Comparative analysis of image processing techniques for obstacle avoidance and path deduction

R Rajavarshini¹, S Shruthi², P Mahanth³, Boddu Chaitanya Kumar ⁴and A Suyampulingam⁵

Department of Electrical and Electronics Engineering
Amrita School of Engineering, Coimbatore
Amrita VishwaVidyapeetham, India
Email:¹cb.en.u4eee17136@cb.students.amrita.edu,²a_suyampulingam@cb.amrita.edu,
³cb.en.u4eee17126@cb.students.amrita.edu

Abstract. The growing need for automation has a significant impact on our daily lives. Automating the essentials of our society like transportation system has plenty of applications like unmanned ground vehicles in military, wheel chair for disabled, domestic robots, etc., There are driving, braking, obstacle tackling etc., to a transportation system that can be automated. This paper particularly focuses on automating the obstacle avoidance which provides intelligence to the vehicle and ensures a high degree of safety and is performed using image processing algorithms. Edge based detection, image segmentation, and Machine Learning based method are the three image processing techniques used to detect and avoid obstacles. Haar cascade classifier is the machine learning method where Haar cascade analysis is performed for better accurate results with justifying graphs and parametric values obtained. A comparison of the three image processing algorithms is also tabulated considering obstacle size, colour, familiarities and environmental lightings and the best image processing algorithm is inferred.

1. Introduction
Unmanned vehicles are being used to a large extent these days. The simplicity and efficiency of unmanned vehicles in the transportation world is the reason for its success. GPS tracking, steering control, obstacle avoidance, traffic signal and sign boards recognition are the major aspects of a self-driving automobile [1]. The project is focused on the obstacle detection and obstacle avoidance of an unmanned vehicle.

Obstacle detection and avoidance can be done using 2D-LIDAR, RADAR, vision cameras, ultrasonic sensors, Infra-Red sensors, etc., [2] Image processing is a cost-effective and high accurate way of obstacle detection. Vision of an automobile can be given by a camera and the knowledge to understand the obstacles and avoiding it can be provided using the image processing algorithms. Processing of digital images and extracting its features like edges, texture, etc., compressing the image, analyzing the image and then obtaining the results are the parts involved in image processing [3]. Image processing can be performed using different libraries and computer languages. This project comprises image processing algorithms using OpenCV library in python.
Obstacle detection is the first step to avoid obstacles. Videos recorded using a camera provide the necessary input required for processing. The python code, built with relevant image processing techniques process the input and mark the obstacles. This is used in wide range applications like domestic robots, unmanned ground vehicle for military purposes, self-driving automobiles, automatic wheel chair for disabled, etc., [4]. Image processing techniques using OpenCV library has various primary concepts like the edges, colour segmentation, threshold segmentation, histograms of oriented gradients, machine learning, etc., [5]. All these concepts can be used to obtain the common goal of obstacle detection and avoidance. The project performs three concepts from image processing to obtain obstacle avoidance.

Machine learning is the study of computer-based algorithms that improves its accuracy through experience and by the use of data. Paul Viola and Michael Jones had proposed Haar feature-based cascade classifier in 2001[6], which provides effective object detection used in many projects all over the world. The project has used this algorithm for detection and avoidance of obstacles [10]. The edge-based detection and colour-based detection which are the basics of image processing algorithms are also developed in the project to attain the goal of obstacle avoidance [11]. These algorithms are purely based on feature extraction for detection of obstacles. The aim of the edge detection is to partition an image based on abrupt changes in intensity segmentation. The thresholding supports colour-based algorithm in obtaining the obstacles from the floor based on colour differences in the image. The paper thoroughly analyzes the proposed methods to perform obstacle avoidance using these three algorithms and discusses their advantages and disadvantages.

2. Edge based detection

 Edge detection is one of the methods that are used for the obstacle avoidance. The mathematical models that detect points in a digital image, where the brightness of the image goes through a distinct and a prominent change comprised of the edge detection. The result of applying edge detection in the image is the identification of the boundaries of the objects which are marked in white colour (pixel value = 255) and remaining in black colour (pixel value = 0). The different edge detection methods can be classified into two broad groups, zero-crossing based and search based. They primarily differ in the nature of the smoothing filter involved and the technique applied to measure the strength of the edges.

The canny edge detection algorithm is used for this method. It is a multi-stage algorithm which has the capability of detecting a high range of edges in the image. The stages are noise reduction, intensity gradient, gradient magnitude thresholding, double thresholding, and hysteresis thresholding. It can be used to obtain significant data that is structural from an image and another important aspect of this algorithm is that the quantity of data to be processed is reduced to a large amount.

2.1 Execution of the algorithm

The first step of the execution is the selection of input image and applying the canny edge detector on it. The input image is shown in Figure 1 a has a floor on which random obstacles are placed. The canny detection is used to obtain the edges of obstacles in the image which is the base work. Figure 1 b. shows canny edge detection performed on the image, Segregation of the pixel values as 0’s and 255’s happens. The threshold values for edge detection are fixed, and lines based on these values are drawn. The image gets filled from the bottom row of pixels and when edges are encountered, the filling is stopped as in Figure 1 c. Obstacles are not filled, and boundaries of obstacles can be identified approximately. The filled area is the path obtained for travelling. Erosion is used to look for small differences in a large homogenous pool and negate the difference to improve the overall accuracy. Smoothing is used to smoothen the boundaries obtained.

Array sets of continuous columns containing maximum filled row values is created. The clear path should ideally contain no obstacle and should have enough space for the vehicle to travel. Lengths of
such arrays are checked and should be greater than half of the height of the entire image. A transparent rectangular area is marked as shown in Figure 1d and it is considered to be the safe region. The system is expected to encounter no obstacle if it stays in the safe region. The trajectory line showing direction in which the vehicle should travel is also marked as in Figure 1d.

![Figure 1. Edge-based detection: a) Input image b) Canny edge detected c) Side-filled d) Deducted path](image)

### 3. Image segmentation

A digital image is divided into sets of pixel segments and this process is defined as Image segmentation. When the representation of an image is simplified and more meaningful, analysis becomes easy. The idea is to segment every pixel to a corresponding primary colour based on the colour spectrum. Pixels that share common base colour properties come under a common label. There are two types of image segmentation, Semantic and Instance segmentation.

Simple thresholding is a semantic segmentation method applied on an image. In simple thresholding, if the value of the pixel value is greater than the threshold value, a particular value is assigned to it, if not, another value is assigned. The threshold in OpenCV is used to allocate pixel values by using a threshold value. There is a lot of threshold functions available in OpenCV library like thresh binary, thresh_binary_inv, thresh_trunc, thresh_tozero, thresh_tozero_inv, etc., and thresh binary and its inverse are used in the execution of algorithm. The area along the points that constitute the boundary of image is a contour and it can be used for object detection.

### 3.1 Execution of the algorithm

The input image in Figure 2a consist of a floor on which different obstacles are placed. The first step is to identify the pixel value of the road from the bottom part of frame, where an obstacle does not lie. The pixel value obtained in RGB is converted to grey image. This pixel value obtained in grey scale is set as the threshold value. The next step is to analyse the other pixels in the frame. Any pixel value above the threshold value or below the threshold value is converted to white colour (pixel value = 255). The idea is to make the region containing threshold value insignificant (in this case the road) and to highlight the regions only containing the non-threshold pixels (everything other than the road) as shown in Figure 2b. The next stage is finding the contours, the boundaries of all obstacles marked as white in the frame. Every contour is marked around the enclosed area that is qualified as an obstacle. Figure 2c. shows
bounding boxes marked for all the identified contours in the image. The bounding boxes present in the left side of the frame are grouped and same is done on the boxes in the right. The maximum distance between any two detected obstacles in left and right half of the image is considered as the clear path. Finally a transparent rectangular area, the safe region is marked as shown in Figure 2d and the trajectory line showing direction in which the vehicle should travel is also marked as in Figure 2d.

3.2 Pre-Processing
The input image contains obstacles placed on different floorings. The unwanted features of the floor like the patterns, colours, irregularities, etc., can cause mispredictions in the colour and edge based detections. The input image is processed before using so that the texture of the floor does not affect the obstacle detection in the previously mentioned methods and this step is called pre-processing. The pre-processing involves two stages.

3.2.1 Dilation. Dilation is a morphological operation that enhances the image features. Dilation is the function requires two inputs, an image to be dilated, and a two dimensional structuring element. It adds the pixels to the boundaries of objects in an image approximately increasing the area.

3.2.2 Blurring. This stage is used to reduce the noise in the image. When an image with a low pass kernel uses convolution, the sharp edges are blurred and it is called as blurring the image. There are different types of blurring techniques that include averaging, Gaussian filtering, median filtering and bilateral filtering.

4. Machine learning based Algorithm
Machine learning is the recently popular method used for obstacle avoidance. Haar cascade is an algorithm based on machine learning, where haar-like features are extracted for object recognition. The areas in which Haar cascade classifier is commonly implemented are obstacle detection and facial recognition.

The classifier is trained with the data set that belongs to the category of positive images that should be detected by classifier and the negative images that needs to be ignored. The images of toy cars are considered as positive dataset in this algorithm execution, negative dataset can contain anything other than cars, empty floors, roads and the obstacles that looks like car in shape can be used for relevance. Once the model is trained with these images, it can be applied to detect the obstacles on the floor.
4.1 Analysis

4.1.1 Training dataset. The training dataset consist 735 images out of which 423 are negative images and 312 are positive images. The training can be improved by varying the boosting parameters that are, number of stages, number of threads, acceptance ratio break value, boost type, minimum hit rate, maximum false alarm rate, weight trim rate, maximum depth. The tree depth is considered as the varying parameter for better cascade detection. Four different categories of sets with tree depth ranging from (1, 4) are tested to choose the optimum value that can be used as the tree depth. This is done to make sure that the number of features evaluated at each stage is kept minimal as tree depth is greater than four features extracted and increases the training time required to train the cascade.

The number of stages is set to be 15. Values are chosen in such a way that the training does not stop before acceptance ratio break values reaches -1. The other parameters are Number of threads = 7, feature type: HAAR, Haar feature type: Basic, Boost type: GAB, Minimum hit rate = 0.995, Weight trim rate = 0.95, Maximum false alarm rate = 0.5, Maximum weak tree = 100, Maximum depth weak tree = 3, Number of positive images = 423 and number of negative images = 312.

4.1.2 Testing dataset. The testing dataset consist of entirely total different images of 214 which includes 141 positive images and 73 negative images and different testing dataset is also chosen for better analysis. After testing the different tree depths parametric values such as Precision, Accuracy, F1score, and Recall are calculated as shown in table 1.

The decision tree with depth = 3 is chosen to be the optimum tree depth value. The parameter precision has been considered to come up with the optimal tree depth because, it is quantified based on the true positive in the predicted positives. The false positive of the cascade needs to be maintained minimum for better detection and so higher the precision lower the false positive rate. Hence tree depth = 3 has higher precision and is the optimum cart tree depth but, accuracy of the classifier is very less. Hence, higher number of negative dataset is collected to improve the accuracy of the classifier and a total of 1264 images (540 positives and 724 negative images) are trained in the cascade. The parametric values of the well trained classifier is shown in Table 2.

The Receiver Operating Characteristic (ROC) curve gives the behavior of a classifier as the discrimination threshold is varied and the minimum neighbour’s parameter is varied to observe false positive and true positive percentages. The ROC curve is plotted based on the performance curve which is the relation between the false positive and true positive. The ROC curve is plotted in Figure 3 and it is inferred that the value of true positive percentage is close to double the value of false positive percentage at majority of the points. Hence the rate of true positive is higher than false positive, the cascade can detect better.

| Classifiers | Precision | Accuracy | F1 score | Recall |
|-------------|-----------|----------|----------|--------|
| Depth =1    | 0.863     | 0.479    | 0.612    | 0.475  |
| Depth =2    | 0.872     | 0.410    | 0.523    | 0.333  |
| Depth =3    | 0.948     | 0.383    | 0.393    | 0.248  |
| Depth =4    | 0.888     | 0.383    | 0.333    | 0.205  |

Table 1. Parametric values of cascade models
4.2 Execution of the algorithm

The classifiers are trained based on a dataset of positive and negative images. The input image for machine learning based method is selected as the toy cars placed on the floor. As the classifier is trained to identify only toy cars, anything other than toy cars cannot be identified. The image before detecting is converted to grey scale as in Figure 4 a. The obstacles are detected using OpenCV detect function, the xml file of the trained cascade is loaded and the co-ordinates of the boundaries of the obstacles are obtained as shown in Figure 4 b.

Correlation tracker function from digital library is used to keep the track of the obstacles in the subsequent frames. The identified co-ordinates from the obstacles are added into the tracking dictionary with unique car ID provided for each obstacles. The co-ordinates of the obstacles in the subsequent frames are compared with the existing car ID in the dictionary. The co-ordinates of the obstacles that are inferred to be new are added and co-ordinates that are expired removed from the tracking dictionary. The lower half of the frame is taken into consideration and the frame is divided into two halves left and right. The distance between the last obstacle in the left half and the first obstacle present in the right half is calculated and hence the clear path is obtained. Figure 4 c shows transparent rectangular area, which is the safe region to travel where the trajectory line marked for direction.

### Table 2. Parametric values of trained classifier

| Classifier   | Precision | Accuracy | F1 score | Recall |
|--------------|-----------|----------|----------|--------|
| Trained classifier | 0.873     | 0.534    | 0.450    | 0.304  |

Figure 3. ROC Curve

![ROC Curve](image-url)
5. Comparative analysis
The results are obtained for the three image processing techniques, Machine learning based, Image segmentation and Edge based detections by executing in OpenCV software. The results are validated by considering the output of the algorithms when performed on different floorings, size of obstacles: small, large, medium, familiarity of obstacles, environmental lightings, area of obstacles detected, multiple detection. The comparative analysis of the three algorithms are tabulated in Table 3.

The computational time taken on a common video is considered and the accuracy of obstacle detection and avoidance in the video for all the three algorithms is maintained 100%. The frames per second and the total time taken for a single frame is calculated for precise quantization alongside with the time taken for the stop case scenario where the vehicle is not able to pass through the free path. The results are listed in Table 4.

The algorithms are analysed with respect to the different parameters in the comparison matrix and 30 different videos have been run and tested in order to be able to come up with an accurate analysis. The Machine Learning algorithm turns out to be the best of the three in terms size, colour of obstacles, texture of floorings, environmental lightings etc., The computational time is reduced in the machine learning algorithm and the overall performance is obtained best by this algorithm. The algorithm requires training to a large extent and the performance of the algorithm is impacted by the training. The obtained merits and demerits of all three algorithms are also listed.

Table 3. Comparison Matrix

| Algorithm       | Performance on Different floorings | Orientation | Size | Unfamiliar obstacle | Environment lighting | Area of obstacle that can be determined | Multiple detection |
|-----------------|------------------------------------|-------------|------|---------------------|----------------------|-----------------------------------------|--------------------|
| ML based        | Good                               | Partially Depends on accuracy | Yes   | Yes                 | Yes                  | No                                      | High               |
|                 |                                    |             |      |                     | High                 | Medium                                  | High               |
|                 |                                    |             |      |                     |                      | Approximate Position                     |                    |
| Edge based      | Fair                               | Floors with very sharp patterns can distract the algorithm | Yes | Yes             | Yes                  | Yes                                     | High               |
|                 |                                    |             |      |                     |                      | Low                                     | High               |
|                 | Poor                               | Flooring should not have multicolour patterns | Yes | Yes             | Yes                  | Yes                                     | Low                |
|                 |                                    |             |      |                     |                      | Approximate Position                     |                    |
| Image segmentation |                                  |             |      |                     |                      | It also considers the shadows of the obstacles | Yes               |
Table 4. Quantitative Analysis of Time taken

| Algorithms                  | Processing time for a video (sec) | Time taken to process a frame (sec) | Average frame per sec (fps) |
|-----------------------------|----------------------------------|-----------------------------------|---------------------------|
| ML based                    | 32.5625                          | 0.052                             | 52.3412                   |
| Edge based detection        | 98.00                            | 0.1568                            | 9.2720                    |
| Image Segmentation          | 23.5635                          | 0.0376                            | 43.4213                   |

5.1 Merits and Demerits

5.1.1 Edge based detection.
Merits
- Wide range of obstacles can be detected.
- The colour of the obstacle does not impact the accuracy.
- It executes and produces results in finite time.
Demerits
- It is an exhaustive search algorithm.
- Large memory and computation power are required which results in lesser efficiency and high computational time.
- Sharp patterns of the floor texture has an impact on accuracy.
- If background of the frame has other obstacles then that is also processed.
- Accuracy of marking obstacles is lost as the algorithm also considers the shadow as an object.

5.1.2 Image segmentation.
Merits
- Wide range of obstacles can be detected.
- Very less space and computational time is required for execution of algorithm.
- It executes and produces results in finite time
Demerits
- The colour of the obstacle has an impact on the accuracy.
- A single colour floor should be used for better results.
- If background of the frame has other obstacles and colours then that is also processed
- Accuracy of marking obstacles is lost as the algorithm also considers the shadow as an object.

5.1.3 Machine learning based method
Merits
- Can be trained as per the requirement of path.
- This can identify and distinguish the obstacle wisely using its features.
- High accuracy while detecting the obstacles and avoiding it and so the floor texture doesn’t have an impact.
- Detection can work on all environmental lightings, with any size and any background colours.
- This is very useful if only specific obstacles need to be tackled.
- Computational time is lesser than edge detection-based method but higher than image segmentation, because it identifies the obstacle with the features, which are loaded to the trained cascade.
Demerits
- All obstacles cannot be detected, only objects trained using the cascade are detected.
- In some case it does not exhibit completeness, when the cascade is not trained to a higher accuracy value.

6. Conclusion
There are braking, GPS tracking, allowing it to discover the optimal path, obstacle avoidance, driving examples from the ocean of options that can be automated in an unmanned vehicle. The focus of this paper is to automate the obstacle detection and avoidance. All obstacles present in the path are detected and the clear path after consideration of the obstacles, is deduced. The three methods that are based on image processing: Machine learning based method, Image segmentation, and Edge based detection, have been used for the obstacle avoidance, detection and the comparative study has been done. The three algorithms are implemented by simulation and are tested with different set of videos alongside considering different parameters like environment, obstacle size, orientation, environmental lighting. The results obtained from three methods are analysed to determine the best method which has more accuracy and fast speed response.

The machine learning algorithm turns out to be the best among all the three methods because the outer parameters like environmental lighting, size, orientation do not have an impact on it whereas for other two methods, the parameters do impact on the efficiency. The colour of the obstacle has also an impact on the accuracy in image segmentation method but for machine learning algorithm it depends only on training and also it completes the processing in a good amount of time. However, the processing time can be further reduced by using more powerful processor, so considering all these, it can concluded that machine learning algorithm is the most reliable one. The comparative analysis performed and valuable insights come in handy while deploying it for real-time applications. This analysis can be made a part of any unmanned vehicle after resolving the compatibility issues. When working models are made, it becomes ready to be implemented in various environments and in different contexts.

7. References
[1] McCall J C and Trivedi M M 2006 Video-Based Lane Estimation and Tracking for Driver Assistance: Survey, System, and Evaluation IEEE Transactions on Intelligent Transportation Systems 7 20–37
[2] Huang K, Wang L, Tan T and Maybank S 2008 A real-time object detecting and tracking system for outdoor night surveillance Pattern Recognition 41 432–44
[3] Latha M, Poojith, Reddy A, Amarnath B V, and Kumar G 2014 Image Processing in Agriculture International Journal of Innovative Research in Electrical 2 6
[4] Tangsukhsant W and Wada C 2018 Static Obstacle Detection along the Road with a Combined Method 2018 11th Biomedical Engineering International Conference (BMEiCON)
[5] Fouad A M, Sharkawy R M and Onsy A 2019 Fixed Obstacle Detection for Autonomous Vehicle 2019 IEEE Conference on Power Electronics and Renewable Energy (CPERE)
[6] Viola P and Jones M 2015 Rapid object detection using a boosted cascade of simple features Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001
[7] Sen R, Siriah P and Raman B 2011 RoadSoundSense: Acoustic sensing based road congestion monitoring in developing regions 2011 8th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks
[8] Kalyan S S, Pratyusha V, Nishitha N and Ramesh T K 2020 Vehicle Detection Using Image Processing 2020 IEEE International Conference for Innovation in Technology (INOCON)
[9] Schmidt A and Kasiński A 2007 The Performance of the Haar Cascade Classifiers Applied to the Face and Eyes Detection. Advances in Soft Computing 816–23
[10] Phuc L T, Jeon H J, Truong N T and Hak J J 2019 Applying the Haar-cascade Algorithm for Detecting Safety Equipment in Safety Management Systems for Multiple Working Environments *Electronics* **8** 1079

[11] Modi S B, Chandak P, Murty V S and Hall E L 2001 *Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision*

[12] Saji M, Sridhar M, Rajasekaran A, Kumar R A, Suyampulingam A and Krishna Prakash N 2020 IoT-Based Intelligent Healthcare Module *Advances in Intelligent Systems and Computing* 765–74

[13] A. E A, Sugathan S K and Ho A T 2008 Receiver Operating Characteristic (ROC) Graph to Determine the Most Suitable Pairs Analysis Threshold Value *Advances in Electrical and Electronics Engineering* - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008

[14] Vinuchandran A V and Shanmugasundaram R 2017 A real-time lane departure warning and vehicle detection system using monoscopic camera 2017 *International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*

[15] Ameer S, Shah M A, Khan A, Song H, Maple C, Islam S U and Asghar M N 2019 Comparative Analysis of Machine Learning Techniques for Predicting Air Quality in Smart Cities *IEEE Access* **7** 128325–38

[16] Yustiawati R, Husni N L, Evelina E, Rasyad S, Lutfi I, Silvia A, Alfarizal N and Rialita A 2018 Analyzing Of Different Features Using Haar Cascade Classifier 2018 *International Conference on Electrical Engineering and Computer Science (ICECOS)*

[17] Lijun X 2011 Moving object segmentation based on background subtraction and fuzzy inference 2011 *International Conference on Mechatronic Science, Electric Engineering and Computer (MEEC)*

[18] Baskar A and Gireesh Kumar T 2017 Facial Expression Classification Using Machine Learning Approach: A Review. *Advances in Intelligent Systems and Computing* 337–45

[19] Bhavana T, Nithya M and Rajesh M 2017 Leader-follower co-ordination of multiple robots with obstacle avoidance 2017 *International Conference On Smart Technologies For Smart Nation (SmartTechCon)*

[20] Nehru M and Padmavathi S 2017 Illumination invariant face detection using viola jones algorithm 2017 4th *International Conference on Advanced Computing and Communication Systems (ICACCS)*

[21] Fazeen M, Gozick B, Dantu R, Bhukhiya M and González M C 2012 Safe Driving Using Mobile Phones *IEEE Transactions on Intelligent Transportation Systems* **13** 1462–8