Automated Traction using Object Detection on Low End Systems

Chandan Rajpurohit
Department of Computer Engineering
K.C. College of Engineering and Management Studies and Research
Thane, India

Rait Pravleen Singh
Department of Computer Engineering
K.C. College of Engineering and Management Studies and Research
Thane, India

Harish Unnikrishnan
Department of Computer Engineering
K.C. College of Engineering and Management Studies and Research
Thane, India

Sonal Balpande
Department of Computer Engineering
K.C. College of Engineering and Management Studies and Research
Thane, India.

***Abstract*—There are many available embedded systems like (Raspberry pi) which are used to make IOT devices, Robots, etc . While such an application of this type of systems are to make autonomous system (Robot) which can travel without any human intervention to a specified destination. This are done using object detection and navigation algorithms. Object detection uses computer vision models like YOLO, Mobile net, etc, which provide a good performance on this type of system at the same time uses the memory of this systems to fullest, Hence the motive of this paper is to use this Object detection mechanisms effectively by using some extra sensors for autonomous traction. The autonomous traction uses specific modules to differentiate the type of area the system is in and navigate in collaboration with object detection.**

Keywords—Computer Vision, Automated traction, Algorithms, Machine Learning.

I. Introduction

Future technology is advancing daily and robots have made their way into human life providing a comfortable living environment and ensuring safety. They are useful in locations where person himself cannot enter [1]. With decision making capability they can also be utilized in rescue and research purposes. The system, proposed in this paper follows to objective to lower the amount of processing required by the object detection system at the same time not hampering the automated traction systems decision making capabilities. One of the powerful component of this type of systems are object detection , this object detection is achieved by using pre trained or custom trained models of YOLO, Mobile net SSD, these models can be modified for embedded system to work faster but while the increase in performance these models require memory and processing of the embedded system to the fullest. Hence here our system tries the lower the memory used by these models by coupling them with specific sensors and triggering them when ever needed. Our system uses a simple raspberry pi camera to capture images from, a usonic to check for obstacles in the path and Mobile net SSD pre trained model for object detection.The usonic works continuous to detect of an obstacle is present in the path of the system accordingly passes this information to system , after which system decides to trigger object detection or not. After the whole object detection mechanism the results are passed on the the automated traction module. Here in the module uses an api to get the direction such that the system can move in the right path to react the correct destination. This the module which decides weather the system moving in a smaller or a larger area. After receiving the information about the object in path by the Object detection mechanism the automated traction module decides weather it can change its current path and find a new path to reach destination or there is a dead end, In case of a dead end the module decides to backtrack to the source, else tries to find a new path . This process to find a new path is achieved using a virtual compass module which keeps track of the turns taken by the system and helps the system to taken a new turn such that a new path to the destination is found.

II. Literature Review

The systems developed till now had object detection which where tested in ideal scenarios. Detecting Objects in real life is not much easy as considered in the idea situations .In real file conditions for object detection needs to detect many dimensions of the surrounding like depth, height, lighting variation ,various contrast area's, This can be made with training object detection models with a huge data set, but in turn this will make these models to perform less or hog memory on small, low end or embeded system's. Hence we can use various sensor's coupled with object detection models to boost the performance of such system's[2].

Raspberry pi type of embeded systems have a very small amount of memory on board , this amount of memory is only sufficient to run essential tasks on the system, if a colored image is used for the process of features extraction then to process all dimensions of the image will become again a memory hogging task hence will cause others tasks to be deprived of the memory hence to boost the process the system can make use of a detailed monochrome image for feature extraction and hence will eventually require less memory to process. [3].

One of the problem with this small/low end systems is that this systems do not come with integrated cooling systems. If this systems are used to develop projects like of Object Detection and navigating them this systems have use a camera by default through which this systems can capture images/frames and proceed with further object detection process. The above process of capturing images is done contentiously which generated significant heat in these systems following which if the object detection is performed continuously on the system it cause cause overheating which may result in hardware damage or degradation on the processing power of this system, Hence we can create an event based trigger for object detection mechanism which will be triggered when ever the sensors attached to the system detects some abnormal behaviour in surrounding [5].

With inclusion of additional sensor's on the system to help the object detection triggering mechanism addition of a standard web camera on system will cause the system to get bigger in size. Instead of a standard web camera to be used in the system we can use a spy camera like on which comes with Raspberry pi system this camera's are smaller in size, easy to handle and is capable of capturing a good quality of images also an record videos at a good frame rates [6].

The system mentioned in this paper is capable of avoiding the obstacle in its path using an object detection algorithm. The raspberry pi camera module on board will detect the obstacle and using image processing / computer vision algorithm detect the system and send response back to the raspberry pi , hence according to the response it will find a new obstacle free direction to destination of the system. This direction change can be done using an api support which will provide the system with current co-ordinates and the new direction co-ordinates [7].

III. Requirement Specification

Abbreviations and Acronyms

SSD - Single Shot Detected.

YOLO - You Only Look Once.

Usonic - Ultrasonic Sensor

Pdrange - Pre-defined Range

Scope

The use of technology in every field is increasing day-by-day with this enhancement in technology we want to increase intelligence of system which will be used in future The Automation and Navigation Algorithms explain in this paper will help make this work efficiently without any human interaction also object detection will make it more precise and efficient. The Algorithms are made in python language which increases its scope of integration with various systems especially on Raspberry Pi. Techniques used for the working of Object Detection and Automated Traction simultaneously will increase scope of this application in various areas of technology.

IV. Methodology

The Fig. 1. Object Detection flow explain how and when to trigger the object detection. First the system uses an usonic to sense the surrounding area so as to detect if any object is in a pdrange of the sensor. If there is no object in this range then the sensor continues to scan the surrounding for such a event. If there is an object detected in the pdrange of the sensor then the sensor sends an alert to the system.

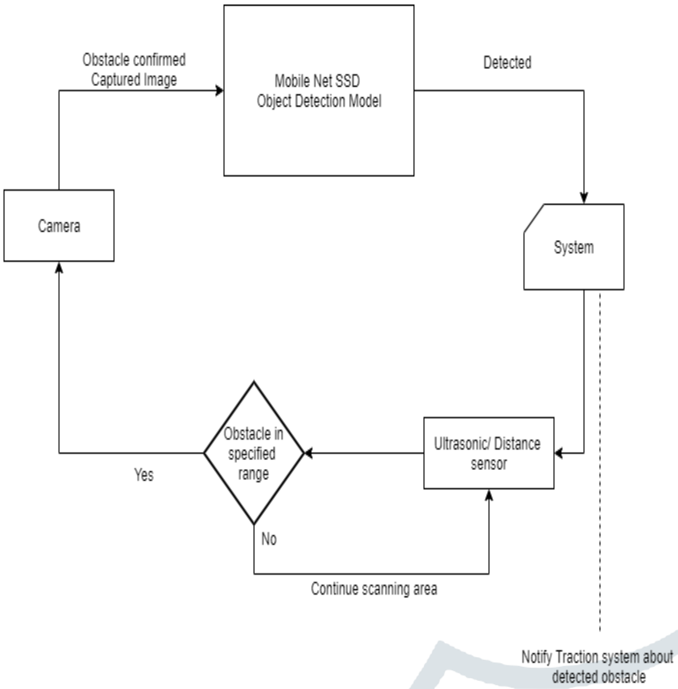

Fig. 1. Object Detection flow

Once an alert is received form the sensor module the system triggers object detection mechanism which will be powered by Mobile net SSD, so the camera starts capturing images , for which the system uses popular image processing library called OpenCv. the images are taken at full camera's resolution in RGB mode so that the object detection model can work on all the dimension of images which might not be possible on only monochrome images. After capturing of images system processes the images one by one and finding if any object is in the images the object detection system tries to find if there is a dead-end in the image or an static obstacle in the path of the system and forwards this information to automated traction which will help the automated traction to decide weather to find an new path to destination or to backtrack to the source.

This mechanism of triggering the object detection mechanism only when sensor senses an object helps the system to save processing power and only use it whenever required.

Automated Traction uses Navigation module which is divided into two parts

1. This part of the module deal with fetching the current and destination co-ordinates from an API and match it with the on board GPS to maintain its correct path towards the destination.
2. This part of the module deals with maintaining the path of the system such that it does not collide while moving.

Navigation module parts mentioned above are referred to as Large Area Navigator and Small Area Navigator. Small Area Navigator does not perform any api calls but it uses the data from the api which is fetched by Large Area Navigator. Large Area Navigator helps Automated Traction work more efficient. Large Area Navigator uses direction api such as graphopper to get path co-ordinates in a JSON format which can we decoded by the system easily to get path.

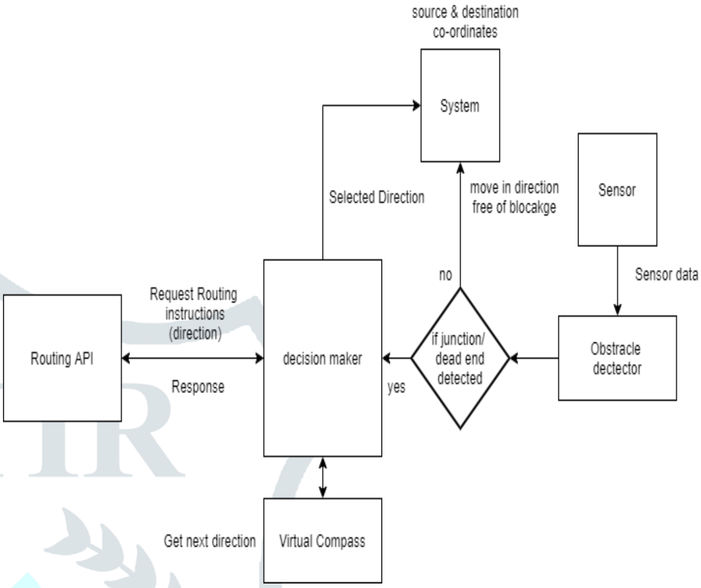

Fig..2. Automated traction

Large Area Navigator:

This is part gets the input from the GPS which gives the current location co-ordinates. And provides the these co-ordinates to API

The output from the API will be set of instructions regarding the turns in roads. For e.g., take left turn and go 4km forward. This decision is also taken by checking the object detectors status

Small Area Navigation:

This module runs continuously and terminates when system reaches to destination.

This module calls large area navigator in regular interval of time to get further instructions regarding navigation. If system reaches the destination the large area navigator send stop command to small area navigator.

Small area navigator takes decisions based on the obstacle decision. If obstacle comes on front of system, it stops its motion and takes decision whether to turn left or right.

System maintains a virtual compass so that system can't take random turns and get totally out of the path narrated by navigation api, which means while taking turns system have to consider the direction also. For which system uses a virtual compass module to keep track of turns to take.

Virtual Compass:

1. There will be a global state constant called compass angle which indicates the degree turn. By default it is 0 degree.
2. System can only take a turn in 90 degrees.
3. If system takes a left turn compass angle's value get updated by 90 degrees and for right turn compass angle's value get updated by -90 degrees.
4. When system reaches at a junction it checks the compass angle's value, it per-calculates that on taking which action (left or right turn) the compass angle will become zero or 360 degrees else value come close to zero or 360 degrees. system always tries to turn to that direction due to which compass angle become zero or 360 degrees.
5. This mechanism enables system to maintain its motion in straight direction; this prevents the system from getting distracted from its straight path because of blocks in path.
6. In case system encounters a dead end where there is no path in forward, left or right, then it will backtracks and keeping checking in left and right direction. If sensor detects no blockage in left or right direction it takes that turns respectively.

V. FUTURE WORK

On this Application the future work can be consider as we can use neutral networks to train the system such that there is no need to define a specific pdrange the system can adapt to the changes in the distance of objects in path and change its detection range accordingly. Training Object detection models for various categories of image will increase the detection and traction ability of this application. This application can be used or developed for High End Systems. Also the system instead of using on board processing of image can use a cloud system to send the images to a cloud function which in return processes images to find out the objects in the image and return this information to the automated traction system. The Traction Algorithm can be improvised for better result and collision free traction.

VI. CONCLUSION

Applications using such kind of system will be able give better performance on both Low End and High End Systems. Automation Traction with Object Detection will also be helpful in various automobile industries, Health care centers and will make change in various manual stuffs. According to this Traction model both small area and large area navigation is possible. Operating it in manual and autonomous mode also make it a better application. Object detection helps in better traction and also help in collision free traction. Since multiple robotics models need high processing power and memory, hence this type of usage of sensors to trigger the object detection on certain intervals of time will allow outer tasks to run on the machine simultaneously also allow the object detection to work when ever required .This system is successfully operating on Low End device i.e. Raspberry Pi Model 3B with 1GB RAM. Hence such memory and processing efficient model can be integrated with many applications to give excellent result.

REFERENCES

[1] Anushka Gupta, Shailesh Tanwar, Vishvander Singh, M. Lavanya and V. K. Mittal,"Design of a GPS-Enabled Multi-modal Multi-Controlled Intelligent spy bot, "Conference on Advances in Computing, Communications and Informatics (ICACCI)" 2016

[2] Arnold W. M. Smeulders, Dung M. Chu, Rita Cucchiara ,"Visual Tracking: An Experimental Survey","IEEE" 2014

[3] Danilo Avola, Luigi Cinque, Cristiano Massaroni and Gian Luca Foresti, "A Multipurpose Autonomous Robot for Target Recognition in Unknown Environments , "IEEE" 2016

[4] Julien Marot, Salah Bourennane, "Raspberry Pi for Image Processing Education" ,"European Signal Processing Conference (EUSIPCO)" 2017

[5] Dhanashree Vijayrao Madhekar, Prof. Mrinal Rahul Bachute , "Real Time Object Detection and Tracking using Raspberry Pi ", "International Journal of Engineering Science and Computing" 2017

[6] Amruta Nikam, Akshata Doddamani, Divya Deshpande, Shrinivas Manjramka , "RASPBERRY Pi BASED OBSTACLE AVOIDING ROBOT", "International Research Journal of Science and Technology" 2017

[7] Zeeshan Shaikh, Priyanka Gaikwad, Nikhil Kare,Swapnali Kapade,Prof.M.V.Korade, "A IMPLEMENTATION ON - SURVEILLANCE ROBOT USING RASPBERRY-PI TECHNOLOGY", "International Research Journal of Engineering and Technology" 2018