ISSN: 2278-0181
RTICCT - 2019 Conference Proceedings

Automated Vehicle: Autonomous Driving using SVM Algorithm in Supervised Learning

Dr. K. Velmurugan Professor (Senior Grade)[1],
B. Mathumitha[2] , B. Merylen Jenow[3], R. Thamizh Oviyam[4]
Department of Computer Science and Engineering,
Anjalai Ammal Mahalingam Engineering.College,
Koyilvenni, Thiruvarur-614 403, Tamilnadu, India.

***Abstract*:- Determination of suitable path for autonomous vehicle that is collision free between initial and end position through a workspace in presence of obstacle is challenging for autonomous vehicle design. Conceptual wise the word "Autonomous" associated with a machine means "a system that works without any intervention of human being" hence in today's era majority of the jobs are replaced by robots for performing numerous tasks i.e. manipulating objects from one place to another, path covering with detection of safe paths and wielding etc. Extra feature of bumpiness detection is added to the existing system. A* search algorithm is used for path planning and decision making for autonomous vehicles in urban environment enable self-driving car to find safest, most convenient an beneficial route. Finally for classification of bumpiness, the proposed approach SVM classification algorithm is applied for classification of road severity.**

Keyword:- Machine Learning, Support Vector Machine Classification, Supervised Learning, Obstacle Detection, Path Planning, Bumpiness Detection ,A* search algorithm.

I. INTRODUCTION

An autonomous vehicle is the one that can guide itself without human conduction. This kind of vehicle has become a concrete reality and may pave the way for future systems where computers take over the art of driving. Since the 1990s – a time when autonomous driving was still only found in science fiction books or films – BMW engineers and technicians have been working on driver assistance systems. In the next decade, the car industry will change more drastically than it has over the past 30 years, because today we are standing at the entrance to a new era – of highly automated driving. Fully-driverless tech is still at an advanced testing stage, but partially automated technology has been around for the last few years. Functions that make partial automation possible are already a reality and installed in the latest BMWs on the street. Semi-autonomous driving assistance systems, such as the Steering and Lane Control Assistant including Traffic Jam Assistant, make daily driving much easier. They can brake automatically, accelerate. With the remote-controlled parking function, BMW made it possible to pull into tight spots without a driver for the first time. With so much investment and interest in driverless technology, it's easy to assume that self-operating cars are imminent, but they're much further away than we might think. Using sensors to read the road like humans had their own limitations. Sensors are vulnerable to extreme sunlight, weather or even defective traffic lights. Henceforth analysing the images came into existence. For this process we are in need of machine learning.

In this paper we have used supervised learning to classify the level of bumpiness for rural environment. The image is obtained and transformed into a grid file. The Grid image file is processed using the CV2 which is available as a separate module in python especially for processing the image. A* search algorithm is used for finding the minimum path in the grid. The minimum path is provided as a result.

For an autonomous vehicle 3 modules are required. They can be listed as 1) Obstacle detection module 2) Path planning 3) Bumpiness detection. We have discussed the obstacle detection module in section 1, on how it processes the image and marking the position of the obstacle. Path planning module is described in section 2, about how it finds the minimum path to the goal with A* search algorithm. Bumpiness detection is depicted in section 3, on how it adjusts speed according to the rate of bumpiness of the road. Section 4 describes the architecture of the system and finally section 5 concludes the paper with a summary of the work

II. RELATED WORK

From Dixit VV, Chand S, Nair DJ (2016) Autonomous Vehicles: Disengagements, Accidents and Reaction Times. PLOS ONE 11(12): e0168054. Fully automated cars will allow drivers to be driven by an informatics system in their own vehicle, which facilitates the drivers to engage in non-driving related activities. However, under unfortunate situations of system failure, the drivers are expected to react in an appropriate and timely manner to resume manual driving. It is essential to understand the causes for disengagements and the resulting driver reaction times as the AV technological development is at full pace. The data from the first year of reporting from the autonomous vehicle trials in California have provided insights into disengagement and accident exposure, perception-reaction time and trust. Autonomous miles driven were found to have significantly high correlation and trends with accidents, suggesting that this quantity can be potentially used as a measure of exposure for disengagements and accidents.

From "Autonomous vehicles and their impact on road infrastructure and user safety" by Anna Skarbek- Zabkin, Marcin Szczepanek.The rapid development of autonomous cars technology and the interest of companies and corporations in the creation of their own vehicles of this type indicates that intelligent technologies on the road are the future of transport. It is a chance to improve road safety and turn towards ecological thinking about transport. In addition, autonomous mobility improves the mobility of people with disabilities and older people. The most serious problem of common infrastructure for conventional and autonomous vehicles is to solve legal problems. This solution requires the development of common rules for self-aware vehicles for all countries, making ethical decisions, and who is at fault for causing an accident: vehicle manufacturer, system manufacturer used in the vehicle, or driver – passenger. The commencement of cooperation at the government level in the field of road automation in Poland was announced by the Minister of Infrastructure and Construction Andrzej Adamczyk and the Minister of Energy Krzysztof Tchorzewski during the international conference "Autonomous future of the road transport". It was organized by the Motor Transport Institute in Warsaw. Over four hundred representatives of the automotive industry and the business world as well as scientists have registered to participate in the meeting. Among them are guests from Israel, the United States, Germany and Belarus. First, the autonomous cars will appear in enclosed areas, i.e. it will be testing under special supervision. However, their full commercial use is planned for around 2025. Then all guidelines for self-steering cars will be created.

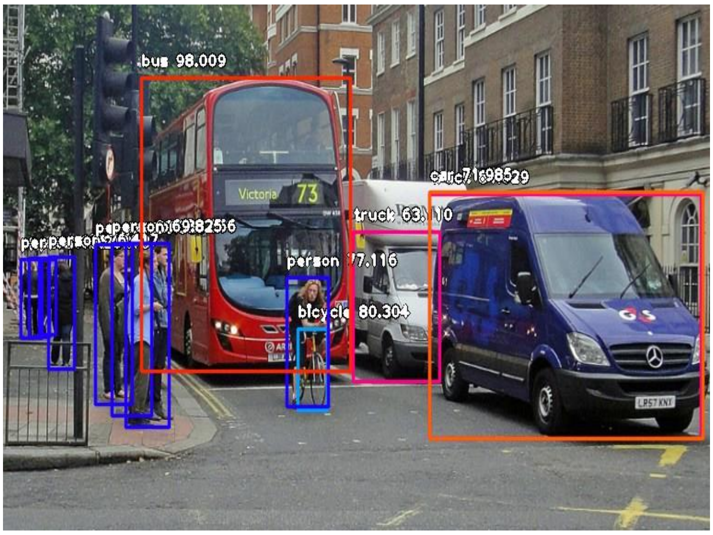
III. PROPOSED SYSTEM

1. Obstacle Detection

Object detection refers to the capability of computer and software systems to locate objects in an image/scene and identify each object. Object detection has been widely used for face detection, vehicle detection, pedestrian counting, web images, security systems and driverless cars. There are many ways object detection can be used as well in many fields of practice. Here we have used Image AI, a python library that lets programmers and software developers easily integrate state-of-the-art computer vision technologies into their existing and new applications, using just few lines of code. To perform object detection using **ImageAI,** all you need to do is

1.Install Python on your computer system
2.Install **ImageAI** and its dependencies
3. Download the Object Detection model file
4. Run the code which contains the codes to invoke the methods in objection detection class (which is as few as 10 lines)
5.After detecting the objects mark those positions in black colour.

After doing these steps obstacle marked grid image is sent for path planning.



2. PATH PLANNING

The grid image file with the obstacle positions marked is received. Using the A* search algorithm.

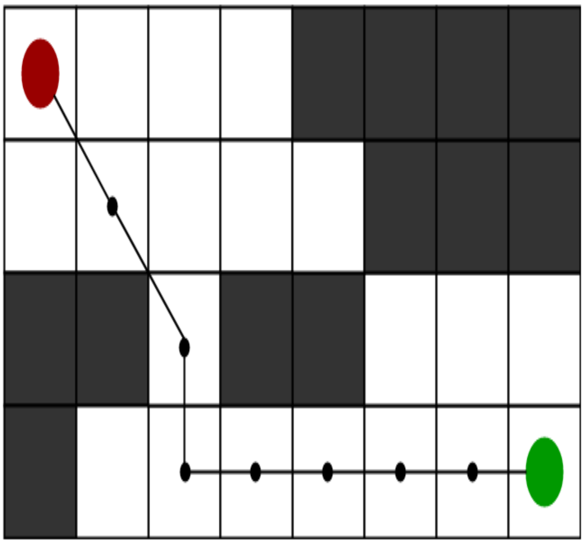
Algorithm to find the minimum path:

//A* search algorithm

1. Initialize the open list

2. Initialize the closed list put the starting node on the open list (you can leave its f at zero)
3. while the open list is not empty
a) find the node with the least f on the open list, call it "q"
b) pop q off the open list
c) generate q's successors and set their parents to q

d) for each successor
i) if successor is the goal, stop search successor.g = q.g + distance between successor and q
successor.h = distance from goal to successor successor.f = successor.g + successor.h
ii) if a node with the same position as successor is in the OPEN list which has a lower f than successor, skip this successor.
iii) if a node with the same position as successor is in the CLOSED list which has a lower f than successor, skip this successor
otherwise, add the node to the open list .
end (for loop).
e) push q on the closed list end (while loop)



What A* Search Algorithm does is that at each step it picks the node according to a value-‘**f**’ which is a parameter equal to the sum of two other parameters – ‘**g**’ and ‘**h**’. At each step it picks the node/cell having the lowest ‘**f**’, and process that node/cell.

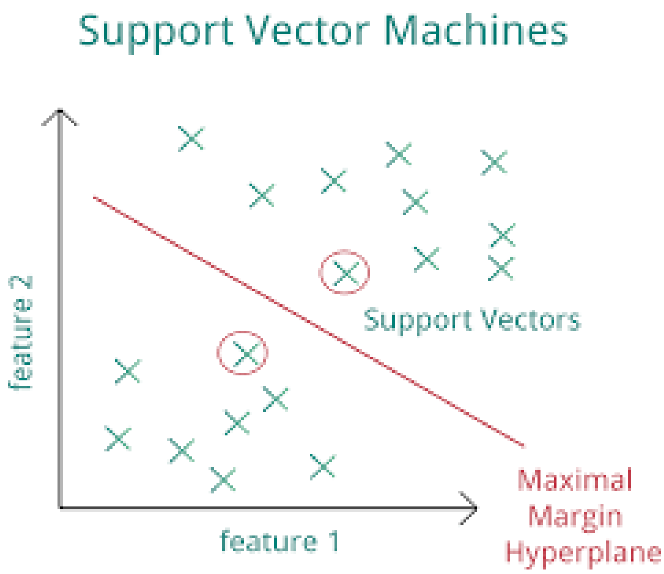
g = the movement cost to move from the starting point to a given square on the grid, following the path generated to get there.

h = the estimated movement cost to move from that given square on the grid to the final destination. This is often referred to as the heuristic, which is nothing but a kind of smart guess.
Here we use Euclidean distance to find **h**.
h = sqrt ((current_cell.x – goal.x)2 +(current_cell.y – goal.y)2
)

3. BUMPINESS DETECTION

For Bumpiness level detection Support Vector Machine (SVM) algorithm is used. Support vectors are data points that are closer to the hyper plane and influence the position and orientation of the hyper plane. Using these support vectors, we maximize the margin of the classifier. Deleting the support vectors will change the position of the hyper plane. These are the points that help us build our SVM. In SVM, we take the output of the linear function and if that output is greater than 1, we identify it with one class and if the output is -1, we identify is with another class. Since the threshold values are 1 and -1 in SVM, we obtain this reinforcement range of values ([-1,1]) which acts as margin. To implement the SVM algorithm we use the Scikit learn library and just call the related functions to implement the SVM model. The number of lines of code reduces significantly too few lines than the normal program without using the library.

SVM algorithm is used to draw a hyper plane between the classified set of classes. In our case the classes would be fast and slow. Based on the ups and downs the classification is done. According to this classification the speed control system is instructed about the rate of speed that should be followed.

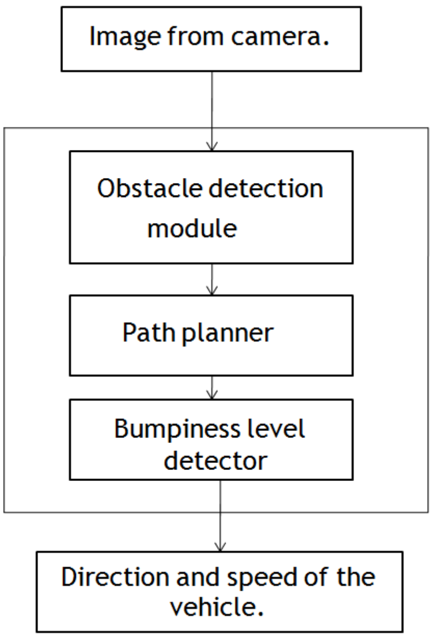


4. ARCHITECTURE

The system architecture comprises of above described modules with input and output of the system.

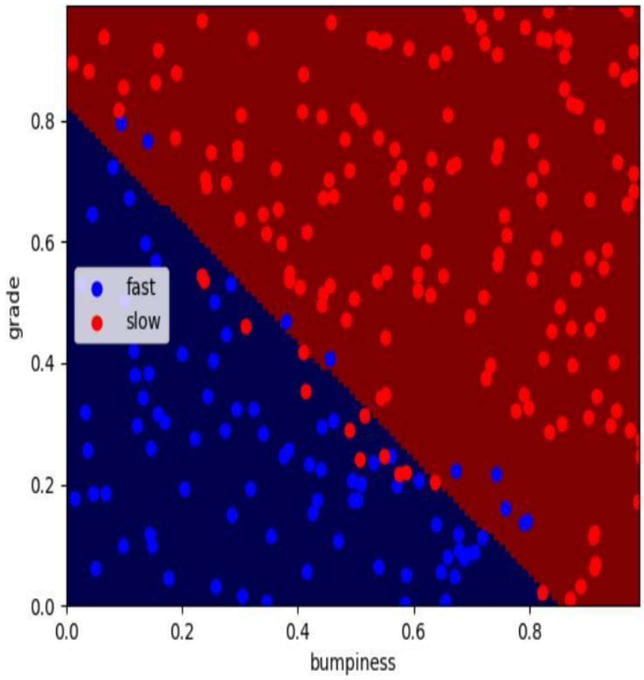
The input is given a frame of the video stream from the camera. That frame is processed by the system for detecting the obstacle and finding an obstacle free minimum path.

The system architecture is depicted as follows:



IV. RESULT AND IMPLEMENTATION

Classifying the bumpiness level with the SVM algorithm had resulted in an accuracy of 92% which is optimal than any of the classification algorithms.



The length of the code is also reduced in obstacle detection with the help of ImageAI module which adds efficiency to the system.

V. CONCLUSION

These labeled data were then used as training and testing data in a supervised learning algorithm (SVM) to establish a decision boundary. This decision boundary can then be used to automatically classify bumpiness level as high or low. In this paper we only

used two features, but this concept can be extended to many additional features. The data were limited in this article for illustration purposes.

At present, in case of SVM algorithm works in real-time on on-board computers. In urban environment, the operating speed of cars is one magnitude greater, so it requires special computers designed for autonomous driving for real-time running. Further possibility for speeding up our method can be the differential processing of roads.

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We would like to express our very great appreciation to Dr.K.Velmurugan, Head of Computer Science Department for his valuable and constructive suggestions during the planning and development of this research work. His willingness to give his time so generously has been very much appreciated. We would also like to extend our thanks to the technicians of the laboratory of the Computer Science department for their help in offering us the resources in running the program.

REFERENCES

[1] TU-Automotive, **"Driverless vehicles will continue to dominate auto headlines to automotive [online],"**April,2016,available: http://analysis.tuauto.com/autonomous- car/driverless-vehicles-willcontinue-dominate- auto-headlines [Accessed:10-April-2018].

[2] L. Fridman, D. E. Brown, M. Glazer, W. Angell, S. Dodd,B.Jenik,J. Terwilliger, J. Kindelsberger, L. Ding, S.Seaman,H.Abraham,A.Mehler,A.Sipperley, Pettinato, B. Seppelt, L. Angell, B. Mehler, and B. Reimer, **"Mit autonomous vehicle technology study: Large-scale deep learning based analysis of driver behavior and interaction with automation,"** Nov 2017, available: https://arxiv.org/abs/1711.06976.

[3] WHO, **"Global status report on road safety 2015. world health organization,"** 2015.

[4] Sean O'Kane (The Verge), **"History of autonomous car [online],"**available: http://research.omicsgroup.org/index.php/ History of autonomous car [Accessed: 21- April-2018].

[5] D. A. Pomerleau, **"Alvinn: An autonomous land vehicle in a neuralnetwork,"** in Advances in neural information processing systems, 1989,pp. 305–313.

[6] Q. Chen, U. Ozguner, and K. Redmill, **"Ohio state university at the 2004 darpa grand challenge: developing a completely autonomous vehicle,"** IEEE Intelligent Systems, vol. 19, no. 5, pp. 8–11, 2004.

[7] K. Mahelona, J. Glickman, A. Epstein, Z. Brock, M. Siripong, andK. Moyer, **"Darpa grand challenge,"** 2007.

[8] N. Spinrad, **"Google car takes the test,"** Nature, vol. 514, no. 7523, p.528, 2014.

[9] Michael McAleer (The Irish Times), **"Audi's self-driving a8: drivers can watch youtube or check emails at 60km/h [online],"** July 11, 2017,available:https://www.irishtimes.com/life -and-style/motors/audi-sself-driving-a8- drivers-can-watch-youtube-or-check-emails-at-60km-h-1.3150496 [Accessed: 19-April-2018].

[10] Andrew J. Hawkins (The Verge) ,**“Waymo is first to put fully self-driving cars on us roads without any safety driver in a car [online]”.** Nov7,2017,available: https://www.theverge.com/2017/11/7/16615290/waymo-self-driving-safety-driver-chandler- autonomous. [Accessed:19-April-2018].

[11] P. G. Trepgnier, J. E. Nagel, M. T. Dooner, M. T. Dewenter, N. M. Traft, S. Drakunov, P. Kinney, and A. Lee, **“Control and systems for autonomously driven vehicles,”** Feb. 28 2012, U.S Patent 8,126,642.

[12] J. Guivant, E. Nebot, and S. Baiker, **“Autonomous navigation and map building using laser range sensors in outdoor applications,”** Journal of robotic systems, vol. 17, no. 10, pp. 565–583, 2000.

[13] A. Paul, R. Chauhan, R. Srivastava, and M. Baruah**, “Advanced driver assistance systems,”** SAE Technical Paper, Tech. Rep., 2016.

[14] V. V. Dixit, S. Chand, and D. J. Nair, **“Autonomous vehicles: disengagements, accidents and reaction times,”** PLoS one, vol. 11, no. 12, p. e0168054, 2016.900.

[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al., **“Human-level control through deep reinforcement learning,”** Nature, vol. 518, no. 7540, p. 529, 2015.

[16] C. J. C. H. Watkins, **“Learning from delayed rewards,”** Ph.D.dissertation, King’s College, Cambridge, 1989.