A Differential Evolution Based Axle Detector for Robust Vehicle Classification

Deepak Dawar and Simone A. Ludwig
Department of Computer Science
North Dakota State University
Fargo, ND, USA
{deepak.dawar,simone.ludwig}@ndsu.edu

***Abstract*—Video based vehicle classification is gaining huge grounds due to its low cost and satisfactory accuracy. This paper presents a robust vehicle classification system. The system in its essence, aims to classify a vehicle based on the number of circles (axles) in an image using Hough Transform which is a popular parameter based feature detection method. The system consists of four modules whereby the output of one module feeds the next in line. We test our system on single lane highway and street traffic. When the information about the problem at hand (changing weather conditions, camera calibration parameters etc.) is limited or is dynamic, determining the Hough Transform set-up parameters manually becomes time consuming, challenging, and may often lead to false detections. This calls for finding the appropriate parameter-set dynamically according to the situation, which inherently is a global optimization problem. Differential Evolution has emerged as a simple and efficient global optimizer, and we couple it with Hough Transform to improve the overall accuracy of the classification system. We test five different variants of DE on varied videos, and provide a performance profile of all the variants. Our results demonstrate that employing DE indeed improves the system's classification accuracy (at the expense of extra compute cycles) making the system more reliable and robust.**

***Index Terms*—Differential evolution, shape detection, hough transform, vehicle classification.**

I. INTRODUCTION

Automatic vehicle classification has emerged as a significantly important element in the myriad web of traffic data collection and statistics. Regulations on road side construction for pertinent reasons, increasing vehicle density, and cost of overlaying roads are some of the factors calling for ever more efficient utilization of our existing transportation networks. A part of the solution to these pressures lies in vehicle classification systems that compute the number and type of vehicles passing a particular street or highway. This information has an evident impact on the cost and efficiency of the transportation system; road thickness decision being one of the many advantages this system has to offer. Many video based classification systems have been proposed in the past with their own advantages and disadvantages. These systems can be primarily distinguished by the type of sensors they use, most common of which are magnetic, laser, pressure, single or multiple cameras, etc. Magnetic and laser sensors tend to have a higher classification accuracy but at the same time have high equipment and installation costs, and are intrusive techniques. Computer vision based vehicle classification systems are generally attributed with low cost and accuracy, and are an active area of research. We propose a video based vehicle classification system that determines the type of vehicle based on the number of axles and distance between them. We use Hough Transform, a parameter based feature detection method, to detect the axles. The quality of the detected circles is sensitive to appropriate settings of these parameters. Since the process is time consuming and it may not be fruitful to adjust these parameters manually every time, there is always a motivation to do a parameter search by attaching a machine learning algorithm to discover an optimized set.

Differential Evolution (DE) [1], proposed by Storn and Price in 1995, is a robust real parameter optimizer in the family of evolutionary algorithms. DE has become quite popular lately and has been subjected to rigorous analysis in the past decade. It has been applied to a multitude of benchmark problems to ascertain its efficacy, and at the same time has proved quite effective in solving a broad range of real life scientific and engineering problems [2]. To add to its acclaim, DE secured first position among evolutionary algorithms at the First International Contest on Evolutionary Optimization in May 1996 [3]. One of the major reasons for its popularity lies in its simplicity as it works with a few control parameters namely the scaling factor (F), the crossover rate (Cr), and the population size (NP).

We employ DE as the real parameter optimizer to find the best suited parameters for accurate circle detection, which is a crucial part of our vehicle classification system. We show that the use of DE, apart from removing the need for setting Hough Transform parameters manually, also has the added advantage of improving the accuracy of the axle detection module, thereby improving the robustness of the overall classification system. On the other side, the process of finding the optimal Hough Transform parameters does add an extra computational cost making it a obvious case of trade-off between speed and accuracy.

The focus of this work is to propose a new system of classifying vehicles, and investigate the utility of DE to improve its classification accuracy. Due to limited space, this work keeps the former part succinct and describes the later in detail. To the best of our knowledge, no axle based vehicle classification system in the current form has been proposed before.

The rest of this paper is structured as follows. Section II describes the related work. Section III outlines and explains the proposed classification system with all its features. In Section IV, results and their analysis are presented, and Section V concludes the paper.

II. Related Work

Vehicle classification is a difficult problem to tackle. Categorizing vehicles comprehensively is quite an arduous task given the variety of vehicles and similarities between them at the same time. Different shapes and sizes within a single vehicle category adds to the dilemma. On top of this we have drastically changing weather conditions, shadows, camera noise, occlusions, etc., which make the task even more challenging. Many attempts have been made to solve this classification problem using real time (online) and recorded (offline) video. In [5], the authors describe a vehicle tracking and classification system that could classify moving objects as humans or vehicles without classifying vehicles into further subcategories. A parameterized three dimensional model for vehicle classification was presented in [6]. The model was based on the shape of a common sedan, the assumption being that in regular traffic conditions, cars are more likely to be encountered than trucks or other vehicles. In [7] and [8], the authors developed three dimensional models for various vehicles like sedans, wagons, etc., and then compared the projections of these models with features of the detected object in the image. This model was parameterized and improved in [9]. In their award winning paper [10], the authors proposed a video based detection and classification system that modeled vehicles as rectangular patches with dynamic behavior. They used vehicle dimensions, i.e. length and height, to classify vehicles into two categories: cars and non-cars. Camera orientation played a big role in determining the height of the vehicle in this case. For example, the vehicle's height was computed as a combination of width and height as it was not possible to separate the two using only the vehicle boundaries and camera parameters.

Vehicle detection, which is an indispensable part of the classification system, has been generally approached through background subtraction models. In [11]-[14], the authors used background subtraction models for the vehicle detection task. An approximated background is subtracted from the current frame to extract the foreground object, and the background is updated over time. The important challenge for the background subtraction scheme, apart from being relatively computationally expensive, is the determination of the background, which may change with changing environmental conditions, and which may affect the heuristic thresholding that the scheme utilizes.

Lately, for vehicle counting and to circumvent the problems associated with background subtraction models to some extent, time-spatial image generation models have been proposed [15]-[17]. These models aim to detect a moving object that crosses a virtual line on the video frame. For the moving objects that pass this virtual line, a time-spatial image is generated and a count of the vehicles is approximated by the number of blobs detected in that image.

Feature based techniques [5], [10] are quite popular for classifying the detected objects. These methods make use of direct or indirect geometric and statistical features extracted from the pertinent frame which is usually constructed using background subtraction models discussed above. The larger the number of features used for classification, the smaller the misclassification error, but at the same time the higher the computational load. The classification performance of these models highly depend upon the chosen background model and its adaptation through the thresholding measure used. Moreover, the performance may start to degrade if the data statistics of the dynamically updated background inches closer to the detected objects.

This work proposes an axle based vehicle classification system. The main emphasis of this work is to investigate the feasibility of using axles to classify vehicles. Identifying axles in an image is essentially a circle detection problem. Circle detection holds high significance in image analysis as is evident from its vast applications in the manufacturing goods industry, military, etc. [4]. This problem has been tackled with different approaches most common of which are:

- *Deterministic* - Hough Transform based methods [18].
- *Geometric Hashing and template matching* [19], [20].
- *Stochastic* - Simulated annealing [21], Genetic Algorithms (GA) [22], etc.

The listed methods have shown important results with some limitations. For example, template matching has shown much promise [23], but it struggles to deal with pose invariance generated from complex models. Hough Transform based methods are the most common and popularly used [24], but are relatively computationally expensive. A number of methods have been proposed to overcome this shortcoming [25]-[27]. A GA based circle detector was presented in [28], which could detect multiple circles on real images, but failed to detect the ones with less than perfect configurations. The authors in [29] proposed an optimization method as an automatic circle detector, which was a combination of DE and simulated annealing. It could detect only one circle on synthetic images and also had the drawback of converging to sub-optimal solutions.

After weighing the pros and cons of all these methods we choose Hough Transform for our investigation. The main reasons for this choice, apart from its good success rate and popularity, was its relative ease of use, simple setup, and open availability of relevant APIs for testing.

The choice of Hough Transform as the circle detection method brings another challenge to the front. It is a parameterized method that works on thresholds. The quality and number of detected circles depend largely upon the parameter thresholds, which may vary given changing intensities, illumination of pixels and other relevant features of the image. Manual settings of these parameters could prove difficult as these settings will have to be adjusted for different scenarios of traffic. To solve this problem, we use DE as the parameter optimizer and attach it to the circle detection method. We describe the details in the next section.

III. The proposed System

We present a video based vehicle classification system that categorizes vehicles based upon the number of axles and distance between them. As already mentioned, the focus of this work is to propose the idea of a different way of vehicle classification and test Differential Evolution's utility as a parameter optimizer in the process. The process essentially entails extracting relevant frames from a given video sequence, detecting axles as circles, computing distance between the farthest axles, and then classifying the detected vehicles. The proposed system, for now, works for a single traffic lane with the camera mounted on the sideways that captures the side view of the moving vehicle. A black-box description of the system is represented by Figure 1.

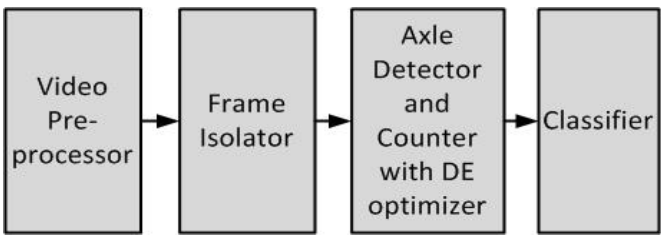


Fig. 1: Modular overview of the axle count based vehicle classification system.

A. Video Pre-processor

The video pre-processor is an optional sub-system. The main utility of this module is to reduce the video size (frame size) from the recorded/captured resolution to the one set by the user. The higher the resolution of the video, the greater is the computational cost. A low resolution video, however, will be detrimental in achieving good detection accuracy. So the frame resolution should be kept within an acceptable range.

B. Frame Isolator

This module is responsible for locating the important frames within the captured video, i.e. the frames that contain the potential vehicles in them. This module assumes fairly high importance in the sense that the more potential frames this module misses to isolate from the video, the less the number it sends to the next module for axle detection thereby reducing the overall accuracy of the system. The important frames are extracted using the background subtraction model with the background being learned and updated dynamically. Background subtraction is a relatively popular technique for frame differencing. The idea is simple. The image data of the two frames isolated at different times is compared and the difference is converted into a useful metric. The resultant metric is then evaluated against a threshold. There are many popular methods for representing image data in terms of metrics. We use the histogram representation of image data. We compare the histograms of the current background and current frame, and then apply the Chi-Square metric [30], to compare the similarity of the frames. The Chi-Square metric is calculated as:

$$D(H_B, H_C) = \sum_I \frac{(H_B(I) - H_C(I))^2}{H_B(I)} \tag{1}$$

where I is the pixel intensity, H_B and H_C are the histograms for the background and current frames, respectively, and D is the resultant Chi-Square score. A low value of this score represents a better match and vice-versa. Another well-known metric that we experimented with in conjunction with Chi-Square metric is the Bhattacharyya distance [31]. Using a combination of these two metrics added robustness to this module, but at the same time slowed down the frame isolation process to some extent. Thus, for now, we employ only the Chi-Square metric to make a decision of either discarding or sending the current frame to the next module. The data statistics of the frames, and in some cases their difference or both, are fed to a model which returns a float value. If this value is above a certain threshold, a significant difference between the frames has been detected, and the current frame is sent to the axle detector and counter module, which is described in the next section. Details of this and the next module are kept succinct due to paucity of space.

C. Axle Detector and Counter with DE optimizer

This module is responsible for counting the number of vehicles in a given frame, their axles and distance between the axles. Hough Transform is used to detect the circles. Being a parameter based detection method, Hough Transform requires that the user provides some information about the circles that need to be detected. For example, the edge detector component requires a threshold to be set for the quality of edges detected. The higher this threshold, the fewer the number of circles that are detected. The important parameters for Hough Circle detection are:

- Accumulator threshold
- Edge detection threshold
- Inverse ratio of resolution
- Minimum distance between detected centers
- Minimum radius of detected circles
- Maximum radius of detected circles

D. DE optimizer

This section is the primary focus of the work presented in this paper. All the parameters mentioned in the previous section are integers. These parameters can be tuned manually for a given scenario but the same set may show less than satisfactory performance on other test subjects. Thus, there is always a motivation to automate the process, and for that reason we employ DE to perform the parameter search. This, of course will require more compute cycles but would, at the same, improve the accuracy and robustness of the system as a whole. We test 5 DE variants to gauge their ability to perform this task effectively, and suggest the one which performs the best in terms of number of function evaluations used.

DE, being a real parameter optimizer, has to be modified to work with integer values. This essentially makes the task a combinatorial optimization problem. Truncating the real values to integer values seems a straight forward solution to this problem, but it has shown to be characteristically unstable in some cases [32]. Many novel approaches have been proposed to make DE perform the combinatorial optimization tasks and have yielded good results [33]-[34]. We utilize the approach suggested in [34] to convert integer values to float values and vise-versa, keeping all other properties of the DE variants unchanged.

After a potential frame is selected from the video, it is sent to the axle detection module. In real world applications, in general, apart from the distance between the camera and the road, other calibration parameters are usually known to the designer. This may help in determining a region of interest of the image where the vehicles are most likely to be detected. It would be computationally prudent to perform the detection and analysis on this region instead of the whole frame. As this work is primarily focused on testing the axle detection and counting approach (examining DE's effectiveness at the same time), we have steered clear of having to specify the calibration parameters of the camera and the captured scene. Instead, we have used video sequences where the distance between camera and the road is not fixed. This approach, though being relatively computationally expensive, tests the robustness of the system, and DE in particular by expanding its search space.

The fitness function for DE to optimize is kept simple. There is a cost associated with circles which are detected but are not aligned horizontally within a certain threshold. This addition of cost is based on the assumption that all the axles of the vehicle are likely to be horizontally aligned. The special case of raised axles is not considered here. Another cost is added if the radii of the detected circles differ more than a certain set threshold. This again is based on the assumption that all the axles of a vehicle are more likely to be of the same radius. There is a minimum distance between the centers that is specified and a cost is added if some circles are found to be closer than that distance. This is done to discourage DE from finding circles which are very close to each other. In mathematical form our model is represented as:

$$f(x) = (C_M)^2 \times (g(x) + h(x) + r(x)) \tag{2}$$

where

$$g(x) = (\frac{1}{C_A + \epsilon} + (C_T - C_A)) \tag{3}$$

$$h(x) = (\frac{1}{C_R + \epsilon} + (C_T - C_R)) \tag{4}$$

$$r(x) = (\frac{1}{C_D + \epsilon} + (C_T - C_F)) \tag{5}$$

and
C_M - maximum number of axles/circles to be detected in a frame; in our case we have fixed it to 10
C_T - total number of circles detected in a frame
C_A - number of horizontally aligned circles detected
C_R - number of detected circles having almost same radius
C_D - number of circles having their centroids satisfactorily distant from each other
ϵ - a very small number to avoid divide by zero error

There can certainly be many more sophisticated ways to improve this model but for our purposes we have kept it simple.

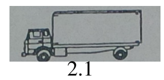
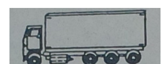
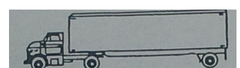
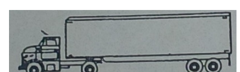
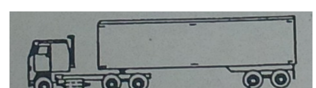
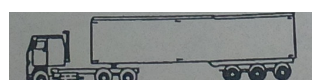
S. No.	Vehicle	Vehicle Class
1	 2.1	Passenger Vehicle
2	 2.2	Truck Type I
3	 2.3	Truck Type II
4	 2.4	Truck Type III
5	 2.5	Truck Type IV
6	 2.6	Truck Type V
7	 2.7	Truck Type VI
8	 2.8	Truck Type VII

Fig. 2: Vehicle outlines and their associated classes.

E. Classifier

Classifying vehicles based on the number of axles and distance between them does away with the need to compute other attributes of the vehicle like height, width, area, solidity, etc. Computing these additional features may improve the classification accuracy but not without increasing the computational cost. Also, the length of a vehicle can be fairly approximated as the distance between the farthest axles. Our approach also does away with the need for employing a specialized classification algorithm, for now, as there are only two features involved. We use a simple Decision Tree classifier. In future if the need arises, we might consider using a more sophisticated classifier. The current decision classes that we have experimented on, are shown in Figure 2. It should be noted that for this scheme to be fruitful, the distance between the camera and the road need to be fixed beforehand which should be considered a part of the camera calibration process. We, however, have experimented with varying distances as already mentioned and for the reasons stated in the previous section.

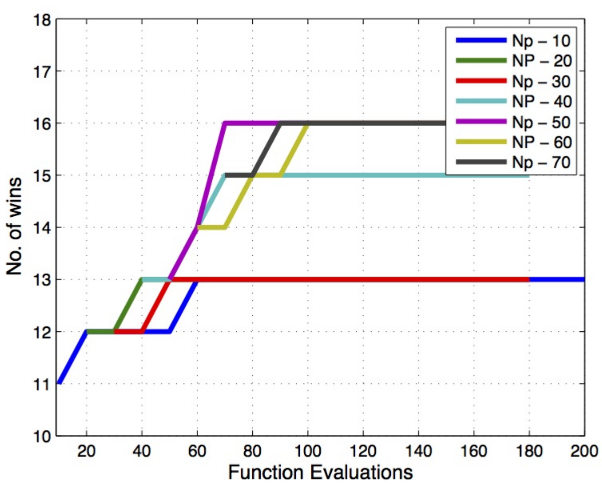


Fig. 3: Performance of DE/Rand1/bin using multiple population sizes with increasing function evaluations.

IV. Results

The performance of the system with and without the DE optimizer is presented. The videos captured were of single lane highways and streets. The traffic flow was chosen to be moderate. Table I presents the performance of five variants of DE on 18 test frames isolated from multiple video sequences, which are the outputs of the frame isolator module. The respective column value includes the best value achieved by the DE variant alongside a binary number which is shown as 1 if the DE variant was able to find the equivalent number of axles with the same centroids in the frame, i.e., excluding the false positives. The binary number is substituted as 0 otherwise. The results of a superior manually tuned parameter setting is also presented. We fixed the crossover rate (Cr) to 0.9 and scaling factor (F) to 0.5 as suggested in [35]. The maximum function evaluations was set to 300.

It is clear that DE/Rand/1/bin emerges as the best strategy among the DE variants. To improve upon the accuracy and speed of detection, we further experimented with multiple population sizes to see if that actually impacts the system's performance. The motivation is to investigate if a lower value of the population size *NP*, and for that matter fewer function evaluations, produces the same results as shown in Table I, or would a higher *NP* produce better results. *NP* cannot be too high so as to exacerbate the performance making the system untenable. At the same time, it cannot be too low as this might seriously degrade the accuracy. In essence, this problem presents the classical accuracy versus speed dilemma and we try to find the critical and harmonious set of parameters that lead to acceptable performance on this particular problem. The results are enumerated in Table II.

TABLE III: Effect of increasing NP and function evaluations on success rate. Saturation point is reported at 50-70 combination.

Population Size (NP)	Success Rate %	Saturation FEs
10	66	60
20	72	40
30	72	50
40	83	70
50	88	70
60	88	100
70	88	90

We tested multiple *NP-FEs* combinations leading to a generally expected result, i.e., performance improves with an increase in *NP* and subsequent increase in *FEs*. Table II also points out an interesting observation, i.e., an increase in the number of solutions after a certain number does not necessarily lead to an improvement in performance of DE. This phenomenon has also been corroborated by some recent publications [35], [36], though, their domain was real parameter optimization on benchmark functions. This result insinuates that there is a critical value of *NP* after which an increase in the number of solutions does not necessarily lead to an improved performance.

Figure 3 summarizes the results presented in Table II. We performed the parameter search with population size ranging between 10 and 70 with an increment of 10. Figure 3 shows that the success rate of a population size improves with an increase in function evaluations. This is an expected outcome. But after a point, increasing the population size does not improve the success rate. On similar lines, an increase in function evaluations does not offer an added advantage after a certain limit as the success rate saturates. We found that the best set of control parameters that lead to the highest accuracy (88%) among the combinations compared is: F=0.5, Cr=0.9, *NP*=50 with 70 *FEs*. Increasing *NP* above this value does not yield better results. Figure 4 visually depicts the results obtained for manual settings (left aligned in the sub-figures) as compared to DE/Rand/1/bin optimized set (right aligned in the sub-figures) discovered. Due to space constraints, only 16 of total frames are presented. Axles detected by both methods are represented by red circles.

It is imperative to note that manually setting circle detection parameters can be tedious and depends upon the scenario at hand. At the same time it is quick. Our results show that manually setting the parameters leads to a low success rate (55%) if the weather conditions and other factor are changed. At the same time, attaching an DE optimizer to the circle detection system can slow down the detection process but

TABLE I: A comparison of five variants of DE in detecting the number axles and their centers in 18 frames isolated from multiple video sequences. The values presented indicate the best/minimum value obtained by the variant along with a binary number (successful detection is represented as 1 and 0 otherwise).

Vehicle No.	No. of Axles	Manual Setting	DE/Best/1/bin	DE/Rand/1/bin	DE/RandToBest/1/bin	DE/Best/2/bin	DE/Rand/2/bin
1	2	1	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)
2	2	1	52.00 (1)	52.00 (1)	36.33 (0)	52.00 (1)	36.33 (0)
3	2	1	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)
4	2	1	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)
5	2	1	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)
6	2	1	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)
7	2	1	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)
8	4	1	36.33 (0)	29.00 (1)	29.00 (1)	29.00 (1)	29.00 (1)
9	5	1	29.00 (0)	25.00 (1)	25.00 (1)	25.00 (1)	25.00 (1)
10	5	0	25.00 (1)	29.00 (0)	29.00 (0)	29.00 (0)	36.33 (0)
11	2	0	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)
12	2	1	52.00 (1)	52.00 (1)	52.00 (1)	154.00 (0)	154.00 (0)
13	2	0	52.00 (1)	52.00 (1)	154.00 (0)	154.00 (0)	203.00 (1)
14	2	0	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)
15	2	0	154.00 (0)	203.00 (0)	52.00 (0)	52.00 (1)	52.00 (1)
16	2	0	136.33 (0)	52.00 (1)	52.00 (1)	154.00 (0)	152.00 (0)
17	2	0	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)	52.00 (1)
18	2	0	152.00 (0)	203.00 (0)	154.00 (0)	52.00 (1)	203.00 (0)
–	Wins	10	13	15	13	14	13
–	Loses	8	5	3	5	4	5
–	Suc. Rate (%)	55	72	83	72	77	72

TABLE II: Wins, Loses, and Success Rate of multiple NP-FEs combinations for DE/Rand/1/bin tested on 18 vehicular frames isolated from multiple video sequences.

Vehicle No.	10-10	10-20	10-30	10-40	20-20	20-30	20-40	20-50	30-30	30-40	30-50	30-60	40-40	40-50	40-60	40-70	50-50	50-60	50-70	50-80	60-60	60-70	60-80	60-90
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	1	1	1	1
9	0	0	0	0	0	1	1	1	0	0	1	1	0	0	0	0	0	0	1	1	1	0	1	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
12	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	1	1	0	1	1	1
16	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Wins	11	12	12	12	12	13	13	13	12	12	13	13	12	12	14	15	13	13	16	16	15	14	16	16
Loses	7	6	6	6	6	5	5	5	6	6	5	5	6	6	6	3	5	5	2	2	3	4	2	2
Suc. Rate (%)	61	66	66	66	66	72	72	72	66	66	72	72	66	66	77	83	72	72	88	88	72	77	88	88

yields a much higher success rate (88%). The system of classification that we propose, therefore, may be suited more for off-line detection and classification of vehicles where the video is pre-processed to some extent to reduce its size etc.

V. Conclusions

This work presents an axle count based vehicle classifier. Our system consists of four modules namely video preprocessor, frame isolator, axle detector, and classifier. The output of one module feeds the other in the same sequence. We used the background subtraction technique in our frame isolator module to extract pertinent frames from a video sequence. Axle detection is performed with Hough Transform, which is a well-known feature detection method in the image analysis domain. Hough Transform for circle detection works on parameters that are dependent on the image data and type of problem that is being addressed. Manually setting these parameters can be tricky, tedious, and often produces less than satisfactory results (as shown in this paper) if the weather conditions and related circumstances change. We therefore use

Fig. 4: Results obtained through manual settings (left aligned) of Hough Transform parameters vs the best settings obtained for DE/Rand/1/bin (right aligned).

a combinatorial version of Differential Evolution to optimize the parameter set.

This approach yields much higher accuracy as shown by the results we achieved. We initially tested five different variants of DE, and concluded that DE/Rand/1/bin is most suitable for this task reaching a steady success rate of 83% while excluding the false positives. We further investigated the plausibility of DE/Rand/1/bin to further its accuracy and speed. For this we tested this variant with multiple population sizes (*NP*) - *FEs* combinations. We found that F=0.5, Cr=0.9, and *NP*=50 with 70 *FEs* yields an accuracy of around 88%, and increasing *NP* further does not yield any better results. Our current system is designed to be used as an offline vehicle classifier.

To make the system perform as an online classifier, a few changes need to be made. For example, by careful camera calibration, it is possible to specify a region of interest in the test frame where the probability of finding the axles is quite high given various assumptions about inclination of the road. This will reduce the computing load considerably. If there is enough information available about the scene, it is possible to initialize DE with good values to begin with. These and other modifications are planned as future work. In addition, future work includes developing the system further by employing a parallel version of DE to increase its overall speed to make it work online, and extending it to classify multiple lane traffic.

References

[1] R. M. Storn and K. V. Price, "Differential evolution - A simple and efficient adaptive scheme for global optimization over continuous spaces," International Computer Science Institute, Berkeley, CA, USA, ICSI Technical Report 95-012, Mar. 1995.

[2] S. Das and P. N. Suganthan, "Differential evolution - A survey of the state-of-the-art," IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 4-31, Feb. 2011.

[3] R. M. Storn and K. V. Price, "Minimizing the real functions of the ICEC 1996 contest by differential evolution," in Proc. IEEE Int. Conf. Evol. Comput., 1996, pp. 842-844.

[4] L. F. D. Costa and R. M. Cesar Jr., "Shape Analysis and Classification," CRC Press, Inc. Boca Raton FL, U.S.A, 2000.

[5] A. J. Lipton, H. Fujiyoshi, and R. S. Patil, "Moving target classification and tracking from real-time video," in Proc. IEEE Workshop Applications of Computer Vision, 1998, pp. 814.

[6] D. Koller, "Moving object recognition and classification based on recursive shape parameter estimation," in Proc. 12th Israel Conf. Artificial Intelligence, Computer Vision, and Neural Networks Dec. 27-28, 1993, pp. 359-368.

[7] K. D. Baker and G. D. Sullivan, "Performance assessment of model based tracking, in Proc. IEEE Workshop Applications of Computer Vision, Palm Springs, CA, 1992, pp. 28-35.

[8] G. D. Sullivan, "Model-based vision for traffic scenes using the ground plane constraint, Phil. Trans. Roy. Soc. (B), vol. 337, pp. 361-370, 1992.

[9] G. D. Sullivan, A. D.Worrall, and J. M. Ferryman, "Visual object recognition using deformable models of vehicles, in Proc. Workshop on Context-Based Vision, Cambridge, MA, Jun. 1995, pp. 75-86.

[10] S. Gupte, O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos, "Detection and Classification of Vehicles," in IEEE Transactions on Intelligent Transportation Syatems, vol. 3, no. 1, pp. 37-47, Mar. 2002.

[11] E. Rivlin, M. Rudzsky, M. Goldenberg, U. Bogomolov, and S. Lapchev, "A real-time system for classification of moving objects, in Proc. International Conference on Pattern Recognition, vol. 3, pp. 668-691, 2002.

[12] L. Xie, G. Zhu, Y. Wang, H. Xu, and Z. Zhang, "Robust vehicles extraction in a video-based intelligent transportation system, in Proc. IEEE International Conference on Communications, Circuits and Systems, vol. 2, pp. 887-890, 2005.

[13] J. Cao and L. Li, "Vehicle objects detection of video images based on gray-scale characteristics, in Proc. IEEE International Conference on Education Technology and Computer Science, 2009, pp. 936-940.

[14] B. T. Morris and M. M. Trivedi, "Learning, modeling, and classification of vehicle track patterns from live video, IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 3, pp. 425-437, 2008.

[15] L. Anan, Y. Zhaoxuan, and L. Jintao, "Video vehicle detection algorithm based on virtual-line group, in Proc. IEEE Asia Pacific Conference on Circuits and Systems, 2006, pp. 1148-1151.

[16] J. Wu, Z. Yang, J. Wu, and A. Liu, "Virtual line group based video vehicle detection algorithm utilizing both luminance and chrominance, in Proc. IEEE Conference on Industrial Electronics and Applications, 2007, pp. 2854-2858.

[17] Y. Hue, "A traffic-flow parameters evaluation approach based on urban road video, International Journal of Intelligent Engineering and Systems, vol.2, no.1, pp. 33-39, 2009.

[18] H. K. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of Hough transform methods for circle finding," Image Vision Comput. vol. 8, no. 1, pp. 71-77, 1989.

[19] J. Iivarinen, M. Peura, J. Srel, and A. Visa, "Comparison of combined shape descriptors for irregular objects," in 8th Proc. British Machine Vision Conf., Cochester, UK, pp. 430-439, 1997.

[20] G. A. Jones, J. Princen, J. Illingworth, and J. Kittler, "Robust estimation of shape parameters," in Proc. British Machine Vision Conf., 1990, pp. 43-48.

[21] G. Bongiovanni, P. Crescenzi, and C. Guerra, "Parallel Simulated Annealing for Shape Detection," Computer Vision and Image Understanding, vol. 61, no. 1, pp. 60-69, 1995.

[22] G. Roth and M.D. Levine, "Geometric primitive extraction using a genetic algorithm," IEEE Trans. Pattern Anal. Machine Intell. vol. 16, no 9, pp. 901905, 1994.

[23] M. Peura and J. Iivarinen,"Efficiency of simple shape descriptors", Advances in Visual Form Analysis. World Scientific, Singapore, pp. 443451, 1997.

[24] H. Muammar, M. Nixon, "Approaches to extending the Hough transform," in Proc. Int. Conf. on Acoustics, Speech and Signal Processing ICASSP 89, vol. 3, pp. 1556-1559, 1989.

[25] D. Shaked, O. Yaron, and N. Kiryati, "Deriving stopping rules for the probabilistic Hough transform by sequential analysis," Comput. Vision Image Understanding vol 63, pp 512-526, 1996.

[26] L. Xu, E. Oja, and P. Kultanen, "A new curve detection method: Randomized Hough transform (RHT)," Pattern Recognition Lett. vol. 11, no 5, pp. 331-338, 1990.

[27] J. Becker, S. Grousson, and D. Coltuc, "From Hough transforms to integral transforms," in Proc. Int. Geoscience and Remote Sensing Symp., IGARSS 02, vol. 3, pp. 1444-1446, 2002.

[28] V. Ayala-Ramirez, C. H. Garcia-Capulin, A. Perez-Garcia, and R. E. Sanchez-Yanez, "Circle detection on images using genetic algorithms," Pattern Recognition Letters, vol. 27, 652-657, 2006.

[29] S. Das, S. Dasgupta, A. Biswas, and A. Abraham, "Automatic Circle Detection on Images with Annealed Differential Evolution," in 8th International Conference on Hybrid Intelligent Systems, pp. 684-689, 2008.

[30] G. W. Corder and D. I. Foreman, "Nonparametric Statistics: A Step-by-Step Approach," Wiley, New York, 2014.

[31] F. Goudail, P. Rfrgier, and P. Delyon, "Bhattacharyya distance as a contrast parameter for statistical processing of noisy optical images," JOSA A, vol. 21, no. 7, pp. 1231-1240, 2004.

[32] G. Onwubolu, and D. Davendra, "Differential Evolution: A Handbook for Global Permutation-Based Combinatorial Optimization," Springer-Verlag, Heidelberg, 2009.

[33] G. Onwubolu, and D. Davendra, "Scheduling flow shops using differential evolution algorithm," Eur. J. Oper. Res. vol. 171, no. 2, pp. 674-692, 2006.

[34] D. Davendra, and G. Onwubolu, "Forward Backward Trasformation," Differential Evolution A Handbook for Global Permutation-Based Combinatorial Optimization, pp. 37-78, Springer, Heidelberg, 2009.

[35] A. K. Qin, Xiaodong Li, "Differential Evolution on the CEC-2013 Single-Objective Continuous Optimization Testbed," IEEE Congress on Evolutionary Computation, Cancun, Mexico, June 20-23, 2013.

[36] A. P. Engelbrecht, "Fitness Function Evaluations: A Fair Stopping Condition?," IEEE Symposium Series on Computational Intelligence, Orlando, Florida, U.S.A, Dec 9-12, 2014.