An Interactive Robotic Design for Object Detection and Follow up Action

Rupashri Barik
Asst. Prof., Dept. of IT,
JIS College of Engineering,
Kalyani, Nadia, W.B., India.

Arkya Pratim Banerjee
Student, Dept. of IT,
JIS College of Engineering,
Kalyani, Nadia, W.B., India.

Mrinmoy Mitra
Student, Dept. of IT,
JIS College of Engineering,
Kalyani, Nadia, W.B., India.

ABSTRACT
The discussion of this paper is to develop an robotic model for hunting, tracking [7] and picking or collecting specific target-object with its autonomous capability. Autonomous Robot has been in use in multiple domains for example in surveillance, in battlefield [2], in remote locations where it is not impossible for human beings to reach. But most of the autonomous robot that exists today tends to be tele-operational [22]. This project aims to achieve the very fact to make the robot autonomous. The robot will be able to operate in an unsupervised manner and reach to its target-location. And here the addition of mechanical hand gives it some extra benefit of picking and collecting objects. The project aims in the application of Machine Learning [10] and fuzzy logic to attain this functionality. This project is intended to exploit the feature of neural network and make it capable of learning in an unsupervised environment and carry out its task autonomously.

Keywords
robotics, mechatronics, haar wavelet, haar-like features, haartraining, OpenCv.

1. INTRODUCTION
This project aims at developing a prototype of an Autonomous Robot [1]. The first part of the project deals with the physical constructions of the Robot, the design of it and the components layout. During the first phase of the construction it is required to look at the various off the shelf components available in the market that can be effectively used for a compact design. The Robot thus built should be able to support two different kinds of mode.

- Tele-operator mode
- Autonomous mode

1.1Tele-operator mode [3]
In tele-operator mode the robot can be controlled by human using radio frequency away from the robot's location, vision control [8] and arm control system will be the key term of this operation.

1.2 Autonomous mode
In autonomous mode [4] the Robot would be able to detect the target and then track the target location. After reaching, it would try to pick the target and then comes back to its origin-location.

1.3 Framework for further development
The second aim of this project is to develop the Robot in a manner such that following iterations will not require developing the Robot from scratch. The program running on the Robot can be further enhanced and the software developed for the Robot can be further modified to add more features to it. One can also enhance the physical components of the Robot if required to make it an intelligent Robot.

2. LITERATURE SURVEY
Development of an autonomous robot model is a multi-disciplinary approach. This model involves the application of concepts and principles from following domain of work.

- The Robotic/Mechatronics [23]Framework
- Digital Image processing
- Artificial Intelligence & Machine Learning[10]

Under most of the cases the term Robotics and Mechatronics have been used interchangeably.

Robotics is the branch of technology that deals with the design, construction, operation, structural disposition, manufacture and application of robots and computer systems for their control, sensory feedback [6], and information processing.

Mechatronics is a multidisciplinary field of engineering, that is to say it rejects splitting engineering into separate disciplines. Originally, mechatronics just included the combination between mechanics and electronics mechatronics.

Digital Image Processing [12] is the use of computer algorithms to perform image processing on digital images. It allows a much wider range of algorithms to be applied to the input data and can avoid problems such as the build-up of noise and signal distortion during processing. Since images are defined over two dimensions (perhaps more) digital image processing may be modeled in the form of multidimensional systems.

In particular, digital image processing is the only practical technology for:

- Classification
- Feature extraction
- Pattern recognition[14]
- Projection
- Multi-scale signal analysis

Some techniques which are used in digital image processing include:

- Pixelization
- Linear filtering
- Principal components analysis
- Independent component analysis
- Partial differential equations
- Neural networks
- Wavelets

Digital image processing has wide applications in intelligent transportation systems, such as automatic number plate recognition and traffic sign recognition.

3. OBJECT DETECTION MODEL

The object detection technology used in this project is based on the work of Paul Viola and Michael Jones namely the "Rapid Object Detection using a Boosted Cascade of simple classifiers "[24]. This technology has been implemented in OpenCv [11] namely under Haa r[16] Training.

The Rapid Object Detection Framework made use of two concepts, one was the representation of the image as an integral image for detecting Haar like features and second was constructing a classifier by selecting a small number of features using Adaboost [18] learning algorithm.

In this section we are going to look at two different things, the first is to show how to make use of HaarTraining in OpenCv for object detection and then we will have an insight into how the HaarTraining works.

3.1 Haar Training with OpenCv

In machine learning [15] one thing that is made most use of is the training set. The training set consist of the samples in the form (x,y) where y=f(x). For example let consider that are a set of balls of different sizes , may be some of the balls are of diameter 2cm , some of 5cm and some of 10cm and we want to classify these balls as small , medium and large.

x (in cm)	y (scale)
2	small
5	medium
5	large

So given a size of a ball in diameters, classify them in one of the groups above y=f (size of the ball).

These training set serves as a vital role in Haartraining. For detecting an object in the given frame it is required to make use of a training set consisting of the positive samples and the negative samples. The best way to create a positive sample is to record a video of the object of interest, for the negative sample record another video in the similar environment but without the object of interest. The videos should be of approximately 1 min. Make sure that the video was shot in different illumination environment at different angles. Once the video has been recorded, it is required to feed that video to objectRec.exe that was developed for the sake of this project. The objectRec.exe will process frame by frame of the current video in which the region of interest encapsulating the object of interest is marked. The objectRec.exe will automatically move the positive frames to the positive folder and the negatives frame to the negatives folder. Figure 1 shows the object of interest, the object that is to be detected and figure 2 shows the region of interest encapsulating object of interest in the current frame of the positive video sample. Please refer to objectRec.cpp to know the details of using it.

Fig 1: Object of Interest

Fig 2: Region of interest encapsulating object of interest

The objectRec.exe will approximately create 1500 samples from the positive and negative frame.

The figure 3 below shows the positives and negatives folder consisting of frames from the positive sample video and the negative sample video as marked and moved by objectRec.exe

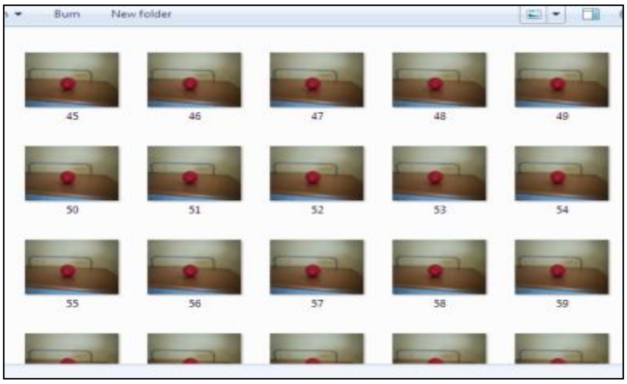


Fig 3: The positives and the negatives folder consisting of the frames created by objectRec.exe

Once the above steps have completed it is required to pack the marked images in a vector file. To do so, make use of opencv_createsamples.exe that comes inbuilt into openCv. Issue the following command opencv-createsamples -info -positives.txt -vec positives.vec -w 24 -h 24

Figure 4 below shows the opecv_createsamples creating the vector file. In the above statement -w and -h stands for height and width of the sample. For each object of interest mark The vector file thus created will be responsible for use in haartraining which will be our next step.

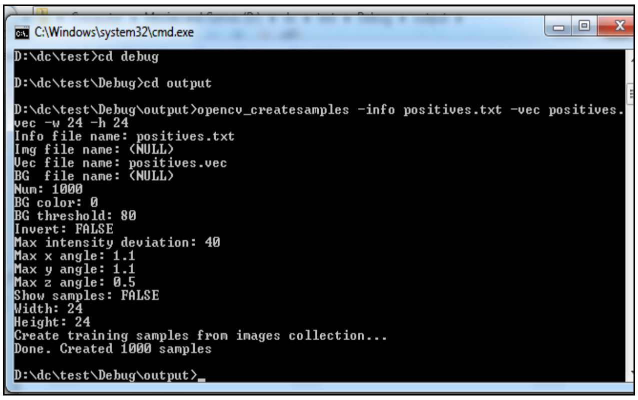


Fig 4: opecv_createsamples creating the vector file

Now that the vector file has been created it is required to train the system with the positive and the negative samples so that it can detect our object of interest in the moving frame. To train the system use the command opencv_harrtraining –data trainout –vec positives.vec –bg negetives.txt –npos 100 –nneg 1000 –nstages 20 Where the -data option specifies where to place the .xml file which is the trained classifier, -vec is our vector file from the previous create samples step, -bg is a txt file containing a listing of the negative images, -npos is the number of positive images, -nneg is the number of negative images, and nstages is the number of classifier stages to train.

This is the most time consuming step in the entire process and will take approximately 48hrs to complete. The number of stages can be increased but should be at least 14.

The figure 5 below shows the opencv _haartraining creating the classifiers and the output would be trainout.xml file that would be used for detecting.

Once this step completes outputfile.xml would be generated. This xml file a classifier that will help in determining if a given frame does consist the object of interest or not and if so where is it located (the x and y co-ordinates) in the given frame. Please refer to objectDetect.cpp to know about the object detection in a given frame using the xml file that we created in the last step.

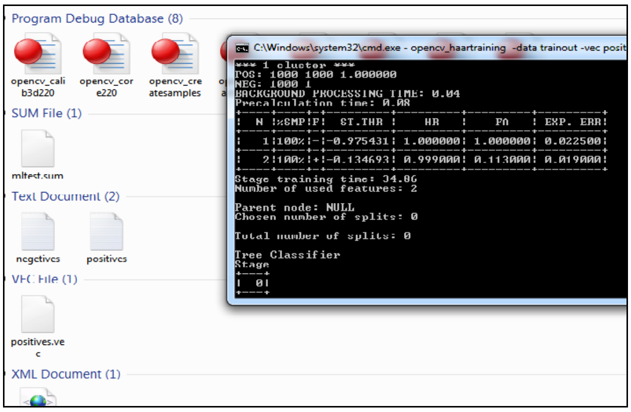


Fig 5: opencv_haartraining

3.2 Insight the Haar Trainning

Haar wavelet [16] is a sequence of rescaled function commonly "Square Shaped" and together forms a wavelet family. It is as like Fourier analysis and it allows a target function over an interval to be represented in terms of orthonormal function basis.The Haar wavelet's mother wavelet function ψ (t) can be described as:

$$\Psi(t) = \begin{cases} 1 & ,0 \le t < \frac{1}{2} \\ -1 & ,\frac{1}{2} \le t < 1 \\ 0 & ,otherwise \end{cases}$$

Its scaling function

$$\varnothing(t) = \begin{cases} 1 & ,0 \le t < 1 \\ 0 & ,otherwise \end{cases}$$

The 2X2 Haar matrix association with Haar wavelet is:

$$H_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

We can transform any sequence of even length into a sequence of two component vectors with the discrete wavelet transform. Right multiplication of each vector with a matrix H_2 can produces result of one stage of the Haar wavelet transform. .

Haar like features [16] are used in object recognition. In the detection phase of Viola-Jones object detection framework, a window of target size is moved over the input image and Haar like feature is calculated for each sub section of image.

After that difference is compared to a learned threshold for separating target object Haar like features are therefore organized in classifier cascade to form a strong learner of classifier .Viola-Jones used sum area table which is called as integral image.

Integral image is defined as two dimensional look up tables in the form of a matrix with the same size of original image (in relation to the element's position) computing sum of rectangular areas in the image in spite of its position on scale using :

Sum = pt_4 - pt_3 - pt_2 + pt_1 [pt_n belongs to integral image]

After getting the Haar like features we use adaboost [17] learning algorithm to produce strong classifier as output.

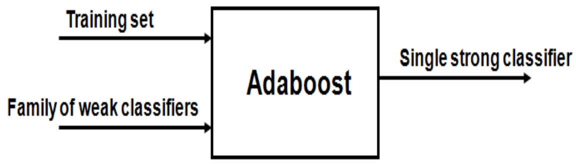


Fig 6: Block diagram of Adaboost

Adaboost is an abbreviation of the term adaptive boosting [17] this machine learning algorithm is formulated by Yoav Freund and Robert Schapire. It can be used with other algorithms to improve performance.

Adaboost generates as well as call a new weak classifier in each of a series of rounds t= 1, .., T and for each call a distribution of weight p_t is updated. For each round the weight of each incorrectly classified examples are increased and the weights of each correctly classified example are decreased.

The algorithm for the binary classification task [20] defines

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t\, h_t(x)\right)$$

Here, H(x) is final output classifier.

The equation to update the distribution D_t is constructed

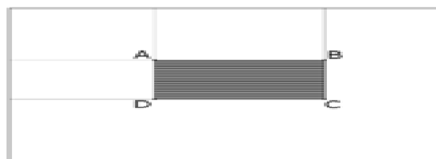
$$-\alpha_t y_i h_t(x_i) \begin{cases} < 0, y(i) = h_t(x_i) \\ > 0, y(i) \neq h_t(x_i) \end{cases}$$

The value of any point (x,y) in the summed area table is the sum of all the pixels above and to the left at (x,y) , inclusive

$$I(x,y) = \sum_{\substack{x' \le x \\ y' \le y}} i(x',y')$$

The summed area table can be computed efficiently in a single pass over the image using the value in the summed area table

$$I(x,y) = i(x,y) + I(x-1,y) + I(x,y-1) - I(x-1,y-1)$$



After the computation of sum area table, evaluating any rectangle can be done in fixed time with just four array reference

$$\sum_{\substack{A(x) < x' \le C(x) \\ A(y) < y' \le C(y)}} i(x',y') = I(A) + I(C) - I(B) - I(D)$$

4. ALGORITHM

The second aim of this project was to pick up the detected object by the Robotic arm [18]. The need for such an algorithm lies in the necessity that if this technology is implemented in to real use moving vehicle for transportation, it should be able to pick up the object of interest which is detected previously. Now however this concept needs to extended and the algorithm must be modified to make it suitable for real life applications, none the less this algorithm serves as the foundation for all of it.

```
Loop till we receive frame from the device
Convert the frame to gray frame
Process the frame for detecting blobs
For each blob in frame
Loop:
//. we will have only one blob for the object under
consideration
   Discard the blobs less than a particular size
   If the blob is greater than the minimum size
     Check the shape of the blob // for multiple object
                                        detection
        If the blob is quadrilateral // the target object
          Save the coordinates of the Blob
          Draw a quadrilateral corresponding to the
                blob on the frame
          Stop the camera of the robot
          Orient the robot based on the blobs co-ordinates
                motor functioning
            Start the camera
Continue:
```

The algorithm shows how to pick the object detected. Please note that here blob refers to the object that needs to be detected and encapsulated in a rectangle on the frame after being detected.

```
Frame f;  // Frame represents an image captured from the
video device
while(f=getFrame())
{
    f  = convertToGray(f); //convert to a gray  frame
             blobCounter bc = new blobCounter(); //let us
                    say that we have blobcounter class
             blobs  = bc.processFrame(f); //process each
                                      frame for blobs
             for each blob in blobs  // there can be multiple
                                    blobs in a frame
               {
                if(size(blob) > minimumsize) //discard all
               the blobs that are less than a particular size
               if(checkShape(blob)==qudrilateral) // only
                        quadrilaterals are considered
                   {
                     coordinates  = getCoordinates(blob);
                              // get (x,y) of the qudrilateral
                     drawRectangle(frame,coordinates);
                              // draw rectangle on the frame
                       // Now we have to pick  the object
                     UGV.stop_camera() // stop the camera
                               until the UGV orients itself
                    if(size(blob)<maximum_allowed_Size)
                      UGV.moveForward(); // UGV is at
                             safe  distance from obstacle
                     if(size(blob)>maximum_allowed_Size
                               &&  left_oriented())
                      UGV.moveLeft(); //blob occupies
                     most of  3rd and 4th partial quadrant
                     if(size(blob)>maximum_allowed_size
                                  &&  right_oriented())
                       UGV.moveRight();//blob occupies
                        most of 4th and 3rd partial quadrant
                    send(values)//microcontroller get excited
                           }
             UGV.Start_camera(); // start the camera back ;
             }
}
```

Fig 7: Algorithm for picking up, here blob refers to the rectangle around the detected object using cascade

The algorithm given in Figure 8 below explains the same thing but in a more lucid way.

Fig 8: The same algorithm in an easy way.

4.1 Explanation to the algorithm*:*

The first step in the entire process is to grab the frame from the camera mounted on the Robot. A forge.Net supplies libraries that can be easily used to stream the data from the camera into the calling program. One might also use the cvCapture () of the opencv library to take frames from the camera.

The next step is to convert the colored frame that we received to a gray frame. Converting the frame to a gray frame reduces the computational overhead. The cascade file developed earlier in the object detection step needs to be loaded before calling for the frame from a camera. For each frame thus grabbed and converted detect the objects in each frame using the cascade file. If an object is detected it is highlighted using a rectangle encapsulating the region of interest in the given frame. Alongside the co-ordinates are also measured.

The third step is to discard all rectangles less than a given size. A detected object too far is not a concern for the Robot. An object far way will have a small rectangle drawn on the screen; these small rectangles will be discarded. An object too close will have a larger rectangle on the screen. If the rectangle is of maximum size that indicates the Robot is too close to the object and so that it can pick it up. So it requires that the detected rectangle size is checked between the maximum and minimum.

The fourth step is to detect the orientation of the object. That is to say which quadrant the object occupies the most. For example if the (x, y) and (x1,y1) are two end points on the length of the rectangle and x and x1 are greater than zero then the object is right oriented and the Robot must move towards the left until and unless the object goes out of scope. That means for each object determined on the screen,

For each frame we, find the orientation of the object and move the Robot accordingly until and unless the objet goes out of scope.

5. RESULTS & DISCUSSIONS

This with the various test results that were achieved during the process of development of this project are discussed here.

The figure 9 shows the vector file and the cascade file built during the process of haartraining and the contents of the cascade file as an xml document.

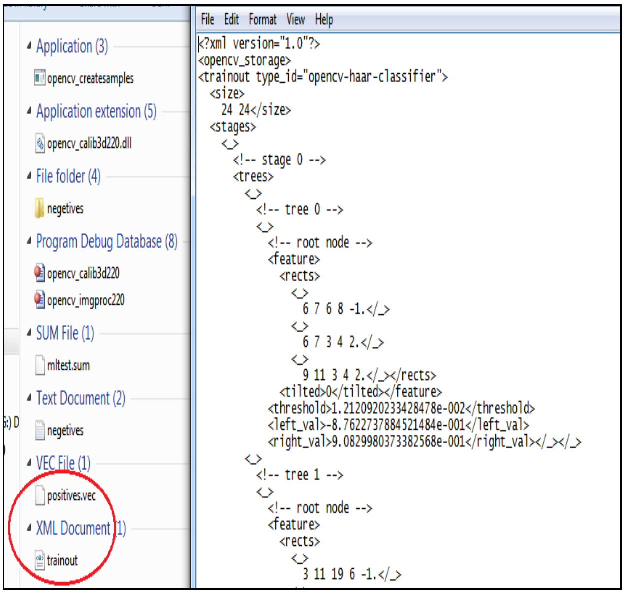


Fig 9: The vector file and cascade file in red, the contents of cascade file as an xml document

The figure 10 shows the positives and negatives file created by the objecRec.exe during processing of the positives and the negatives sample.

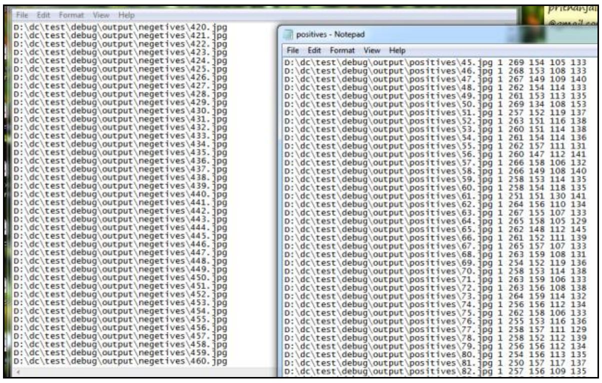


Fig 10: The positives and negatives file created by the objecRec.exe

The figure 11 shows the layout structure of the dependencies in case the project developed needs to be recompiled.

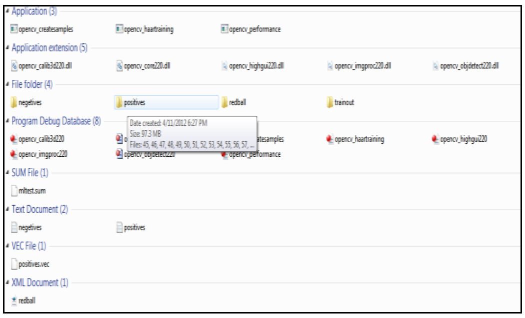


Fig 11: The layout structure for haartraining

Figure 12(a) & (b) are the images of the Autonomous Robot developed are below

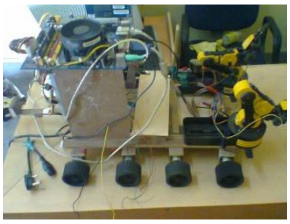


Fig 12(a): Side view on Robo Scorpion

Fig 12(b): Top view on Robo Scorpion

The mechanism of improving the performance of the entire autonomous robotic model is discussed here. The first section will deal with reducing the time of HaarTraining and the second section will deal with working with multiple cascades.

5.1.1 *Reducing HaarTraining time*

The haartraining.cpp that comes as a part of the opencv package has a lot of code in OpenMP.Open Multi-Processing (OpenMP) is an application programming interface (API) that supports multi-platform shared memory multiprocessing programming in C, C++, and Fortran, on most processor architectures and operating systems,including Linux, Unix, AIX, Solaris, OSX,and Micro soft Windows platforms.

Once Microsoft visual c++ compiles opencv with OpenMP it will create a HarrTraining.exe file. This file is capable of parallelizing the instructions. Haartraining with this newly compiled program reduces the time by approximately 21%.

The screenshots were taken in two different systems for certain library dependency.

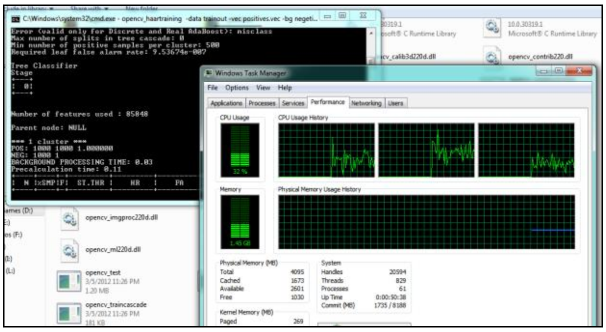


Fig 13: The default haartraining.exe utilizing 32% of the total capacity

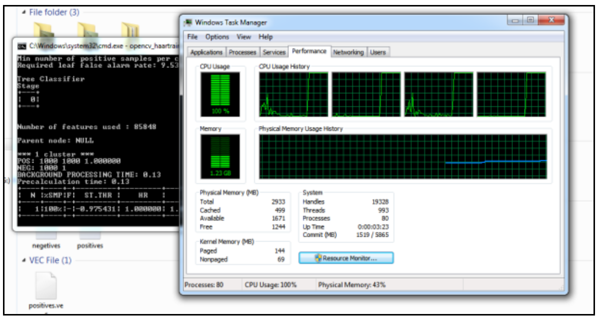


Fig 14: The recompiled HaarTraining.exe

From the two figures 13 and 14, we see that in the first case the program was only utilizing 32% of the total processing capacity, in the second case it is utilizing 100% of the total processing capacity. This overall reduces the training time. In first case the time taken was approximately 48hrs, in the second case it was reduced to around 18Hrs.

6. CONCLUSION

The domain of developing an Autonomous Robot is challenging as well as an interesting field. A field that is mixed with electronics and mechanics is enhanced with the help of image processing and machine learning. Machine learning can be applied in different ways as and when required. For example the bypass of object could also have been developed using Linear Regression of multiple variables using training and test mechanism however that concept is rather a big project in itself and out of scope of this document.

The project was developed using a general purpose web camera, however it might be considered that using night vision cameras and thermal sensors are feasible enough to gain more information about the surrounding of the robotic model. Transducers could have been also used to gain information about the distance between the robotic model and the object.

This is however a prototype of what might be achieved in real scale. This project in one way created a simulation of a bigger picture of Autonomous Robot, its capabilities and how this can be achieved to be used extensively in real life applications. The object detection algorithm can itself be separated from this project and it might find its use in a security monitoring system or in analysis of CCTV's. The object detection mechanism was one of the mile stones in this project and it was capable of executing in real time.

One might also consider attaching different kinds of payload to this system. For example like shooting at a particular trajectory from a given remote distance.

None the less this project was able to deliver the requirements that were expected of it. This project not only served at developing an Autonomous robot model but also creating a framework that will enhance the robotic model in each iteration. It served as prototype, or a generic device that finds its use in various kinds of fields and with the framework used, a scratch development won't be necessary.

7. ACKNOWLEDGMENT

Heartily we want to thank Prof. Somsubhra Gupta for his knowledge sharing and whole hearted support to complete this job.

8. REFERENCES

[1] Junmin Wang; Steiber, J.; Surampudi, "Autonomous Ground Vehicle Control System for High Speed and Safe Oepration ". American Control Conference, pp 218- 223, 2008.

[2] Comito, A.; Woody, M.W.; Perry, J.L.; Ziegler, W.F., Jr. "Advanced Eletronics Applied To Army Ground Vehicle". Vehicular Technology Conference, pp 26-30, 1981.

[3] Change-Hoi Kim; Kyung-Min Jeung; Yong-Soo Choi; Sung-Uk Lee; Chang-Hwoi Kim; "Design Of The Unmanned Ground Vehicle", ICCAS, pp 1094-1097, 2011.

[4] Balakirsky, S.; Lacaze, A.; "World Modelling and Behaviour Generation for Autonomous Ground Vehicle", ICRA 2000, Vol 2 , pp 1201 -1206, 2000.

[5] Yunji Wang; Chen, P.; Yufang Jin; "Trajectory Planning for Unmanned Ground Vehicle", SMC 2009, pp 4341-4346, 2009.

[6] Changhui Yang; Hongo, H.; Tanimoto, S "A New Approach for In vehicle Camera Obstacle Detection by Ground Movement Compensation", ITSC 2008, pp 151-156, 2008.

[7] Liguo Weng; Song, D.Y. "Path Planning and Path Tracking Control of Unmanned Ground Vehicle", SSST 05, pp 262-266, 2005.

[8] Bristeau, P.-J.; Petit, N.; "Navigation System for Ground Vehicles using Temporarily Interconnected Observer", ACC 2011, pp 1260-1267, 2011.

[9] Stromy Attaway, "Matlab: A Practical Introduction to Programming and Problem Solving", Butterworth-Heinman, 2nd Edition, 2012.

[10] Ethem Alpaydm," Introduction to machine Learning", MIT-PRESS, 3rd Edition, 2006.

[11] Gary Bradski and Adrian Kaehler, "Learning OpenCV", O'Reilly, 2nd Edition, 2007.

[12] Mark J Burge, Wilhelm Burger, "Principles of Digital Image Processing Core Algorithms", Springer, 2nd Edition, 2007.

[13] Beba. P Mandal," Pattern Recognition and Machine Intelligence", Springer, 4th International Conference, Jun 27-July1 2011.

[14] Nils J. Nilsson, "Introduction to Machine Learning", Stanford University Press, 1st Edition, 2010.

[15] Radomir S. Stankovic, Bogdan J. Falkowski "The Haar wavelet transform: its status and achievements", 27 November 2000.

[16] Sungji Han, Youngjoon Han and Hernsoo Hahn "Vehicle Detection Method using Haar-like Feature on Real Time System" 2009.

[17] Yoav Freund Robert E. Schapire "A Short Introduction to Boosting" Journal of Japanese Society for Artificial Intelligenc"e, 14(5):771-780, September, 1999. (In Japanese, translation by Naoki Abe).

[18] Alana Lafferty, Professor Rodney Brooks and Dr. Una May O'Reilly "A Robust Inexpensive Multi-Purpose Robotic Arm", May 20, 2010.

[19] Nghia Tran; Giraud-Carrier, C.; Seppi, K.; Warnick.S, "Cooperation-based Clustering for Profit-maximizing Organizational Design", IJCNN'06, 2006.

[20] Robert Laganiere, "Opencv Image Processing Han dbook",PACKT, 1st Edition , 2011

[21] Nils J. Nilsson, "Introduction to Machine Learning",Stanford University Press, 1st Edition,2010

[22] Sebastian Blumenthal1, Dirk Holz1, Thorsten Linder1, Peter Molitor2, Hartmut Surmann2 and with Surveillance Viatcheslav Tretyakov1 "Teleoperated Visual Inspection Unmanned Ground and Aerial Vehicles".

[23] T. Mori, "Mechatronics," Yasakawa Internal Trademark Application Memo, 21.131.01, July 12, 1969.

[24] Viola, P.; Jones, M. "Rapid Object Detection Using a Boosted Cascade of Simple Features" May 2004.

[25] Radomir S. Stankovic, Bogdan J. Falkowski "The Haar wavelet transform: its status and achievements", 27 November, 2000.

[26] Sungji Han, Youngjoon Han and Hernsoo Hahn "Vehicle Detection Method using Haar-like Feature on Real Time System", 2000.

[27] Yoav Freund Robert E. Schapire "A Short Introduction to Boosting" Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999.(In Japanese, translation by Naoki Abe).

[28] Alana Lafferty, Professor Rodney Brooks and Dr. Una May O'Reilly "A Robust Inexpensive Multi-Purpose Robotic Arm" May 20, 2000.