8th IEEE International Conference Humanoid, Nanotechnology, Information Technology
Communication and Control, Environment and Management (HNICEM)
The Institute of Electrical and Electronics Engineers Inc. (IEEE) – Philippine Section
9-12 December 2015 Water Front Hotel, Cebu, Philippines

Machine Vision for Traffic Violation Detection System through Genetic Algorithm

Aaron Christian P. Uy, Rhen Anjerome Bedruz, Ana Riza Quiros, Argel Bandala, Elmer P. Dadios
Gokongwei College of Engineering
De La Salle University
Manila, Philippines
christianparayno_14@yahoo.com, elmer.dadios@dlsu.edu.ph

Abstract— **This paper presents a machine vision algorithm to detect traffic violations specifically swerving and blocking the pedestrian lane. The proposed solution consists of background difference method, and focuses on the genetic algorithm of the system to detect these violations. The general process is as follows: a capture picture is to be subtracted first by the reference image, then the genetic algorithm is run to find the violator, and finally a display is outputted with the corresponding type of violation. The machine vision traffic violation detection system was found to have an average convergence of about 8 iterations, within an average of less than 300 generations. These results show that the algorithm is well-suited for real time implementation in traffic detection system. Provided the system inputs were captured photos from a CCTV camera, whereas the outputs were cropped pictures of the car that was detected to have such violations mentioned earlier.**

Index Terms— **genetic algorithm, traffic violation detection, swerving, background difference method**

I. Introduction

The increasing number of cars in cities can cause high volume of traffic, and implies that traffic violations become more critical nowadays. This causes severe destruction of property and more accidents that may endanger the lives of the people [1]. To solve the alarming problem and prevent such unfathomable consequences, traffic violation detection systems are needed. For which the system enforces proper traffic regulations at all times, and apprehend those who does not comply. A traffic violation detection system must be realized in real-time as the authorities track the roads all the time [2]. Hence, traffic enforcers will not only be at ease in implementing safe roads accurately, but also efficiently; as the traffic detection system detects violations faster than humans. Consequently, a traffic violation detection system may be first realized by detecting the most common traffic violations, which are swerving and blocking the pedestrian lane. White line barriers are drawn in the road to specify the lanes that will have their corresponding directions, once a car drifts from one lane to another such that they will turn in an intersection, it is called swerving. On the other hand, blocking of pedestrian lane occurs when a car halted within the pedestrian lane.

Presented in [3] is the implementation of real-time traffic violation detection in a monitoring stream which utilized simultaneous video stream from different cameras using parallel computing techniques. Another approach of implementing real-time traffic violation detection was seen in [4], as they used video-based traffic detection through an improved background-updating algorithm, thereafter track the moving vehicles by feature-based tracking method.

The application of genetic algorithm for traffic violation detection system was not that widespread with researchers, as the typical technique used in such system is image processing related topics. Yet, there were quite few papers that used genetic algorithm for image matching like Ke's, and Li's research [5], which could be correlated with the traffic violation detection system.

This paper employs a machine vision for traffic violation detection system using genetic algorithm. The said system is assumed to have predefined parameters to acquire consistent results; hence a real-time processing is attained in the system. In addition, the violation detection system realized in this paper is limited in detecting swerving and blocking the pedestrian lane violations. The proceeding chapter discusses the process of the system implementation and the analysis of the results.

II. Genetic Algorithm

One of the functional types of machine learning is Genetic Algorithm (GA). As the name suggest it is a stimulated term from biology that adapts its behavior with accordance with the natural processes of evolution [6]. The nature of this algorithm is through evolving such that an optimal solution is obtained. Hence, it can be called as an evolutionary intelligence [7]. Theoretically, genetic algorithms are a composition of population of sequences called chromosomes. Relating to biology, chromosomes are composed of alleles, which in machine learning may indicate the answer for the given problem, thus every chromosome is a probable solution. Chromosomes may be in the form of alphabet, binary numbers [8]. Therefore, GA's are search algorithms and adapts the concept of survival of the fittest. And the answer to the problem converges to a single value, that is, the algorithm chooses the best solutions from initially random population to a generated and defined class system [9] [10].

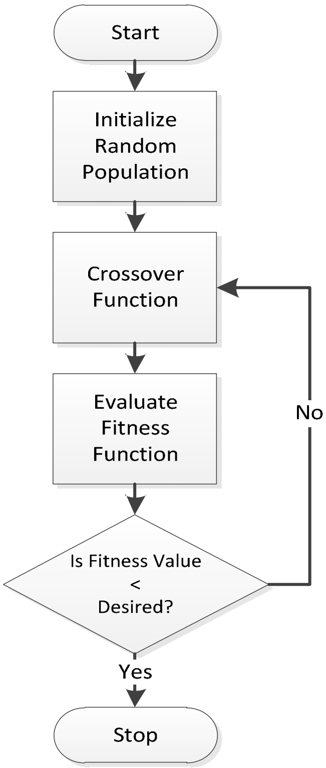

Figure 1. Genetic Algorithm Flowchart

Figure 1 shows the flowchart of the genetic algorithm of the violation detection system used in this research. There is an initial population of solutions that serve as the first generation in the algorithm. Every solution is a set of coordinate points in a single photo that represent the desirable cropped image. After which, the crossover is evaluated amongst the surviving chromosomes to generate another set of solutions. Then the fitness function will filter a classed system, and if the answer is greater than the desirable, the resulting filtered population will be looped to the crossover block and so on until the answer converges to the desirable value and the algorithm will stop.

The GA employed in this program will start by initializing random chromosomes of images in a photo in black and white format. The number of chromosomes to be initially created will depend on the dimensions of the whole photo captured from a CCTV camera, as the sizes of an average car in different cameras differ. The initial population is then evaluated using fitness function confined for the specific problem. This fitness function evaluates each chromosome, whether they are fit or not and will be sorted from best to worst fitness, then will proceed to reproduction. The procreation of the next generation will depend on genetic operation of solely the crossover to bear children to be part of the new population. After some generations, the fitness value eventually falls below the desired amount and the GA converges to an answer.

III. Processes in Detecting Traffic Violations using Machine Vision

The focus of this paper is to implement a machine vision for traffic violation detection system through genetic algorithm. Specifically, the system must detect swerving and blocking the pedestrian lane. It is implemented through Matlab in a computer with a clock processor of 3.4GHz and an 8GB of RAM, to have a nearly optimal program runtime. The input data to this system is the captured photos, constantly gathered from a specific traffic CCTV camera, which has a dimension of 480 by 850 pixels. Figure 2 shows a sample photo gathered from a CCTV camera. Whereas, the output data of this system are cropped pictures of the cars that has a violation, and along with it, is the violation description. The dimensions of cropping will be constant throughout the experimentation and is estimated as the average size of cars in the image, for this case it would be 116 by 161 pixels.

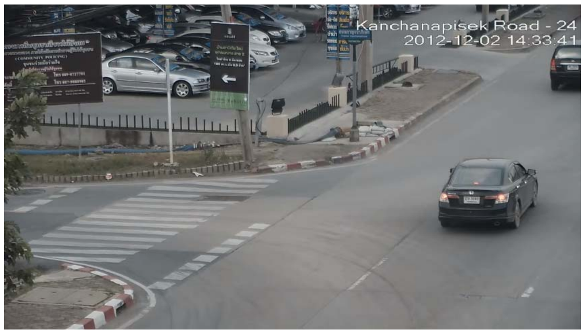

Figure 2. Sample Photo Captured from a CCTV Camera

The process comes with detecting the cars in the photo then recognizes what area in a road they covered based from the reference image, and realize genetic algorithm whether they had violated any of the specified traffic rules or not. The initial approach to the implementation of the said system is to introduce a basic image processing technique to derive the necessary parameters needed by the genetic algorithm. And the method used is the background differencing, and that a reference image is initialized to the system shown below in Figure 3.

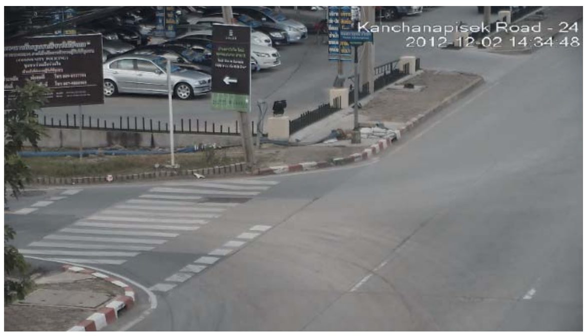

Figure 3. Reference Image

Every captured photo is first subtracted by the reference picture before processed in the genetic algorithm. Shown in Figure 4 is the integrated system. Since this paper emphasizes

978-1-5090-0360-0/15/$31.00 ©2015 IEEE

8th IEEE International Conference Humanoid, Nanotechnology, Information Technology
Communication and Control, Environment and Management (HNICEM)
The Institute of Electrical and Electronics Engineers Inc. (IEEE) – Philippine Section
9-12 December 2015 Water Front Hotel, Cebu, Philippines

the application of genetic algorithm to the system, it does not concentrate on the operation of the background difference method but tackles more on genetic algorithm.

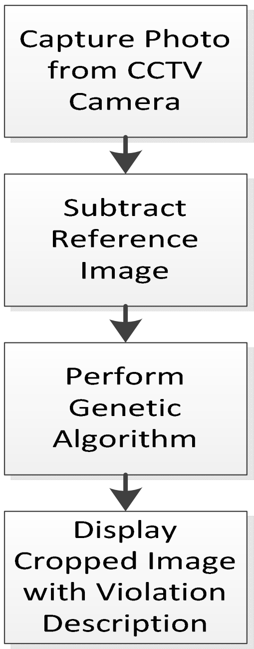

Figure 4. Block Diagram of the Implemented System

The genetic algorithm is applied thereafter. The solution works converting the image from colored to black and white, then apply GA to detect whether the captured cars in the road violates. The population of chromosomes will be a set of cropped image with equal dimensions from the derived pictures of cars. Then each will be evaluated by the fitness function, and check the remaining chromosomes for the answer. If not, the remaining individuals will undergo crossover, and the process is repeated until an answer is found. When a cropped image of car is outputted, a display will show the corresponding type of violation.

A. Fitness Function

The fitness function to be used in this paper is shown below. Equation 1 suggests an average of the total white pixels over the target image, this refer to the ratio of white to the total pixels that will show whether the area beneath each car detected contain white line/s that symbolizes either pedestrian lane or the boundary line between lanes in the road. Each image chromosomes is subjected iteratively to this equation. The selection process would be every chromosome that reaches a certain threshold to detect whether there is a violation/s incurred. The chromosome passed in the fitness test will then proceed to the crossover process to reproduce more solutions.

$$ratio = \frac{1}{C*R}\sum_{i=0}^{C}\sum_{j=0}^{R}(I) \qquad \text{(Equation 1)}$$

Where:

Ratio= white to total pixel ratio of pixels in a chromosome image
R= number of row vectors in the chromosome image
C= number of column vectors in the chromosome image
I= chromosome images

B. Reproduction

The first generation of the population is set randomly. After which, the reproduction of the next generations will come from crossover between the surviving image chromosomes. The surviving chromosome shall come after fitness evaluation. This is to ensure a class system is generated within the remaining population to converge to the answer. The reproduction process would be converting each image chromosome to a bit stream of 20 bits. This is so since the cropped image came from a 480 by 850 pixel image, and to be able to represent them with equal number of bits per dimension, each is represented by 10 bits. As 2^{10} equates to a total of 1024 while 2^{9} only equates to 512, and that when only 9 bits is used for each dimension, 512 spaces is not enough to accommodate all the rows with 850 slots. Thus, 10 bits is used to represent the row and similarly 10 bits is used to represent the column for a total of 20 bits. Correspondingly, a concatenation is done to simplify the matrix to a single row vector, and the sequence of concatenation is done consistently by having the first 10 bits be the row matrix of the image then followed by 10 bits for the column matrix of the image. Note that the image chromosome be represented by 20 bits is actually a location from the whole image. And since it is predefined in this system that a constant cropped picture of 116 by 161 pixel image is gathered with respect to the location, it can be said then that the 20 bits represents the image chromosome.

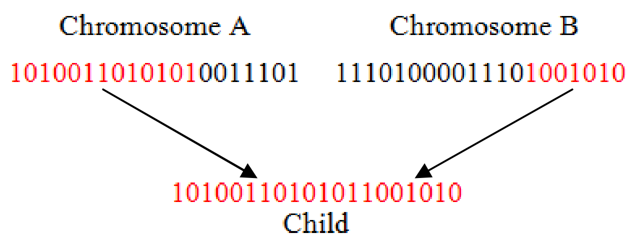

Figure 5. Sample Crossover for n=13

The crossover is done by having a random integer, n, from 1 to 20 that shall assign how many bits from chromosome A is developed to a child, while 20-n bits from chromosome B. Figure 5 shows a sample crossover between chromosome A and B. Finally, when such crossover process was verified in the system, the genetic algorithm effectively converged. Hence, the said procedure will be implemented in the reproduction scheme in the genetic algorithm.

C. Stopping Criteria

The stopping criterion is the one that differentiates the type of violation, for this case, swerving and blocking the pedestrian lane. The threshold of white is to total pixel ratio in blocking the pedestrian lane is way more than that of the threshold of

978-1-5090-0360-0/15/$31.00 ©2015 IEEE

white is to total pixel ratio in swerving. As blocking the pedestrian lane possesses greater amount of white pixels, due to the multiple white lines that correspond to the pedestrian lane, than in swerving violation, which only has a single white line to be detected by the system. In fact, it was found out to be any ratio greater than 0.1 for blocking the pedestrian lane, and ratio in the middle of 0.01 to 0.1 for swerving. As the 0.01 threshold mean the noise level floor that was either created within the image due to some process made by the system, or other external effects.

IV. Experiment Results

When the machine vision for traffic violation detection system was implemented, the process relating the image processing from the gathered photo from CCTV camera up to the output of the system is described as follows: First the system captures and takes a photo from the CCTV camera, then subtract from it the reference image, which gives the significant properties of the cars in the picture that is needed by the GA. The genetic algorithm is employed thereafter to know and detect whether each car in the scene violates or not. Finally, the system outputs cropped pictures of the car violators similar to the figures shown in the latter part. The internal process of the genetic algorithm was discussed in the preceding chapter.

Shown in the proceeding discussions are the following different inputs and the output of the system was gathered and analyzed, since not only the cropped picture of the car violated was outputted but also the crossover performance and the best fitness plot was taken note of.

Figure 6. First Original Data Captured

Figure 7. Output Picture from the System with Figure 6 as Input

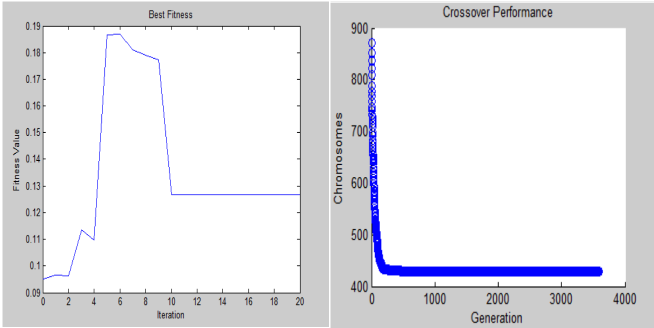

Figure 8. Best Fitness Plot and Crossover Performance for Figure 7

Figure 9. Output Picture from the System with Figure 6 as Input

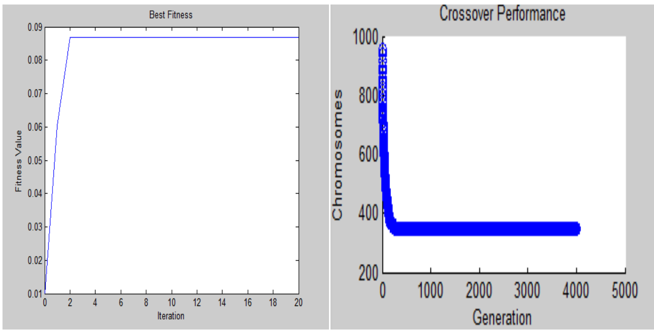

Figure 10. Best Fitness Plot and Crossover Performance for Figure 9

Shown in Figure 6 is the first data captured from the CCTV camera. Implementing the system given in Figure 6 as input obtains an output cropped picture shown in Figure 7 and 9 respectively. Figures 8 and 10 only describe the GA properties employed in deriving the corresponding outputs. The program run time was found to be 15.55 seconds.

The best fitness plot in Figure 8 shows the fitness value of the chromosomes every iterations, the shape of the curve overshoots but stabilized at the 10^{th} iteration at a value of 0.127, which proves the fitness value for the blocking the pedestrian lane comes greater than 0.1. On the other hand, the crossover performance plot in Figure 8 shows the number of chromosomes through each generation, the plot shows a convergence in the 587^{th} generation which provides the saturation of the remaining chromosomes in the system. This only means that the chromosomes reproduced for the next generations will be the same all throughout, and indeed the plot

978-1-5090-0360-0/15/$31.00 ©2015 IEEE

shows the convergence which provides that the solution is derived by using genetic algorithm. The crossover performance curve is similar to all other detections whether blocking the pedestrian lane or swerving as every detections of violation means that the system converges to an answer, for this case it is the cropped photo of the car that violated.

The best fitness plot in Figure 10 shows the fitness value of the chromosomes every iterations for the detection of swerving, the shape of the curve slowly stabilized at the 2nd iteration at a value of between 0.01 and 0.1 as expected. This proves that the fitness value for swerving do lie between the said range, which in this case lied on 0.087. On the other hand, the crossover performance plot in Figure 10 shows a similar curve in that of blocking the pedestrian lane as explained earlier but converged at the 364th generation. The early convergence for swerving was reasonable because of the iterations took for detecting swerving was less than the blocking of pedestrian lane.

To check the consistency of the system, 3 more input figures are tested to the system and shown below, and one of which includes a scene wherein there are no violators, so as to ensure the reliability of the system.

Figure 11. Second Original Data Captured

Figure 12. Output Picture from the System with Figure 11 as Input

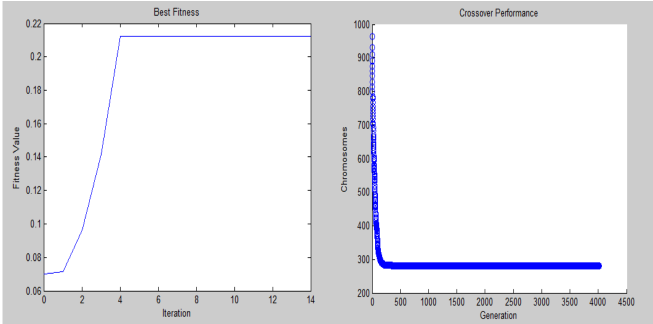

Figure 13. Best Fitness Plot and Crossover Performance for Figure 12

Shown in Figure 11 is the second data captured from the CCTV camera. Implementing the system given in Figure 11 as input obtains an output cropped picture shown in Figure 12. Figure 13 only describes the GA properties employed in deriving the corresponding output. The program runtime was measured to be 11.64 seconds.

The best fitness plot in Figure 13 similarly shows the fitness value, the shape of the curve slowly approach and stabilized at the 4th iteration at a value of 0.213, which again proves the fitness value for the blocking the pedestrian lane lies above 0.1. On the other hand, the crossover performance plot in Figure 13 shows the number of chromosomes through each generation, the plot converged in the 430th generation which provides again the saturation of the remaining chromosomes in the system, with the same reason for any violation detected.

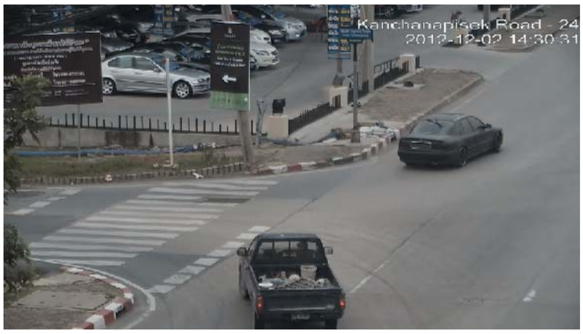

Figure 14. Third Original Data Captured

Figure 15. Output Picture from the System with Figure 14 as Input

978-1-5090-0360-0/15/$31.00 ©2015 IEEE

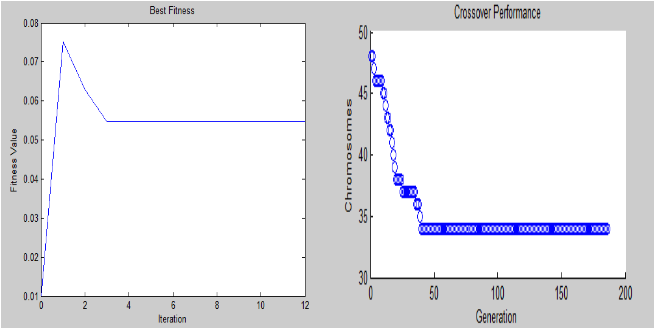

Figure 16. Best Fitness Plot and Crossover Performance for Figure 15

Shown in Figure 14 is the third data captured from the CCTV camera. Implementing the system given in Figure 14 as input obtains an output cropped picture shown in Figure 15. Figure 16 only describes the GA properties employed in deriving the corresponding output. The program runtime was determined to be 10.10 seconds.

The best fitness plot in Figure 16 likewise show the fitness value, the shape of the curve overshoots and stabilized at the 3rd iteration at a value of 0.055, which once again proves the fitness value for the swerving lies in between 0.01 to 0.1. On the other hand, the crossover performance plot in Figure 16 shows the number of chromosomes through each generation, the plot converged in the 41st generation which once again provides the saturation of the remaining chromosomes in the system, with the same reason for any violation detected.

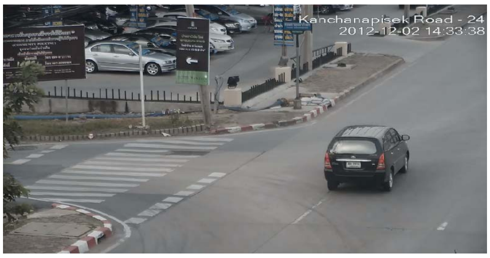

Figure 17. Fourth Original Data Captured

The fourth data input to the system shown in Figure 17 is a view of which no cars have violated in the road. As expected, the system did not have any output, which signifies and proves the operations of the system stated as planned, which was the system shall only output the cropped pictures of cars that has violated, if there is any. In addition, the runtime took 8.9 seconds to execute the whole program.

To sum it all up, the best fitness plots in either violation is said as either with overshooting curve or slowly approaching curve. Since, there is no specific distinct curve for the fitness plot for a particular type of violation, yet only one thing can be noted as both curves converge at a single value.

Meanwhile, the crossover performance plot is alike for any violations detected that it is converging dramatically at a value without overshooting, since genetic algorithm provides a sequence of evaluation after every reproduction in the system which prohibits the next population to be greater than the present population. And for the case of constant population for convergence only means that the chromosomes reproduced for the next generations will be the same all throughout as mentioned earlier, and the converged value is said to be random and insignificant since it only represents the number of chromosomes left after repeated evaluation of fitness function. Moreover, the plot shows the convergence means that a solution is derived, as every detections of violation means that the system converges to an answer.

V. Discussion and Analysis of Results

Integrating all the results and mapping the functionality of the overall system was evident in a sense that every possible input data scene was considered. The system was able to successfully detect the all the following violations under blocking the pedestrian lane and swerving. Moreover, since the system is made to constantly gather photos from a CCTV camera, the system avails itself once it is done processing the current photo.

Observing the characteristics of the type of violation in general versus the convergence, has shown that the system detects swerving violations faster than blocking the pedestrian lane violations due to the fitness requirement for swerving was less than the blocking of pedestrian lane. Thus, the system took less iterations, generations, and program runtime to converge in detecting swerving. The first data in Figure 6 provides 10 iterations and 587 generations shown in Figure 8 for the system to detect blocking the pedestrian lane violation while 2 iterations and 364 generations shown in Figure 10 for the system to detect swerving. Furthermore, comparing the second and the third data in Figures 11 and 14 respectively, shows a separate type of violations in each scene, has presented the difference in runtime for convergence other than the iterations and generations, 11.64 seconds for blocking the pedestrian lane violation while 10.10 seconds for swerving violation. Hence, it can be established that swerving detection is easier to converge than blocking the pedestrian lane detection.

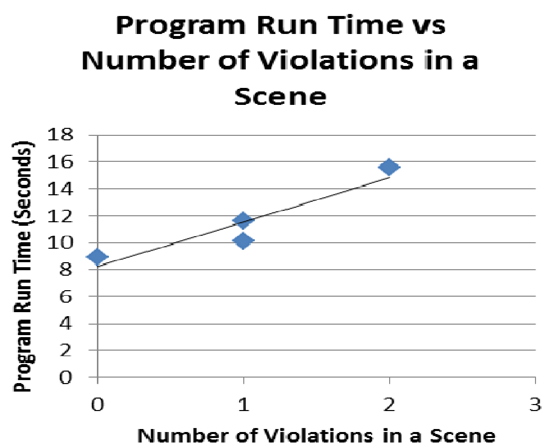

Figure 18. Scatter Plot of the Program Run Time from Each Input Data

978-1-5090-0360-0/15/$31.00 ©2015 IEEE

Another significant parameter to consider is the program runtime, which evaluates the real-timeliness of the system. Figure 18 shows the dependence of the program runtime with respect to the number of violations in the scene. As each violation undergoes testing, it is indeed that the system will took longer runtime for single photo with two car violators in it than that of single photo containing only one violator. While, it is obvious that a capture photo with no violations in it takes the least time for data processing. Assuming an average of violators occur for blocking the pedestrian lane and swerving are equal, the average runtime would be approximately 11.55 seconds. Hence, this system is capable of processing at least 5 photos in a minute. Provided that the system is implemented in with same computer specification set in part III.

VI. Conclusions and Recommendations

The designed genetic algorithm was effectively able to detect the type of violation specified on this paper which are blocking the pedestrian lane and swerving respectively. The convergence of detection for the two kinds of traffic violations mentioned is dissimilar, since there each has a different threshold condition. The system provides detection for both violation but detects swerving violations faster than blocking the pedestrian lane violations. Further, the system is able to process one data at a time. Also, the program runtime is somewhat slow, and can be improved by using a computer with high speed processor specifications.

Future research about the application of the designed genetic algorithm for other advanced image processing techniques. Since, this may improve the program runtime of the system by neglecting other unnecessary steps done in a background difference method. A machine vision algorithm may be done instead to provide more intelligence in the system.

References

[1] Y. Artan, O. Bulan, R. P. Loce and P. Paul, "Passenger Compartment Violation Detection in HOV/HOT Lanes," in *2014 IEEE Winter Conference inApplications ofComputer Vision*, Steamboat Springs, CO,USA, 2015.

[2] X. Zhang, Y. Tian, L. Liang, T. Huang and W. Gao, "MACRO-BLOCK-LEVEL SELECTIVE BACKGROUND DIFFERENCE CODING FOR SURVEILLANCE VIDEO," in *2012 IEEE International Conference on Multimedia and Expo*, Beijing, China, 2012.

[3] G. Ou, Y. Gao and Y. Liu, "Real-TimeVehicularTrafficViolationDetectioninTrafficMonitoringStream," in *2012 IEEE/WIC/ACM* , Beijing, China , 2012.

[4] X. Wang, L.-M. Meng, B. Zhang, J. Lu and K.-L.Du, "A Video-based Traffic Violation Detection System," in *MEC*, Shenyang, China, 2013.

[5] Y. Ke and D. L. Yuhao Li, "Image Matching using Genetic Algorithm on GPU," in *IEEE*, Tianjin, China, 2011.

[6] S. Sumathi, T. Hamasapriya and P. Surekha, Evolutionary Intelligence: An Introduction to Theory and Applications with Matlab, Berlin: Springer, 2008.

[7] G. E. U. Faelden, J. M. Z. Maningo, R. C. S. Nakano, A. A. Bandala and E. P. Dadios, "Blind Localization Method for Quadrotor-Unmanned Aerial Vehicle (QUAV) Utilizing Genetic Algortihm," in *7th IEEE International Conference Humanoid, Nanotechnology, Information Technology Communication and Control, Environment and Management (HNICEM)*, Puerto Princesa, Palawan, Philippines, 2013.

[8] S. C. Barrido and E. P. Dadios, "Online Robot Tracking Using Genetic Algorithms", in *Intelligent Control, 2002 Proceedings of the 2002 IEEE International Symposium on Intelligent Control*, Vancouver, Canada, 2002.

[9] M. A. Ferreira and H. K. Lee, "Multiscale Modeling: A Bayesian Perspective," Springer, Chicago, 2007.

[10] M. Cepin, Assessment of Power System Reliability: Methods and Applications, Chicago: Springer, 2011.

978-1-5090-0360-0/15/$31.00 ©2015 IEEE