2014 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)
July 6-11, 2014, Beijing, China

Fuzzy Adaptive Cruise Control System with Speed Sign Detection Capability

Raazi Rizvi[1], Shivam Kalra[2], Chirag Gosalia[3], Shahryar Rahnamayan[1]

***Abstract*— Advanced Driver Assistance System (ADAS) is one of latest innovations in the auto-mobile industry and has become a premium feature in many luxury vehicles. ADAS assists drivers by integrating multiple safety and convenience features into a single system. Current ADAS technology usually comprises of an Adaptive Cruise Control (ACC) system in combination with one or more warning/prevention systems. Such as lane departure, collision avoidance, and parking assist systems. This paper outlines a fuzzy logic based ADAS with integrated speed sign detection (SSD) capability. The described system improves safety of the vehicle by dynamically adjusting the speed of the ACC in accordance with the speed limit of the road. The proposed ADAS system will be helpful in reducing speeding violations and enhancing smoother cruise control in heavy traffic conditions. All system design, implementation and testing was done using the MATLAB development environment, and TORCS virtual car simulator.**

I. Introduction

DISTRACTED driving is one of the major causes of collisions in North America [1]. The United Nation (UN) reported in 2004 that on average, 1.2 million people were killed and 20-50 million people were injured throughout the world in the road crashes each year [2]. In US driver distraction and speeding account for 42% of auto-mobile related fatalities [3]. Over the past decade auto-mobile manufacturers have implemented adaptive cruise control systems (ACC) in lieu of a conventional cruise control. ACC systems introduce intelligence into vehicles through embedded controllers and sensors technologies which help maintain a safe distance between cars. Control is usually accomplished by a proportional-integral-derivative (PID) based system as reported in [8]. Auto-makers are slowly transitioning from ACCs to advanced driver assist systems (ADAS). ADAS comprise of one or more warning, prevention, or convenience systems such as, adaptive ACC, Lane Departure Warning System (LDWS) and/or Collision Avoidance System (CAS).

Vehicle automation systems such as ADAS help the risk of accidents, improves vehicle safety, optimizes fuel consumption, enhances the overall comfort of auto-mobiles [3]. Research regarding ADAS have been ongoing for the past decade [17] and is still growing because of the increasing demand for safer vehicles. By continuously monitoring the road for potentially dangerous situations, ADAS either alerts or assists drivers to avoid collisions. The primary benefit of ADAS is that it is not inhibited by factors such as fatigue, stress, or distraction. Implementation of an ADAS mitigates much of the risk associated with driving by verifying most of the drivers checks and assessments through an embedded system. Reports regarding existing systems can be found in [13]–[30].

[1]S. Rizvi and S. Rahnamayan are with Department of the Electrical, Computer, and Software Engineering, University of Ontario Institute of Technology, Oshawa, Ontario, Canada (email: {syed.rizvi, shahryar.rahnamayan}@uoit.ca).

[2]S. Kalra is a software developer at Klick Health, Toronto, ON, Canada (email: skalra@klick.com).

[3]C. Gosalia is a software developer at RSI, Oshawa, ON, Canada (email: chirag.gosalia@outlook.com).

Traditionally an ADAS has been aimed at maintaining a minimum safe distance between two vehicles to avoid collisions by dynamically adjusting the speed, and monitoring other vehicles on the road. Some available systems rely on the global positioning system's (GPS) data to determine the current road speed [9]–[11]. The problem with relying on GPS is that it must be regularly updated in order to maintain accuracy; and also for some streets and roads, speed limit data is either not available or inaccessible (loss of signal) in the corresponding databases. Recent reports and surveys have suggested the implementation of variable speed limits in both the EU and Canada [5][6]. In order to circumvent the current availability and validity limitations of GPS, speed sign detection (SSD) system is gaining popularity. Camera based systems which are already used for lane detection, distance measurement, and etc. can also determine the speed limit of the current road by reading posted speed limit signs. We propose a fuzzy logic based ACC in combination with a SSD system to create a novel advanced ADAS. The proposed system's primary function is to reduce distracted driving and speeding which are leading causes of collisions. A Canadian survey reported that 75% respondents felt that a self-monitoring system would be successful in reducing the speeding by alerting drivers when they exceed the current road speed limit [7].

II. Main Framework

The proposed ADAS comprises of a fuzzy logic based adaptive cruise control (ACC), and video based speed sign detection system (SSD). Inputs to the ADAS are readings from distance sensors and camera, which provide the inter-vehicle gap size and video data for the road. The system architecture is shown in Fig. 1. Inputs to the system are are the distance between vehicles, and video stream data. Correspondingly the system outputs are throttle and brake. Throttle and brake are the control parameters for the vehicle,

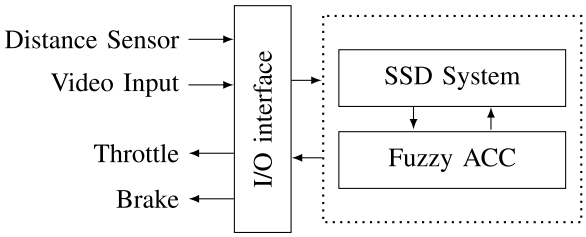


Fig. 1: System architecture for the proposed ADAS

and can be further analyzed to generate warnings as well, but in this paper, these parameters are used solely vehicle control.

One of the primary task of the ACC consists of being able to maintain a specific safe distance to vehicle(s) ahead of it. As such control process in which drivers attempt to maintain a safe distance between their car and the proceeding vehicle by accelerating/ decelerating is referred as car-following [28]. In order to obtain a comfortable driving experience with an ACC system, it is imperative that car-following behavior of ACC system must resemble the human driver. Partha Chakroborty and Shinya Kikuchi concluded that fuzzy logic is a significant choice for implementing the car-following controller as it is based on approximate reasoning which mimics the behaviour of human drivers [31].

III. Fuzzy Logic Controller

A. Background

Drivers face the daunting task of making complex decisions in a short time-frame. As such this leaves no margin for an error as a difference of fraction of a second could mean life or death. There are various factors which a driver must consider in order to make a safe decision. Crucial among these is, maintaining a safe distance between the proceeding vehicle. A fuzzy logic controller (FLC) which mimics the humanistic reasoning can demonstrate a safe real-time decision making process in complex scenarios [25]. The fuzzy logic based controller was introduced by Lotfi Zadeh for systems which were difficult to control using conventional methods. FLC's are based on Zadeh's fuzzy sets [32]. Since their introduction, FLCs have been used in many applications of intelligent transportation systems. C. Kim used fuzzy logic-based method to make an optimal decision for an autonomous vehicle [33].

The primary role of the FLC in the proposed ADAS is to maintain a safe distance to proceeding vehicles while adhering to posted speed limits. The FLC consists of three primary stages which include: 1) a fuzzification of antecedent for each rule, 2) inference rules based on fuzzy operators and aggregating the results, and 3) de-fuzzification of the aggregated fuzzy-set from previous step into "crisp" output values [26].

For ACC applications, fuzzy based systems are generally preferred because: 1) they are based on approximate reasoning and do not exhibit definitive behaviour compared to mathematical models used in probabilistic approaches [8], and 2) they are more flexible in maintaining or changing the model as per the requirements. FLC systems use $if-then$ rules and human friendly linguistic variables to express the knowledge, unlike PID controllers which are dependent on a fixed mathematical model. Some hybrid approaches have been attempted but have demonstrated varying results [8][27][29][30].

B. Proposed Algorithm for the Controller

The proposed controller uses fuzzy logic to adjust vehicle speed while driving. The FLC adjusts the $throttle$ and $brake$ values, which control the acceleration and deceleration of the vehicle. Values for both the $throttle$ and $brake$ range from 0 to 1, where 0 means no brake or throttle has been applied; relatively a value of 1 means maximum brake or throttle applied on vehicle, as outlined in the following equations:

$$throttle_{actual} = throttle_{max} * throttle \quad (1)$$

$$brake_{actual} = brake_{max} * brake \quad (2)$$

Input signals are d, s, s_l, p_t, and p_b. They are defined as follows:

$$d = \text{distance from proceeding vehicle} \quad (3)$$

$$s = \text{current speed of vehicle} \quad (4)$$

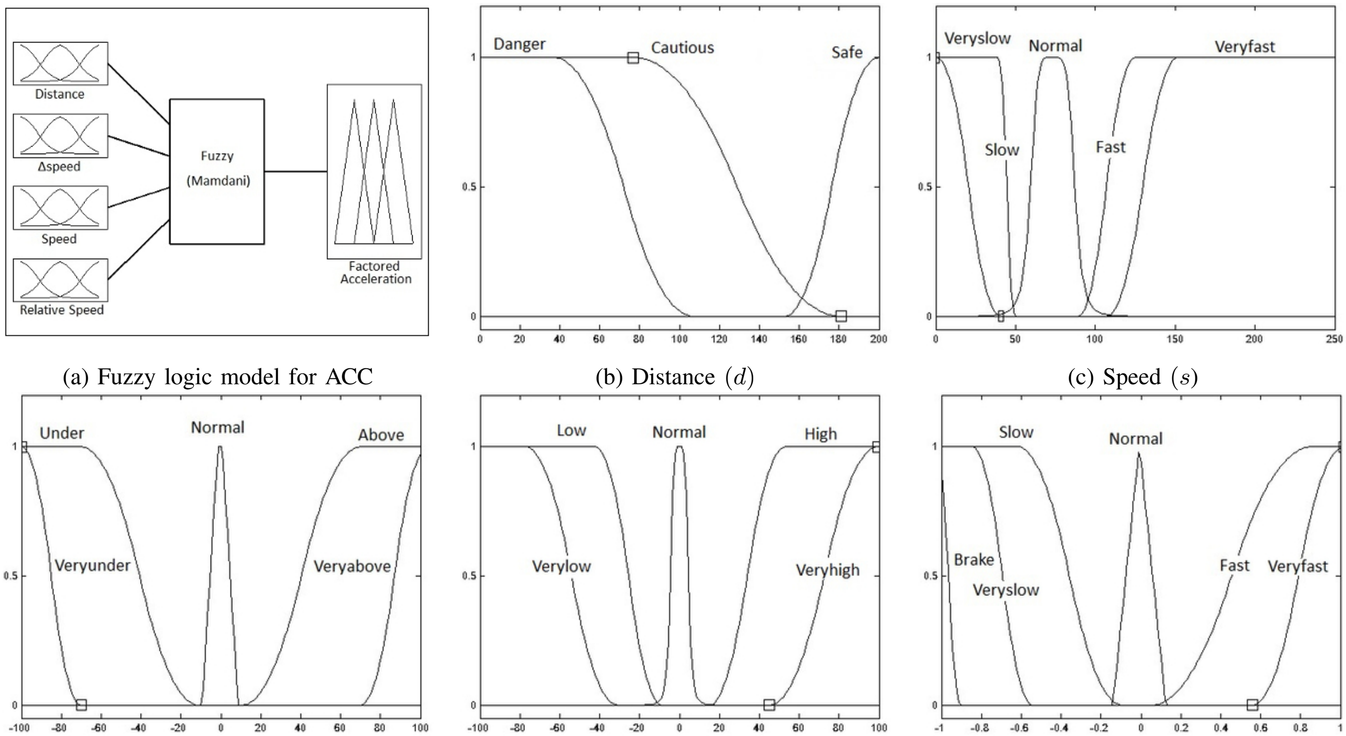
$$s_l = \text{current road speed limit} \quad (5)$$

$$p_t = \text{depression value of acceleration pedal} \quad (6)$$

$$p_b = \text{depression value of brake pedal} \quad (7)$$

As in most cruise control systems, control of the vehicle is split between the fuzzy logic controller (FLC) and the driver. The driver always has the capability to override the FLC. In cases, where the driver decides to accelerate or slow down they push the accelerator or brake pedal on the vehicle which results increases the values of p_b and p_t. If the values of p_t or p_b are greater than 0 the ADAS reverts control of the throttle and brake from the FLC to the driver. Conversely if p_b and p_t are equal to 0, the ADAS sets them equal to the values calculated by the FLC (outlined in algorithm 1). The algorithm is invoked for every new set of data retrieved from the sensors. In practical implementations, the ADAS would update the FLC with sensor data according to a specific sample rate, which would be dependant on the sensor sample rate.

In Algorithm, 1 line 9, $evalfis$ function is the core component of the FLC. $evalfis$ decides acceleration based on four primary factors: 1) distance from vehicle proceeding vehicles, 2) the current road speed limit, 3) the current vehicle speed, and 4) the speed of proceeding vehicle. The $evalfis$ function evaluates the underlying fuzzy model, based on its inputs, and outputs factored acceleration ranging from -1 to 1 (where negative value corresponds to brake and positive to throttle).



(a) Fuzzy logic model for ACC (b) Distance (d) (c) Speed (s)

(d) Delta speed (Δs) (e) Relative speed (s_r) (f) Factored acceleration (f_acc)

Fig. 2: Fuzzy logic model and corresponding membership functions for the proposed system.

Algorithm 1 Algorithm for Fuzzy Logic Controller
Input: d, s, s_i, p_t, p_b
Output: $throttle, brake$

$throttle \leftarrow 0$
$brake \leftarrow 0$
if p_i >0 *or* p_b >0 **then**
  $throttle \leftarrow p_i$
  $brake \leftarrow p_b$
else
  $\Delta s \leftarrow \frac{\Delta d}{\Delta t}$
  $s_r \leftarrow (s - s_I)$
  $acc_f \leftarrow evalfis(d, s_r, s, \Delta s)$
  if $acc_f > 0$ **then**
    $throttle \leftarrow acc_f$
  else
    $brake \leftarrow |acc_f|$
  end if
end if

C. Inputs/Outputs and Membership Functions

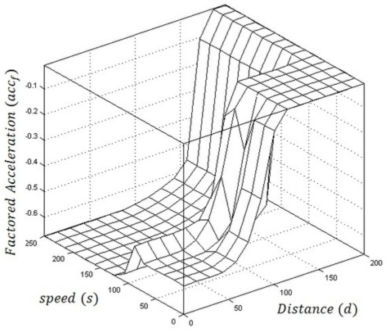
The fuzzy logic model $(evalfis)$ used by the FLC accepts 4 inputs $(d, s_r, s, \delta s)$ and produces a single output (acc_f). The inputs were carefully chosen to provide all the necessary information for controlling a vehicle under varying situations. The membership functions for these inputs are shown in Fig $2.(b)-(e)$, and were initially chosen to be symmetric shapes with 50% overlap. Tuning procedures were applied during which left and/or right spread and/or overlapping of member functions were changed. This process was continued until the desired accuracy was achieved through simulation. The input variables are defined as:

1) Distance (d): Fig. $2.(b)$, is measure of distance between front car and ACC equipped car; it is measured using distance sensor attached to car.
2) Speed (s): Fig. $2.(c)$, is current reading on the speedometer of the vehicle.
3) Delta Speed (Δs): Fig. $2.(d)$, is the rate of change of the distance between ACC equipped car and any vehicle in front as outlined in Algorithm 1.
4) Relative Speed (s_r): Fig. $2.(e)$, is difference between current speed of vehicle (s) and road speed limit (s_l).

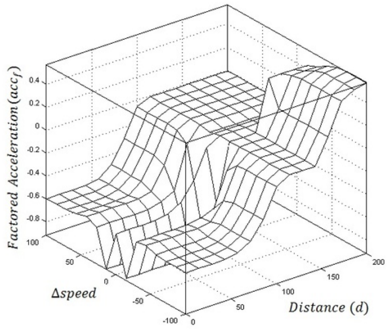
Evaluating the fuzzy logic model we obtain the factored acceleration (acc_f) which determines the throttle and brake values for the vehicle. The membership function of the output is shown in Fig $2.(f)$. The 3D plots in Fig. 3 shows the resultant factored acceleration (acc_f) caused by varying any two of the inputs. It can be inferred from Fig. 3 that: 1) system output and input relationship is very complex; 2) system behaviour changes as result of small changes to defined rules and/or membership functions. Thus verifying the approximation and indefinite behaviour of fuzzy systems.

D. Defuzzification

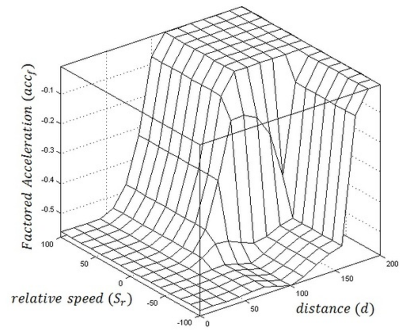
De-fuzzification is the process of obtaining 'crisp' or quantified value from fuzzy sets following rule inferencing and aggregation. The FL model evaluates all applicable rules,



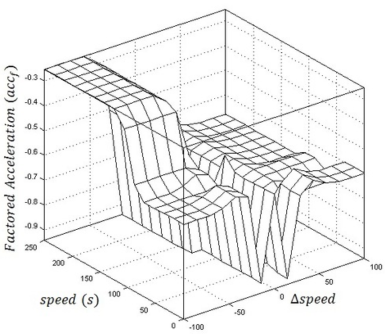
(a) Speed (s) and Distance (d)



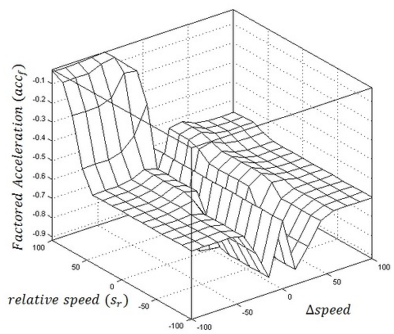
(b) Delta speed (Δs) and Distance (d)



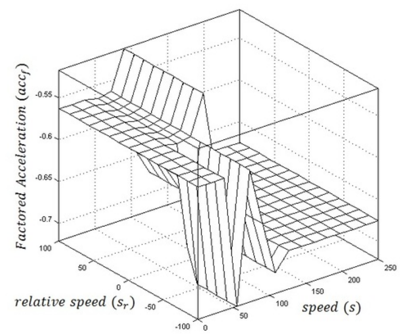
(c) Relative speed (s_r) and Distance (d)



(d) Speed (s) and Delta speed (Δs)



(e) Relative speed (s_r) and Speed (s)



(f) Speed (s) and Relative speed (s_r)

Fig. 3: Surface plots showing variation in output based on each of the two inputs.

based upon the membership functions and input values, a final result. The result from the rule base is then mapped to the output membership function output (acc_f). The mapped results from each and every rule are subsequently aggregated and transformed into a 'crisp' values of acc_f using a centroid based de-fuzzification technique.

$$x^* = \frac{\int \mu_i(x) * x * dx}{\int \mu_i(x) * dx} \tag{8}$$

The centroid de-fuzzification is expressed in Eq. (8), where x^* is the de-fuzzified output, $\mu_i(x)$ is the aggregated membership function and x is the output variable.

IV. Speed Sign Detection

Recognizing speed limits from North American traffic signs is a challenging task. Conventional approaches, such as [34] and [35], utilize the colored background or border of traffic signage to isolate the sign from its surroundings. However, North American speed signs are black and white, thus cannot be easily detected using conventional color based segmentation methods. Another difficulty is the fact that the reflective surface of the speed sign reflects into the camera. Thus color of the sign highly dependant on the time of day and the presence of external light sources (such as headlights). It is more effective to consider the shape profile of the sign as suggested in [38] to localize specific shape descriptors.

A. *Proposed Algorithms for Sign Detection and Speed Limit Recognition*

The approach shown below focuses on extracting the speed sign's rectangular shape through pixel-wise operations, rather than color based segmentation.

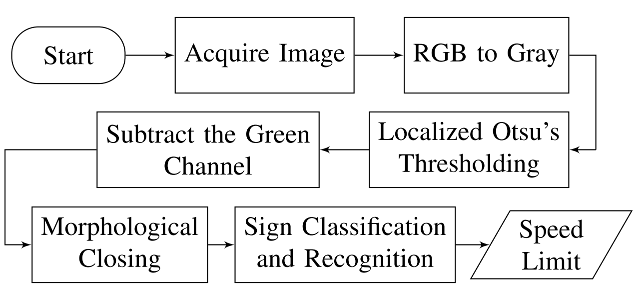


Fig. 4: The speed sign detection and speed limit recognition algorithm

B. *Image Thresholding*

After the image, I, has been acquired by the camera, it is split into n blocks using Eq. (9). Each block is thresholded using Otsu's method as presented in [37].

$$Block_i = \sum_{x=ni}^{n(i+1)} \sum_{y=ni}^{n(i+1)} I(x, y), i = 0, \ldots, n-1. \tag{9}$$

A binary image is reconstructed from these blocks and is subtractively masked with the green channel of I to remove noise from the image. Filtering out any foliage such as leaves and branches which like speed signs are highly reflective. Fig. 5 shows the result of the filtering out the green channel significantly reducing the noise.

Fig. 5: (a) shows the original image and (b) shows the locally thresholded image with the subtracted green channel

C. Morphological Operations

After thresholding the image I, the task of isolating the speed sign becomes easier. The speed sign is a relatively small object compared to the rest of the image, but it has a distinct rectangular shape which is constrained by Eq. (10). This relationship originates from standardized speed sign regulations which can be found at [4]:

$$L_{sign} > 0.8 * W_{sign} \tag{10}$$

Morphological operations such as $filling$ and $opening$ are used to process the image even further. $filling$ operator fills any holes in enclosed within the image boundaries, and the $opening$ operator morphs all objects in the image with the rectangular structural element (SE) as shown in Fig. 6. These operations are explained in detail in [38]. The SE is dictated by the image size and follows the same constraints as Eq. (10) and is defined in Eq.s $(11)-(12)$. After opening the image with the structural element, SE, the sign is extracted from the image by comparing all the rectangles, based on criteria in Eq.s $(13-14)$. Where the $\frac{Area_{filled}}{Area_{box}}$ is the measure of how much each potential sign from Fig. $6.(b)$ conforms to its bounding box (a rectangle taken from two opposite corners). Eq. (14) shows how well the the bounding box of potential signs conforms to the constraint Eq. ((10)). After potential signs have met the constraints posed in Eq.s ((13) and (14)), they proceed to the detection stage.

$$\begin{aligned} Length_{SE} &= 0.05 * Length_{image} && (11)\\ Width_{SE} &= 0.05 * Width_{image} && (12)\\ \left|\frac{Area_{filled}}{Area_{box}} - 1\right| &\leq 0.2 && (13)\\ \left|\frac{Y_{box}}{X_{box}} - 0.8\right| &\leq 0.1 && (14) \end{aligned}$$

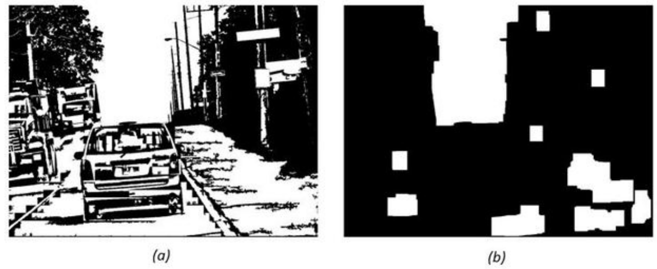


Fig. 6: (a) shows the filled holes image and (b) shows the image after it has been opened with the square SE.

D. Speed Limit Recognition

After thresholding and morphological operations are preformed and a number of candidate solutions (regions of interests) are detected. In order to recognize the speed sign from these potential candidates, an artificial neural network (ANN) is employed to classify them based on speed limit. The results classification are as follows: $i)$ Is the candidate is a speed sign, and $ii)$ what is the detected speed limit. After the speed limit is extracted from the sign the information is relayed to the ACC system.

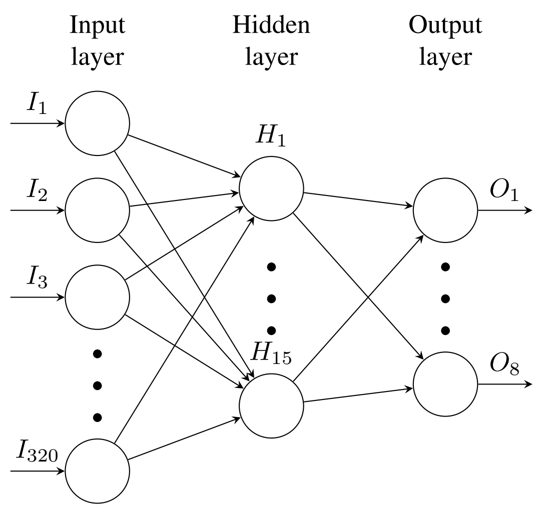


Fig. 7: ANN diagram for speed sign recognition.

The individual candidate regions (i.e., rectangular boxes) are thresholded and converted to binary images. It then re-sized to 16×20 pixels images which are flattened out to 1-D array consisting of 320 bit values which are subsequently fed into a trained ANN model. The trained ANN compares the input bits and evaluates them based on the known input-output relationships to match it to the corresponding speed limit if applicable. The ANN outputs a 8 values between [0,1] value corresponding to a specific speed limit (10,30,40,50,60,70,80,100). The candidate is rejected if none of the output values from ANN is greater than 0.4. A candidate whose ANN output has at least one value that is greater than or equal to all other values for all candidates is classified as a speed sign. The candidates classified as road signs have their ANN outputs analysed and compared with the last detected road sign to make a hard decision on the posted limit.

The ANN is trained using set of 150 images, out of which 20% of images do not contain speed signs and remaining images consist of equivalent distribution among 8 of the speed signs. Images in the training set are taken in different light and weather conditions. Within some of the images, noise and other deformations have been added to diversify the training set. The ANN has a single hidden layer, ans is a Sigmoid neural network with 15 nodes in hidden layer and 8 output nodes. The structure of the ANN is shown in Fig. 7.

E. Test data set and measured accuracy

For this study a series of 40 videos (captured during 20 day and 20 nights times) were used to measure the accuracy of the SSD. The proposed algorithm was run for every 10 frames of a video sequence and provided the current road speed (if detected) to the fuzzy controller. In the event that no change in road speed was detected the system simply used the most recently detected speed limit. The accuracy of the system is measured at almost 80% with a false positive rate of 16%.

Fig. 8: (a) shows the successfully recognized speed sign, and (b) shows an image with a false positive recognition.

The high false positive detection rate can be attributed mainly due to misclassification of speed signs by the ANN. On average there were 5 potential speed sign candidates each time the algorithm was run classified as traffic signs. The systems accuracy can be greatly improved by validating detected speed limits against ones obtained via GPS [12]. Researchers looking to improve the performance of our proposed algorithm can consider utilizing support vector machines (SVM) approach to narrow down the potential sign candidates rather than using the fixed constraints.

V. Simulation Results

The simulation of system was conducted using an open source car-racing platform, Open Racing Car Simulator (TORCS). TORCS API provides users with the ability to control a simulated car within a game by controlling variables: 1) steer, 2) throttle, and 3) brake. It also allows users to query the current state of car and other external parameters of the game's environment such as speed, coordinates of other cars, and road geometry.

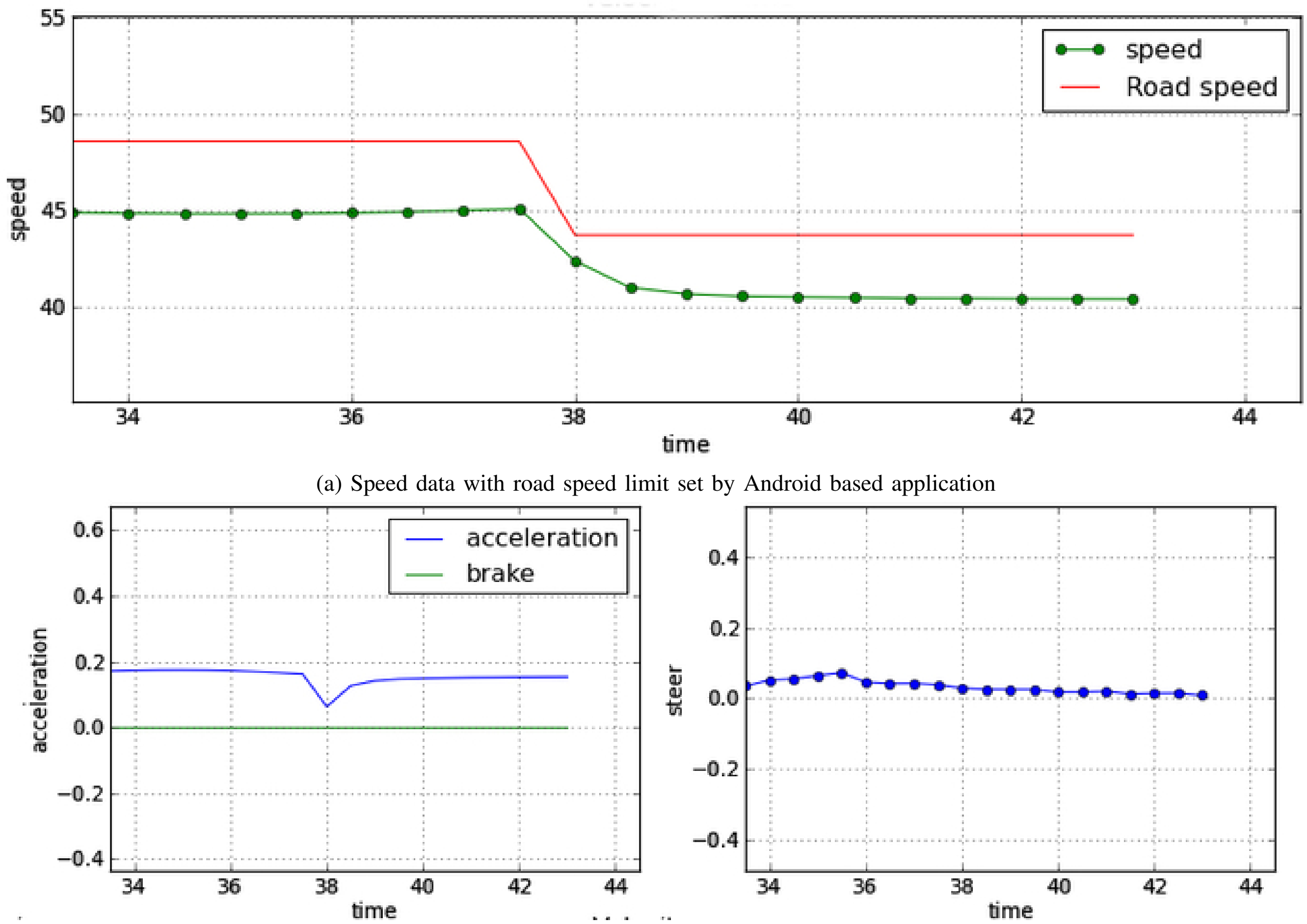
The simulation is conducted on the x86 64 bits GNU/Linux machine installed with TORCS software. The system consists of Intel Core i7-2640M, 8 GB RAM and NVIDIA NVS4200M 1 GB VGA memory. The system is able to run TORCS up-to 60 FPS with human, fuzzy logic, and stock AI controlled cars.

Fig. 9: Yellow colored car is controlled by human driver whereas white and blue colored car is autonomously controlled by fuzzy logic controller.

Fig. 10: A human controlling a vehicle within TORCS environment to test the fuzzy logic ACC.

The simulation is carried out by developing the proposed fuzzy logic model to control the car within the TORCS environment. Open source C++ fuzzy logic library, Fuzzy-lite, is used to implement the fuzzy logic controller for the car used within the simulator. The objective of the simulation is to test the ability of the fuzzy logic controller to adapt under various traffic conditions and scenarios while not exceeding the specified road speed limit. An autonomous vehicle AI was implemented with the FLC to mimic real world driving situations.

In order to simulate the the input from the SSD an android application was developed to allow users to input the current road speed, thus simulating the SSD system.



(a) Speed data with road speed limit set by Android based application

(b) Throttle and Brake data

(c) Steer data

Fig. 11: Plots of different data sets collected by running the FL based car under TORCS race simulator.

The credibility of fuzzy ACC system was tested simulating it under AI based traffic situations as well with a human controlled vehicle. Fig. 10 shows the setup for the simulation with driving wheel and throttle-brake pedal which allowed human driver to control one of the car within the TORCS environment and test the behaviour of fuzzy logic car in real time. Unpredictable situations were tested using the user controlled car to test FLC with unpredictable scenarios that a vehicle would normally be found in real world situations. The simulation results of the FLC are shown in Fig. 11.

An important evaluation criterion for the proposed ADAS was how well the system mimics a human driver and the comprehensive simulations showed that the system preformed very well and all the transitions were smooth and gradual rather than multiple small abrupt adjustments.

The simulation results are positive as the overall system preformed smoothly according to changes to vehicle conditions and speed limit changes. As shown in Fig. 11.(a), fuzzy car is able to maintain speed well within the road speed limit, around $t = 36$ when road speed limit is changed. The reaction of fuzzy car to match with changed road speed limit is very gradual and smooth. The fuzzy ACC system operates in real time and quickly to responds to changing scenarios. For example, one of the test cases confirmed that the fuzzy logic controlled car immediately stopped when the human controlled car abruptly stopped.

VI. Conclusions and Future Work

Advanced driver assist systems are moving to the point where machine vision is being employed in lieu of conventional sensors because of its ease of use and many diverse applications. This paper has proposed a FL based ADAS with integrated SSD to detect road speed limits. The proposed ACC has proved very effective in simulation as it operated autonomously in the presence of another car without a driver. The SSD used well established morphological methods in order to detect road signs and speed recognition by an ANN. The proposed methods might be improved upon by implementing boosted methods with larger training data set, additionally the accuracy of the system may be improved by implementing a differential evolutionary algorithm to determine the optimal shape parameters for the rectangle using techniques presented in [41][42][43].

Mapping of the membership functions and rule-set for each input variable as well as the baseline in the current implementation is based on the knowledge elicited from simulations and 3D plots. This could be further improved by using AI techniques, i.e. analysing the trends in empirical data from Neural Networks or Reinforcement based learning approaches.

VII. Appendix A: Designed Fuzzy Rules

1) If (Distance is Cautious) and (Deltaspeed is veryunder) and (Cspeed is vslow) then (Facceleration is normal) (1).
2) If (Distance is Cautious) and (Deltaspeed is veryunder) and (Cspeed is slow) then (Facceleration is normal) (1).
3) If (Distance is Cautious) and (Deltaspeed is veryunder) and (Cspeed is normal) then (Facceleration is normal) (1).
4) If (Distance is Cautious) and (Deltaspeed is normal) and (Cspeed is vslow) then (Facceleration is normal) (1).
5) If (Distance is Cautious) and (Deltaspeed is normal) and (Cspeed is slow) then (Facceleration is normal) (1).
6) If (Distance is Cautious) and (Deltaspeed is normal) and (Cspeed is normal) then (Facceleration is normal) (1).
7) If (Distance is Cautious) and (Deltaspeed is normal) and (Cspeed is fast) then (Facceleration is slow) (1).
8) If (Distance is Cautious) and (Deltaspeed is normal) and (Cspeed is veryfast) then (Facceleration is Veryslow).
9) If (Distance is Cautious) and (Deltaspeed is above) and (Cspeed is slow) then (Facceleration is normal) (1).
10) If (Distance is Cautious) and (Deltaspeed is above) and (Cspeed is normal) then (Facceleration is slow).
11) If (Distance is Cautious) and (Deltaspeed is above) and (Cspeed is fast) then (Facceleration is slow) (1).
12) If (Distance is Cautious) and (Deltaspeed is above) and (Cspeed is veryfast) then (Facceleration is Veryslow) (1).
13) If (Distance is Cautious) and (Deltaspeed is under) and (Cspeed is vslow) then (Facceleration is normal) (1).
14) If (Distance is Cautious) and (Deltaspeed is under) and (Cspeed is slow) then (Facceleration is normal) (1).
15) If (Distance is Cautious) and (Deltaspeed is under) and (Cspeed is normal) then (Facceleration is normal) (1).
16) If (Distance is Cautious) and (Deltaspeed is under) and (Cspeed is fast) then (Facceleration is normal) (1).
17) If (Distance is Cautious) and (Deltaspeed is veryabove) and (Cspeed is normal) then (Facceleration is slow) (1).
18) If (Distance is Cautious) and (Deltaspeed is veryabove) and (Cspeed is fast) then (Facceleration is Veryslow) (1).
19) If (Distance is Cautious) and (Deltaspeed is veryabove) and (Cspeed is veryfast) then (Facceleration is Veryslow) (1).
20) If (Distance is Danger) and (Deltaspeed is veryunder) and (Cspeed is vslow) then (Facceleration is slow) (1).
21) If (Distance is Danger) and (Deltaspeed is veryunder) and (Cspeed is slow) then (Facceleration is normal) (1).
22) If (Distance is Danger) and (Deltaspeed is veryunder) and (Cspeed is normal) then (Facceleration is slow) (1).
23) If (Distance is Danger) and (Deltaspeed is normal) and (Cspeed is vslow) then (Facceleration is Veryslow) (1).
24) If (Distance is Danger) and (Deltaspeed is normal) and (Cspeed is slow) then (Facceleration is slow) (1).
25) If (Distance is Danger) and (Deltaspeed is normal) and (Cspeed is normal) then (Facceleration is slow) (1).
26) If (Distance is Danger) and (Deltaspeed is normal) and (Cspeed is fast) then (Facceleration is Veryslow) (1).
27) If (Distance is Danger) and (Deltaspeed is above) and (Cspeed is slow) then (Facceleration is Veryslow) (1).
28) If (Distance is Danger) and (Deltaspeed is above) and (Cspeed is slow) then (Facceleration is Veryslow) (1).
29) If (Distance is Danger) and (Deltaspeed is above) and (Cspeed is normal) then (Facceleration is Veryslow) (1).
30) If (Distance is Danger) and (Deltaspeed is above) and (Cspeed is fast) then (Facceleration is brake) (1).
31) If (Distance is Danger) and (Deltaspeed is above) and (Cspeed is veryfast) then (Facceleration is brake) (1).
32) If (Distance is Danger) and (Deltaspeed is under) and (Cspeed is vslow) then (Facceleration is slow) (1).
33) If (Distance is Danger) and (Deltaspeed is under) and (Cspeed is slow) then (Facceleration is slow) (1).
34) If (Distance is Danger) and (Deltaspeed is under) and (Cspeed is normal) then (Facceleration is slow) (1).
35) If (Distance is Danger) and (Deltaspeed is under) and (Cspeed is fast) then (Facceleration is brake) (1).
36) If (Distance is Safe) and (Deltaspeed is veryunder) and (Cspeed is vslow) then (Facceleration is fast) (1).
37) If (Distance is Safe) and (Deltaspeed is veryunder) and (Cspeed is slow) then (Facceleration is fast) (1).
38) If (Distance is Safe) and (Deltaspeed is veryunder) and (Cspeed is normal) then (Facceleration is veryfast) (1).
39) If (Distance is Safe) and (Deltaspeed is normal) and (Cspeed is vslow) then (Facceleration is normal) (1).
40) If (Distance is Safe) and (Deltaspeed is normal) and (Cspeed is slow) then (Facceleration is normal) (1).
41) If (Distance is Safe) and (Deltaspeed is normal) and (Cspeed is normal) then (Facceleration is normal) (1).
42) If (Distance is Safe) and (Deltaspeed is normal) and (Cspeed is fast) then (Facceleration is normal) (1).
43) If (Distance is Safe) and (Deltaspeed is above) and (Cspeed is slow) then (Facceleration is normal) (1).
44) If (Distance is Safe) and (Deltaspeed is above) and (Cspeed is normal) then (Facceleration is normal) (1).
45) If (Distance is Safe) and (Deltaspeed is above) and (Cspeed is fast) then (Facceleration is slow) (1).
46) If (Distance is Safe) and (Deltaspeed is above) and (Cspeed is veryfast) then (Facceleration is Veryslow) (1).
47) If (Distance is Safe) and (Deltaspeed is under) and (Cspeed is vslow) then (Facceleration is fast) (1).
48) If (Distance is Safe) and (Deltaspeed is under) and (Cspeed is slow) then (Facceleration is normal) (1).
49) If (Distance is Safe) and (Deltaspeed is under) and (Cspeed is normal) then (Facceleration is fast) (1).
50) If (Distance is Safe) and (Deltaspeed is veryabove) and (Cspeed is normal) then (Facceleration is slow) (1).
51) If (Distance is Safe) and (Deltaspeed is veryabove) and (Cspeed is fast) then (Facceleration is Veryslow) (1).
52) If (Distance is Safe) and (Deltaspeed is veryabove) and (Cspeed is veryfast) then (Facceleration is brake) (1).
53) If (Distance is Danger) then (Facceleration is brake) (1).
54) If (Distance is Cautious) and (Rspeed is verylow) then (Facceleration is Veryslow) (1).
55) If (Distance is Danger) and (Rspeed is verylow) then (Facceleration is brake) (1).
56) If (Deltaspeed is under) and (Rspeed is veryhigh) then (Facceleration is fast) (1).
57) If (Deltaspeed is veryunder) and (Rspeed is veryhigh) then (Facceleration is veryfast) (1).
58) If (Deltaspeed is veryabove) and (Rspeed is veryhigh) then (Facceleration is Veryslow) (1).
59) If (Deltaspeed is above) and (Rspeed is veryhigh) then (Facceleration is slow) (1).
60) If (Distance is Cautious) and (Rspeed is low) then (Facceleration is Veryslow) (1).
61) If (Distance is Danger) and (Rspeed is low) then (Facceleration is brake) (1).

62) If (Distance is Safe) and (Rspeed is low) then (Facceleration is normal) (1)
63) If (Distance is Safe) and (Deltaspeed is under) and (Rspeed is high) then (Facceleration is fast) (1).

REFERENCES

[1] Govt. of Canada, A Quick Look at Speeding Crashes in Canada, 2008. [Online]. Available: http://www.tc.gc.ca/eng/motorvehiclesafety/tp-tp2436-rs200807-menu-158.htm. [Accessed: 12-Dec-2013].
[2] M. M. Peden, World Report on Road Traffic Injury Prevention. World Health Organization, 2004.
[3] Govt. of Canada, Transportation in Canada 2011.
[4] Ontario Provincial Govt. Signs, Regulation 615 Highway Traffic Act, 1990 [Online].
[5] Parker, M. R., Sung, H.-Y., Dereniewski, L. J., and Wade-Trim, "Posted Speed Limits and Speed Limit Setting Practices in British Columbia Final Report Wade-Trim Review and Analysis of Posted Speed Limits and Speed Limit Setting Practices in British Columbia", pp. 26-28, 2003. [Online] http://www.e-laws.gov.on.ca/html/regs/english/elaws_regs_900615_e.htm
[6] ETSC, Fact Sheet, pp. 1-6, 2008. [Online] http://www.etsc.eu/documents/Speed%20Fact%20Sheet%207.pdf
[7] Transport Canada and Motor Vehicle Safety, Road Safety in Canada, 2011. [Online]. http://www.tc.gc.ca/eng/motorvehiclesafety/tp-tp15145-1201.htm
[8] K. Osman, M. F. Rahmat, and M. A. Ahmad, Modelling and controller design for a cruise control system, 5th Int. Colloq. Signal Process. Its Appl., no. 1, pp. 254-258, Mar. 2009.
[9] Carsten, Oliver MJ, and F. N. Tate. "Intelligent speed adaptation: accident savings and costbenefit analysis." Accident Analysis & Prevention 37.3 (2005): 407-416.
[10] Hegeman, Geertje, Karel Brookhuis, and Serge Hoogendoorn. "Opportunities of advanced driver assistance systems towards overtaking." EJTIR 5.4 (2005): 281-296.
[11] Venhovens, P. J. "The application of advanced vehicle navigation in BMW driver assistance systems." IV: vehicle navigation systems and advanced controls (1999).
[12] A. Bargeton and F. Moutarde, Joint interpretation of on-board vision and static GPS cartography for determination of correct speed limit, arXiv Prepr. arXiv , pp. 1-5, 2010.
[13] B. van Arem, C. J. G. van Driel, and R. Visser, The Impact of Cooperative Adaptive Cruise Control on Traffic-Flow Characteristics, IEEE Trans. Intell. Transp. Syst., vol. 7, no. 4, pp. 429-436, Dec. 2006.
[14] N. Benalie, W. Pananurak, S. Thanok, and M. Parnichkun, Improvement of adaptive cruise control system based on speed characteristics and time headway, IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 2403-2408, Oct. 2009.
[15] C. Yoon, S. Jang, and M. Park, Road sign tracking for adaptive cruise control under nonlinear conditions, Electron. Lett., vol. 45, no. 23, p. 1165, 2009.
[16] A. Vahidi and A. Eskandarian, Research advances in intelligent collision avoidance and adaptive cruise control, IEEE Trans. Intell. Transp. Syst., vol. 4, no. 3, pp. 143-153, Sep. 2003.
[17] Holve, R., P. Protzel, and K. Naab. "Generating fuzzy rules for the acceleration control of an adaptive cruise control system." Fuzzy Information Processing Society, 1996. NAFIPS. 1996 Biennial Conference of the North American. IEEE, 1996.
[18] Bengtsson, Johan. Adaptive cruise control and driver modeling. Department of Automatic Control, Lund Institute of Technology, 2001.
[19] Ko, Sang-Jin, and Ju-Jang Lee. "Fuzzy logic based adaptive cruise control with guaranteed string stability." Control, Automation and Systems, 2007. ICCAS'07. International Conference on. IEEE, 2007.
[20] M. Sotelo and D. Fernandez, Vision-based adaptive cruise control for intelligent road vehicles, Intell. Robot. , pp. 6469, 2004.
[21] W. Pananurak, S. Thanok, and M. Parnichkun, Adaptive cruise control for an intelligent vehicle, 2008 IEEE Int. Conf. Robot. Biomimetics, pp. 17941799, Feb. 2009.
[22] M. Padash, M. Tavassoli, A. Khoei, and K. Hadidi, A Sophisticated Algorithm for Using Fuzzy Logic Controllers in Adaptive Cruise Controller Systems, Univers. J. Electr. Electron. Eng., vol. 4, no. 1, pp. 134 141.
[23] T. Yi, Z. Xin, and Z. Liang, Fuzzy-Genetic Control Strategy of Hybrid Electric Vehicle, 2009 Second Int. Conf. Intell. Comput. Technol. Autom., pp. 720723, 2009.
[24] L. Jing-liang and L. Zhao-du, Design of ACC Controller Based on Double Level Mamdani Reasoning, 2009 WRI Glob. Congr. Intell. Syst., no. 2, pp. 233237, 2009.
[25] J. E. Naranjo, C. Gonzlez, J. Reviejo, R. Garca, and T. De Pedro, Adaptive Fuzzy Control for Inter-Vehicle Gap Keeping, vol. 4, no. 3, pp. 132142, 2003.
[26] A. Shaout, R. McGee, and S. Awad, An embedded fuzzy logic prediction cruise control system, 2012 8th Int. Comput. Eng. Conf., pp. 715, Dec. 2012.
[27] L. Cai, A. B. Rad, S. Member, and W. Chan, An Intelligent Longitudinal Controller for Application in Semiautonomous Vehicles, vol. 57, no. 4, pp. 14871497, 2010.
[28] J. E. Naranjo, C. Gonzlez, R. Garca, and T. De Pedro, ACC + Stop & Go Maneuvers With Throttle and Brake Fuzzy Control, vol. 7, no. 2, pp. 213225, 2006.
[29] K. Passino, Fuzzy Control, vol. 20103237. 2010, pp. 5515527.
[30] B. Wu, The Embedded Driving-assistance System on Taiwan i TS-1, pp. 33823387, 2008.
[31] P. Chakroborty and S. Kikuchi, Evaluation of the General Motors based car-following models and a proposed fuzzy inference model, Transp. Res. Part C Emerg. Technol., vol. 7, no. 4, pp. 209235, Aug. 1999.
[32] L. A. Zadeh, Outline of a new approach to the analysis of complex systems and decision processes, IEEE Trans. Man Cybern., vol. SCM-3, no. 1, p. 28 44, 1973.
[33] C. Kim and R. Langari, Adaptive analytic hierarchy process-based decision making to enhance vehicle autonomy, IEEE Trans. Veh. Technol., vol. 61, no. 7, pp. 3321 3332, 2012.
[34] B. Hoferlin and K. Zimmermann, Towards reliable traffic sign recognition, in IEEE Conference In Intelligent Vehicles Symposium, 2009, p. 324 329.
[35] X. Gao, L. Podladchikova, D. Shaposhnikov, K. Hong, and N. Shevtsova, Recognition of traffic signs based on their colour and shape features extracted using human vision models, J. Vis. Commun. Image Represent., vol. 17, no. 4, p. 675 685, 2006.
[36] M. Fu and Y. Huang, A survey of traffic sign recognition, Wavelet Anal. Pattern Recognit. (, no. July, pp. 1114, 2010.
[37] H. A. N. Liu, D. Liu, and J. Xin, Real-time recognition of road traffic sign in motion image based, no. November, pp. 8386, 2002.
[38] C. Fang, S. Chen, and C. Fuh, Road-sign detection and tracking, Veh. Technol. IEEE , vol. 52, no. 5, pp. 13291341, 2003.
[39] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Trans. Syst. Man. Cybern., vol. 9, no. 1, p. 62 66, 1979.
[40] P. Soille, Morphological Image Analysis: Principals and Applications, Springer-Verlag, p. 173 174, 1999.
[41] S. Rahnamayan, H.R. Tizhoosh, M.M.A Salama, Opposition-Based Differential Evolution, IEEE Transactions on Evolutionary Computation, Vol. 12, Issue 1, Feb. 2008, pp. 64-79.
[42] S. Rahnamayan, H.R. Tizhoosh, M.M.A Salama, Opposition versus Randomness in Soft Computing Techniques, Journal on Applied Soft Computing - Elsevier, Volume 8, Mar. 2008, pp. 906-918.
[43] S. Rahnamayan, G.G. Wang, M. Ventresca, An Intuitive Distance-Based Explanation of Opposition-Based Sampling, Journal on Applied Soft Computing - Elsevier, Vol. 12, Issue 9, Sept. 2012, pp. 2828-2839.