et al. used the multi-step method for vehicle detection [23]. The bottom shadow is extracted to adjust the region of interest of the vehicle and the average energy models and marginal areas were created.

Chang et al. proposed a vision-based vehicle detection system by creating a cascade of powerful classifiers for vehicle detection trained in response to changes in traffic environments [24].

The aforesaid methods introduced algorithms in some specific applications. These methods have lower accuracy. Shadow was not evaluated in many of these methods.

A two-step detection algorithm based on a combination of Harr and Histogram of oriented gradients features was proposed in [25]. This algorithm takes advantage of target vehicles, and the prospect region of interest can be extracted using the Harr attributes.

Some methods use edge specifications for detection. There are many methods for edge detection, that are based on particle swarm optimization algorithm (PSO) [26], deep learning method [27], bacterial foraging optimization algorithm (BFO) [28] and its fuzzy [29], PSO algorithm for noise images [30], genetic algorithm [GA] [31, 32], ant colony optimization (ACO) [33, 34], and neuro-fuzzy processing [35]. Deep learning has significant advantages compared to traditional edge detection algorithms.

Compared to the existing edge detection methods, this paper proposes a method that is unique in terms of its usage of ResUNet-a architecture – that is characterized by its high accuracy - for vehicle detection. Edge detection is used in the proposed method's algorithm to increase its speed and reduce computations. Moreover, the proposed method has the ability to removes shadows which is expected to increase accuracy

3. PROPOSED METHOD

3.1. Architecture of ResUNet-a

The ResUNet-a architecture model demonstrates the better performance of the current segmentation architectures of the deep CNN [12]. UNet block is the backbone and has a symmetrical and expansive direction for the accurate location of objects. The structure of the ResUNet-a consists of two parts: the encoder and the decoder as shown in Fig. 1.

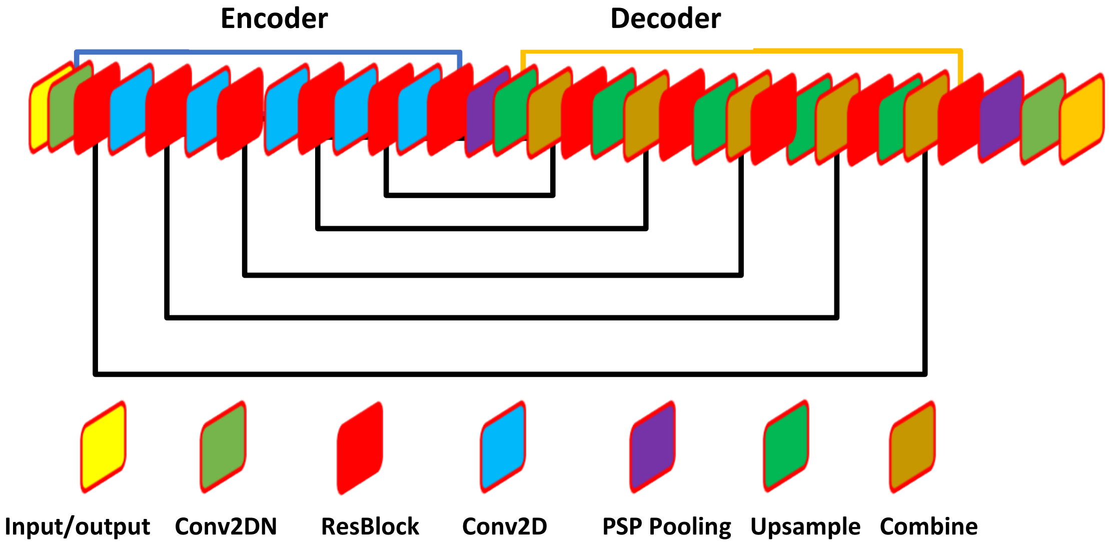


Fig. 1. Architecture of ResUNet-a.

The encoder includes the input layer, Conv2D layer, Conv2DN layer and ResBlock-a layer. The input layer receives the input image. Conv2D layer is a standard 2D convolution layer with kernel size equal to 3 and padding that equals 1. The Conv2DN layer is a 2D convolution layer in which the kernel size is 3 and padding is 1, followed by a batch of a normalized layer. ResBlock layer follows the physiology of the remaining units (the units placed between the connections,) i.e., between the first and last layers. Instead of having a single residual branch consisting of two consecutive convolution layer branches, there are up to four parallel branches so that the input is processed simultaneously in several fields of view. The feature map size determines the input property of the dilation rate.

The decoder includes PSP Pooling layer, upsample layer, combine layer and output layer. Pooling layer scans a pyramid that delivers information using maximum pooling at four different scales. The first scale is maximum pooling global. In the second scale, the feature map is divided into four equal areas and maximum pooling is performed in each area. The same applies to the subsequent two scales. As a result, this layer encapsulates feature information.

The up-sample layer contains a prototype of the feature maps, which is then followed by the Conv2DN layer. The size of the feature map is doubled and the number of filters in the feature map is halved. The combine layer receives two inputs identical to the number of filters in each feature map. This layer connects them and outputs the same number of filters as the input feature maps.

The output layer is a multitasking layer that produces a total of four output layers. These layers consist of the extent mask, the boundary mask, the distance mask and the reconstructed image of the input image.

3.2. ResUNet-a for Vehicle Detection

The pseudocode for the proposed method is:

```
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 ResUNet-a, the input training data set is set to $s=\{(x_n.z_n)\}$
STEP 5: The edge detection step:
STEP 5-1: The feature re-extraction and sampling and pixels are divided into edge and non-edge pixels with:
$E_{k.l} = \begin{cases} 1 & p_{k,l}^{(N)} \geq T \\ 0 & \text{otherwise} \end{cases}$
STEP 5-2: Threshold step:
The extraction of threshold with Pirahansiah's single threshold algorithm
While increasing PSNR>0.1 is true do
  r = [a, b]. (in the first step a = 0 and b = 255)
  Find the mean and standard deviation for all of the pixels in  the image range r
  The mean value of range [$t_2$, b) is set as the threshold value of the partial range.
  a = $t_1$+1, b = $t_2$-1
end while
$t_1$ = mean, $t_2$ = mean+1
Obtain new image with multilevel threshold
STEP 6: Shadow removal
STEP 7: detect the vehicle with the Gaussian mixing method
END
```

Fig. 2 exhibits the block diagram of the proposed method.

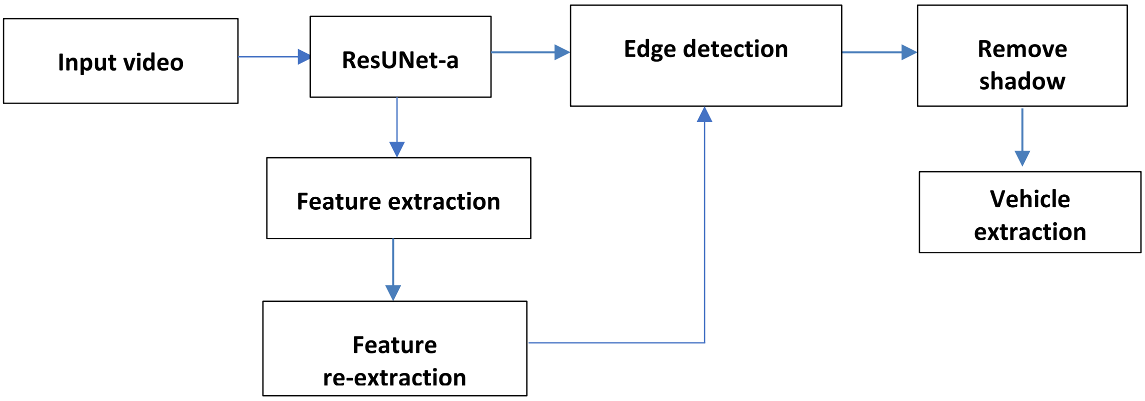


Fig. 2. Block diagram of the proposed method.

ResNet series is used to extract features because the accuracy of this architecture is high. To detect 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 re-extraction and sampling. So, features can be mapped to the edge pixel space. For this purpose, we need to introduce a threshold and compare it with the matrix pixels to reach the edge pixels. All pixels are divided into edge and non-edge pixels. Threshold can increase the reliability of extracting the edge. This is especially important in cases where there is noise because, in many cases, the noise is confused with the edge. Threshold is a problem that needs to be addressed in applications of pattern recognition methods. In addition, it has a significant important impact on the later stages of computer vision, such as optical character recognition, image splitting, object detection and tracking especially in vehicle detection. Thresholds can also be used as a way to separate the foreground from the background. It is essential to choose appropriate threshold value. If the threshold value is too low, the noise will be extracted as an edge, and when the threshold value is high, fewer edges will be detected. To prevent such a problem, the optimal threshold is used instead of a fixed threshold. Using a single threshold means using a value of thresholds that turns the image into a black-and-white image. Using a binary image is effective in increasing the computation speed or reducing the storage space. Detection accuracy can be increased using a maximum of a suitable threshold value. The single threshold condition is as follows [10]:

$$E_{k.l} = \begin{cases} 1 & p_{k,l}^{(N)} \geq T^s \\ 0 & p_{k,l}^{(N)} \leq T^s \end{cases} \tag{1}$$

where $P_{k,l}^{(N)}$ and T^s are the are the values of pixel k,l and threshold, respectively. A threshold algorithm based on the single threshold method of Pirahansiah's single threshold algorithm was used [36]. First, the peak signal-to-noise ratio (PSNR) value is calculated for each pixel of the image. The initial matrix is equal to the mean value of the matrix. Two random values were selected, and the mean value was divided by these two values. If the obtained PSNR value is between these two values, then this value is selected as the threshold.

The same Pirahansiah's single threshold method can be used to separate foreground and background to provide a traffic and accuracy system with reasonable accuracy and speed. In this case, this model specifies that the individual pixels are part of the background or foreground.

Shadow removal is a critical step in improving vehicle detection. Therefore, to increase the accuracy of vehicle detection, it is necessary to remove the shadow. Several methods can remove the shadow: chromaticity, physical, geometry and textures. All shadow recognition approaches have different contributions, and can have strengths and weaknesses. Some methods are accurate but cannot be generalized to different environments. The physical method improves the color accuracy of shadow models by local adaptation, but fails when the spectral properties of objects are similar to the background. The small-area texture-based method is powerful for pixels whose neighborhoods have textures, but it takes more time to implement, and is computationally more expensive. Therefore, the texture of the large area gives the method the most accurate results, but has a significant computational load.

In order to remove the shadows in the proposed method, the color combination and contour properties are used to identify the shadow. According to the background extraction in the previous steps, the moving object, which is the vehicle, is extracted. At this stage, the moving object is detected along with the shadow. The connected domain is marked to be compared to the entire foreground area. The moving target contour is extracted, and the shadow direction is checked according to the contour trend. A weighted image hypergraph is subdivided into hypergraphs using color and location. Then, the pixels in these sub-hypergraphs are classified as shadow or non-shadow pixels based on color and local gradient information. Eventually, the points detected as shadows are removed.

The Gaussian mixing modeling is used to detect the vehicle. This method is one of the image processing methods that leads to better results of vehicle detection when using the detection method with deep learning.

4. EXPERIMENTAL RESULTS

In this section, the performance of the proposed method is described. For this purpose, we need videos that can test the proposed method’s performance on problematic factors such as weather conditions, traffic, shadows and misdiagnosis of other moving objects.

One of the criteria for evaluating the output performance of edge detection with deep learning is the Shannon entropy function [37]. The goal of this criterion is to measure information. Higher entropy values are due to greater randomness and less information. This criterion can describe with [10]:

$$H(I) = -\sum_{i=0}^{L} q_i \log q_i \tag{2}$$

where I and q_i are the desired image and the frequency of the pixel i, respectively.

To find the effectiveness of the edge detection with ResUNet-a, the simulations have been done and evaluated on images from Berkeley segmentation dataset exhibited in Table 1. This database is a collection of images that contains an empirical basis for research on image segmentation and boundary recognition**.**

The obtained values show – as depicted in Table 1 - that the lowest entropy value is obtained in the edge detection with ResUNet-a, which is valid for all images in this dataset. This method can find meaningful edges. In [27], a deep learning method is used to find the edge whose entropy values are higher because using architecture ResUNet-a can achieve better results.

Table 1. The entropy values for the outputs of some of the edge detectors on Berkley segmentation dataset.

Edge detection method	Image				
	35010	42049	118035	135069	119082
PSO [26]	0.7213	0.6811	0.6992	0.8112	0.8365
Deep learning [27]	0.6882	0.6243	0.6836	0.7210	0.7991
BFO [28]	1.6124	1.5216	1.4245	1.2124	1.4642
Fuzzy+BFO [29]	0.6331	0.5999	0.5876	0.6675	0.7999
PSO for noisy [30]	0.6995	0.6778	0.6476	0.7889	0.8999
GA [31]	0.8211	0.8322	0.8836	0.9214	0.9214
ACO [32]	0.7715	0.7722	0.7765	0.8833	0.8987
Neuro-Fuzzy [33]	0.6110	0.6561	0.5788	0.6689	0.7989
Edge detection with ResUNet-a	0.4527	0.528	0.4852	0.5385	0.6522

Road-traffic monitoring datasets may be appropriate for this purpose. This dataset is used to detect vehicles and consists of three videos in separate frames. The movie "M-30" is on a sunny day with medium resolution. The second movie "M-30-HD" is in the same place as the previous movie but on a cloudy day. The third set is "Urban1" on a low-resolution street. The baseline video also uses the CDnet2014 dataset that shows a highway [38]. Fig. 3 demonstrates the part of the detection results of the proposed method. In this figure, the green boxes represent the positive sample accurately detected by the method.

Fig. 3(b) shows detected edges of the frame. In Fig. 3(c), a vehicle with an incomplete appearance is seen, which is shown in Figs. 3(d) and 3(e) as the proposed method recognizes it. Fig. 3(f) shows that moving vehicles were detected, but parked vehicles were not detected. Figs. 3(g) and 3(h) show that it can make a correct detection even in the presence of shadows.

From Fig. 3, it can be concluded that this method has succeeded in detecting vehicles in: i) cloudy conditions, ii) incomplete appearance of the car and iii) the presence of traffic in vehicles where vehicles are stacked and only part of the car is visible in the image. This method does not detect stationary vehicles, such as those parked on the side of the road.

Fig. 4 shows the shadow removal function in the proposed method. This video contains a background shadow, which is combined with the shadow in one of the vehicles as an example. The simulation result presented on this figure shows that the shadow removal was performed correctly.

The vehicle discrimination rate η, the vehicle detection rate ϕ, and the F-measure popular metrics were used to evaluate vehicle detection performance quantitatively.

The criteria used for the study are expressed by the following equations [18]:

$$accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{3}$$

$$\eta = \frac{TN}{TN + FP}. \tag{4}$$

$$\phi = \frac{TP}{TP + FP}. \tag{5}$$

$$F\text{-}measure = \frac{2\eta\phi}{\eta + \phi}. \tag{6}$$

where TP is true-positive samples, TN is true-negative, FP is false-positive and FN is false-negative. F-measure is a popular metric used to quantify vehicle detection performance.

Table 2 demonstrates the vehicle detection performance of the proposed method compared to the state-of-the-art methods.

From the results shown in Table 2, it can be observed that the proposed method – compared to other methods - has achieved the highest values of F-measure and η.

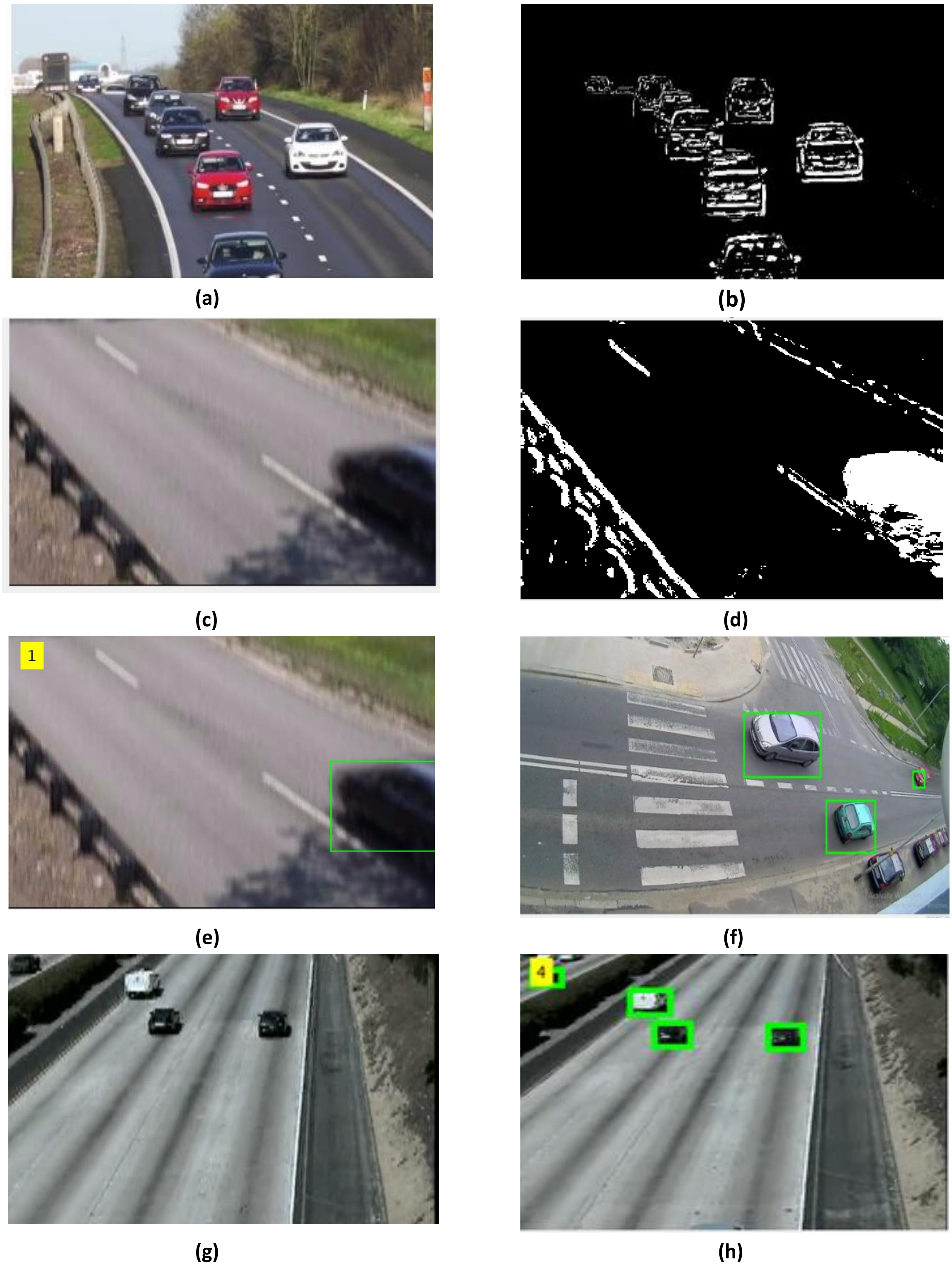
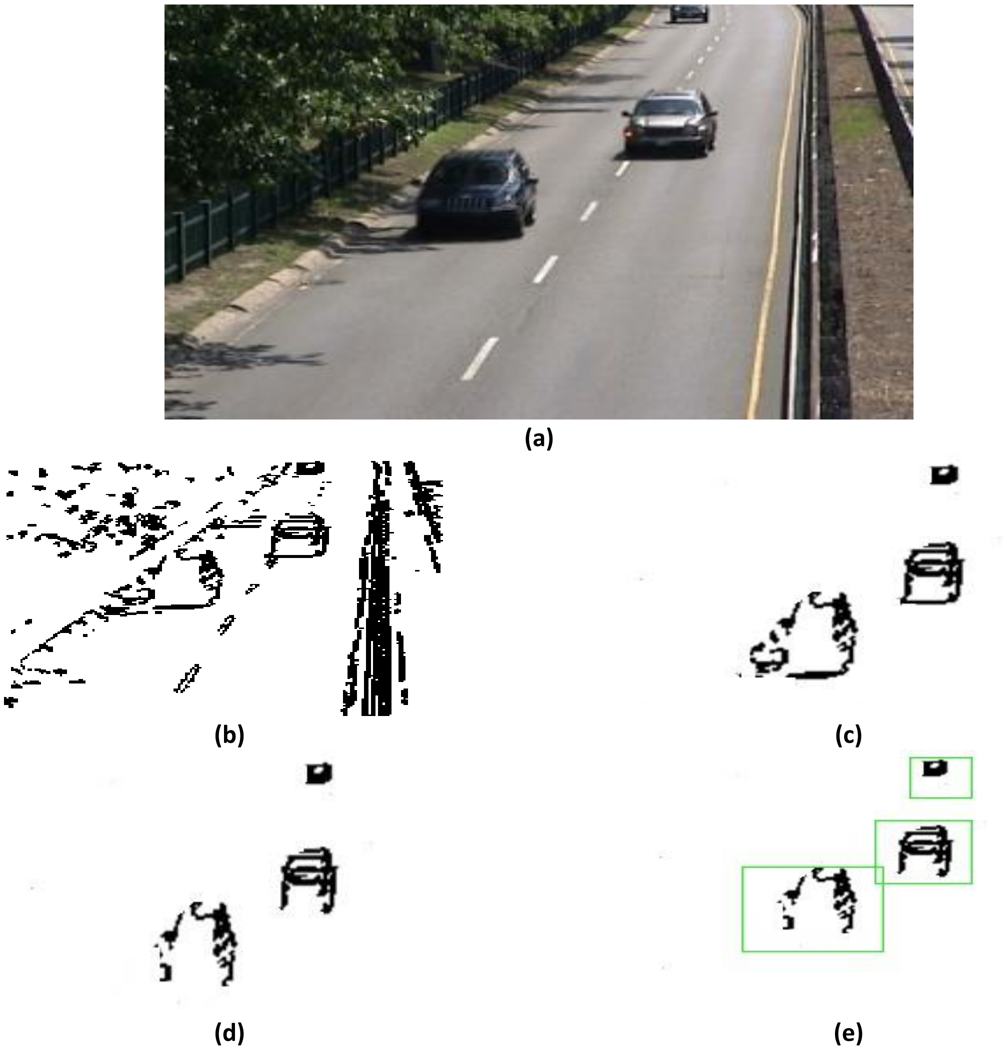


Fig. 3. Detection results using the proposed method: a) frame of the video; b) detection for vehicles in a; c) imperfect vehicle; d) detection for vehicle in c; e) vehicle detection on d; f) failure to detect parked vehicles by the proposed method; g) existence of shadows; h) vehicle detection on g.



(a)

(b)

(c)

(d)

(e)

Fig. 4. Shadow removal using the proposed method: a) frame of the video with shadow; b) edge detection for a; c) background removal; d) shadow removal; e) vehicle detection.

Table 2. Vehicle detection performance of the proposed method against state-of-the-art methods.

Method	F-measure	η	ϕ
SSD-VGG16 [19]	0.7603	0.7406	0.7811
YOLOv4 [21]	0.7792	0.7406	0.8220
YOLOv3 [21]	0.8192	0.8251	0.7992
YOLOv4+HNEM [26]	0.8798	0.8106	0.9621
Active Learning [3]	0.8300	0.7430	0.9400
Gabor+SVM [22]	0.8524	0.7742	0.9481
Vehicle shadow +ROI entropy [23]	0.7852	0.6737	0.9410
Harr+online boosting [24]	0.8125	0.7042	0.9600
Harr and HOG features[25]	0.8521	0.7539	0.9796
Proposed method	0.8992	0.8369	0.9715

The value of F-measure has increased because the value of η has increased compared to the other methods. The amount of ϕ has decreased compared to the method of Harr and HOG features, but increasing the amount of η - compared to this method - has led to an improvement in F-measure.

After the proposed method, the method of YOLOv4+HNEM features has the highest values with F-measure = 0.8798, η = 0.8106 and ϕ = 0.9621. The results obtained using the proposed method show that the values of the evaluated criteria have been improved compared to the rest of methods. The criterions of F-measure and η values have reached 0.9264 and 0.8854, respectively. The proposed method achieves ϕ = 0.9715 compared with ϕ = 0.9796 of the Harr and HOG method. This criterion is very close to the value in the method of Harr and HOG which has the highest value.

Table 3 displays the time and accuracy values for the road-traffic monitoring GRAM dataset. In this table, another criterion that was considered is the computational time which is evaluated by adding a value processing time for each video frame in each iteration.

Table 3. Time and accuracy obtained by the proposed and state-of-the-art methods.

Method	Parameter	Video		
		M-30	M-30 HD	Urban1
Haar Cascade [14]	Time [s]	0.08-0.13	0.30-0.44	0.02-0.06
	Accuracy	43%	75%	40%
SSD [15]	Time [s]	4-7	11-14	2.6-5.6
	Accuracy	22%	70%	69%
Mask R-CNN [9]	Time [s]	2.4-3	2.4-3	2.4-3
	Accuracy	89%	91%	46%
YOLOv3 [10]	Time [s]	1-1.8	1-1.8	1-1.8
	Accuracy	82%	86%	91%
Proposed method	Time [s]	2.1-2.9	2.1-2.9	2.1-2.9
	Accuracy	92%	93.5%	91.5%

The proposed method is the best in terms of accuracy with 92.3%. After the proposed method, the method Mask YOLO V3 has the highest accuracy. The results show that the fastest detector is Haar Cascade, but it offers an average accuracy of 52.7% which has the lowest accuracy and cannot be used in many applications. Auspicious results promise higher accuracy and higher performance. Since there is a trade-off between accuracy and speed, this method has created a good balance between the two criteria. Eliminating the shadow increases the calculations of the proposed method, which in turn increases the time required to perform the steps.

The confusion matrix of the proposed method is shown in Table 4.

Table 4. Confusion matrix of the proposed method.

Video	M-30	M-30 HD	Urban1
Vehicle	290	339	798
Detected vehicles	267	317	731

From Table 4, it can be seen that the proposed method detected 267, 317, and 731 vehicles for M-30, M-30HD, and Urban1, respectively.

Fig. 5 shows the average accuracy on the road-traffic monitoring GRAM dataset obtained by the proposed method compared to other methods while Fig. 6 shows the average time to run the proposed method in comparison with other methods.

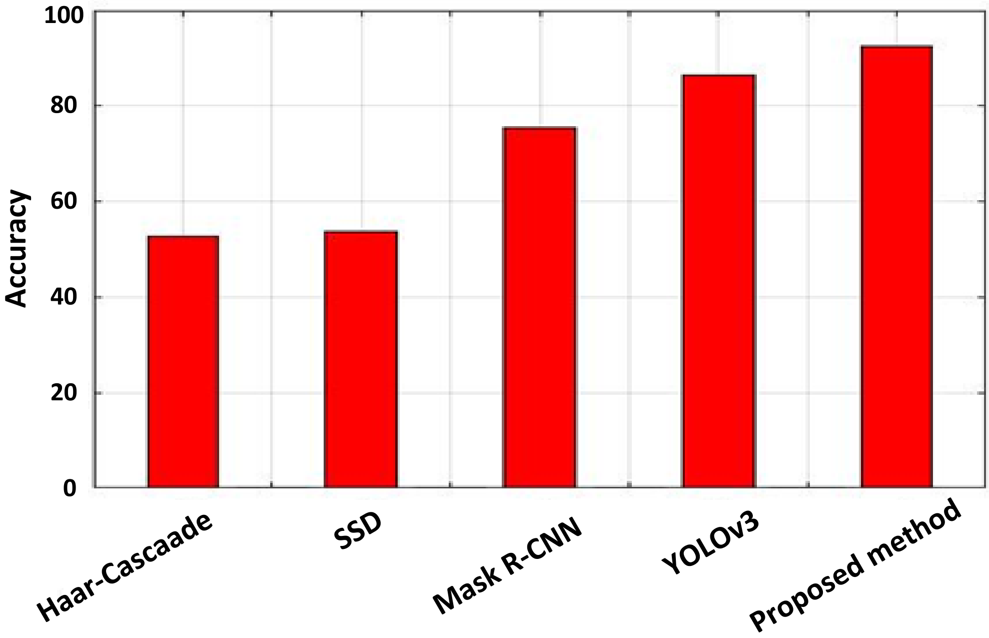


Fig. 5. Average accuracy by various methods run on the road-traffic monitoring GRAM dataset.

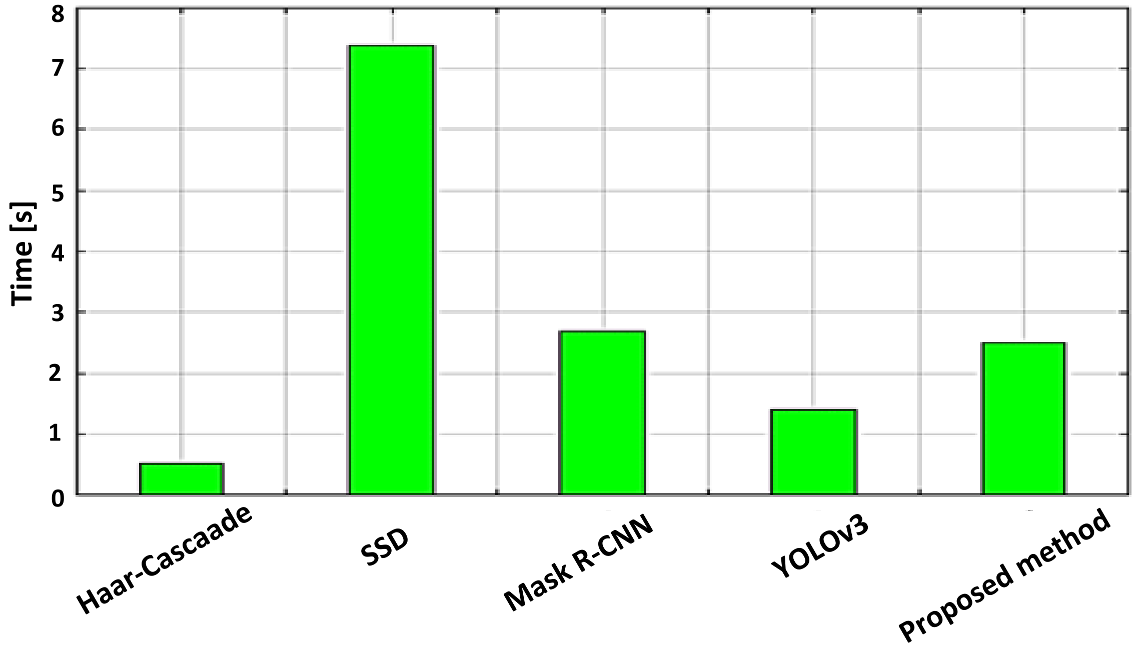


Fig. 6. Average time by various methods run on the road-traffic monitoring GRAM dataset.

The values of Figs. 5 and 6 are the average values obtained from Table 3 to study the results more accurately. For example, for the proposed method, the values of accuracy are 92%, 93.5%, 91.5% for M-30, M-30HD, and Urban 1 datasets, respectively, and its average is 92.3%.

The results show that the proposed method is in a good position in terms of speed. The two faster algorithms have lower accuracy. To increase the accuracy of the proposed method, shadow removal was performed, which led to an increase in calculations. From the results exhibited in Figs. 5 and 6, it can be concluded that the proposed method - which is based on deep learning and edge detection - can create a reasonable balance between speed and accuracy.

The proposed method can perform various tasks such as: edge detection, background subtraction and threshold techniques to provide a suitable video-based surveillance technique.

5. CONCLUSIONS

In this paper, a method for vehicle detection based on one of the deep CNN architectures was presented. The ResUNet-a architecture was used to detect the vehicle because this architecture has high accuracy. Using edge detection in the proposed method, led to reduction in the computational load and a decrease in entropy values. The results of evaluating the proposed method - on videos with different resolutions, presence of clouds and busy streets - show an improvement in F-measure and accuracy. Shadow removal is also another advantage of this method, which led to improved results and increased accuracy. Moreover, the accuracy of detection in complex conditions - such as climate change, presence of noise and shadows and low image resolution - has increased. This method was also able to detect vehicles in cloudy conditions, incomplete appearance of the vehicle, and the presence of traffic in a situation where vehicles are stacked in a row and only part of the car is visible in the image. Compared to the well-known methods, the proposed method had the highest accuracy. In terms of speed, the proposed method was average. Therefore, a future work, the authors intend to focus on improving the time of running the proposed method.

REFERENCES

[1] J. Hua, Y. Shi, C. Xie, H. Zhang, J. Zhang, "Pedestrian-and vehicle-detection algorithm based on improved aggregated channel features," *IEEE Access*, vol. 9, pp. 25885-25897, 2021.

[2] P. Solic, R. Colella, L. Catarinucci, "Proof of presence: novel vehicle detection system," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 44-49, 2019.

[3] D. Li, M. Prasad, C. Liu, C. Lin, "Multi-view vehicle detection based on fusion part model with active learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 5, pp. 3146-3157, 2020.

[4] X. Hu, X. Xu, Y. Xiao, H. Chen, S. He, "SINet: a scale-insensitive convolutional neural network for fast vehicle detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 1010-1019, 2018.

[5] H. Liu, J. Ma, T. Xu, W. Yan, L. Ma, "Vehicle detection and classification using distributed fiber optic acoustic sensing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1363-1374, 2019.

[6] S. Pasban, S. Mohamadzadeh, J. Zeraatkar-Moghaddam, A. Shafiei, "Infant brain segmentation based on a combination of VGG-16 and U-Net deep neural networks," *IET Image Processing*, vol. 14, no. 17, pp. 4756-4765, 2021.

[7] R. Nasiripour, H. Farsi, S. Mohamadzadeh, "Visual saliency object detection using sparse learning," *IET Image Processing*, vol. 13, no. 13, pp. 2436-2447, 2019.

[8] H. Zamanian, H. Farsi, S. Mohamadzadeh, "Improvement in accuracy and speed of image semantic segmentation via convolution neural network encoder-decoder," *Information Systems and Telecommunication*, vol. 6, no. 3, pp. 128-135, 2018.

[9] A. Gheitasi, H. Farsi, S. Mohamadzadeh, "Estimation of hand skeletal postures by using deep convolutional neural networks," *International Journal of Engineering,* vol. 33, no. 4, pp. 552-559, 2020.

[10] Z. Dorrani, M. Mahmoodi, "Noisy images edge detection: ant colony optimization algorithm," *Journal of AI and Data Mining*, vol. 4, no. 1, pp. 77-83, 2016.

[11] Z. Dorrani, H. Farsi, S. Mohamadzadeh, "Image edge detection with fuzzy ant colony optimization algorithm," *International Journal of Engineering*, vol. 33, no. 12, pp. 2464-2470, 2020.

[12] F. Diakogiannis, F. Waldner, P. Caccetta, C. Wu, "ResUNet-a: a deep learning framework for semantic segmentation of remotely sensed data," *Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 94-114, 2020.

[13] F. Waldner, D. Foivos, "Deep learning on edge: extracting field boundaries from satellite images with a convolutional neural network," *Remote Sensing of Environment*, vol. 245, pp. 111741, 2020.

[14] J. Fu, C. Zhao, Y. Xia, W. Liu, "Vehicle and wheel detection: a novel SSD-based approach and associated large-scale benchmark dataset," *Multimedia Tools and Applications*, vol. 79, pp. 12615–12634, 2020.

[15] Z. Wu, J. Sang, Q. Zhang, H. Xiang, B. Cai, X. Xia, "Multi-scale vehicle detection for foreground-background class imbalance with improved YOLOv2," *Sensors*, vol. 19, no. 15, pp. 3336-3345, 2019.

[16] P. Mahto, P. Garg, P. Seth, J. Panda, "Refining Yolov4 for vehicle detection," *International Journal of Advanced Research in Engineering and Technology*, vol. 11, no. 5, pp. 2-10, 2020.

[17] Y. Wang, L. Wang, Y. Jiang, T. Li, "Detection of self-build data set based on YOLOv4 network," *IEEE Accesses*, pp. 640-642, 2020.

[18] S. Du, P. Zhang, B. Zhang, H. Xu, "Weak and occluded vehicle detection in complex infrared environment based on improved YOLOv4," *IEEE Access*, vol. 9, pp. 25671-25680, 2021.

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

[20] W. Qiong, L. Sheng-Bin, "Single shot multiBox detector for vehicles and pedestrians detection and classification," *2nd International Seminar on Applied Physics*, *Optoelectronics*, vol. 2, pp. 1-7, 2017.

[21] S. Ren, K. He, R. Girshick, J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis*, vol. 39, no. 6, pp. 1137–1149, 2017.

[22] W. Zhu, J. Miao, J. Hu, L. Qing, "Vehicle detection in driving simulation using extreme learning machine," *Neuro Computing*, vol. 128, pp. 160–165, 2013.

[23] Y. Chong, W. Chen, Z. Li, W. Lam, C. Zheng, Q. Li, "Integrated real-time vision-based preceding vehicle detection in urban roads," *Neuro Computing*, vol. 116, pp. 144–149, 2013.

[24] W. Chang, C. Cho, "Online boosting for vehicle detection," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics*), vol. 40, no. 3, pp. 893–902, 2010.

[25] Y. Wei, Q. Tian, J. Guo, W. Huang, J. Cao, "Multi-vehicle detection algorithm through combining Harr and HOG features," *Mathematics and Computers in Simulation*, vol. 155, pp.130-145, 2019.

[26] S. Djemame, M. Batouche, "Combining cellular automata and particle swarm optimization for edge detection," *International Journal of Computer Applications*, vol. 57, no. 14, pp.16-23, 2012.

[27] S. Sarkar, V. Venugopalan, K. Reddy, J. Ryde, "Deep learning for automated occlusion edge detection in RGB-D frames," *Journal of Signal Processing Systems,* vol. 88, no. 2, pp. 205-217, 2017.

[28] S. Rajaraman, A. Chokkalinga, "Chromosomal edge detection using modified bacterial foraging algorithm," *Bio-Science Bio-Technology*, vol. 6, pp. 111–122, 2014.

[29] O. Verma, M. Hanmandlu, A. Sultania, A. Parihar, "A novel fuzzy system for edge detection in noisy image using bacterial foraging," *Multidimenal Systems and Signal Processing*, vol. 24, no. 1, pp. 181–198, 2013.

[30] A. Khunteta, D. Ghosh, "Multiresolution edge detection using particle swarm optimization," *International Journal of Engineering Science and Application*, vol. 1, no. 1, pp. 11-17, 2017.

[31] M. Pavlovic, V. Nikolic, M. Simonovic, "Edge detection parameter optimization based on the genetic algorithm for rail track detection," *Facta Universitatis, Series: Mechanical Engineering*, vol. 17, no. 3, pp. 333-344, 2019.

[32] S. Abdulla, "Functional electrical stimulation assisted cycling exercise optimized by multi-objective genetic algorithm," *Jordan Journal of Electrical Engineering,* vol. 7, no. 2, pp. 108-129, 2021.

[33] S. Ari, D. Ghosh, P. Mohanty, "Edge detection using ACO and F rati," *Signal, Image and Video Processing,* vol. 8, no. 4, pp. 625-634, 2014.

[34] M. Asgari, F. Pirahansiah, M. Shahverdy, M. Fartash, A. Prabhu, D. Ravichandran, N. Rokhhman, E. Subanar, E. Marlisah, R. Yaakob, M. Sulaiman, "Using an ant colony optimization algorithm for image edge detection as a threshold segmentation for OCR system, " *Journal of Theoretical and Applied Information Technology*, vol. 95, no. 21, pp. 5654-5664, 2017.

[35] M. Yüksel, "Edge detection in noisy images by neuro-fuzzy processing," *AEU-International Journal of Electronics and Communications*, vol. 61, no. 2, pp. 82–89, 2007.

[36] R. Guerrero-Gomez-Olmedo, R. Lopez-Sastre, "Adaptive image thresholding based on the peak signal-to-noise ratio," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 8, no. 9, pp. 1104-1116, 2014.

[37] H. Liu, Q. Li, I. Wang, "A deep-learning model with learnable group convolution and deep supervision for brain tumor segmentation," *Mathematical Problems in Engineering,* vol. 2010, pp. 1-11, 2021.

[38] M. Babaee, D. Dinh, G. Rigoll, "A deep convolutional neural network for video sequence background subtraction," *Pattern Recognition*, vol. 76, no. 1, pp. 635-649, 2018.