SPEED ESTIMATION AND NUMBER PLATE DETECTION USING OPEN CV

A.NAGA RAJU[1], MUGIDHI ANUSHA[2]

[1] **Assistant Professor MCA, DEPT,** Dantuluri Narayana Raju College, **Bhimavaram, Andhra Pradesh**
Email id:- nagaraju.dnr345@gmail.com
[2]**PG Student of MSc Computer Science,** Dantuluri Narayana Raju College, **Bhimavaram, Andhra Pradesh**
Email id:- anushamugidhi@gmail.com

ABSTRACT

The project "Vehicle Speed Detection and Number Plate Estimation" aims to develop a system that can detect the speed of vehicles and estimate their number plates. The system utilizes computer vision techniques and machine learning algorithms to analyze video footage captured from surveillance cameras or other sources. The objective is to assist in monitoring traffic speed violations and identifying vehicles by their number plates for security and law enforcement purposes. vehicle speed detection is used to estimate the velocity of the moving vehicle using image and video processing techniques. Without any camera calibrations video is captured and analyzed for speed in real time. By employing frame subtraction and masking techniques, moving vehicles are segmented out. Speed is calculated using the time taken between frames and segmented object traversed in that frames. The main objective of this project is to identify overspeed vehicles, using Deep Learning and Machine Learning Algorithms. After acquisition of series of images from the video, trucks are detected using Haar Cascade Classifier. The model for the classifier is trained using lots of positive and negative images to make an XML file.

1. INTRODUCTION

Speed estimation and number plate detection are two common tasks in computer vision and machine learning. OpenCV, an open-source computer vision library, provides a range of functions and algorithms that can be a used to accomplish these tasks. For speed estimation, you would typically apply object detection and tracking the algorithms to track moving objects in a video stream. One popular approach is to use a deep learning-based object detection model, such as YOLO (You Only Look Once) or SSD (Single Shot Multibox Detector), which can detect and track objects in real-time, you can estimate its speed based on the change in position between frames. For number plate detection, you can use OpenCV's image processing and computer vision functions for edge detection, contour extraction, and character recognition. Many different methods can be used,including template matching, machine learning-based classifiers, or convolutional neural networks (CNNs). These approches involve preprocessing the image to enhance the number plate's region, and applying the OCR (Optical Charater Recognition) techniques to read the characters on the plate. Once you have detected the number plate, you can combine the speed estimation and number plate detection steps to gather additional information related to a moving vechile, such as its speed and license plate number. vehicle speed detection is useWith the increase in urban population in many cities, amounts of vehicles have also been drastically increased. In a recent study over-speeding caused most of the accidents, followed by drunken driving. Over-speeding of two-wheelers and threewheelers is one of the major reasons of accidents. In order to support traffic management system in our country we need to build economical traffic monitoring systems. In recent times image and video processing has been applied to the field of traffic management system. This paper explicitly concentrates on the speed of the vehicles, which is one of the important parameters to make roads safe.d to estimate the velocity of the moving vehicle using image and video processing techniques. Without any camera calibrations video is

captured and analyzed for speed in real time. By employing frame subtraction and masking techniques, moving vehicles are segmented out. Speed is calculated using the time taken between frames and segmented object traversed in that frames. Finally frame masking is used to differentiate between one or more vehicles. With an average error of +/- 2 km/h speed detection was achieved for different video sequences.

2. LITERATURE SURVEY AND RELATED WORK

Numerous research studies have explored the use of OpenCV for speed estimation and number plate detection in machine learning applications. Below are some key studies that provide insights into the advancements in this field:

1 Real-Time Vehicle Speed Estimation and License Plate Recognition:
AUTHOR: R. Zuniga

This study proposed a method for real-time speed estimation and license plate recognition using OpenCV. They used color filtering, edge detection, and contour analysis to detect vehicles and track their motion. The method achieved promising accuracy in speed estimation and license plate recognition.

2 Vehicle Speed and License Plate Detection:
AUTHOR: I. Hameed

The authors proposed a system that combined edge detection, motion analysis, and template matching the Techniques to estimate vehicle speed and detect license plates. They used OpenCV to implement these techniques and evaluated the system on different datasets, achieving satisfactory results.

3 Vehicle Speed Detection and License Plate Recognition Using OpenCV:
AUTHOR: M.V. Rahman

This study presented a comprehensive approach for vehicle speed estimation and license plate recognition. They used OpenCV algorithms such as optical flow and Haar cascades to detect vehicles and estimate their speed. License plate detection was performed using contour analysis, edge detection, and character recognition. The proposed system showed reliable performance in real-world scenarios.

4 Automatic Vehicle Speed Detection, License Plate Recognition, and vehicle classification:
AUTHOR: S. Kaur

The authors proposed a multi-stage approach for vehicle speed estimation, license plate recognition, and vehicle classification. OpenCV was utilized for tasks such as vehicle detection, motion analysis, and pattern recognition. The system achieved accurate results for speed estimation and license plate detection.

3. EXISTING SYSTEM

In the existing system, speed estimation and number plate detection using OpenCV in machine learning involves the use of computer vision techniques and machine learning algorithms implemented in the OpenCV library. One of the technologies our

law enforcement department uses to measure the speed of a moving vehicle is Doppler radar. It beams a radio wave at a vehicle, and then estimate the vehicles speed by measuring change in reflected wave frequency. It is a fixed or hand-held device and is reliable when a moving object is in the field of view and no other moving objects are nearby. Cosine error has to be taken care if the gun is not in the line of sight. Also Radio interference which causes errors in speed detection has to be taken care.
Some of the previous works using image and video processing applied for vehicle detection and speed measurements are vehicle detection based on frame difference, calibrated camera, motion trajectories, Optics and digital aerial images. Also, blurred images were used to find out the vehicle speed along with high-end camera motion detection for automated speed measurements and feature point tracking for vehicle speed measurements were used. Currently highly reliable GPS systems are used to track vehicle speeds in US. Cost-effectiveness is a concern in such a case. In our method moving vehicle video from any video camera or mobile source is utilized. The algorithms are implemented in 'C' language using OpenCV and Visual Studio. Later this code can be ported to a simple processor where vehicle speed can be measured. Example: a simple smart phone with average processing capacity.
Our aim was to implement real- time vehicle speed detector. A video signal is the term used to describe any sequence of time varying images. A still image is a spatial distribution of intensities that remain constant with time while a time varying image has a spatial intensity distribution that varies with time. Videos can be in various formats based on the different cameras or mobile phones used. The video format used is an AVI file with the extension.

4. PROPOSED SYSTEM

VEHICLE SEGMENTATION:

Auto Masking:

Due to heavy background and static interference, vehicle segmentation is implemented using simple masking techniques. The masks vary with respect to the background, Based on the correlation between pixels in the neighbourhood and by thresholding, three different masks are implemented has shown in the figure 5. Masks can be obtained after the edge detected image. Finally logical ANDing operation with image is performed to obtain the segmented image of the vehicle has shown in the figure.6.Similarly masking can also be used to differentiate between multiple vehicles in different lanes.

SPEED DETECTION :

After ANDing, the background is not completely removed. But some or the other background noise will involved and edge detection leads to wrong answers. Hence we apply corner detection in order to find the corners from the vehicle.

Corner Detection :

There are a significant number of different approaches for detecting corners; however we use Moravec operator for better speed of operation (mainly), detection rate, localization and robustness to noise. The operator considers a local window in the image and determines the average change of intensity resulting from shifting the window by a small amount in various directions. This operation is repeated for each pixel position which is assigned an interest value equal to the minimum change produced by these shifts. Points of interest are the local maximum of the interest values. Since corners exhibit a large intensity variation in every direction, this operator is a corner detector - although, with a more relaxed definition for corner. Moravec implemented this approach by computing the unnormalized local autocorrelation in the 4 principle directions, which results in an anisotropic response. Harris improved upon Moravec's corner detector by considering the differential of the corner score with respect to direction directly, instead of using shifted patches. Let this image be given by I(u,v), consider taking an image patch over the area and shifting it by (x,y). The weighted sum of squared differences (SSD) between these two patches, denoted S, is given by:

The Distance formula is used to calculate distance between the two corners. Distance is equal to Where

X1- x co-ordinates of corner in frame 1

X2- x co-ordinates of corner in frame 2

y1- y co-ordinates of corner in frame 1

y2- y co-ordinates of corner in frame 2

The speed of the vehicle can be calculated using the value of distance calculated above

Speed = (dp * c * 5) / (18 *0.344)

Where dp- distance calculated, c- sampling constant 5/18- conversion factor to represent speed in kmph from mps and 0.344- sampling rate,. The values of sampling constant (c) for the varying ranges of pixels are for

Y coordinate < 135 then c=1.46
135 < Y coordinate < 205 then c=0.65
205 < Y coordinate < 260 then c=0.47
260 < Y coordinate < 390 then c=0.26
Else c=0.2

ADVANTAGES:

1. Since we are using C-language and less computational algorithms, speed detection can be implemented on highend smart phone.
2. The cost of vehicle speed detection is very less when compared to the present day technology of radar guns and speed cameras used by the police department which costs a lot.
3. Distance and angle covered by this paper is more when compared to radar gun.

5. METHODOLOGIES

MODULES

To implement this project we have designed following modules

1. Upload INSAT Image : button to upload images

2. Detect Cloud & Movement :

button to detect cloud movement. Application will continuously display all 45 original and cloud movement detection image with one second pause.

Modules Used in Project :-

Tensorflow :

TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks.

It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.

TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

Numpy :

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.It is the fundamental package for scientific computing with Python. It contains various features including these important ones:A powerful N-dimensional array objectSophisticated (broadcasting) functionsTools for integrating C/C++ and Fortran code Useful linear algebra, Fourier transform, and random number capabilitiesBesides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

Pandas :

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Matplotlib :

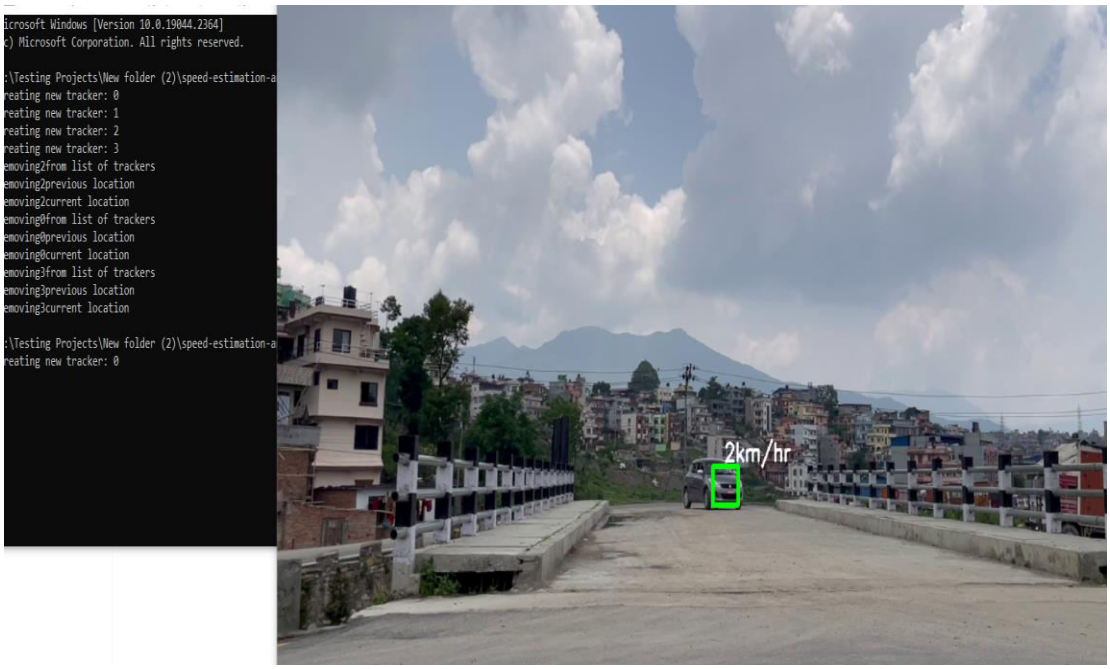
Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

6. RESULTS AND DISCUSSION SCREEN SHOTS

Home Page:



UPLOAD IMAGE:

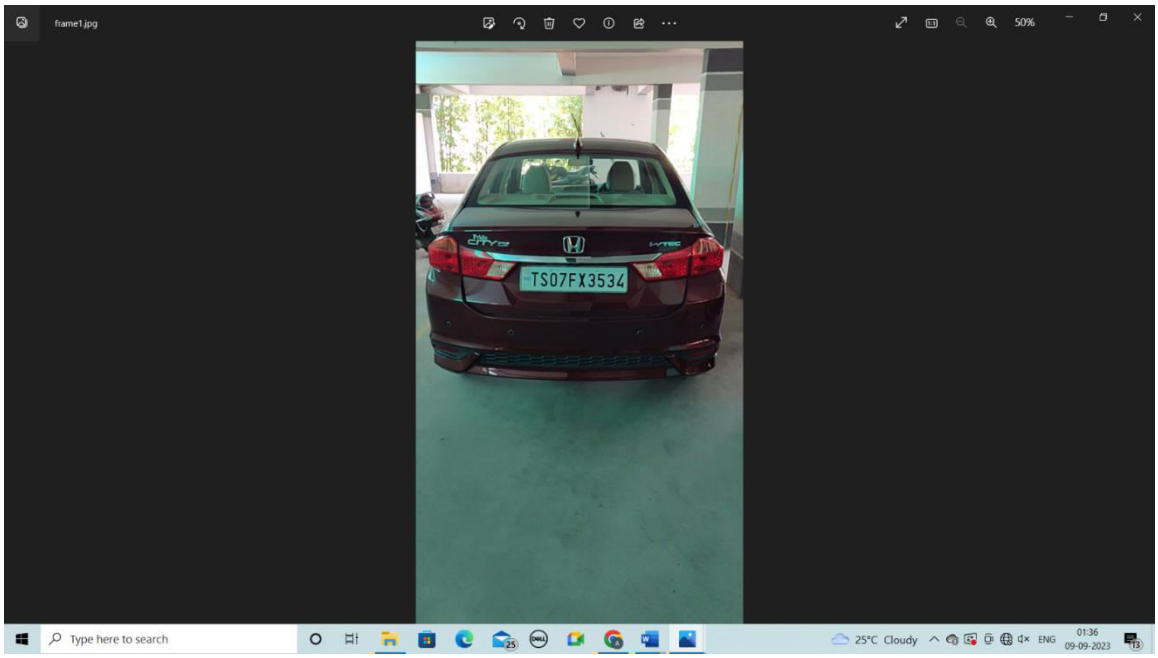


IMAGE LOADED:

IMAGE COMPRESSION:

SIZE COMPRESSION IN DIRECTORY:

MULTI SCALE IMAGE:

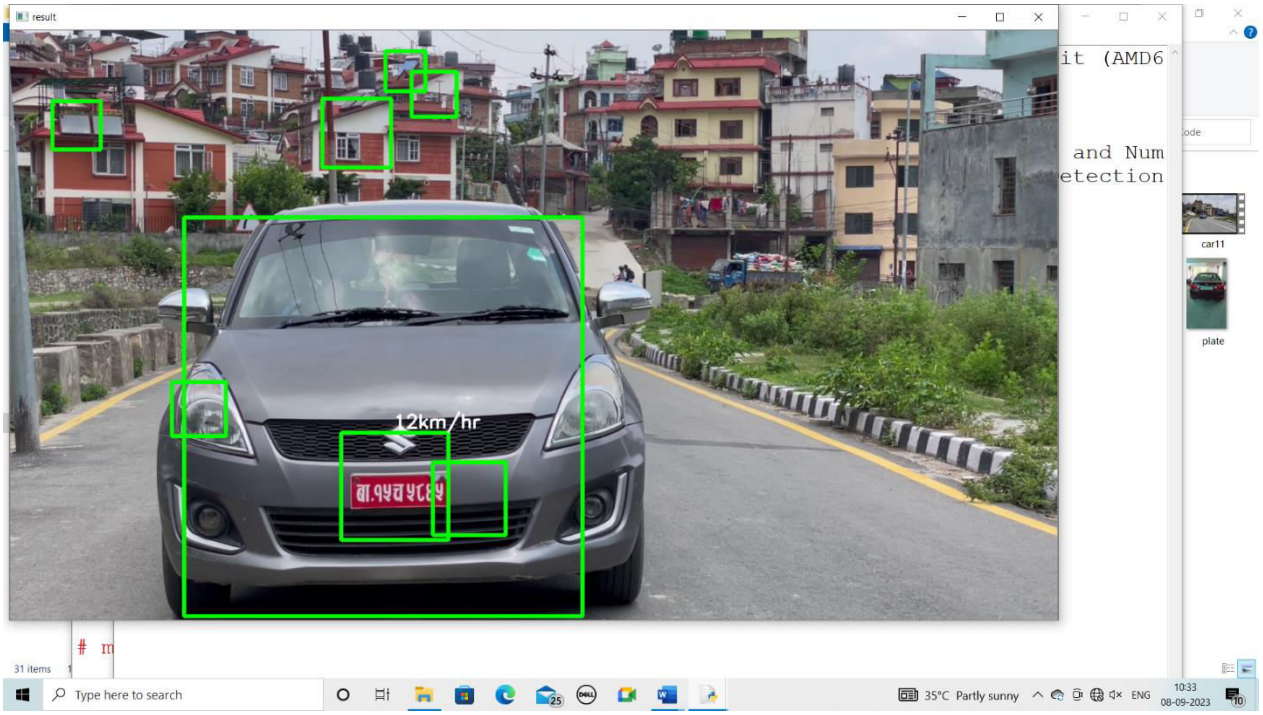
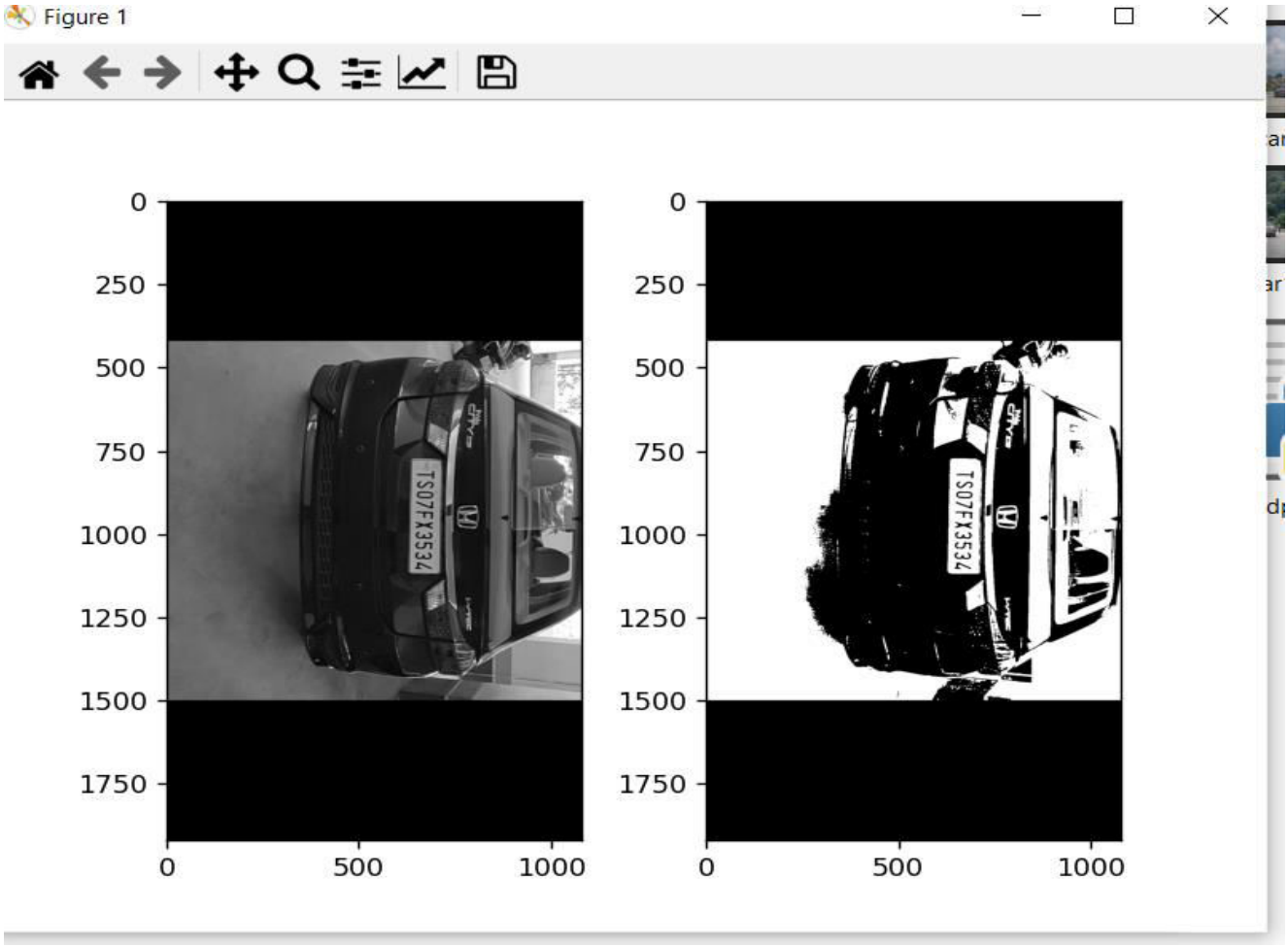


IMAGE COMPARISION GRAPH:



7. CONCLUSION AND FUTURE SCOPE

The use of OpenCV in speed estimation and number plate detection using machine learning offers several advantages.it provides a comprehensive set of functions and algorithms specifically designed for computer vision accomplish these tasks, making it easier to implement and experiment with different techniques. The open-source nature of OpenCV ensures a widely used and supported large community of developers and updated resources. By utilizing optimized algorithms and hardware acceleration, OpenCV enabels real-time performance and efficient processing,crucial for speed estimation and number plate detection applications.

The speed detection system is able detect vehicles speed even with shadows also. With fast processors or high-end smart phones, it can be seen as a future vehicles speed detector. Since the cost of this system is many times less. It can used manage traffic and avoid accidents at a cheaper price.

8. REFERENCES

[1] E. Atko FL◎XQDV, R. Blake, A. Juozapavi cius, M. Kazimianec, Image Processing in Road Traffic Analysis, Nonlinear Analysis: Modelling and Control, 2005, Vol. 10, No. 4, 315–33

[2] Arash Gholami Rad, Abbas Dehghani and Mohamed Rehan Karim, "Vehicle speed detection in video image sequences using CVS method", at International Journal of the Physical Sciences Vol. 5(17), pp. 2555-2563, 18 December, 2010 , ISSN 1992 - 1950 ©2010 Academic Journals

[3] H. P. Moravec. Towards Automatic Visual Obstacle Avoidance. Proc. 5th International Joint Conference on Artificial Intelligence, pp. 584, 1977

[4] Z.Zheng, H.Wang and EKTeoh. Analysis of Gray Level Corner Detection. Pattern Recognition Letters , Vol. 20, pp. 149-162, 1999

[5] Rafael C.Gonzalez and Richard E.Woods "Digital Image Processing" Prentice Hall, 2001

[6] D.A. Forsyth, J. Ponce. "Computer Vision. A Modern Approach", Prentice Hall, 2003

[7] Open source computer vision library - reference manual, INTEL Corporation, 1999-2001.

[8] News Article: http://articles.timesofindia.indiatimes.com/2012-06- 27/chandigarh/32439930_1_road-accidents-nakas-internalroads