**IAES International Journal of Artificial Intelligence (IJ-AI)**
Vol. 11, No. 2, June 2022, pp. 405~413
ISSN: 2252-8938, DOI: 10.11591/ijai.v11.i2.pp405-413 

# Vehicles detection and counting based on internet of things technology and video processing techniques

**Marwa A. Marzouk[1], Amr Abd El Azeem[2]**
[1]Department of Information Technology, Matrouh University, Mersa Matruh, Egypt
[2]Milatery Technical College, Computer Engineering, Cairo, Egypt

**Article Info**

***Article history:***

Received Sept 9, 2021
Revised Feb 9, 2022
Accepted Feb 21, 2022

***Keywords:***

Image processing
Internet of things
Motion detection
Traffic density

**ABSTRACT**

Recent studies have proven that vehicle tracking and detection play an important role in traffic density monitoring. Traffic overcrowding can be effectively controlled if the number of vehicles expected to pass through a congested intersection can be predicted ahead of time. To overcome such impact of traffic congestion the proposed system presents a framework, using motion detection algorithms and “ThingSpeak” internet of things (IoT) platform which is used in to calculate traffic density, the proposed system capturing video with wireless internet protocol (IP) cameras and broadcasting it to the server where motion detection algorithms as background subtraction are used to obtain a quick overview of traffic density, To save cost and improve the solution, the suggested system utilizes image processing techniques as well as the IoT analytic platform “ThingSpeak” to monitor traffic density. Finally, the suggested method is used to manage traffic flow and avoid traffic crowded. The results of the studies show that the integration of IoT-based technologies with a modified background subtraction technique is effective. This method might be enhanced further to detect vehicles that break traffic laws. We may also improve this system by detecting the presence of emergency vehicles (including an ambulance or fire truck) and granting priority to those cars.



***Corresponding Author:***

Marwa A. Marzouk
Department of Information Technology, Matrouh University
Mersa Matruh, Egypt
Email: mabdelazeem@nctu.edu.eg

## 1. INTRODUCTION

As a result of the growing population, there is a greater demand for vehicles, which increases traffic congestion in various places [1], [2]. So reliable vehicle detection is a critical step of vehicle recognition a smart traffic system can improve, road safety and reduce traffic congestions. For on-road vehicle detection, imaging technology has advanced significantly. Nowadays, cameras are available at a lower cost., are compacted and of high quality. Vehicle detection advances using the internet of things (IoT) have proven their utility in practically every aspect of our everyday lives and can be regarded as a tool for traffic management via a central server [3], [4].

Because of, image-databases and live video information are becoming more and more, because of, image-databases and live video information are becoming more and more, Widespread and intelligent, they are exceptionally important [5], [6]. For the monitoring of traffic data. image and video processing modules are utilized, with motion detection algorithms that identify automobiles as moving blobs and track them for several frames difference, and a modified background subtraction technique is employed in this paper for daylight vehicle recognition. Through frame pre-processing using smoothing with red green blue (RGB) data format to remove noise and unwanted objects such as rain and clouds and to get more information from the frame. Also, the pre-processed frames are used for background modeling to provide a statistical description of the entire background scene using non-recursive frame differencing technique with 4 frames difference and scaled frame to minimize the modeling time. To extract foreground objects using 2 pre-calculated thresholds (strong threshold and weak threshold) not only one threshold to give an accurate decision about moving vehicles in the video sequence then count the number of vehicles.

Finally, the proposed system intends to construct an automatic vehicle counting system that can analyze video captured using wireless internet protocol (IP) cameras on roadways near traffic intersections/junctions, followed by vehicle counting. Data from traffic monitoring will be analysed using the ThingSpeak channel. ThingSpeak channel is an IoT analytics platform with the ability to collect real-time data and visualize the data in the form of charts. We build an IoT-based traffic signal monitoring and control system image processing on this framework. For traffic density monitoring, manual signaling mode, and automatic signaling mode, the system is developed and simulated. IoT technology has recently been applied in a wide range of applications.

In this section, we'll look at some of the work that's been done using IoT in traffic management. A great deal of research is being done to address the issues of vehicle detection and tracking [7]. In [8] to establish the expected needed timing of traffic lights, an array of infrared sensors is used to count the number of vehicles on each traffic lane of the road and record the information on the cloud via a bluetooth connectionuses clustering techniques based on the k-nearest neighbors (KNN) algorithm. Bluetooth needs data transfer from access points close to the sensor array, which adds to the system's complexity. Using the clustering technique also adds to the cloud computing system's overhead, causing delays in decision-making and traffic light timing changes. night vehicle detection approach represented by Nath and Deb [3] as template matching because it contains information about the number of libraries of templates, the approach is ineffective, and approximating correlation is a challenging process. Kim *et al.* [9] and Zhang *et al.* [10] proposed approaches for night vehicle identification based on tracking and lighting pairing. an IoT-based traffic control system was proposed in [11] for image processing, this system utilizes MATLAB software, and for delivering vehicle data sets, it used a Wi-Fi module. If direct data transfer through the cloud is utilised instead of a Wi-Fi transceiver, the system may be made more effective. In [12] a traffic control system was built that utilized a wireless transmitter to send photos directly to the main server. Then there's the server, where the process is more effective if the data being communicated isn't in the form of images. Rather, the processed output information is delivered directly, saving a significant amount of time and communication weight. In [13] the use of networking and embedded technologies to solve traffic congestion problems is described. Using Raspberry Pi, routers, an ultrasonic sensor, and email servers, the author created an alarm system. The intelligent traffic system model was presented by Badura and Lieskovsky [14] the cameras positioned at the intersections scan and monitor their respective domains. For general image analysis, the collected data is instantly transferred to a topology independent data delivery system photoelectric sensors, according to Salama *et al.* [15] might be used to regulate traffic lights. The precise locations for sensor placement are one of the most important aspects. This is mostly due to the traffic control department's need to track cars' movements at specified times, particularly during busy times.

## 2. RESEARCH METHOD

A new method is being developed to reduce the effects of traffic congestion by combining image processing and IoT technology. This system uses wireless IP cameras at traffic intersections to capture video and send it to a server, where the vehicle density on the road is computed using video and image processing techniques. The total number of cars will be spotted and estimated after obtaining a quick look at traffic conditions. The suggested system was developed by using the MATLAB programming environment. The ThinSpeak channel would be used to do traffic control analyses [16]. The next two subsections go through all of the specifics of how the work was completed.

### 2.1. Vehicle detection using image processing

#### 2.1.1. Modified background subtraction algorithm for vehicle detection

Detecting moving objects in a video stream is an important and important topic of computer vision research. Background subtraction is a common method for identifying moving vehicles in a video frame segment that clearly distinguishes from the background model. Developing a decent background subtraction method is difficult for a variety of reasons. It should, first, be sensitive to variations in lighting. Second, it must be avoided to detect non-stationary background noise objects for instance snow, moving leaves, rain, and shadows cast by moving objects [17], [18]. Figure 1 represents a block diagram of video processing.

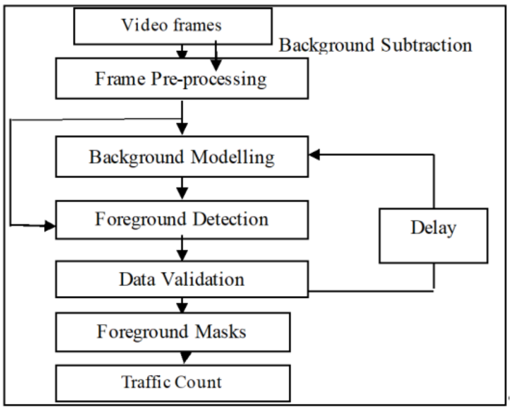
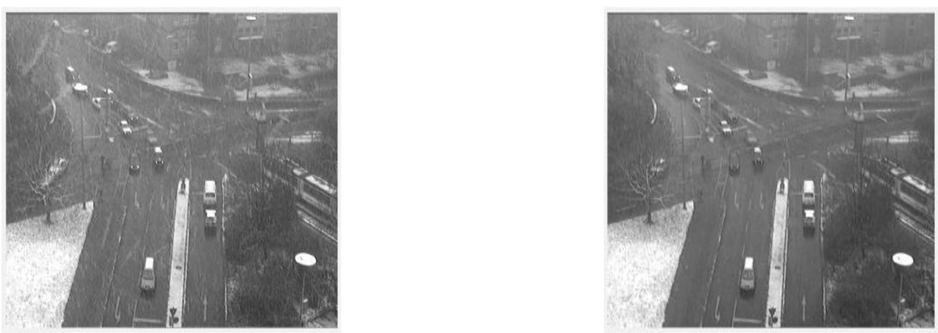



Figure 1. Flow diagram of video processing

```
Function Main_Function()
For each Frame in FarmesSequence
     Call FramePreprocessing(Frame)
     Call FrameScaling(Frame)
     Frame_Number:=Frame_Number +1
     Add Frame to Frame_Array_List
     If Frame_Number>4 then
         Loop
               I:=0
               Get Last 4 Frames
               New_Frame=Average(Last_4_Frames)
               Add New_Frame to New_Frame_Array_List
               If New_Frame_Array_List.Count()>2 then
                        Call Frame_Differencing(New_Frame_Array_List[i]
New_Frame_Array_List[i+1])
                                        I:=1+1
                                        End if
                      End Loop
               End If
End Loop
End Main Function
```

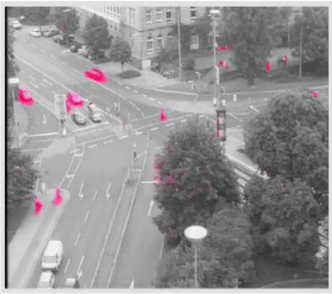
**a. Preprocessing**

To increase the detection of moving vehicles, the video must be pre-processed. To get more information from the frame by using smoothing with RGB data format to remove noise, Figure 2 in this paper for instance, by spatial and temporal smoothing, snow can be cut out of a video as shown in Figure 2(a). After the recognition of small moving objects, morphological processing of the frames can be used to remove them as shown in Figure 2(b). Such as moving leaves on a tree as shown in Figure 3 by comparing video frame that shows the objects detected as moving in Figure 3(a) with video frame in Figure 3(b) after excluding the moving objects by using the morphological processing.



(a) (b)

Figure 2. Comparing the video frame in (a) while it snowed in the traffic scene with the same video frame in (b) after excluding the snow streaks after spatial and temporal smoothing


(a)

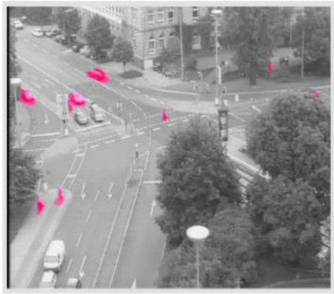

(b)

Figure 3. Comparing video frame that shows the objects detected as moving in (a) with video frame in (b) after excluding the moving objects by using the morphological processing

```
Function FramePreprocessing(InputFrame) returnes Frame
Set OutputFrameArray[frame_width][frame_height] edgex=(frame_width/2)
edgey==(frame_height/2)
for x from edgex to frame_width-edgex
    for y from edgey to frame_height-edgey
       Set colorArray[frame_width][frame_height]
       for fx from 0 to frame_width
           for fy from 0 to frame_height
                        colorArray[fx][fy]:=Inputframe[x+fx-edgex][y+fy-edgey]
sort all entries in colorArray[][]in ascending prder
OutputFrameArray [x][y]:=colorArray[farme_width/2][frame_height/2]
Return Frame
End FramePreprocessing Function
```

**b. Background modeling**

Background modeling uses the new video frame to calculate and update a background model. Recursive techniques do not maintain a buffer for background estimation. Compared with non-recursive techniques, recursive techniques require less storage, but any error in the background model can linger for a much longer period of time. The preprocessed frames are used for background modeling to provide a statistical description of the entire background scene, using non-recursive frame differencing technique with 4 frames difference and scaled frame to minimize the modeling time to extract foreground objects and, using 2 pre-calculated thresholds (strong threshold and weak threshold) not only one threshold to give an accurate decision about moving objects in the video sequence.

**c. Foreground detection**

The background model is compared to the input video frame, and relevant foreground pixels out from the input frame are identified. Then, foreground detection finds pixels in the video frame that the background model can't explain and outputs them as a binary relevant foreground mask. The most standard method for detecting foreground objects is to see if the input pixel differs considerably from the equivalent background estimate:

$$|I_t\ (x;\ y) - B_t\ (x;\ y)\ | > T \tag{1}$$

A threshold based on normalized data is another common foreground detection method:

$$\frac{|\ I_t\ (x,y) - B_t\ (x,y) - \mu_d\ |}{\partial_d} > Ts \tag{2}$$

Where µd and ∂d are the mean and the standard deviation of It (x, y)-Bt (x, y) for all spatial locations (x, y). Most schemes determine the foreground threshold T or Ts experimentally. Ideally, the threshold should be a function of the spatial location (x, y). For instance, in low-contrast areas, the threshold should be lower. To accentuate the contrast in dark areas such as shadows, one possible change is to use relative difference instead of absolute difference:

$$\frac{|\ I_t\ (x,y) - B_t\ (x,y)\ |}{B_t\ (x,y)} > Tc \tag{3}$$

Using two thresholds is another way to introduce spatial variability. The aim is to detect “strong” foreground pixels with absolute deviations from background estimations that exceed a specific threshold. Then, from strong foreground pixels, foreground regions are produced by integrating surrounding pixels with absolute differences greater than a lower threshold. To expand the area, a two-pass, connected-component grouping method might be applied.

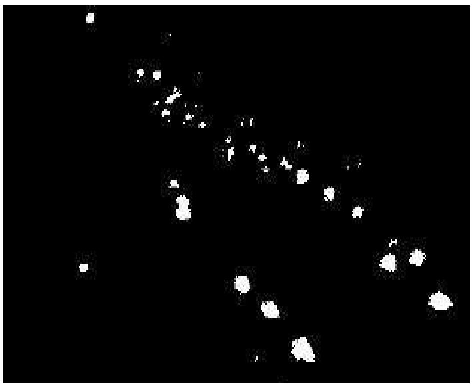
**d. Data validation**

In this step, the candidate mask is examined, and pixels that do not relate to actual moving objects are removed before the final foreground mask is produced. As shown in Figure 4 where the highway sequence processed with a background subtraction method and our method Figure 4(a) input frame, Figure 4(b) background subtraction, and Figure 4(c) motion objects. While most of the background models have three main problems [19], [20]:

i) They ignore any pixel-to-pixel connection. Small false-positive or false-negative portions are irregularly spread across the candidate mask as a result of this problem. To eliminate these portions, common methods use morphological filtering and related component grouping [21].
ii) The rate of background modeling adaptation may not match the moving speed of the foreground objects. To solve these issues by running numerous background models at varying adaption rates and cross-validating between these models regularly to improve accuracy.
iii) Moving leaves or moving object shadows cast non-stationary pixels that are simply mistaken for actual foreground objects. By using sophisticated background modeling techniques like mixture of Gaussians (MoG) and morphological filtering for cleanup, the moving leaves problem can be solvedas shown in Figure 4(b).
iv) Finally, only doing background subtraction on a sub-sampled version of each image is a simplification strategy that speeds up data validation. As a result, a 640 pixel by 480-pixel image can be resized to 160 by 120 pixels, a fraction of the linear dimensions. Because this image is one-sixteenth the size of the original, the background subtraction processing time is also one-sixteenth the time as shown in Figure 4(c).

(a)

(b)



(c)

Figure 4. The highway sequence processed with a background subtraction methode and our method (a) input frame, (b) background subtraction, and (c) motion objects

**e. Traffic count**

In a binary image, blobs are the linked sections. The goal of the blob analysis method is to find spots and/or areas in a picture that differ in brightness or area [22]. As explained in [23] the laplacian of Gaussian is utilized as a formulation for finding blob value using a computer vision approach. The procedure begins with the labeling of an area that is deemed a foreground object, followed by the collection of data into the blob, such as the initial pixel location, x- and y-axis lengths, and pixel area. Figure 5 illustrates a blob area. In Figures 5(a) to (c) are visible object detection processes that are acquired by foreground area detection from the binarization process as Figure 5(d). Setting pixel vector values to point values can help in determining blob area as demonstrated in Figure 5(e).

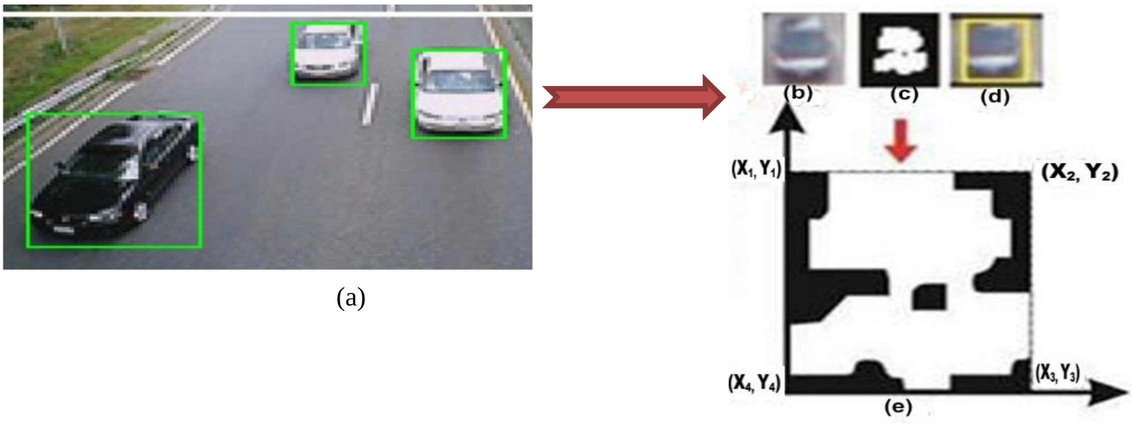



Figure 5. Vechicles counting (a) big frame of the original image, (b) cropping object, (c) foreground segmentation, (d) object detection using the bounding box, and (e) BLOB area in x, y-axis direction

### 2.1.2. Traffic density monitoring on IoT analytics platform

The vehicle density count is provided by joining up for ThingSpeak and creating a channel. Figure 6 shows how the traffic signal timer is set based on the number of vehicles. We added a description and a field to the channel settings before it could be used to contain data [24]. A write accept payments online (API) key is generated by default when we build a channel. When transmitting data to the channel the write API key is used. Figure 7 displays the time and date of the road congestion monitoring analysis on a ThingSpeak channel. Each dot represents the current traffic value and the moment in which this was sent to the channel.

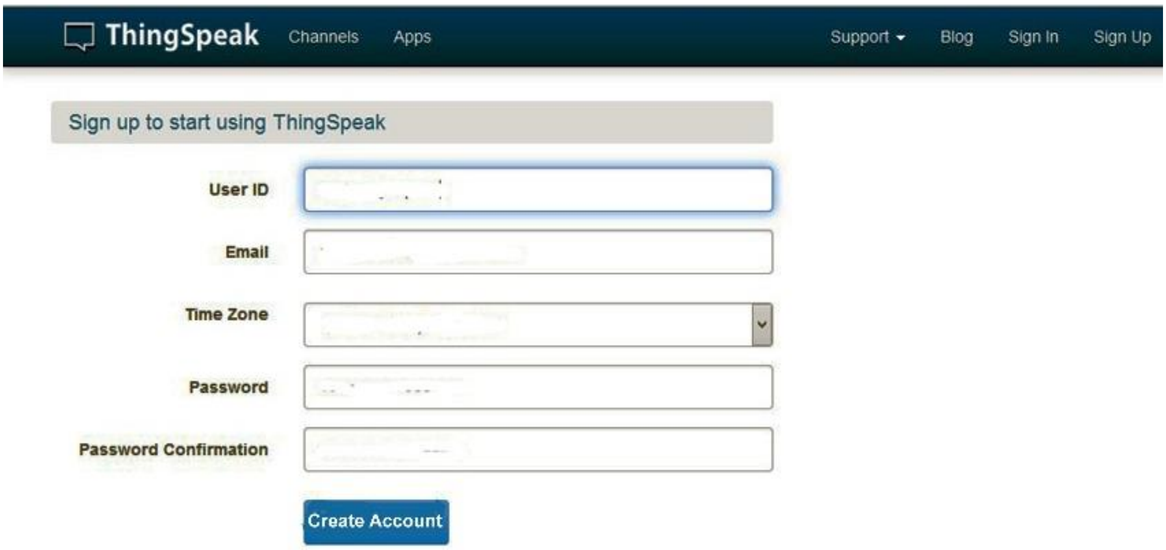



Figure 6. Sign-up windows for ThingSpeak

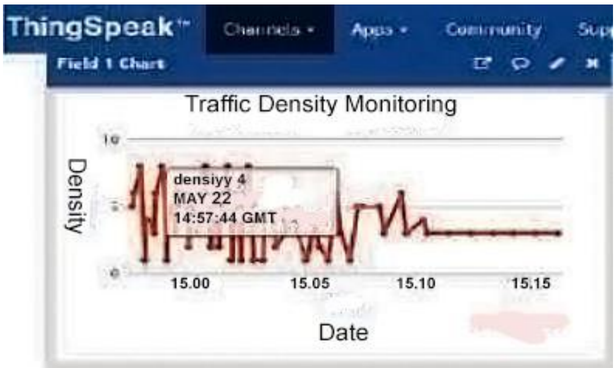



Figure 7. Traffic values

## 3. RESULTS AND DISCUSSION

The proposed method was tested on 8 videos captured using an IP webcam. Measurements in this paper were made in different environments to measure the strength of the algorithm under factors that will decrease the accuracy and performance of the algorithm like moderate rainfall as shown in Table 1, fog as shown in Table 2, and night as shown in Table 3. The data is analyzed by thingSpeak which is an IoT analytics platform with the capability of collecting real-time data and visualizing it in the form of charts. To calculate accuracy in this paper we use the standard accuracy equation [25] Accuracy=(TP+TN)(TP+TN+FN+FP)x100%. Where TP is a true positive, TN is a true negative, FP is a false positive, and FN is a false negative.

Table 1. Performance of the traffic monitoring system in moderate rainfall

| Vid.NO | Vehicles Count | TP | TN | FP | FN | Accuracy |
|---|---|---|---|---|---|---|
| 1 | 6 | 2 | 2 | 0 | 0 | 100 |
| 2 | 10 | 2 | 2 | 0 | 1 | 80 |
| 3 | 8 | 3 | 2 | 0 | 1 | 83 |
| 4 | 9 | 2 | 4 | 1 | 0 | 85.7 |
| 5 | 11 | 4 | 5 | 1 | 1 | 90 |
| 6 | 2 | 1 | 1 | 0 | 0 | 100 |
| 7 | 4 | 2 | 1 | 0 | 0 | 100 |
| 8 | 5 | 1 | 3 | 0 | 0 | 100 |
| Average | | | | | | 93 |

Table 2. Performance of the traffic monitoring system in Fog

| Vid.NO | Vehicles Count | TP | TN | FP | FN | Accuracy |
|---|---|---|---|---|---|---|
| 1 | 5 | 1 | 2 | 0 | 0 | 100 |
| 2 | 4 | 2 | 4 | 0 | 1 | 85.7 |
| 3 | 8 | 3 | 2 | 1 | 0 | 83 |
| 4 | 7 | 4 | 5 | 1 | 1 | 90 |
| 5 | 10 | 2 | 2 | 0 | 1 | 80 |
| 6 | 3 | 1 | 1 | 0 | 0 | 100 |
| 7 | 9 | 2 | 3 | 1 | 0 | 83 |
| 8 | 12 | 1 | 3 | 1 | 0 | 80 |
| Average | | | | | | 88 |

Table 3. Performance of the traffic monitoring system in night

| Vid.NO | Vehicles Count | TP | TN | FP | FN | Accuracy |
|---|---|---|---|---|---|---|
| 1 | 5 | 4 | 4 | 2 | 0 | 80 |
| 2 | 4 | 2 | 4 | 0 | 1 | 85.7 |
| 3 | 8 | 3 | 6 | 1 | 2 | 81.8 |
| 4 | 7 | 3 | 4 | 1 | 1 | 77.7 |
| 5 | 10 | 5 | 4 | 0 | 2 | 81.1 |
| 6 | 3 | 2 | 2 | 1 | 0 | 80 |
| 7 | 9 | 2 | 3 | 1 | 0 | 83 |
| 8 | 12 | 6 | 4 | 0 | 2 | 83.3 |
| Average | | | | | | 82 |

## 4. CONCLUSION

The modified background subtraction technique, noise reduction via morphological analysis, blob detection, and a signal system based on numerous blobs or density a module were created in this study. We also presented a solution for identifying vehicle density using an IoT analytic platform. The suggested approach was tested on eight videos taken using an IP camera. Measurements are taken in a variety of environments to assess the algorithm's strength in the face of factors that might reduce accuracy and performance such as (fog, rain, and night). And based on these measurements the accuracy of rain is about 93% also in fog is 88% however at night the accuracy is about 83% so the proposed system needs to enhance its accuracy at night but the main advantage of the proposed system using image processing, MATLAB software and ThingSpeak platform. that it is possible to execute it at a low cost and with the highest level of accuracy using the suggested technique, we are just monitoring the number of vehicles present at the signal. This method might be developed further to detect vehicles that violate traffic laws also this method might be developed further to identify the presence of emergency vehicles (such as an ambulance or fire truck) and give those vehicles precedence.

## REFERENCES

[1] E. Bekiaris and Y. J. Nakanishi, Eds., *Economic impacts of intelligent transportation systems: innovations and case studies*, 1st Editio. Elsevier, 2004.

[2] Y. A. Syahbana and Y. Yasunari, "Early detection of incoming traffic for automatic traffic light signaling during roadblock using vanishing point-guided object detection and tracking," in *2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2021, pp. 1466–1471.

[3] R. K. Nath and S. K. Deb, "On road vehicle/object detection and tracking using template," *Indian Journal of Computer Science and Engineering*, vol. 1, no. 2, pp. 98–107, 2010.

[4] R. P. Nimkar and C. N. Deshmukh, "Traffic density monitoring and cattle menace alert system using IoT," *International Journal of Research in Engineering and Technology*, vol. 7, no. 8, pp. 96–104, Aug. 2018, doi: 10.15623/ijret.2018.0708012.

[5] N. K. Jain, R. K. Saini, and P. Mittal, "A review on traffic monitoring system techniques," in *Advances in Intelligent Systems and Computing*, Springer Singapore, 2019, pp. 569–577.

[6] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, Apr. 2003, doi: 10.1016/S0262-8856(03)00004-0.

[7] D. Nettikadan and S. R. M. S., "Smart community monitoring system using thingspeak iot plaform," *International Journal of Applied Engineering Research*, vol. 13, no. 17, pp. 13402–13408, 2018.

[8] S. Kumar Janahan, M. R. M. Veeramanickam, S. Arun, K. Narayanan, R. Anandan, and S. Javed Parvez, "IoT based smart traffic signal monitoring system using vehicles counts," *International Journal of Engineering and Technology*, vol. 7, no. 2.21, Art. no. 309, Apr. 2018, doi: 10.14419/ijet.v7i2.21.12388.

[9] H.-K. Kim, S. Kuk, M. Kim, and H.-Y. Jung, "An effective method of head lamp and tail lamp recognition for night time vehicle detection," *World Academy of Science, Engineering and Technology*, vol. 44, pp. 1091–1094, 2010.

[10] W. Zhang, Q. M. J. Wu, G. Wang, and X. You, "Tracking and pairing vehicle headlight in night scenes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 140–153, Mar. 2012, doi: 10.1109/TITS.2011.2165338.

[11] E. Basil and S. D. Sawant, "IoT based traffic light control system using Raspberry Pi," in *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, Aug. 2017, pp. 1078–1081, doi: 10.1109/ICECDS.2017.8389604.

[12] T. Osman, S. S. Psyche, J. M. Shafi Ferdous, and H. U. Zaman, "Intelligent traffic management system for cross section of roads using computer vision," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2017, pp. 1–7, doi: 10.1109/CCWC.2017.7868350.

[13] K. B. Malagund, S. N. Mahalank, and R. M. Banakar, "IoT based smart city traffic alert system design," in *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, Aug. 2016, pp. 1–6, doi: 10.1109/ICCUBEA.2016.7860146.

[14] S. Badura and A. Lieskovsky, "Intelligent traffic system: cooperation of MANET and image processing," in *2010 First International Conference on Integrated Intelligent Computing*, Aug. 2010, pp. 119–123, doi: 10.1109/ICIIC.2010.41.

[15] A. S. Salama, B. K. Saleh, and M. M. Eassa, "Intelligent cross road traffic management system (ICRTMS)," in *2010 2nd International Conference on Computer Technology and Development*, Nov. 2010, pp. 27–31, doi: 10.1109/ICCTD.2010.5646059.

[16] J. M. S. Waworundeng, D. Fernando Tiwow, and L. M. Tulangi, "Air pressure detection system on motorized vehicle tires based on IoT platform," in *2019 1st International Conference on Cybernetics and Intelligent System (ICORIS)*, Aug. 2019, pp. 251–256, doi: 10.1109/ICORIS.2019.8874904.

[17] S. González Díaz, "Fall detection system for the elderly by means of artificial vision (in Spanish)." 2017.

[18] B. Garcia-Garcia, T. Bouwmans, and A. J. Rosales Silva, "Background subtraction in real applications: Challenges, current models and future directions," *Computer Science Review*, vol. 35, Art. no. 100204, Feb. 2020, doi: 10.1016/j.cosrev.2019.100204.

[19] R. Ke, Y. Zhuang, Z. Pu, and Y. Wang, "A smart, efficient, and reliable parking surveillance system with edge artificial intelligence on IoT devices," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 8, pp. 4962–4974, Aug. 2021, doi: 10.1109/TITS.2020.2984197.

[20] A. Rosenfeld and D. Weinshall, "Extracting foreground masks towards object recognition," Nov. 2011, doi: 10.1109/iccv.2011.6126391.

[21] Q. Zhang, J. Lin, Y. Tao, W. Li, and Y. Shi, "Salient object detection via color and texture cues," *Neurocomputing*, vol. 243, pp. 35–48, Jun. 2017, doi: 10.1016/j.neucom.2017.02.064.

[22] A. Alharbi, A. Aloufi, E. Hamawi, F. Alqazlan, S. Babaeer, and F. Haron, "Counting people in a crowd using Viola-Jones algorithm," *International Journal of Computing, Communication and Instrumentation Engineering*, vol. 4, no. 1, pp. 57–59, Feb. 2017, doi: 10.15242/IJCCIE.IAE1216010.

[23] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, "High-level semantic feature detection: a new perspective for pedestrian detection,"

in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 5182–5191, doi: 10.1109/CVPR.2019.00533.
[24] S. Pasha, "Thingspeak based sensing and monitoring system for IoT with Matlab analysis," *International Journal of New Technology and Research*, vol. 2, no. 6, pp. 19–23, 2016.
[25] A. S. More and D. P. Rana, "An experimental assessment of random forest classification performance improvisation with sampling and stage wise success rate calculation," *Procedia Computer Science*, vol. 167, pp. 1711–1721, 2020, doi: 10.1016/j.procs.2020.03.381.

## BIOGRAPHIES OF AUTHORS

**Marwa A. Marzouk** Ph.D. Institute of Graduate Studies and Research Department of information technology, Alexandria University, 2017. M.sc, Institute of Graduate Studies and Research Department of information technology, Alexandria University, 2011. Post graduate diploma, Institute of Graduate Studies and Research Department of information technology, Alexandria University, 2008. Bachelor of Science and Education-Alexandria University.interested in Computer Vision, Image Processing, and Information Retrieval. She can be contacted at email: mabdelazeem@nctu.edu.eg.

**Amr Abd El Azeem** Milatery Technical College, Computer Engineering, Cairo, Egypt. Master in Cyber security from arab academy for science, technology and maritime transport. interested in Computer Vision, Image Processing, network security, cyber security, internet of things and Information Retrieval. He can be contacted at email: amro_832002@hotmail.com.