PUNE INSTITUTE OF COMPUTER TECHNOLOGY,

DHANKAWADI PUNE-43.

A

Seminar Report

On

Intelligent Speed Adaptation Using a Self-Organizing Neuro-Fuzzy Controller with Speed Sign Detection Capability.

SUBMITTED BY

Venkatesh Lokare

ROLL NO. 3438

CLASS: TE 4

GUIDED BY

PROF. R. S. Paswan

ISO 9001 : 2008 Certified

COMPUTER ENGINEERING DEPARTMENT

Academic Year:

PUNE INSTITUTE OF COMPUTER TECHNOLOGY,

DHANKAWADI PUNE-43.

CERTIFICATE

ISO 9001 : 2008 Certified

This is to certify that Mr. Venkatesh Lokare Roll No. 3438 a student of T.E. (Computer Engineering Department) Batch 2014-2015, has satisfactorily completed a seminar report on "Intelligent Speed Adaptation Using a Self-Organizing Neuro-Fuzzy Controller with Speed Sign Detection Capability." under the guidance of prof. R. S. Paswan towards the partial fulfillment of the third year Computer Engineering Semester II of Pune University.

Prof. R. S. Paswan	Prof. G. P. Potdar
Internal Guide	**Head of Department Prof.**
	Computer Engineering

ACKNOWLEDGEMENT

It gives me great pleasure and satisfaction in presenting this seminar on "Intelligent Speed Adaptation Using a Self-Organizing Neuro-Fuzzy Controller with Speed Sign Detection Capability". I would like to give special thanks to my seminar guide Prof. R.S. Paswan for her invaluable advice and wholehearted cooperation without which this seminar would not have seen the light of day.

I would like to express my deep sense of gratitude towards Dr. G.P Potdar, Head of Department of Computer Engineering, Pune Institute of Computer Technology, Pune, for his cooperation and encouragement.

I would like to thank all those, who have directly or indirectly helped me for the completion of the work during this seminar.

Venkatesh Lokare

CONTENTS

Abstract:

Every year there are millions of cases of traffic accidents pertaining to pedestrians and road users, mainly caused due to lack of diligence from the car driver causing a serious threat to road safety. 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. Park autonomously, drive on highways, and take some decisions such as lane changing, car following, and overtaking are some of the features which have been implemented. These methodologies are implemented using a hybrid neuro-fuzzy algorithm. Through this paper we would like to put into light a new method of autonomously adapt the speed of the car by learning from a human driver and using anticipation. The implementation of the system is a special fuzzy neural network namely: the Generic Self Organizing Fuzzy Neural Network using the Yager inference scheme (GenSoFNN(Yager)).This experiment is to elaborate and anticipate the speed constrains during curved tracks. The described system will also improve 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 and increasing efficiency during obscure curves..

KEYWORDS:
Adaptive systems, Artificial neural networks, Fuzzy Control, Fuzzy neural networks, Autonomous Vehicles, Object Detection.

Image List

1. INTRODUCTION:

1.1 MOTIVATIONAL SURVEY

In a world report on road traffic injury prevention published by the World Health Organization (W.H.O.), it was estimated that 1.2 million people are killed due to road crashes each year, and over 50 million are injured [1]. The Projections indicate that the figures will increase to about 65% more than the current scenario in the next 20 years. The global cost of road crashed and injuries which have to be later recovered, is estimated to be US$ 518 billion per year. There are various causes of these accidents, which are attributed to human error, alcohol, bad weather, heavy traffic or bad infrastructures. Some of these causes like the road infrastructure, the traffic congestions and maintenance of roads and footpaths, can be managed by measures, but the primary cause is human error, which is independent with the problem of road security. Autonomous driving systems have resulted from researches to decrease the risk of human error and provide a less risky environment to travel around in.

1.2 APPLICATIONS AND IDENTIFICATION OF CHALLENGES IN DOMAIN

The Intelligent Autonomous Vehicle has the primary purpose of modeling the human driver in abilities such as perception of the environment, learning and reasoning, all of which are key aspects pertaining to the activities performed by the human brain. While there have been several researches[1] made which have been proved effective on highways, with simple scenarios such as lane marking, slow deviations in curvature, no crossing, automated driving in more diverse and complex environments is still far away from being implemented. An average person who has to overcome arduous tasks such as, drive in a city with stops, pedestrian crossing, lane changing, accelerating or slowing down according to the environment changes, to overcome during his everyday commute to and fro from work , trying to automate these tasks raises a lot of problems. Even on the already implemented Highways Automated Systems, the defined longitudinal control of the vehicle uses simple behaviors such as accelerating until the speed limit, braking or decelerating if there is an obstacle in front of the vehicle, accelerating for overtaking, performing smooth stops in front of stop signs and traffic signals, but more complex behaviors, such as anticipating the curves, have not yet been implemented. Intelligent Speed Adaptation are systems which can regulate the vehicle speed on roads, based on the various environmental factors such as traffic, weather, road conditions and various other environmental constrains. These obtuse obstacles have been overcome by the Intelligent Speed Adaptation and the results have been tested and proved efficient in several countries, but these efficient systems only react to the speed limit of the road, some more complex systems being able to adapt their speed to the road and weather conditions are in the making, and are soon going to be seen around in the coming future. But none of them can slow down when the car dangerously reaches a curve, which can cause the vehicle to go off the road and create an accident. This phenomenon mainly occurs due to moving at a speed greater than the allowed banking of the road. Though the anticipation of curves is an important aspect for the security of the driver, very few research has been made on this subject, this topic being quite innovative in the Intelligent Speed Adaptation domain.

1.3 LITERATURE REVIEW

Driving could be modelled as a continuous and tedious decision-making process involving a set of rules that relate sensory input to control output which has to be performed by the autonomous car. But designing these erudite rules is quite difficult and complex, so the simplest way is to learn from human expertise for mapping and extracting the rules and deploying it onto the computer. This approach uses a type of hybrid intelligent system, a Fuzzy Neural Network, the Generic Self Organizing Fuzzy Neural Network mapped with the Yager inference scheme GenSoFNN(Yager)[1]. The fuzzy neural network is a combination of a neural network and a fuzzy system, which provides the esteemed advantage of both these techniques: the learning and generalization of neural networks, and the reasoning strength and ease of interpretation of fuzzy systems. A simulator, namely TORCS (The Open Racing Car Simulator)[1][2], has been used to test the capacity of the GenSoFNN(Yager) to learn how to drive from a human. Successful manoeuvers achieved so far include reverse parking, U-turn, as well as automated driving.

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. 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. In order to circumvent the current availability and validity limitations of GPS, speed sign detection (SSD)[2] 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[5] 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.

2. SURVEY OF MATHEMATICAL MODELS

2.1 PRESENTATION OF THE GENSOFNN(YAGER)

The GenSoFNN(Yager) is a specific Fuzzy neural network, based on the structure of the Generic Self-Organizing Fuzzy Neural Network (GenSoFNN)[4], and using the Yager Inference Scheme to interpret the fuzzy relations of the rules[1].

A. Generic Self-Organizing Fuzzy Neural Network :The Generic Self-organizing Fuzzy Neural Network (fig. 1) is a fuzzy neural network with a generic connectionist structure. This network is able to automatically generate fuzzy rules, from a training data set, and has a strong noise tolerance by using the Discrete Incremental Clustering (DIC) technique[3].

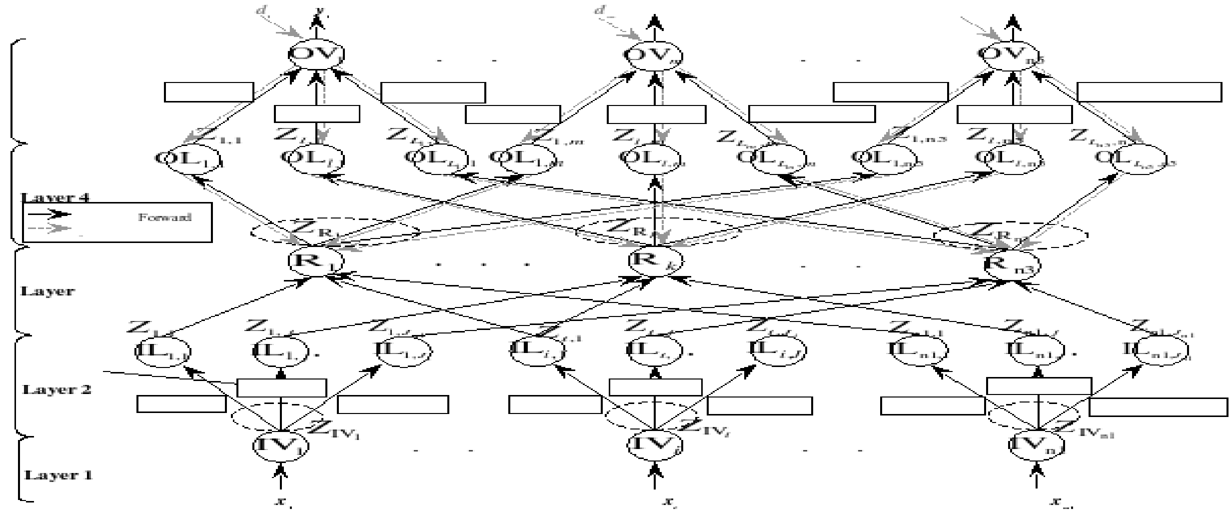


Fig. 1. Schematic structure of the GenSoFNN(Yager)

The GenSoFNN(Yager) consists of five layers of nodes. Each input node has a single input, the vector X = [x1,...,xi,...,xn1] represents the inputs of the GenSoFNN(Yager), and each output node computes a single output, the vector Y = [y1,...,ym,...,yn5] denotes the outputs of the GenSoFNN(Yager) with respect to the X input. In addition, the vector D = [d1,...,dm,...,dn5] represents the desired network outputs required during the parameter learning phase of the training cycle. The input and output labels in layers 1 and 5 are determined by DIC during the self-organizing phase. The nodes of other layers (2, 3 and 4) are created during the rule formation phase. The training phase of the GenSoFNN(Yager) consists of three phases : self-organizing, rule formulation and parameter learning. The negative gradient descent backpropagation algorithm is employed in order to tune the parameters of its fuzzy sets in label layers. B. The Yager Inference Scheme The original fuzzy inference scheme extends the conventional modus ponens rule which states that from the propositions:

P1: IF X is A THEN Y is B
P2: X is A
Thus Y is B.

Let µA(x) and µB(y) be the membership functions of set A and B. The Yager Rule of Inference adopts the second approach, which is based on implication model of fuzzy relation $\mu R(x,y)=\max\{(1-\mu A(x)),\mu B(y)\}$

It should be noted that both formulae correspond to the logical transition of P1 interpreted in different ways. The second formula corresponds to the statement $\neg A \cup B$,whichis essentially the same as $A \rightarrow B$ in crisp logic.

Mapping of Yager Inference Scheme

The mapping of the Yager Inference Scheme into the Generic Self-Organizing Fuzzy Neural Network allows the allocation of the different operations used in the different layers[1].

• Layer 1 (Fuzzyfication): This layer consists of input nodes, which fuzzify the crisp inputs the network receive.
• Layer 2 (Antecedent Matching) : The fuzzified inputs from layer 1 are then compared against their corresponding input labels that form the antecedent section of the fuzzy rules in the GenSoFNN(Yager). The antecedent matching between the inputs and the antecedent section is essentially to compute the negation of membership values of the inputs with respect to the input fuzzy sets.
• Layer 3 (Rule Fulfillment): The third layer of the GenSoFNN(Yager) contains the fuzzy rule base of the network. Each rule node Rk computes the degree of fulfillment (i.e. the overall similarity) of the current inputs with respect to the antecedents of the fuzzy rules it denotes. In a fuzzy relation, the antecedent sections of a fuzzy rule Rk are connected by "AND" conjunctive and therefore operator min is used to compute the aggregated rule fulfillment of Rk.
• Layer 4 (Consequent Derivation): Layer 4 contains output term nodes that represent the output fuzzy sets of the consequent of the rules in layer 3. Each output term may be connected to multiple fuzzy rules indicating that they may have the same consequent. GenSoFNN(Yager) uses implication-based model of fuzzy relation therefore conclusions of parallel rules will have to be combined conjunctively.
• Layer 5 (Output defuzzification): The output nodes are responsible for the defuzzification of the derived fuzzy outputs from the GenSoFNN(Yager) before presenting them as crisp outputs. For each output y_m, the derived fuzzy conclusions for all its output labels are aggregated using a modified center of averaging (COA) technique to produce the final output.

2.2 ADAPTATION OF THE GENSOFNN(YAGER) FOR CONTROL OF THE VEHICLE

The GenSoFNN(Yager) has already proved its efficiency for various applications such as highways driving and a few tactical car maneuvers, such as lane changing, overtaking, car following, and collision avoidance. But if lateral control has been a big focus of research during these last years, longitudinal control has been in its first experiments[1]. The GenSoFNN(Yager) was used to implement longitudinal control of a vehicle, in order to autonomously adjust its speed according to the upcoming curve. The main advantage of using a fuzzy neural network instead of a classical neural network is the ability to extract the rule base from the system, and then to understand the functioning of the network.

Implementation of the Longitudinal Control In order to implement the longitudinal control, two networks are used, one for the throttle , and another for the brake. The inputs used for the both networks are the same: the speed of the car and an anticipation variable

Fig. – The throttle/brake subsystem scheme

The main idea was to calculate anticipation in a variable, which is then fed to the GenSoFNN(Yager). Every time step, the angle of the curve is calculated at a certain distance of the car (the distance depends of the speed, as shown in table I). The returned value is then added to a list, which contains the description of the shape track for 1.6 seconds. This list is then used to calculate the anticipation of the curve (described in algorithm 1). The result value is an angular speed, normalized between −PI and PI.

TABLE I

THE DISTANCE OF THE CURVE, DEPENDING OF THE SPEED

speed (in m/s)	distance (in m)
x < 4.7	5
4.7 < x < 7.8	10
7.8 < x < 10.9	15
10.9 < x < 14.1	20
14.1 < x < 17.2	25
17.2 < x < 20.3	30
20.3 < x < 23.4	35
23.4 < x < 26.6	40
26.6 < x < 29.7	45
29.7 < x < 32.8	50
32.8 < x < 35.9	55
x > 35.9	60

Algorithm 1 Calculate the anticipation variable

Algorithm 1 Calculate the anticipation variable

$$
\begin{array}{l}
\quad anticipation \leftarrow 0 \\
\quad \textbf{for } i = 1 \text{ to } length(curve) \textbf{ do} \\
\qquad \textbf{if } curve[i > 0] \wedge curve[i+1 > 0] \textbf{ then} \\
\qquad\quad anticipation \leftarrow anticipation + \frac{curve[i+1]-curve[i]}{time\_step} \\
\qquad \textbf{else if } curve[i < 0] \wedge curve[i+1 < 0] \textbf{ then} \\
\qquad\quad anticipation \leftarrow anticipation + \frac{curve[i]-curve[i+1]}{tim\_step} \\
\qquad \textbf{else if } curve[i < 0] \wedge curve[i+1 > 0] \textbf{ then} \\
\qquad\quad \textbf{if } -curve[i] > curve[i+1] \textbf{ then} \\
\qquad\qquad anticipation \leftarrow anticipation + \frac{curve[i]-curve[i+1]}{time_step} \\
\qquad\quad \textbf{else} \\
\qquad\qquad anticipation \leftarrow anticipation + \frac{curve[i+1]-curve[i]}{time\_step} \\
\qquad\quad \textbf{end if} \\
\qquad \textbf{else if } curve[i > 0] \wedge curve[i+1 < 0] \textbf{ then} \\
\qquad\quad \textbf{if } curve[i] > -curve[i+1] \textbf{ then} \\
\qquad\qquad anticipation \leftarrow anticipation + \frac{curve[i+1]-curve[i]}{time\_step} \\
\qquad\quad \textbf{else} \\
\qquad\qquad anticipation \leftarrow anticipation + \frac{curve[i]-curve[i+1]}{time\_step} \\
\qquad\quad \textbf{end if} \\
\qquad \textbf{end if} \\
\quad \textbf{end for}
\end{array}
$$

From the algorithm, we can deduce if the curve is soft or sharp (the larger the value is, the sharpest is the curve), and also to know if this is the beginning or the end of the curve. At the beginning, the curve become sharper, then the result is positive, and at the end the curve become softer, and the result is negative. Consequently, the vehicle does not react as soon as it detects a curve, but as a limit is reached (the variable anticipation become bigger and bigger while the car approaches a curve). So the variable depends not only from the sharpness of the curve, but also from the distance the car is from the curve. This allow the system to have the shape of the road for the next 1.6 seconds.

3. PROPOSED MATHEMATICAL MODEL

Let S be the solution set for finding out the solution for Speed Adaptation using a Self-Organizing Neuro Fizzy Controller with Speed Sign Detection Capability.

$S = \{s, e, X, Y, DD, NDD, F, S_C, F_C \mid Ø\}$

s = start state Y = {Ø}

e = end state Y = {T, B}

T = Throttle B = Brake Intensity

X = {set of inputs}

X = { X | X € { Image , Curve}}

$$\text{Image} \in \left\{ \begin{matrix} P_{0,0} & P_{0,1} & P_{0,2} \ldots\ldots & P_{0,n} \\ P_{0,0} & P_{0,1} & P_{0,2} \ldots\ldots & P_{0,n} \\ .. & & & \\ P_{m,0} & P_{m,1} & P_{m,2} \ldots\ldots & P_{m,n} \end{matrix} \right\} \quad n,m \in I^{+} \quad \{\text{Set of Pixels}\}$$

$P_{i,j}$ € (r,g,b) $j < n$, $I < m$

r, g , b € [0-255] (colour integrity)

Curve € $[d_0, d_1, d_2, \ldots]$ (set of distances)

d_i = i th distance from start point

Y = { set of outputs}

= { T , B }

T = Throttle Value (Acceleration)

T € [0 – a] Km / hr a = max speed

B = Break Value

B € [(-T) - 0] Retardation in acceleration

F = {set of functions}

$= \{F_{image}, F_{ANTI}, F_{GENSOFNN}\}$

F_{image} = {function to capture image and read correct speed}

$= \{F_{capture}, F_{THRESH}, F_{MORPH}, F_{RECOG}\}$

$F_{capture}$ = {capture image from camera}

= 1 <image>

= 0 < no image>

F_{THRESH} = {perform thresh holding on image}

$= I_B (0,0,0)$ $I_{R,G,B} < T$

$= I_{R,G,B} (R,G,B)$ $I_{R,G,B} > T$

Where

$I_{R,G,B}$ = intensity of pixel at i , j

I_B = intensity of black pixel.
F_{MORPH} = {morphing performed on Image}
= {filling, opening }
Where.
Filling = {function to fill holes in image}
Opening = {Morph image according to S E }   SE = Structure Element

$L_{sign} > 0.8 * W_{sign}$
$Length_{SE} = 0.05 * Length_{image}$
$Width_{SE} = 0.05 * Width_{image}$

$$\left| \frac{Area_{filled}}{Area_{box}} - 1 \right| \leq 0.2$$

$$\left| \frac{Y_{box}}{X_{box}} - 0.8 \right| \leq 0.2$$

F_{RECOG} = {Recognize image from set of images}
Image = {16 * 20} pixel
1D => 320 bit array

Ip = {10,20,30,40,50,60,70,80,90,100}

Image[k] = Ip[i][x] $0 < i < 7$, $0 < k < 320$

Op = [0 -1] [i]
Op = {Probability of image = Ip[i] }

F_{ANTI} = {function to anticipate curve}
Ip = < Curve>   time step = 1.6

For i = 0 < i < length (curve)

$$P : anti \leftarrow anti + \frac{Curve\,[i+1] - Curve\,[i]}{Time\ step}$$

$$Q : anti \leftarrow anti + \frac{Curve\,[i] - Curve\,[i+1]}{Time\ step}$$

P   Curve [i] > 0 ^ Curve [i+1] > 0
Q   Curve [i] < 0 ^ Curve [i+1] < 0

Curve [i] < 0 ^ Curve [i+1] > 0

Q   - Curve [i] > Curve [i+1]
P   - Curve [i] < Curve [i+1]

$Curve[i] > 0 \wedge Curve[i+1] < 0$

P $Curve[i] > - Curve[i+1]$
Q $Curve[i] < - Curve[i+1]$

$F_{GENSOFNN}$ = { function to train car}

Ip = <speed , Anticipation>
Op = <Throttle , Brake>

P1 : if X is A, then Y is B
P2 : X is A, thus Y is B

Speed = { slow, fast}
Angle of Curvature = { straight, softleft, right, soft right, left, hard right, hard left, extreme right, extreme left }
Throttle = { normal acceleration, slow acceleration, hand acceleration, no acceleration }

$F_{GENSOFNN}$ = { F_{FUZZ} , F_{MATCH} , F_{RULE} , F_{CONSE} , F_{DEFUZZ} }

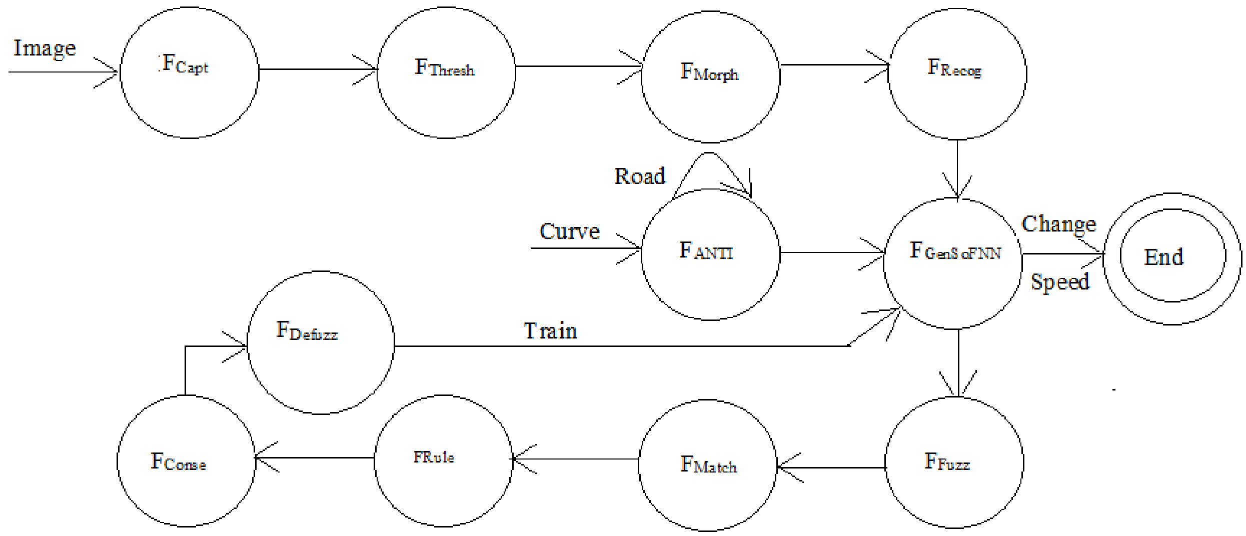
F_{FUZZ} = { Fuzzification of Input }
Speed € {Slow , Fast}

Angle € { straight, softleft, right, soft right, left, hard right, hard left, extreme right, extreme left }

F_{MATCH} = {Antecedent Matching}
If X is A Disjunction
F_{RULE} = {Rule fulfilment }
And R_k Conjunction
$R_1 \wedge R_2 \wedge R_3$ …….
F_{CONSE} = {providing consequents}
Then Y_k Implication
$A_i \rightarrow B_k$
F_{DEFUZZ} = {defuzzification of Op}

T € {normal acceleration, slow acceleration, hand acceleration, no acceleration }

STATE DIAGRAM



Sc = {Success Case}
Case A = Image Input is properly capture
Case B = Curve is properly measured
Case C = Speed is properly changed.

Fc = {Failure Case}

Case D = $\overline{A\ U\ B\ U\ C}$

DD = {deterministic data}
= {image, road, curvature, original speed}

NDD = {non deterministic data}
{position of speed sign on image, Angle of curvature, change in speed}

4. DESIGN AND ANALYSIS OF SYSTEM

Recognizing speed limits from North American traffic signs is a challenging task[2]. Conventional approaches, 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 to localize specific shape descriptors.

4.1 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[2].

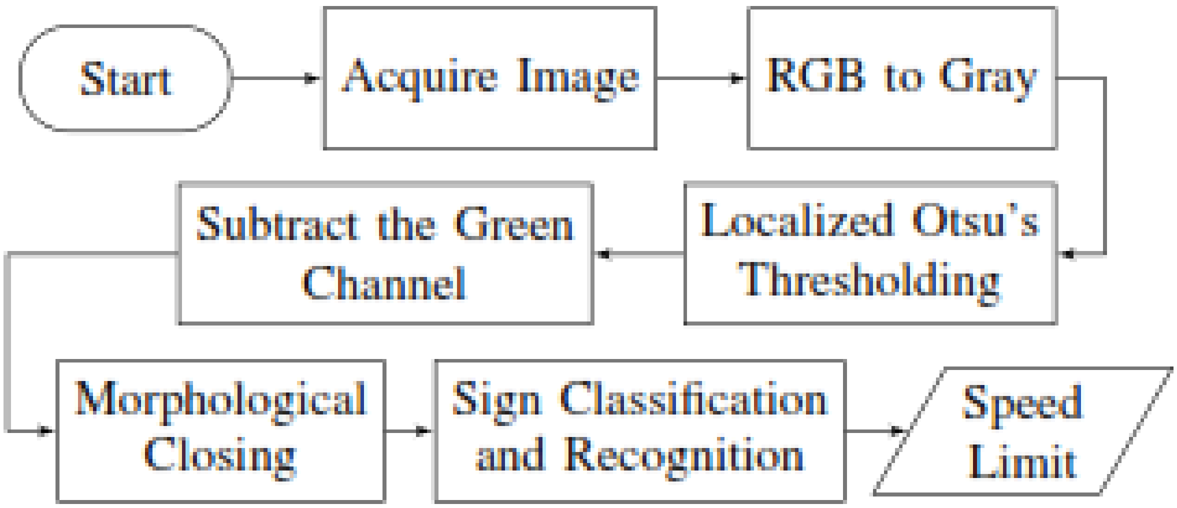


Fig. : The speed sign detection and speed limit recognition algorithm

4.1.1 IMAGE THRESHOLDING

After the image, I, has been acquired by the camera, it is split into n blocks using equation

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

Each block is thresholded using Otsu's method[6] .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. : (a) shows the original image and (b) shows the locally thresholded image with the subtracted green channel

4.1.2 MORPHOLOGICAL OPERATIONS

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)

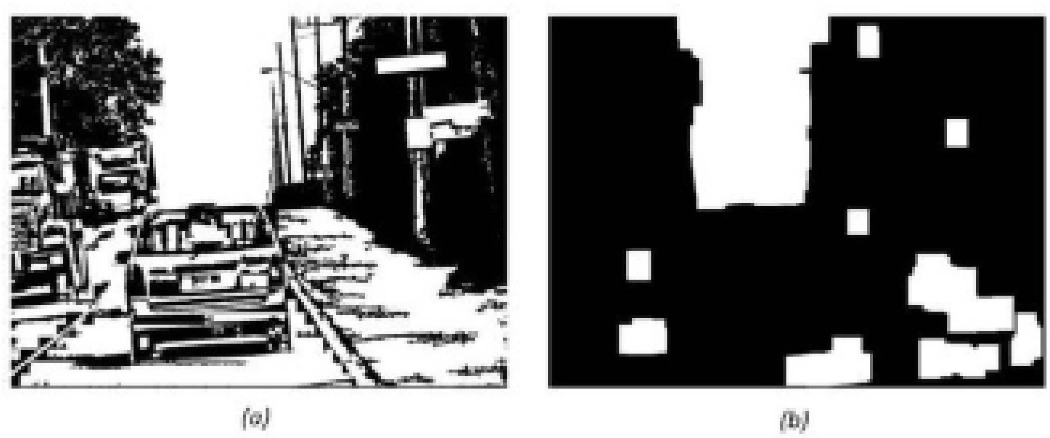


Fig. : (a) shows the filled holes image and (b) shows the image after it has been opened with the square SE.

4.1.3 SPEED LIMIT RECOGNITION.

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[2]. 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.

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.

(a) (b)

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

5. DISCUSSION ON IMPLEMENTATION RESULTS

ON implementation of the anticipation curve algorithm it was brought into light that any straight line after the curve was also treated as a part of the curve. This would lead to cause further speed deceleration by the GenSoFNN(Yager) algorithm. Hence to avoid this phenomenon the new speed can be deduced from the nearby sign boards and the speed of the car can be changed accordingly. As this method slightly reduces acceleration on detection of curve and uses very less breaks the overall efficiency of the car is truly enhanced.

Fig: Experiment Output

6. CONCLUSION AND FUTURE ENHANCEMENT

The results of changing the speed as the roads curvature changes does have a promising vibe towards its implementation in the practical aspects, and the feature of adapting to the speed signs on runtime will thoroughly increase its efficiency

A major point which has been not considered in this proposal as a whole is the mass of the car and the effect it will have on different banked roads, the tuning of these cars will greatly be affected due to the different mass sizes of the cars and various other assumptions would be taken into consideration along with curvature of the road and its original speed.

7. REFERENCES USED:

1. Rizvi, R., Kalra, S., Gosalia, C., Rahnamayan, S. "Fuzzy Adaptive Cruise Control system with speed sign detection capability." Fuzzy Systems (FUZZ-IEEE), 2014 IEEE International Conference.

2 Partouche, D, Pasquier, M. and Spalanzani, A." Intelligent Speed Adaptation Using a Self-Organizing Neuro-Fuzzy Controller" Intelligent Vehicles Symposium, 2007 IEEE.

3 Singh, A. Quek, C. Cho, S.-Y. "DCT-Yager FNN: A Novel Yager-Based Fuzzy Neural Network With the Discrete Clustering Technique" Neural Networks, IEEE Transactions on (Volume:19 , Issue: 4)

4. R.J. Oentaryo, and M. Pasquier. GenSoFNN-Yager: A Novel Hippocampus-like Learning Memory System Realizing Yager Inference. Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN'06), Vancouver, BC, pp. 1684-1691, July 2006

5. N. Benalie, W. Pananurak, S. Thanok, and M. Parnichkun, Improvement of adaptive cruise control system based on speed characteristicsand time headway, IEEE/RSJ Int. Conf. Intell. Robot. Syst., pp. 2403-2408, Oct. 2009.

6. N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Trans. Syst. Man. Cybern., vol. 9, no. 1, p. 62 66, 1979