Project Report

on

# Vehicle speed detection using AI

**ARTIFICIAL INTELLIGENCE FOR DATA SCIENCE**

By

K. MANIDEEP REDDY (2010030309)

I. KAUSTUBH SASTRY (2010030064)

K.V. MANOHAR KARTHIK (2010030197)

M.ABHIRAM (2010030457)

Under the supervision of

# Dr. Arpita Gupta,

# Assistant. Professor



Department of Computer Science and Engineering

K L University Hyderabad,

Aziz Nagar, Moinabad Road, Hyderabad – 500 075, Telangana, India.

March, 2022

**01**

**DECLARATION**

The Project Report entitled “**Vehicle speed detection using AI**” is a record of bonafide work of K. MANIDEEP REDDY (2010030309), I. KAUSTUBH SASTRY (2010030064), K.V. MANOHAR KARTHIK (2010030197), M. ABHIRAM (2010030457) submitted as a requirement for the completion of the course **ARITIFICIAL INTELLI GENCE FOR DATA SCIENCE** in the Department of Computer Science and Engineering to the K L University, Hyderabad. The results embodied in this report have not been copied from any other Departments/University/Institute.

Signature of the Students

K. MANIDEEP REDDY (2010030309)-

I. KAUSTUBH SASTRY (2010030064)-

M. ABHIRAM (2010030457)-

K.V. MANOHAR KARTHIK (2010030197)-

**02**

**CERTIFICATE**

This is to certify that the Project Report entitled “VEHICLE SPEED DETECTION” is a record of bonafide work of K. MANIDEEP REDDY (2010030309), I. KAUSTUBH SASTRY (2010030064), K.V. MANOHAR KARTHIK (2010030197), M. ABHIRAM (2010030457) submitted in partial fulfillment for the award of B. Tech in the Department of Computer Science and Engineering to the K L University, Hyderabad is a record of bonafide work carried out under our guidance and supervision.

The results embodied in this report have not been copied from any other departments/ University/Institute

## Signature of the Supervisor

Dr. Arpita Gupta

Assistant. Professor

## Signature of the HOD Signature of the External Examiner

**03**

**ACKNOWLEDGEMENT**

First and foremost, we thank the lord almighty for all his grace & mercy showered upon us, for  
   
completing this project successfully.

We take grateful opportunity to thank our beloved Founder and Chairman who has given constant encouragement during our course and motivated us to do this Social Internship. We are grateful to our Principal **Dr. L. Koteswara Rao** who has been constantly bearing the torch for all the curricular activities undertaken by us.

We pay our grateful acknowledgement & sincere thanks to our Head of the Department **Dr. Chiranjeevi Manike** for his exemplary guidance, monitoring and constant encouragement throughout the course. We thank **Dr. Arpita Gupta**, of our department who has supported throughout this course holding a position of supervisor.

We whole heartedly thank all the teaching and non-teaching staff of our department without whom we won’t have made this project a reality. We would like to extend our sincere thanks especially to our parent, our family members and friends who have supported us to make this project a grand success.

**04**

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TITLE** | **PAGE NO** |
| 1 | Abstract | 6 |
| 2 | Introduction | 7 |
| 3 | Literature Survey  &  System Requirements –  Hardware and Software | 8 - 9 |
| 4 | Flow Diagram | 10 |
| 5 | Proposed Algorithm | 11 - 13 |
| 6 | Implementation  🡪 Code  🡪 Results | 14 - 17 |
| 7 | Conclusion and Future Work | 18 - 19 |
| 8 | References | 20 |

**05**

**ABSTRACT**

Every year, traffic accidents due to human errors cause increasing amounts of deaths and injuries globally. To help reduce the number of fatalities, in the paper presented here, a new module for Driver Drowsiness detection (DDD) which deals with automatic driver drowsiness detection based on visual information and Artificial Intelligence is presented. The aim of this system is to locate, track, and analyze both the drivers face and eyes to compute a drowsiness index, where this real-time system works under varying light conditions (diurnal and nocturnal driving). Examples of different images of drivers taken in a real vehicle are shown to validate the algorithms used

Drowsy Driving is a deadly combination of driving and sleepiness. The number of road accidents due to Drowsy Driving is increasing at an alarming rate worldwide. Not having a proper sleep is the main reason behind drowsiness while driving. However, other reasons like sleep disorders, medication, alcohol consumption, or driving during night shifts can also cause drowsiness while driving.

**06**

**CHAPTER - 02**

**INTRODUCTION**

“1 in 25 adult drivers report that they have fallen asleep at the wheel in the past 30 days”

If you have driven before, you’ve been drowsy at the wheel at some point. It’s not something we like to admit but it’s an important problem with serious consequences that needs to be addressed. 1 in 4 vehicle accidents are caused by drowsy driving and 1 in 25 adult drivers report that they have fallen asleep at the wheel in the past 30 days. The scariest part is that drowsy driving isn’t just falling asleep while driving. Drowsy driving can be as small as a brief state of unconsciousness when the driver is not paying full attention to the road. Drowsy driving results in over 71,000 injuries, 1,500 deaths, and $12.5 billion in monetary losses per year. Due to the relevance of this problem, we believe it is important to develop a solution for drowsiness detection, especially in the early stages to prevent accidents.

Additionally, we believe that drowsiness can negatively impact people in working and classroom environments as well. Although sleep deprivation and college go hand in hand, drowsiness in the workplace especially while working with heavy machinery may result in serious injuries similar to those that occur while driving drowsily.

**07**

**CHAPTER - 03**

**3.1. LITERATURE SURVEY**

We as a Team, have studied various resources to learn about the different methods and to learn about the uses and effectiveness of the concepts i.e., machine learning concepts We used the available means to avail the resources, we mainly used Internet as the primary source.

In image processing morphological operations highly experimented in improving the appearance. To reduce the noise the MM is also applied it uses structuring element to probe the image and thereby useful information from the image can be obtained and noise can be reduced while preserving the features. This paper is on an experiment in which four morphological operations are working to reduce the noise from the gray scale image and thereby enhancing the quality of the images. In the literal, authors introduce the first step towards developing the Speed Detection Radar, where he explains a new approach in object detection technique, which is “adaptive background subtraction”. Rad A. G. et al. developed a system in which they used video and image processing toolbox which calculates the speed of vehicle. It resulted in average error of speed +7km/h and -7km/h. This system could operate on images with various resolutions and different video sequences. Shedbalkar K.developed a speed estimation technique which was based on extended kalman filter for permanent magnet synchronous. System is developed in MATLAB in SIMULINK model Blockset. Leite A.V. et al. determined a way for estimation of speed in induction motor with sensor less control. Extended kalman filter was used as speed detection technique. This algorithm used reduce order state space model. Kassen N. et al. proposed a vehicle speed estimation technique which was reliable and strong. This helps the user with driving guide and lets him not to join the traffic jam. This approach is based on RF.

From the literature survey we conducted, we learnt different machine learning techniques and datasets available which are related to our project, we have applied the best of those to the project to get the best and accurate results.

**08**

**3.2. SYSTEM REQUIREMENTS**

**SOFTWARE REQUIREMENTS: -**

* Operating system - Windows 10
* Tools – Pycharm

**HARDWARE REQUIREMENTS:** -

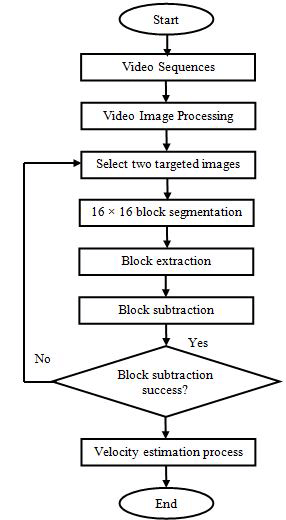
* RAM - 8.00 GB (7.87 GB usable)
* Processor - Intel(R) Core (TM) i5-10300H CPU @ 2.50GHz 2.50 GHz
* System-type - 64-bit operating system, x64-based processor
* Version - 20H2
* Edition - Windows 10 Home Single Language

**09**

**CHAPTER – 04**

**FLOWCHART**

The workflow procedure is as follows



**fig 4.1: showing flowchart of the problem**

**10**

**CHAPTER – 05**

**5.1 PROPOSED ALGORITHM**

## A This method is used for estimating vehicle speed which is coming towards camera by tracking the motion of vehicle through sequence of images. The proposed system mainly consists of steps as shown in fig.1 Firstly, the video is converted into frames. The Background Subtraction is used for moving vehicle detection. Averaging all frames, background without moving object is extracted. Background Subtraction output is applied for Thresholding and Morphological Operation. Connected Component Method isused to detect object and centroid of that object. Centroid is tracked over multiple frames. Velocity is calculated using distance travelled by vehicle and frame rate of video. This system is helpful for traffic parameter surveillance. Fig. 1 shows Block Diagram for proposed method

## Diagram Description automatically generated

## Pre-processing

## The video is recorded using mobile camera having pixels. In pre-processing the video has converted into the frames. The various parameters such as number of frames, frame rate, color format, frame size are extracted

## Moving Vehicle Detection

## Detecting Moving vehicle from video accurately is challenging task. To detect moving object there are various

## approaches such as temporal differencing method, optical flow algorithm, background subtraction algorithm. Temporal differencing method uses two adjacent frames only to get background image. This method has one disadvantage that it cannot detect slow changes accurately. Optical flow algorithm detects object independently using camera motion. Optical flow algorithm is computationally complex, and it is not suitable for real time application. In background subtraction absolute difference between background model and each instantaneous frame is taken to detect moving object. Background model is an image with no moving object.In this work, background subtraction algorithm is used to detect moving vehicle. The background subtraction algorithm

## mainly consist of three stages Background Extraction, Thresholding, Morphological Operations.

## Graphical user interface Description automatically generated

## Fig 5.1.1 Background Substraction

**11**

* **Vehicle Tracking**

It is based on feature tracking. The extracted features are tracked over sequential frames. To identify whether it is same object or other object matching algorithm is used. In object matching algorithm mahalanobis distance is calculated. Mahalanobis distance is used to find similarity and dissimilarity between two groups. It uses covariance of the two groups. When covariance matrix is same as identity matrix then mahalanobis distance is same as Euclidian distance. In object matching the mahalanob is distance between features of object in the previous frame and instantaneous frame is calculated. Some threshold value is set and it has compared with calculated distance. If the distance is less than the threshold value then the object in the previous frame and instantaneous frame is same. According to this, match id has given for each object. The result of this is as shown in fig.7. This match id is tracked over sequential frames.

A picture containing text, scene, way, road

Description automatically generated

* **Speed Determination**

The detected vehicle having match id is tracked over frames. The total number of frames in which same object is present has calculated.

Total Frames Covered= frame n – frame 0

A road with cars on it

Description automatically generated with low confidence

Where, frame 0 is the first frame when object is entered in roi and frame n is last frame when object passed away from roi. Also the real world distance is mapped on the image. The count of total number of frames is then multiplied with duration of one frame which is calculated from frame rate of video. From this the total time taken by vehicle to travel and Distance is fixed it is measured in real world and mapped into image.

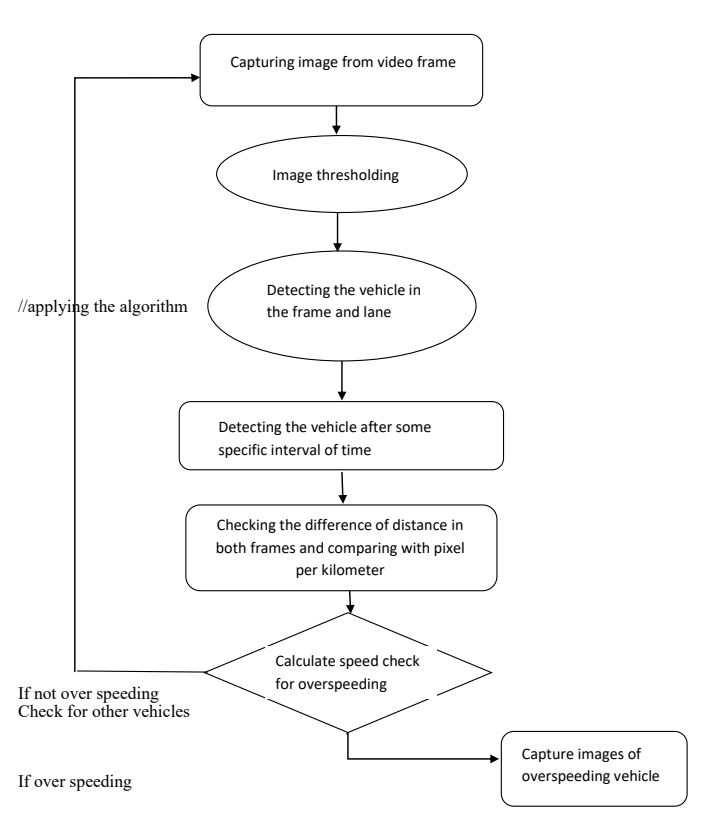
Speed=Distance/ (TF\*Frame rate)

From distance and travelled time of detected vehicle, speed of that vehicle is determined from above formulae.

**12**

**Algorithm:**

In order to initiate the program execution, it will import the following libraries like numpy, OpenCV, play sound, argparse, dlib, distance, timer. The entire algorithm for Vehicle speed detection is shown with the help of a flowchart shown

.  
**Fig 5.1.1 showing the algorithm.**

**13**

**CHAPTER – 06**

**6.1. IMPLEMENTATION**

import math

import time

import cv2

import dlib

carCascade = cv2.CascadeClassifier('myhaar.xml')

video = cv2.VideoCapture('cars.mp4')

WIDTH = 1280

HEIGHT = 720

def estimateSpeed(location1, location2):

d\_pixels = math.sqrt(math.pow(location2[0] - location1[0], 2) + math.pow(location2[1] - location1[1], 2))

# ppm = location2[2] / carWidht

ppm = 8.8

d\_meters = d\_pixels / ppm

#print("d\_pixels=" + str(d\_pixels), "d\_meters=" + str(d\_meters))

fps = 18

speed = d\_meters \* fps \* 3.6

return speed

def trackMultipleObjects():

rectangleColor = (0, 255, 0)

frameCounter = 0

currentCarID = 0

fps = 0

carTracker = {}

carNumbers = {}

carLocation1 = {}

carLocation2 = {}

speed = [None] \* 1000

# Write output to video file

out = cv2.VideoWriter('outpy.avi',cv2.VideoWriter\_fourcc('M','J','P','G'), 10, (WIDTH,HEIGHT))

while True:

start\_time = time.time()

rc, image = video.read()

if type(image) == type(None):

break

image = cv2.resize(image, (WIDTH, HEIGHT))

resultImage = image.copy()

frameCounter = frameCounter + 1

carIDtoDelete = []

for carID in carTracker.keys():

trackingQuality = carTracker[carID].update(image)

**14**

if trackingQuality < 7:

carIDtoDelete.append(carID)

for carID in carIDtoDelete:

print ('Removing carID ' + str(carID) + ' from list of trackers.')

print ('Removing carID ' + str(carID) + ' previous location.')

print ('Removing carID ' + str(carID) + ' current location.')

carTracker.pop(carID, None)

carLocation1.pop(carID, None)

carLocation2.pop(carID, None)

if not (frameCounter % 10):

gray = cv2.cvtColor(image, cv2.COLOR\_BGR2GRAY)

cars = carCascade.detectMultiScale(gray, 1.1, 13, 18, (24, 24))

for (\_x, \_y, \_w, \_h) in cars:

x = int(\_x)

y = int(\_y)

w = int(\_w)

h = int(\_h)

x\_bar = x + 0.5 \* w

y\_bar = y + 0.5 \* h

matchCarID = None

for carID in carTracker.keys():

trackedPosition = carTracker[carID].get\_position()

t\_x = int(trackedPosition.left())

t\_y = int(trackedPosition.top())

t\_w = int(trackedPosition.width())

t\_h = int(trackedPosition.height())

t\_x\_bar = t\_x + 0.5 \* t\_w

t\_y\_bar = t\_y + 0.5 \* t\_h

if ((t\_x <= x\_bar <= (t\_x + t\_w)) and (t\_y <= y\_bar <= (t\_y + t\_h)) and (x <= t\_x\_bar <= (x + w)) and (y <= t\_y\_bar <= (y + h))):

matchCarID = carID

if matchCarID is None:

print ('Creating new tracker ' + str(currentCarID))

tracker = dlib.correlation\_tracker()

tracker.start\_track(image, dlib.rectangle(x, y, x + w, y + h))

carTracker[currentCarID] = tracker

carLocation1[currentCarID] = [x, y, w, h]

currentCarID = currentCarID + 1

#cv2.line(resultImage,(0,480),(1280,480),(255,0,0),5)

for carID in carTracker.keys():

trackedPosition = carTracker[carID].get\_position()

t\_x = int(trackedPosition.left())

t\_y = int(trackedPosition.top())

t\_w = int(trackedPosition.width())

t\_h = int(trackedPosition.height())

cv2.rectangle(resultImage, (t\_x, t\_y), (t\_x + t\_w, t\_y + t\_h), rectangleColor, 4)

# speed estimation

carLocation2[carID] = [t\_x, t\_y, t\_w, t\_h]

end\_time = time.time()

**15**

if not (end\_time == start\_time):

fps = 1.0/(end\_time - start\_time)

#cv2.putText(resultImage, 'FPS: ' + str(int(fps)), (620, 30),cv2.FONT\_HERSHEY\_SIMPLEX, 0.75, (0, 0, 255), 2)

for i in carLocation1.keys():

if frameCounter % 1 == 0:

[x1, y1, w1, h1] = carLocation1[i]

[x2, y2, w2, h2] = carLocation2[i]

# print 'previous location: ' + str(carLocation1[i]) + ', current location: ' + str(carLocation2[i])

carLocation1[i] = [x2, y2, w2, h2]

# print 'new previous location: ' + str(carLocation1[i])

if [x1, y1, w1, h1] != [x2, y2, w2, h2]:

if (speed[i] == None or speed[i] == 0) and y1 >= 275 and y1 <= 285:

speed[i] = estimateSpeed([x1, y1, w1, h1], [x2, y2, w2, h2])

#if y1 > 275 and y1 < 285:

if speed[i] != None and y1 >= 180:

cv2.putText(resultImage, str(int(speed[i])) + " km/hr", (int(x1 + w1/2), int(y1-5)),cv2.FONT\_HERSHEY\_SIMPLEX, 0.75, (255, 255, 255), 2)

#print ('CarID ' + str(i) + ': speed is ' + str("%.2f" % round(speed[i], 0)) + ' km/h.\n')

#else:

# cv2.putText(resultImage, "Far Object", (int(x1 + w1/2), int(y1)),cv2.FONT\_HERSHEY\_SIMPLEX, 0.5, (255, 255, 255), 2)

#print ('CarID ' + str(i) + ' Location1: ' + str(carLocation1[i]) + ' Location2: ' + str(carLocation2[i]) + ' speed is ' + str("%.2f" % round(speed[i], 0)) + ' km/h.\n')

cv2.imshow('result', resultImage)

# Write the frame into the file 'output.avi'

#out.write(resultImage)

if cv2.waitKey(33) == 27:

break

cv2.destroyAllWindows()

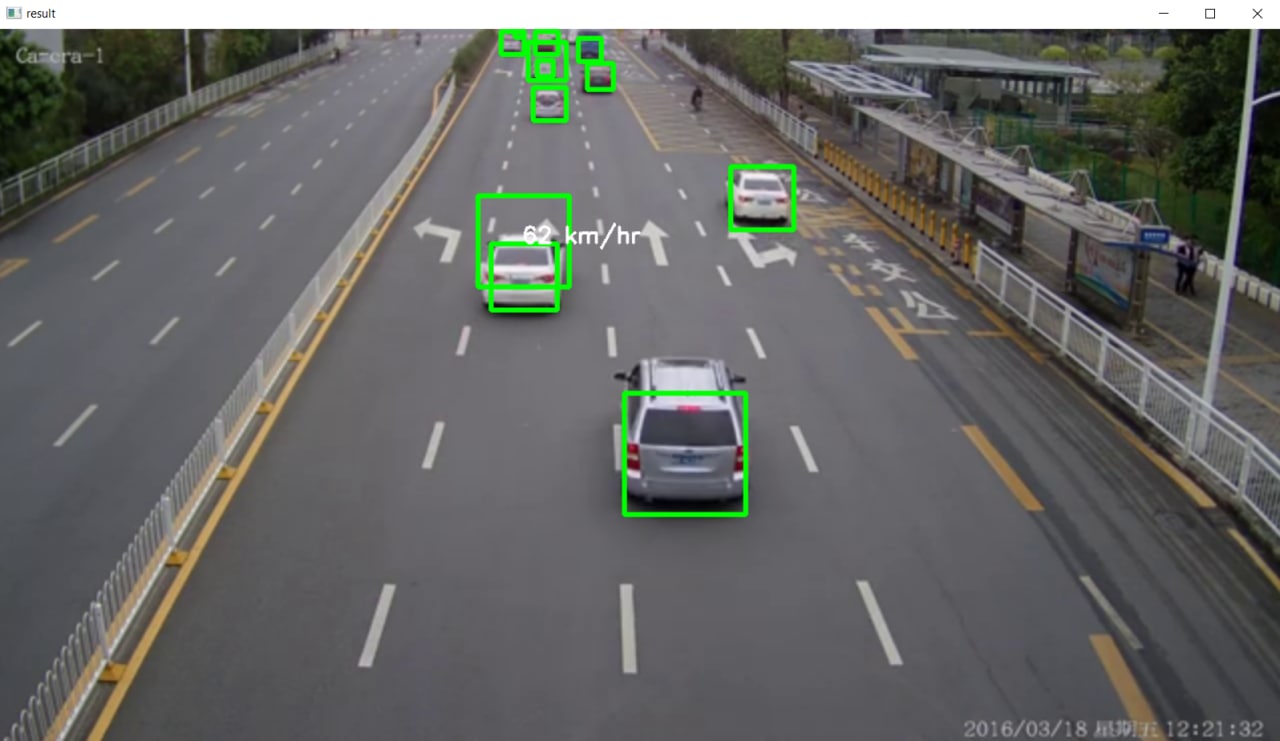
if name == 'main':

trackMultipleObjects()

**16**

**6.2. RESULTS**

we have done the code part for the project in python language, the code part suffices the main objectives of the project, the results are as follows



**6.2.1 Output**

**17**

**CHAPTER – 07**

**7.1. CONCLUSION**

We have presented a model-based vehicle tracking speed detection using AI which can work robustly under most circumstances. The system is general enough to be capable of detecting and classifying vehicles requiring only minimal scene specific knowledge. The proposed method gives better results as compared to previous techniques. Background subtraction is robust against illumination changes in real world. Also by extracting ROI the noise immunity is improved. As the distance is mapped on the image by calculating it from real world. So the calculated speed is approximated to actual speed. Road safety and reducing accidents is a very crucial issue and must be considered at utmost priority. One must abide the rules of maintaining appropriate speed guidelines. Technological tools and tracking devices which help in monitoring the motion and speed of vehicles can help reduce the number of accidents on roads as well as trace the origins of the mishap.

**18**

**7.2. FUTURE SCOPE**

A further developed application of the driver drowsiness detection system may involve In integration of the system directly with the safety systems

* As of now the project’s only aim is to detect the vehicle in the frame and to capture the car from frame to frame and to match with the dataset and provide the output based on them., we would further want to develop our own dataset in order to achieve highly accurate results and to avoid the chance of getting wrong results.
* We would like to develop the system in to an autonomous unit by using the concepts of i.ot and artificial intelligence.
* We would like to develop a machine learning algorithm which would react to the vehicle driven pattern and predict if there will be any crashes based on the driving styles and if there are any chances then it would inform local police stations.

**19**

**CHAPTER – 08**

**REFERENCES**

[1] Vamsi Krishna Madasu and M. Hanmandlu, ―Estimation of Vehicle Speed by Motion Tracking on Image Sequences‖, IEEE Intelligent Vehicles Symposium University of California, San Diego, CA, USA,June 2010.

[2] Guolin Wang, and Deyun Xian Jason Gu ―Review on Vehicle Detection Based on Video for Traffic Surveillance‖ Proceedings of the IEEE International Conference on Automation and Logistics Qingdao, China September 2008

[3] Muhammad Akram Adnan, Nor Izzah Zainuddin ―Vehicle Speed Measurement Technique Using Various Speed Detection Instrumentation‖ IEEE Business Engineering and Industrial

Applications Colloquium (BEIAC), 2013

[4] Chen Peijiang, ―Moving Object Detection Based on Background Extraction‖ IEEE conference, 2009

[5] Weiqiang Wang, Member, IEEE, Jie Yang, Senior Member, IEEE, and Wen Gao, Senior Member, IEEE Modeling Background and Segmenting Moving Objects from Compressed Video‖, IEEE Transactions On Circuits And Systems For Video Technology, VOL. 18, NO. 5, MAY 2008

[6] Chieh-Ling, Huang, Heng-Ning, Ma, ―A Moving Object Detection Algorithm for Vehicle Localization‖, Sixth International Conference on Genetic and Evolutionary Computing, 2012 P.Daniel Ratna Raju, G.Neelima, ―Image Segmentation by using Histogram Thresholding‖, IJCSET Vol 2, Issue 1pp:776-779, 2012.

**20**