Engineering and Technology Journal
Journal homepage: engtechjournal.org

An Efficient Approach for Detecting and Classifying Moving Vehicles in a Video Based Monitoring System

Sajidah S. Mahmood [a*], Laith J. Saud [b]

[a] Computer Engineering Branch, Department of Control and Systems Engineering, University of Technology-Iraq. sajedaalmaster@gmail.com

[b] Computer Engineering Branch, Department of Control and Systems Engineering, University of Technology-Iraq. laithjasim15@yahoo.com

*Corresponding author.

Submitted: 03/07/2019 Accepted: 23/10/2019 Published: 25/06/2020

K E Y W O R D S

Object detection, object recognition, background subtraction, image processing, traffic analysis.

A B S T R A C T

Moving objects detection, type recognition, and traffic analysis in video-based surveillance systems is an active area of research which has many applications in road traffic monitoring. This paper is on using classical approaches of image processing to develop an efficient algorithm for computer vision based on traffic surveillance system that can detect and classify moving vehicles, besides serving some other traffic analysis issues like finding vehicles speed and heading, tracking specified vehicles, and finding traffic load. The algorithm is designed to be flexible for modification to fulfill the changes in design objectives, having limited computation time, giving good accuracy, and serves inexpensive implementation. A 92% of success is achieved for the considered test, with the missed cases being abnormal that are not defined to the algorithm. The computation time, with a platform (hardware and software) dependent, the algorithm took to produce results was parts of milliseconds. A CNN based deep learning classifier was built and evaluated to judge the feasibility of involving a modern approach in the design for the targeted aims in this work. The modern NN based deep learning approach is very powerful and represents the choice for many very sophisticated applications, but when the purpose is restricted to limited requirements, as it is believed the case is here, the reason will be to use the classical image processing procedures. In making choice, it is important to consider, among many things, accuracy, computation time, and simplicity of design, development, and implementation.

How to cite this article: S. S. Mahmood, L. J. Saud. "An efficient approach for detecting and classifying moving vehicles in a video based monitoring system," Engineering and Technology Journal, Vol. 38, No. 06, pp. 832-845, 2020.
DOI: https://doi.org/10.30684/etj.v38i6A.438

1. Introduction

The need for traffic surveillance systems that serves efficient monitoring, analysis, and management of road traffic has increased in the last few decades because of the huge extension in the road networks as well as the enormous increase in number of vehicles. The majority of surveillance systems involve human operators which makes them inefficient [1]. Many factors helped in making surveillance systems cheaper, faster, more flexible, more robust, more autonomous and intelligent, and with more capabilities. Cheap, small size and high computational power computers with improved storage capabilities, are just some of these factors besides the continuously developed approaches [2,3]. Different efforts dealing with traffic monitoring could be found in literature. Current research in traffic operation evaluation for urban roads mainly considers road grade and vehicle speed [4]. Some are computer vision based and use traditional image processing methods [5], and some use sensors approach [6].

Traffic monitoring systems based on visual monitoring serve different purposes and applications. As related to traffic issues in general, they can help in detecting, classifying and tracking vehicles, and also can be used to do other complex tasks related to traffic analysis. A visual traffic surveillance system involves hardware and software. Hardware includes mainly a camera installed on the roadside or above it that captures the video, and a processing system which is mainly a digital computing system. The software part of the system is concerned with processing and analyses. These systems could be portable with a microcontroller attached to the camera for the real-time processing and analyses or just the cameras that transmit the video feed to a centralized computer for further processing [2]. Detection of moving objects remains a very challenging job [7]. Detection and classification of moving vehicles could be achieved by classical (traditional) methods or by modern methods. There are so many approaches to achieve detection and classification using classical methods [8]. Regarding modern methods, deep learning with neural network is very promising for detection and classification purposes when it comes to capability, but very time consuming when it comes to processing time considerations [9]. In this work, an algorithm for a stationary camera based monitoring system is designed to detect, classify, and track moving vehicles besides providing some traffic analysis information. The algorithm is formulated, using traditional methods and concepts (old and relatively new) of image processing, in a way to be with lesser computations and so lesser computation time while still providing good accuracy. A CNN deep learning algorithm is also built and evaluated to judge the feasibility of using it for the intended aims. The rest of this paper introduces a theoretical background, the details of the suggested algorithms, the results obtained, and conclusions about the outcomes.

2. Theoretical Background

Moving objects represent objects whose spatial positions or extents change continuously over time. Moving objects could be, for example, people, animals, or vehicles. Generally, dealing with moving objects involves detecting, classifying, tracking, and analyzing. To achieve these purposes, in case of computer vision and image processing based monitoring systems, there are two main approaches which could be categorized as classical, or traditional, and modern (like the ones based on deep learning).

Converting frames from colored to gray is among the necessary preprocessing steps to be done before other procedures on images are carried out. In case of RGB (Red, Green, and Blue) images, there are two methods to convert RGB to grayscale, average method and weighted (luminosity) method. The weighing method produces a relatively brighter gray image. The conversion for this method is given in Eq. (1):

$$Grayscale = 0.299R + 0.587G + 0.114B \qquad (1)$$

Moving objects detection is a technique used to determine the presence of any moving object in a videoed scene. There are different detection methods for the classical approach. Object detection can be done by various techniques like background subtraction, frame difference, temporal difference and optical flow [10,11]. The principle of background subtraction is to build a model of the static scene (i.e. without the moving objects) called background, and then to compare every frame of the sequence to this background in order to discriminate the regions of motion, called foreground (the moving objects). A procedural view of how this method works is shown in Figure 1 [12]. Frame difference, which is the simplest form of background subtraction, is based on subtracting two consecutive frames and as given in Eq. (2):

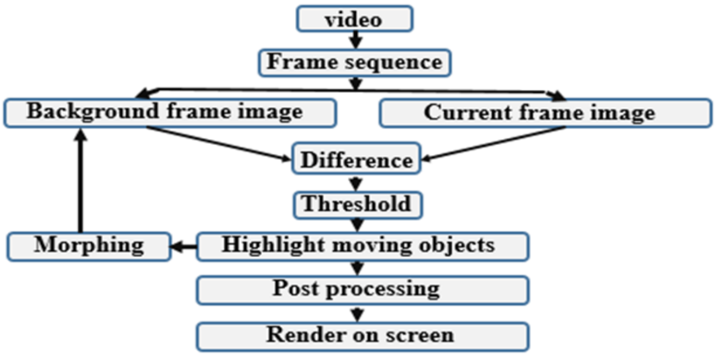

Figure 1: Background subtraction [12]

$$Image = Abs\ (frame_{(i+1)} - frame_{(i)}) \quad (2)$$

Moving objects classification can be done by various techniques like shape based, motion based, color based, and texture based. For shape based technique, convex hull concept could be used to bind the object with a blob. This blob is used later for objects recognition, and then serves classification by using shape size. Convex hull is basically a geometrical issue that could be solved computationally [13]. In this classification context contour concept is assistive. A contour is a closed curve of points or line segments representing the boundaries of an object in an image, and so the shape. Once the contours of the objects in an image are found, many things can be done, like determining the number of objects in an image, classifying their shapes, measuring their size, and stating what they represent. The input to the contour-finding process is a binary image, which is to be produced by first applying thresholding and / or edge detection. In the binary image, the objects to be detected should be white, while the background of the image should be black [14].

After the objects are separated and clarified, each of them is enclosed by a rectangle for which the area, aspect ratio, and diagonal can be calculated and used for distinguishing the objects. The equations for these parameters are:

$$Area = Length \times Width \quad (3)$$

$$Aspect\ ratio = Width \div Height \quad (4)$$

$$Diagonal = \sqrt{((Width)^2 + (Length)^2} \quad (5)$$

The various types of object tracking techniques are generally grouped into three categories; silhouette tracking, kernel tracking, and point tracking [11, 15].

Traffic analysis could cover many things like traffic load, vehicles speed, and heading.

In "modern" approaches or methods, deep learning, neural networks, and artificial intelligence, are subjects that could be linked to achieve a certain purpose in a certain application. Object detection and classification are examples for this purpose, and surveillance system is an example for this application. Convolutional neural network (CNN or ConvNet,) is considered as one of the most popular algorithms for deep learning. Feed-forward artificial neural networks are most commonly applied to the analysis of visual imagery. CNNs use a variation of multilayer perceptron designed to require minimal pre-processing [16]. Such networks are characterized by being multi-layered and so they are large and they are consequently complex. Training and evaluating networks with such characteristics require powerful computing systems [17].

3. The Suggested Algorithm

The algorithm suggested in this paper is a surveillance system algorithm mainly intended for detecting and classifying vehicles in video frames gathered by a static camera used to monitor road traffic. That is besides doing some traffic analysis like finding traffic load and vehicles speed and heading. The algorithm is divided into four main stages each of which in turn has its own internal steps. These stages are shown in Figure 2, and the steps through which the algorithm proceeds are shown in Figure 3.

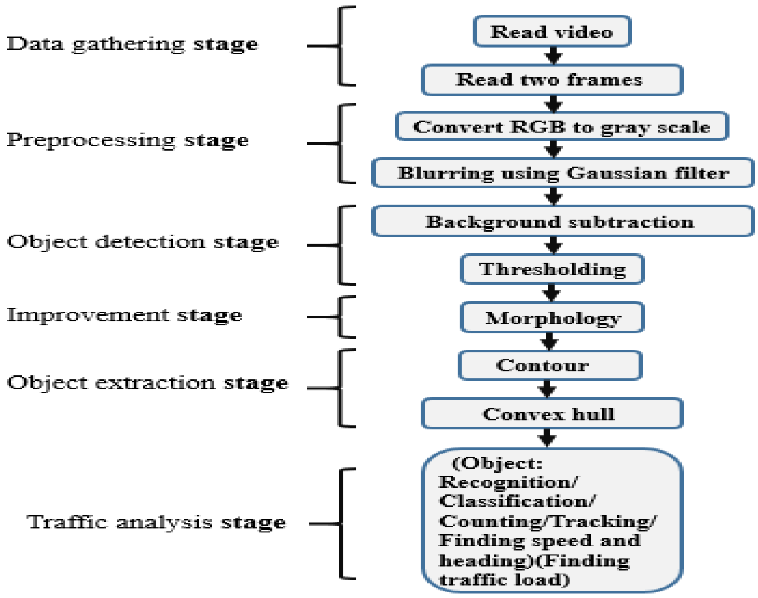

Figure 2: The designed algorithm stages

I. Preprocessing stage

The frames obtained from the captured video will be prepared for processing by: i- Converting them to gray scale in order to reduce the processing time by reducing the amount of information in each frame without affecting the information necessary for next processing stages and steps. ii -Blurring the frame to remove noise by using Gaussian filter.

Algorithm input: two frames
Algorithm output: types of the moving objects detected, counts of the objects, and other traffic analysis data

Step 1: Set a counter for the objects, set a counter for the frames to be read, and set other necessary settings.
Data reading
Step 2: Read two consecutive frames.
Preprocessing
Step 3: convert each frame from RGB to gray scale using the equation:
Grayscale = 0.299R +0.587G + 0.114B
Step 4: apply Gaussian filter (size: 5×5).
Moving objects detection (revelation)
Step 5: background subtraction
Image = Abs (frame $_{(i+1)}$ −frame $_{(i)}$)
Step 6: apply thresholding to the resulting image.
New pixel value = 0 If pixel value in the image is <= 25
New pixel value = 1 If pixel value in the image is > 25
Image refinement:
Step 7: apply dilation (expands image objects and fill the gaps).
Step 8: apply erosion (shrinks image objects).
Object extraction
Step 9: Apply contour (representing the boundaries of an object in an image).
Step 10: apply convex hull (complete the figure of the object and convert the object to a blob).
Object recognition and analysis
Step 11: object recognition (recognize the type of the vehicles objects extracted (classify as motorcycle or car or truck)).
Step 12: object counting (count the number of each type), and do other analysis steps.
Step 13: advance frames counter. Go to step 14 in case frames to be read ended, or step 2 otherwise.
Step 14: end

Figure 3: The steps through which the designed algorithm proceeds

II. Object detection stage

This includes: i- Using background subtraction to detect the moving objects in the scene. ii- Doing proper thresholding to the obtained result to convert the frame image to black and white (binary) and to obtain best representative objects' images. iii- Applying morphology to fill the gaps and so help in obtaining more clear and accurate shapes.

III. Object extraction stage

Object extraction is done by applying two actions; contouring action which will border the moving objects, and convex hull action which will convert the object to a blob. This blob will be enclosed by a rectangle and used later for recognition of objects using shape dimensions.

IV. Analysis stage

The recognition of each object being extracted from the video is done depending on its shape dimensions. A counter is used for each type of vehicle to count the number of vehicles passing of the given type. Other parts of analysis include finding vehicles speed and heading as well as traffic load.
Before presenting tests and results, it is to be stated that:
- The results are obtained using test datasets from Internet represented by two videos each of which being recorded by a static camera monitoring a highway. One sample frame from each video is shown in Figure 4.

Figure 4: Sample frames from videos used in tests

- The classical approach algorithms are implemented with Visual Studio (C++) and OpenCV software, while the modern approach is implemented with MATLAB software.

4. Algorithm Stages with Tests and Results

I. Preprocessing stage

Each image needs to be preprocessed to remove unwanted information and to reduce the amount of data that needs to be processed. Many techniques exist for preprocessing. In this paper, RGB to gray scale conversion is used, and Gaussian blurring of size (5×5).
1) Gray scale conversion: The weighted conversion method given by Eq. (1) is used for gray scale conversion of video frames. A sample result of applying this procedure is shown in Figure 5, where a colored frame is shown before and after applying the converting formula.
The results in this paper are obtained using two test datasets from Internet.

Figure 5: Conversion of an RGB frame to a gray scale frame

Actually these data sets are two videos recorded by static cameras monitoring highways. The algorithms in this paper are developed with C++ using Visual Studio and OPENCV software.
2) Blurring: This step is done via applying Gaussian filter with mask of 5×5 size. The aim of this step is to remove the noise in the image and produce a smooth image more suitable for next processing to get better results. The effect of applying Gaussian filter to the frames is depicted in Figure 6.

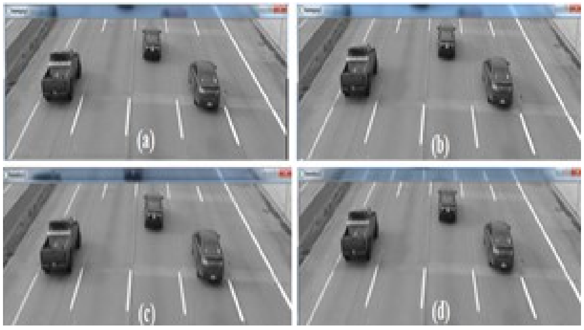

Figure 6: The effect of using Gaussian filter a: grayed frame 1 b: grayed frame 2 c: blurred frame 1 d: Blurred frame2

Trials and tests indicated that the size (5×5) is a good choice. It has better performance than (3×3) and is less time consuming than (7×7). Also the trial tests carried out indicated that increasing the filter window size leads to blurring the image as shown in Figure 7. So, a size of (5×5) is a good compromise.

Figure 7: The distortion caused by large filter sizes

II. Object detection stage

Background Subtraction: Because it is simple and fast, frame subtraction is used here to detect the objects. This is done by subtracting two consecutive frames and as indicated in Eq. (2). Figure 8-a shows the result of applying image subtraction. Background subtraction algorithm needs less computation and performs better and it is more flexible and effective than other detection techniques.
1) Thresholding: The image obtained from the previous step needs to be converted to black and white. This conversion represents a necessity for the subsequent processing stages and helps in reducing processing time. Conversion is done by applying Eq. (6) (thresholding equation).

$$\begin{Bmatrix} NPV = 0 & \text{If PV is} < \text{Th} \\ NPV = 1 & \text{If PV is} >= \text{Th} \end{Bmatrix} \qquad (6)$$

Where; NPV stands for new pixel value, PV stands for pixel value in the image, and Th stands for threshold value. The tests and trials carried out on the data sets used in this work led to a value of 25 for convenient thresholding. The effect of thresholding is shown in Figure 8-b.

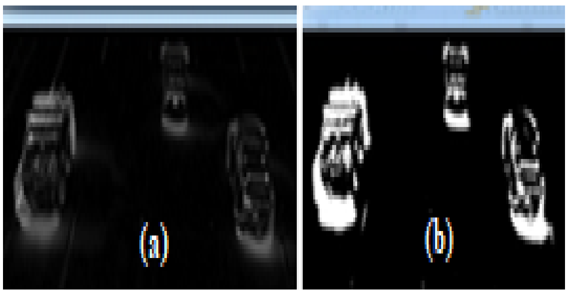

Figure 8: Object detection a: background subtraction b: thresholding

Unsuitable thresholding could, for low threshold value, enlarge the object by adding pixels not belonging to it, or, for large threshold value, wipe away the object.by removing pixels belonging to it. Figure 9 shows a sample for low thresholding, where the objects obtained in the last stage of the process of extracting moving objects Figure 9-d are larger than they are actually.

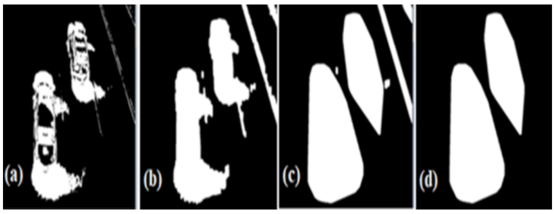

Figure 9: Effect of ignoring shadow problem (Moving objects extraction steps: a: thresholding b: applying contour c: applying convex hull d: image blob)

2) Morphology: After applying background subtraction, image may suffer noise existence, incompleteness in shapes, gaps presence, etc. Morphological filtering is used to deal with this issue. This is achieved here through applying a rectangular filter of (5×5) size to the image pixels. Pixels can be added to (dilation) or subtracted from (erosion) the object boundaries.

III. Object extraction stage

This phase mainly contains two internal steps.

1) Applying contour step: The main purpose of applying the contour in this work is to connect the points and lines to form closed curves, where these lines or points represent the boundary of the detected objects. Applying the contour is shown in Figure 10-a.

2) Applying convex hull step: The main result of applying convex hull will be the convex polygon which is the blob of the extracted object, this is shown in Figure 10-b. At this stage, small objects (unwanted objects) appearing as points or tiny objects are excluded from detection and recognition by the conditions of area, diagonal and aspect ratio. Figure 10-c shows the results after removing unwanted data.

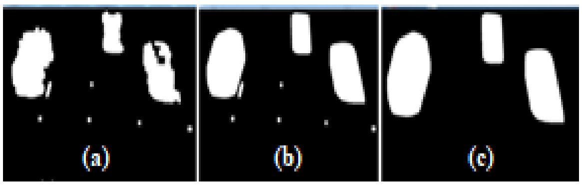

Figure 10: Object extraction a: applying contour b: applying convex hull c: image after refinement

Each of the blobs obtained is enclosed by a rectangle. The rectangles play a vital role in vehicles recognition and classification as will be seen. Once recognition and classification are done, different traffic analysis steps can be done like vehicles counting, finding vehicles heading and speed, and traffic load. The rectangles are also used to indicate the related objects in the original frames as shown in Figure 11.

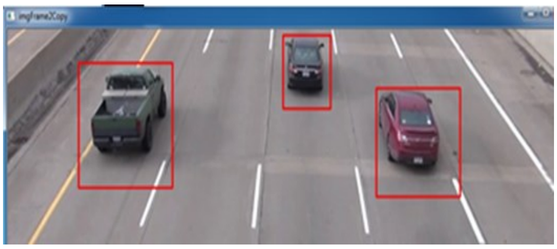

Figure 11: Enclosing recognized vehicles with rectangles

IV. Traffic analysis-classification

The classification idea here is based on comparing the dimensions of the vehicle projection in a specified area within the monitored road, as seen by the camera, with pre-calculated reference dimensions. The idea takes into consideration camera positioning (X-Y-Z location and viewing angle), as this affects the shape and size of the vehicle as seen by the camera and as depicted in Figure 12.

The classifier is intended to be simple and efficient (considering classification objectives here). One privilege here is that the classifier makes use of the outcomes of the previous algorithm steps to find the object dimensions necessary for classification.

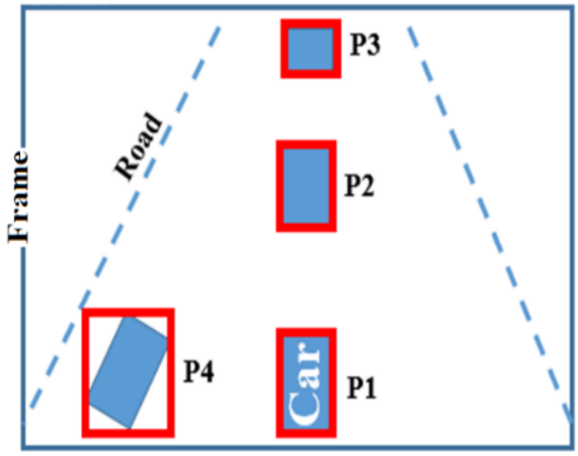

Figure 12: The effect of camera positioning on measured objects dimensions (P1, P2, P3, and P4 represent different locations for a given object)

The main step in recognizing and classifying the vehicles depends on calculating the aspect ratio, diagonal, and area for each rectangle and comparing the data with the pre calculated and saved reference categorizing data.
Comparison equations are given in Eqs. (7)- (9).

$$a >= \text{T-value} >= b \tag{7}$$

Where; a is the maximum aspect ratio value for a given vehicle category, b is the minimum aspect ratio value for a given vehicle category, and T-value is the aspect ratio for the tested rectangle. Similar comparison is done for the diagonal

$$c >= \text{E-value} >= d \tag{8}$$

Where; c is the maximum diagonal value for a given vehicle category, d is the minimum diagonal value for a given vehicle category, and E-value is the diagonal value for the tested rectangle.

$$f >= \text{K-value} >= g \tag{9}$$

Where; f is the maximum area value for a given vehicle category, g is the minimum area value for a given vehicle category, K-value is the area value for the tested rectangle. The equations used for area, aspect ratio, and diagonal are those given in Eqs. (3)-(5). The length and width mentioned in these equations regards the rectangles dimensions. The values for a, b, c, d, f, and g, given in Eqs. (7)-(9) are found according to the dimensions of the projection of the vehicle within the specific monitored area in the video frame. All measures are in terms of pixels.
For the test case of this work the values are listed below keeping in mind that these values depend on many factors that have been previously clarified in this section:
For car: Aspect ratio: a =1.4 and b =1.04, Diagonal: c = 200 and d =184, an Area: f = 20000 and g =17516
For motorcycle: Aspect ratio: a=0.86 and b=0.6, Diagonal: c=160 and d=140, and Area: f=11500 and g=10000.
For trucks: Diagonal: c=297 and d=287, and Area: f=42400 and g=41400.
The values obtained from results are listed in Table 1 considering the aspect ratio, area, and diagonal for the extracted objects blobs.
The Number of objects detected for a given category is given in Table 2. Pictorial results for detection and recognition of motorcycles, cars, and trucks are given in Figures 13-15.
The accuracy of detection and recognition of moving objects in the tests done is about 92%. The accuracy is found as:
(Accuracy = Number of correct detections and recognitions / total number of moving objects.)

Table 1: Aspect ratio, diagonal and area for the blobs of the objects detected

No. of objects	Aspect ratio (Pixel/Pixel)	Diagonal Pixel	Area Pixel^2
Truck1	------	289.933	41610
Car1	1.177	191.552	18104
Car 2	1.092	192.520	18460
Car 3	241	191.314	17880
Car 4	1.322	195.602	18408
Car 5	1.302	195.412	18445
Truck 2	------	289.933	41610
Car 6	1.241	191.314	17880
Car 7	1.302	195.412	18445
Car 8	1.146	194.74	19370
M.cycle1	0.858	148.923	10961

Car 21	1.302	195.412	18445
Truck 13	------	289.933	41610
Truck14	-------	289.933	41610
Car 22	1.241	191.314	17880
Car 23	1.302	195.412	18443
Truck15	-------	289.178	41420

Table 2: Results of object detection and recognition

Object recognized	Cars	Turks	M. cycles	Not recognized
Object count	23	15	7	4
Total objects: 49				

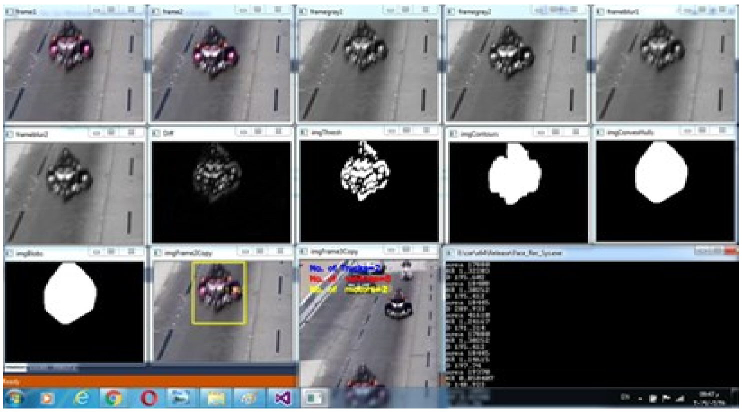

Figure 13: Steps of motorcycles detection and recognition

The cases of undetected or unrecognized moving objects represent abnormal undefined traffic cases, keeping in mind that in the work the source of data is just a single stationary camera with fixed viewing angle

An enhancement to the classifier could be done through different ways like:

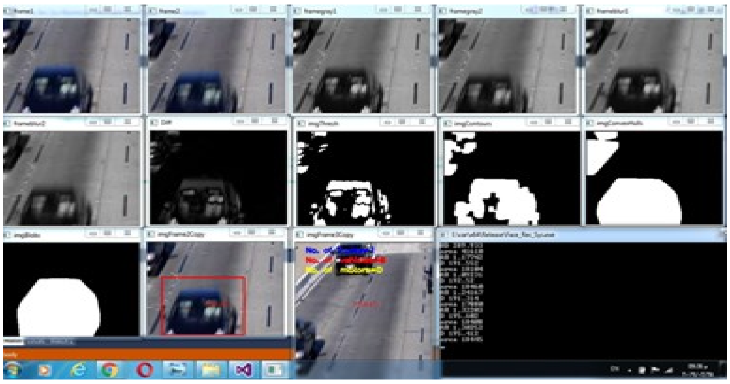

Figure 14: Steps of cars detection and recognition

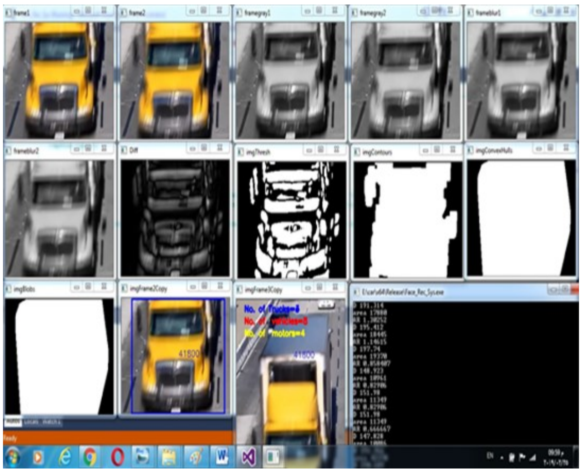

Figure 15: Steps of trucks detection and recognition

- Auto training it to classify the vehicles passing through any desired area on the road instead of pre handmade calculations for a specific area.
- Extending the pre-calculated reference dimensions to cover other vehicles classes or cases.

V. Considering another classifier

For comparison purposes, another classifier which is a CNN based deep learning classifier is built and tested (using data from Internet) for limited job which is classifying cars into two classes, and its performance is evaluated. The classifier is composed of fourteen layers which are of image input, convolution, ReLU, Max pooling, Softmax, fully connected, and classification output types. Figure 16 shows data samples of the two classes data sets used in training and testing the classifier, and Figure 17 shows a sample result of training and testing the algorithm.

Figure 16: Samples of class (1) and (2) images used by the classification algorithm

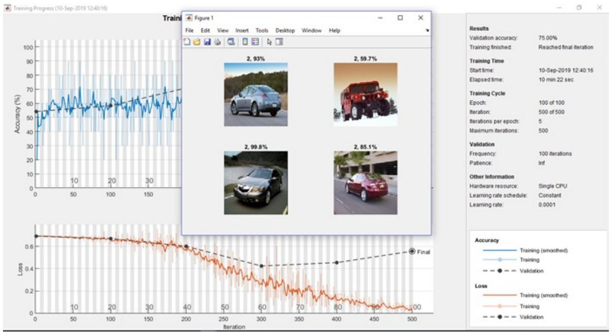

Figure 17: Sample result of the CNN deep learning classifier

Execution of the algorithm took relatively lengthy time as it involves lengthy computations. The average validation accuracy obtained for the tests done was 83%. Besides data set size and contents nature, there are many points affecting the design and performance of NN based deep learning algorithms. A brief lighting of these points is given in section (5/*III*).

VI. Traffic analysis-Vehicle speed measuring

The idea used here for measuring the speed of a given vehicle depends on measuring the time required by the vehicle to pass a certain distance. The idea is explained with the help of the video frame (image) of Figure 18.

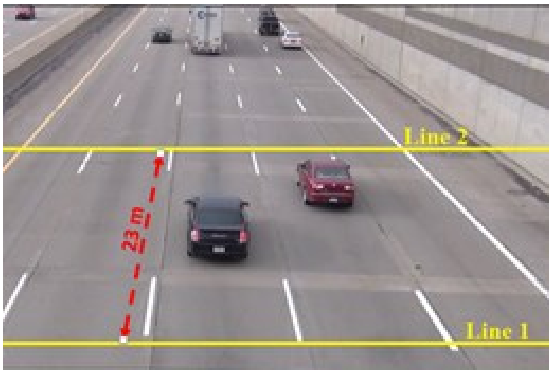

Figure 18: Depicting speed measuring idea

As could be seen, two lines are specified with a spacing distance of 23 meters. This spacing represents the distance passed by the vehicle for which taken time is measured. To get real measurements, time measuring is made depending on the frame counts and as follows:

- The video used in this test is of 32 seconds length.
- The algorithm divides the video into 983 frames, which makes time from frame to frame: ((32/983) = 0.03255 seconds/frame).
- If for a given targeted vehicle, the number of frames required for the vehicle to pass from line 1to line 2 is 20 frames, then the speed will be:

$$Speed = \frac{Distance\ from\ line1\ to\ line\ 2}{One\ frame\ time * Number\ of\ frames\ required\ to\ get\ from\ line\ 1\ to\ line\ 2}$$

$$= \frac{23\ meters}{0.03255 \frac{secondse}{frame} * 20\ frames}$$

= 35.3 meters/second = 127 Km/Hr.

A sample test result is shown in Figure 19.

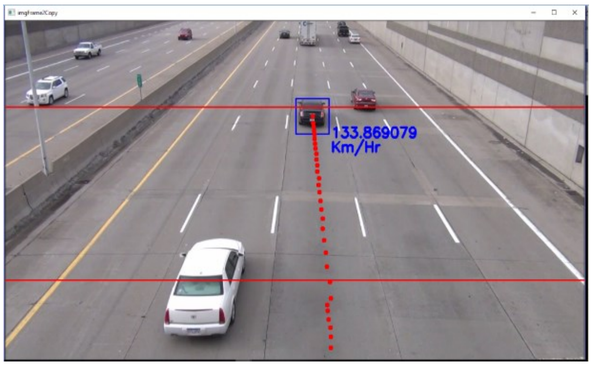

Figure 19: Measuring vehicle speed

VII. Traffic analysis-Vehicle heading

The possible heading cases are shown in Figure 20. The heading of the object depends on the changes in its X-Y position.

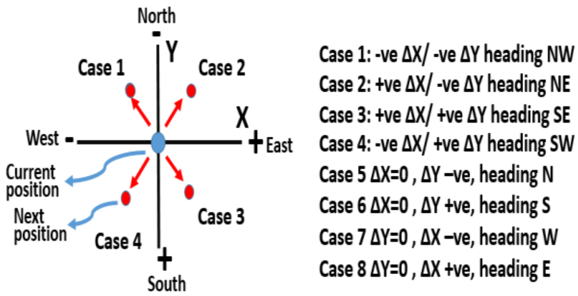

Figure 20: Possible heading cases for the object

The results obtained for three sample vehicles are shown in Table 3 which gives the X-Y changes and the headings, and Figure 21 which depicts the X-Y position for the three considered vehicles as they move.

Table 3: Sample results for finding vehicles heading

Car	ΔY	ΔX	Heading	Lane N0.
Car10	-ve	0	North	Lane3(right)
Car 6	+ve	0	South	Lane2(Left)

VIII. Traffic analysis-Vehicle tracking

The suggested algorithm involves the capability of tracking all recognized moving vehicles in the monitored scene.

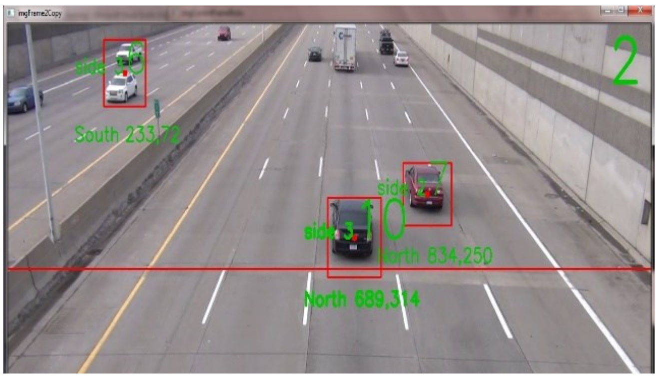

Figure 21: Vehicles heading finding

In Figure 22, a specific car object is chosen to indicate its tracking with a trail marking its path.

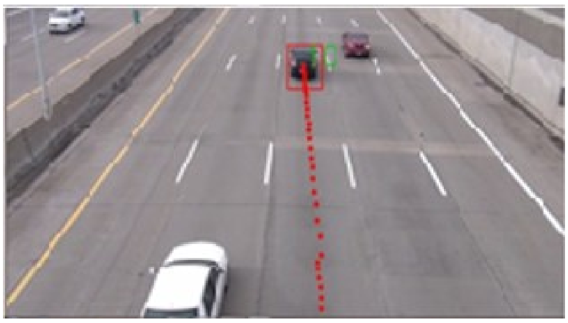

Figure 22: Vehicle tracking

IX. Traffic load

Traffic load is simply done through calculating the number of vehicles passing a hypothetical line on the frame within a given time window. Finding number of vehicles passing and time window defining have already been covered in previous sections.

5. Remarks

I. Achieving design objectives

The results obtained indicate the achievement of the work aim in designing an algorithm capable of detecting moving vehicles in traffic monitoring videos and doing some analysis work like, classifying the vehicles detected, finding vehicles speed and heading, tracking a specified vehicle, and finding traffic load. The algorithm is designed as planned to be computer vision based, use classical image processing procedures, be simple to build and modify, and to take relatively small computation time.

II. Work generality for the classical approach

The key starting point for the work, in the classical approach, is to extract the moving objects in the scene. The extracting process should end with getting fairly precise sub images for the moving objects. Once this point is achieved, a lot of analysis can be done. Some analysis cases have already been used in this work. The algorithm is built of blocks (objects), whether considering programming or functionality. Each of these blocks accomplishes a partial job within the overall job. The algorithm is very flexible to modification. Blocks, and so functions, can be added, removed, or modified to achieve different purposes or to test different procedures and concepts.

III. The modern approach classifier alg.

In general, CNN based deep learning algorithms involve very lengthy computations and so takes lengthy execution time especially when the objectives be complex and wide. Moreover, there are so many factors related to the model design and configuration as well as training for such classifiers, and what makes things more difficult is that there are no solid and fast rules to configure a neural network for a given problem, and it is not possible to calculate analytically the optimal model type or model configuration for a certain dataset. Instead, in the literature there are some accumulated helpful guiding heuristics or techniques or tips.

6. Conclusions

To decide which approach to use, classic or modern, depends on many things among which is the purpose. The modern ANN based deep learning approach is quite powerful and imposes itself as the choice for many of the advanced and sophisticated applications. On the other hand, in case the purpose is restricted to limited requirements, as we consider the case here, the logic will be to use the classical image processing procedures. In taking the right decision it is important to consider, among many things, accuracy, computation time, and simplicity of design, development, and implementation.

References

[1] A.l. Adrian, P. Ismet and P. Petru, "An overview of intelligent surveillance systems development," 13th International Symposium on Electronics and Telecommunications ISETC, Timisoara, Romania, pp. 1–6, 2018.

[2] S. Memon, S. Bhatti, L.A. Thebo, M. M. B. Talpur, and M. A. Memon, "A video based vehicle detection, counting and classification system," *International Journal of Image, Graphics & Signal Processing*, Vol. 10, No. 9, pp. 34-41, 2018.

[3] Y. Li, G. Liu, and S. Chen, "Detection of moving object in dynamic background using gaussian max-pooling and segmentation constrained RPCA," *Computer Science arXiv Articles* [Online], arXiv: 1709.00657, Sep., 2017.

[4] Y. Wang, X. Ban, H. Wang, D. Wu, H. Wang, S. Yang, S. Liu, and J. Lai, "detection and classification of moving vehicle from video using multiple spatio-temporal features," *IEEE Access*, Vol. 7, pp. 80287–80299, June, 2019.

[5] S. Gupte, O. Masoud, R.F.K. Martin, and N.P. Papanikolopoulos, "Detection and classification of vehicles," *IEEE Trans. On Intell. Transp. Syst*., Vol. 3, No. 1, pp. 37–47, March, 2002.

[6] W. Xiao, B. Vallet, K. Schindler, and N. Paparoditis, "Street-side vehicle detection, classification and change detection using mobile laser scanning data," *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 114, pp. 166-178, April, 2016.

[7] A. Raychaudhuri S. Maity, A. Chakrabarti, and D. Bhattacharjee, "Detection of moving objects in video using block-based approach," Springer, 1st Ed, Singapore, Ch. 8, pp. 151-167, 2019.

[8] H.S. Parekh, D.G. Thakore, and U.K. Jaliya, "A Survey on object detection and tracking methods," *International Journal of Innovative Research in Computer and Communication Engineering*, Vol. 2, No.2, pp. 2970-2979, Feb., 2014.

[9] X. Wang, "Deep learning in object recognition, detection, and segmentation," Now Publishers, 1st ed, Hanover, 2016.

[10] S. Johan, and S. Sakhare, "Image processing techniques for object tracking in video surveillance- a survey," Int. Conf. on Pervasive Computing ICPC, Pune, pp. 1-6, 2015.

[11] G. Sindhuja, and R. Devi, "A survey on detection and tracking of objects in video sequence," *Int. Journal of Eng. Research and General Science*, Vol. 3, No. 2, pp. 418–426, 2015.

[12] S. Shah, V. Adhikan, and V. Pokhriyal, "Motion detection algorithm based on background subtraction," *International Journal of Scientific & Engineering Research*, Vol. 4, No. 8, pp. 1945-1948, Aug., 2013.

[13] M. A. Jayaram, and H. Fleyeh, "Convex hulls in image processing: a scoping review," *American Journal of Intelligent Systems*, Vol. 6, No. 2, pp. 48–58, May, 2016.

[14] X. Poda, and O. Qirici, "Shape detection and classification using open CV and Arduino uno," 3rd Inter. Conference on Recent Trends and Applications in Computer Science and Information Technology, Tirana, pp. 128–136, 2018.

[15] S.P. Patil, "Techniques and methods for detection and tracking of moving object in a vide," *Inter. Journal of Innovative Research in Computer and Communication Engineering*, Vol. 4, No. 5, May, 2016.

[16] T. Chilimbi, Y. Suzue, J. Apacible, and K. Kalyanaraman, "Building an efficient and scalable deep learning training system," 11th USENIX Symposium on Operating Systems Design and Implementation, Broomfield, pp. 571–582, 2014.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, Vol. 60, No. 6, pp. 84-90, June, 2017.