iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VII **Month of publication:** July 2022

DOI: https://doi.org/10.22214/ijraset.2022.45634

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VII July 2022- Available at www.ijraset.com

Road Lane Detection using Computer Vision

Prof. Bhasker Rao K[1], Ankit Anand[2], Aditi Soni[3]
[1, 2, 3] *Dayananda Sagar Academy of Technology and Management,Bangalore*

Abstract*: Given a picture captured from a camera hooked up to a vehicle moving on a road during which captured road could or might not be levelled, or have clearly described edges, or some previous acknowledged patterns thereon, then road detection from one image will be applied to search out the road in a picture so it might be used in automation of driving system within the image captured by the vehicle, we are able to use some algorithms for vanishing point detection, exploitation Hough Transformation Space, finding the region of interest, edge detection exploitation Canny edge detection for road detection. We have a tendency to use thousands of pictures of various roads to coach our model so the model might notice the roads as a result within the new image processed through the vehicle.

Keywords*: Patterns thereon, automation of driving system, vanishing point detection, exploitation, Hough Transformation Space, Region of Interest, Canny Edge Detection.

I. INTRODUCTION

The most facing challenge of today's world is automation of manual processes. Automation of vehicle driving is one of the challenging tasks for the world. The main process to automate the driving in vehicles can be road detection by the vehicle. This is the important process because in order to move the vehicle in absolute direction can be implemented once the road is detected by the vehicle. These types of vehicles can be stated as self-driving vehicles. Since road images are to be captured by the vehicle and then generate the output, the images should be captured continuously by the vehicle and process them to find the curves in the road such that vehicles can move in that way. Since road images are continuously captured and processed by the system, this can be classified as a problem of computer vision. Computer vision is the algorithm or group of algorithms running together on instant continuously captured images thus drawing the data insights from the image's information given to a computing machine. Thus, the road detection by a computing machine while capturing images from a moving vehicle can be treated as a problem of computer vision.

II. BACKGROUND WORK

1) *Model 1:* Gaussian filtering to get rid of noise GreyScale image conversion: A greyscale (or grey level) image is solely one during which the sole colours area unit reminder Grey.

The explanation for differentiating such pictures from the other type of colour image is that less info must be provided for every constituent. A `Grey' colour is one during which the red, inexperienced and blue elements all have equal intensity in RGB house, then it's solely necessary to specify one intensity worth for every constituent as critical 3 intensities, required to specify every constituent during a full colour image. Often, the grayscale intensity is held on as AN 8- bit whole number giving 256 doable completely different reminders of Grey from black to white. If the amount of area unit is equally spaced, then the distinction between ordered grey levels is considerably higher than the greylevel physical phenomenon of the human eye. Greyscale pictures area unit quite common, partly as a result of the abundance of today's show and image capture hardware will solely support 8-bit pictures. Additionally, Greyscale pictures area unit entirely spare for several tasks then there's no need to use additional sophisticated and harder-to-process colour pictures.

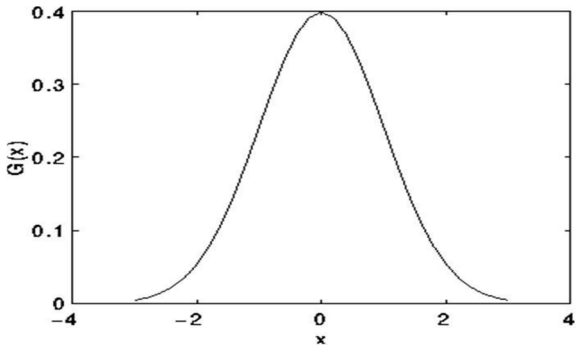
Gaussian Blur in an image:

The Gaussian smoothing operator may be a 2-D convolution operator that's wont to`blur' pictures and take away detail and noise. In this sense it's just like the mean filter, however it uses a distinct kernel that represents the form of a Gaussian (`bell-shaped') hump. This kernel has some special properties that exploit normal distribution.

How Gaussian Blur works:

The normal distribution in 1-D has the form:

©IJRASET: All Rights are Reserved



$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}$$

where in the equation letter of the alphabet is the variance of the distribution. we've additionally assumed that the distribution contains a mean of zero (i.e. it's image centred on the road x=0).

Normal Distribution of information is as follows:

The first step of canny edge detection is to strain any noise within the original image before making an attempt to find and detect any edges. The Gaussian filter is employed to blur and take away unwanted detail and noise. By hard an acceptable five X five mask.

The Gaussian smoothing will be performed using normal convolution methodology. A convolution mask is far smaller than the particular image. Kernel slides over the matrix of image, hard each square. of pixels at a time.

Filter uses a second distribution to perform convolution. The burden of the matrix is targeted at the middle, thus any noise showing within the outside columns and rows are eliminated, because the weight decreases outward from the middle value. The increasing of ordinary deviation reduces or blurs the intensity of noise.

Pre-Processing of chosen Image:

Distortion Correction: Image distortion happens once a camera appears at 3D objects within the globe and transforms them into a second image. This transformation isn't invariably excellent and distortion may result during a modification in apparent size, form or position of AN object. Therefore, we want to correct this distortion to relinquish the camera's correct read of the image. This is often done by computing a camera activity matrix by taking many checkerboard photos of a camera. Example below of a distortion corrected image. Please note that the correction is extremely little in traditional lenses and also the distinction isn't visibly abundant.

Create a binary image: currently that we've the ingenuous image, we will begin our analysis. We'd like to explore completely different schemes so we will clearly see the thing of interest on the road, while ignoring the remainder. I did this in 2 ways.

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

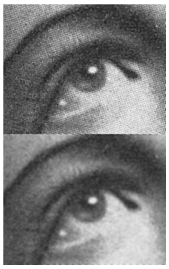


Figure: Describing the first image and ingenuous image

victimization Sobel operator to figure x-gradient: The gradients of a picture are often accustomed to establish sharp changes in

colour a black and white image. It's an awfully helpful technique to sight edges in a picture. For the image of a road, we tend to sometimes have a road line in either yellow or white on a black road so x-gradient are often terribly helpful.

Explore different colour channels: Hue, Saturation and Value (HSV) colour houses are often terribly helpful in uninflected the yellow and line white lines as a result of it isolates colour (hue), quantity of colour (saturation) and brightness (value). we will use the S colour channel within the image.

Birds Eye read Image: when the thresholding operation, we tend to perform a perspective remodel to vary the image to bird's eye read. This can be done as a result of this high read we will establish the curvature of the road and judge the way to steer the automobile. To perform the attitude remodel, I know four supply points that type a trapezoid
on the image and four destination points on the specified road that are parallel to every different transformation. The destination points were chosen by trial and error however once chosen works well for all pictures and therefore the video since the camera is mounted in a very fastened position. OpenCV is often accustomed to perform this. See however clearly the curvature of the road is visible during this read. work curve lines to the bird eye read image: so as to raise an estimate wherever the road is, we tend to use a bar chart of the lowest 1/2 image to spot potential left and right road markings. Modification of this performs to slender down the world within which left and right roads will exist so road road separators or the other noise doesn't get known as a road. Once the initial left and right road bottom points are known. Plot the result known by the system clearly. This plotting is often done filling the house space with clear colour victimization OpenCV.

Thus, Self-Driving automobiles used for road detection are often helpful in detection of roads from a picture captured from an automobile.

Canny Edge Detection: Edges characterize boundaries and are thus a tangle of elementary importance in the image process. Edges in pictures are areas with robust intensity contrasts – a jump in intensity from one element to consequent. Edge police work a picture considerably reduces the number of info} and filters out useless information, whereas conserving the vital structural properties in a picture. A smart edge detection algorithmic program is additionally referred to as the optimum
edge detector. Canny's intentions were to boost the various edge detectors within the image. The primary criterion ought to have a low error rate and separate out unwanted data whereas the helpful data preserve. The second criterion is to stay the lower variation as attainable between the initial image and therefore the processed image. Third criterion removes multiple responses to a grip. Based on these criteria, the smart edge detector initially smoothens the image to eliminate noise. It then finds the image gradient to focus on regions with high spatial derivatives. The algorithmic program then tracks on these regions and suppresses any element that's not at the most victimization non-maximum suppression. The gradient array is currently reduced by physical phenomenon to get rid of streaking and cutting the perimeters Filter out noise.

Convolution Operation: initiative to smart edge detection needs some technique to separate out any noise and still preserve the helpful image. Convolution could be a straightforward mathematical technique to image-processing operators.

I1	I2	I3	I4	I5	I6	I7	I8	I9
I10	I11	I12	I13	I14	I15	I16	I17	I18
I19	I20	I21	I22	I23	I24	I25	I26	I27
I28	I29	I30	I31	I32	I33	I34	I35	I36
I37	I38	I39	I40	I41	I42	I43	I44	I45
I46	I47	I48	I49	I50	I51	I52	I53	I54
I55	I56	I57	I58	I59	I60	I61	I62	I63
I64	I65	I66	I67	I68	I69	I70	I71	I72
I73	I74	I75	I76	I77	I78	I79	I80	I81

K1	K2	K3
K4	K5	K6
K7	K8	K9

Figure: associate degree example little image (left), kernel (right)

Convolution operation: Convolution is performed by slippy the kernel or mask over a grey-level image. The kernel starts from the highest left corner and moves through the entire image at interval image boundaries. every kernel position corresponds to one output element. every element is increased with the kernel cell price and additional along. The output image can have M-m+1 rows and N-n+1 columns, M image rows and N image columns, m kernel rows and n kernel columns. The output image is smaller compared to the initial image. This can be thanks to the lowest and right edge pixels that can't be fully mapped by the kernel thus m –1 right element and n-1 bottom pixels can't be used.

2) Model 2: Sobel Operator: when smoothing the image and eliminating the noise, the consequent step is to seek out the string

strength by taking the gradient of the image. The Sobel operator performs a 2-D special gradient measure on a picture. Then, the approximate absolute gradient magnitude (edge strength) at every purpose is often found by the formula below that is easier to calculate compared to the on top of precise gradient magnitude. Approximate gradient magnitude given below:

$$|\square| = |\square_s| + |\square_y|$$

The Sobel operator uses a combination of 3x3 convolution masks, one estimating the gradient within the x-direction (columns) and therefore the difference estimating the gradient within the y-direction (rows). Sobel

$\square_s$ and Gentile masks shown below each estimate's gradient x direction and y direction severally.

3) Model 3: Finding Gradient angle: Finding the sting direction is trivial once the gradient within the x and y directions are proverbial. However, you'll generate a slip whenever total of *G*s is adequate to zero i.e. *G*s worth in divisor that means scheming arc an of time. So, the sting direction can adequate to ninety or zero degrees rely upon *G*s worth and zero degrees rely upon Christian worth. The formula for locating the sting direction is given below:

$$\theta = tan^{-1}(Gy)$$
$$Gs$$

4) Model 4: Tracing the sting with in the image victimization θ (angle): Once the sting direction is thought, consequent step is to relate the sting direction to a direction that may be copied in a picture. So, if you use the 5x5 matrix to calculate the angle of the sting, the smaller the matrix the less angles would have within the image.

x	x	x	x	x
x	x	x	x	x
x	x	a	x	x
x	x	x	x	x
x	x	x	x	x

By observing the middle component "a", there are four doable directions once describing

G_s and Gentile masks shown below each estimate's gradient x direction and y direction severally.

degrees (in the horizontal direction), 45 degrees (along the positive diagonal), ninety degrees (in the vertical direction), or a hundred thirty five degrees (along the negative diagonal), 180 degrees region is simply an mirror region of 0 degrees region. Therefore, any edge direction calculated is pulled together to the nearest angle. So,

any edge direction falling inside the A and E (0 to 22.5 & 157.5 to

180 degrees) is set to 0 degrees. Any edge direction falling within the D (22.5 to 67.5 degrees) is set to 45 degrees. Any edge direction falling within the C (67.5 to 112.5 degrees) is set to 90 degrees. and eventually, any edge direction falling inside the B (112.5 to 157.5 degrees) is set to 135 degrees.

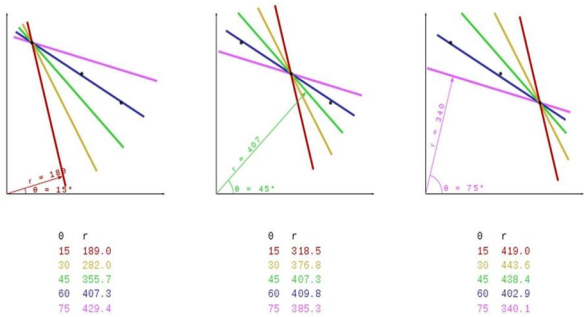


Figure: Tracing the Edge of an Image

The bigger the matrix the larger the range of angles one might get, which suggests the sting angles are a lot more precise.

i.e. can follow the sting higher, however on the down facet it'll be an even bigger computation task as currently the kernel/mask size is greater.

5) Model 5: Non most Suppression: once the sting directions area unit is acknowledged, non-maximum suppression is applied.

Non- most suppression is employed to trace on the gradient within the edge direction and compare the worth perpendicular to the gradient. 2 perpendicular picture element values area unit compared with the worth within the edge direction. If their worth is not up to the picture element on the sting then they're suppressed i.e. their picture element worth is modified to zero, else the upper picture element worth is ready because the edge and therefore the different 2 suppressed with a picture element worth of zero. We would like to mark points on the curve wherever the magnitude is biggest. We will do that by trying to find a most on a slice traditional to the curve (non-maximum suppression). These points ought to type a curve. There are then 2 algorithmic issues: at that purpose is that the most, and wherever is that the next one?

6) *Model 6:* Hysteresis: Finally, physical phenomenon is employed as a method of eliminating streaking. Streaking is that the calling it off of a position contour caused by the operator output unsteady on top of and below the edge. If one threshold, T1 is applied to a picture, and a position has a mean strength adequate to T1, then because of noise, there'll be instances wherever the sting dips below the edge. Equally it'll additionally extend on top of the edge creating a position that seems like a dotted line. To avoid this, physical phenomenon uses two thresholds, a high and an occasional. Any picture element within the image that encompasses a worth larger than T1 is plausible to be a position picture element, and is marked in and of itself straight off. Then, any picture elements that are unit connected to the current edge pixel which have a price larger than T2 also are elite as edge pixels. If you think that by following a position, you would like a gradient of T2 to begin however you do not stop until you hit a gradient below T1. Input and Output of Canny Edge Detection Algorithm:

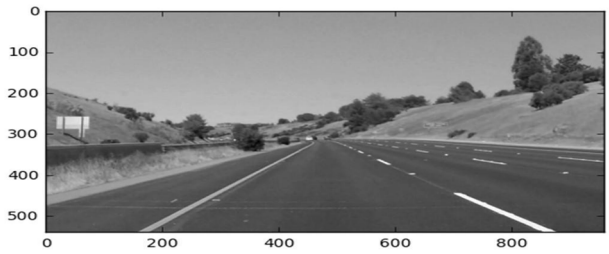


Figure: Input Image to Canny's Process


Figure: Output Image after Canny's Process

Introduction to Hough Transformation: The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform. The classical Hough transform was concerned with the identification of lines in the image, but later the Hough transform has been extended to identifying positions of arbitrary shapes, most commonly circles or ellipses. The Hough transform as it is universally used today was invented by Richard Duda and Peter Hart in 1972, who called it a "generalized Hough transform" after the related 1962 patent of Paul Hough. The transform was popularized in the computer vision community by Dana H Ballard through a 1981 journal article titled "Generalizing the Hough transform to detect arbitrary shapes"

Theory about Hough Transformation Space: In automated analysis of digital images, a sub problem often arises of detecting simple shapes, such as straight lines, circles or ellipses. In many cases an edge detector can be used as a pre-processing stage to obtain image points or image pixels that are on the desired curve in the image space. Due to imperfections in either the image data or the edge detector, however, there may be missing points or pixels on the desired curves as well as spatial deviations between the ideal line/circle/ellipse and the noisy edge points as they are obtained from the edge detector.

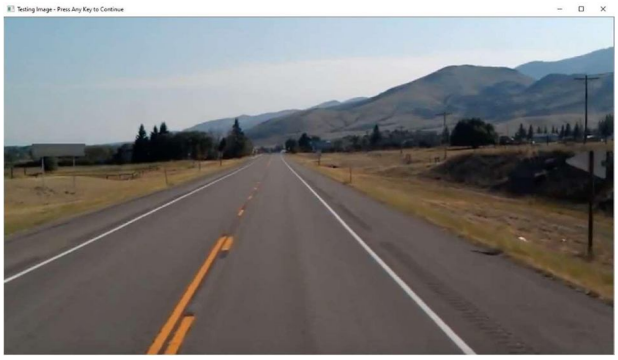
For. The aim of the Hough remodel is to deal with this drawback by creating the potential to perform groupings of edge points into object candidates by performing arts a certain ballot procedure over a collection of parameterized image objects (Shapiro and granger, 304). The only case of Hough remodel is sleuthing straight lines. In general, the straight-line y = Mx
+ b are often drawn to some extent (b,
m) within the parameter area. However, vertical lines cause a drag. they might create infinite values of the slope parameter m. Thus, for procedure reasons, Duda and Hart planned the utilization of the Hermann Hesse traditional type. These reasons, it's usually non-trivial to cluster the extracted edge options to Associate in Nursing acceptable set of lines, circles or ellipses.

III. METHODOLOGY

A. Algorithm

Road Detection from a single image using Computer Vision consists of image insertion, model building and then testing. The model evaluation is done manually by the developer. We divided the complete project into mainly four modules. They are as follows:
Step 1: Selecting the appropriate testing image.
Step 2: Processing the testing image. Step 3: Edge Detection Implementation. Step 4: Hough Transformation.

Step 1: Selecting the appropriate testing image: It is the most important process in the project. Single Image from testing dataset is taken such a way that it reaches our implementation of a model. Each model we implement takes a resultant image as an input and processes it further to produce an output. This selection of images is more important because implementation of each model requires an image input for processing. And if the processing is done, then output is produced. If output produced for the testing image is the same as required, then the resultant image is sent to the next process that we need to develop further. In order to observe the clear output, the best suitable image should be selected such that the testing image should be able to produce the clear required output at the end of processes
For example, we will select an appropriate wanted image such that we could expect the desired output from the image. When we select an image from a group of testing images, it should be clear that the model should work on the
selected image and then appropriate output should be produced in each step and then that output will be
sent as an input to the next step.



Experimental Output 1: Selected image from the testing database

This image is selected because we could expect the output to be highlighted on the right sided road in which the vehicle should move. When a self-driving vehicle is moved by capturing the images, it captures the above type similar image and then absolute road should be detected through next processes. Since, we could detect and validate the absolute road from the above selected image, this image can be considered to be selected for processing by computer vision.
Step 2: Preprocessing the selected image: Preprocessing plays a major role in producing the required output in sufficient required amount of time. Preprocessing of selected images mainly undergoes the Grey scale conversion and smoothing techniques which would be considered as the first process in Canny's process.
The selected image is converted into Grayscale through the open source computer vision package. And then smoothening is applied by implementing the Gaussian Blur algorithm on the selected Grey scale image. A Grayscale image mainly consists of changes in variants from white to black that represent the color mixes of red, green and blue.
Normalization is the main process of Gaussian Blur process conversion which is done through multiplying each intensity of a pixel

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

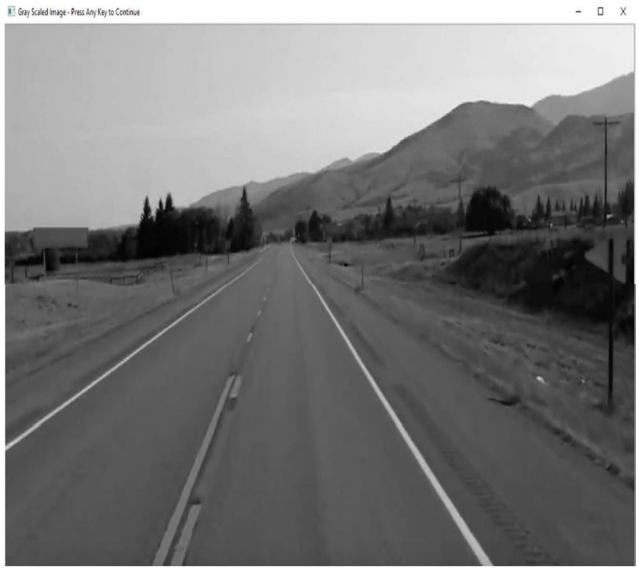
by their corresponding normalized matrix values. Thus, preprocessing is done on the selected image. This conversion of Grayscale image and reducing noise in the image can help by reducing the processing time in the next large processes.
For example, the selected image is sent as an input to a grayscale conversion model and then a grayscale image can occur as an output.

```
def grayImage(img):
    gray = cv.cvtColor(img, cv.COLOR_RGB2GRAY)
    showImage(gray, "Gray Scaled Image - Press Any Key to Continue")
    return gray
```

```
gauss = (1.0/57) * np.array(
        [[0, 1, 2, 1, 0],
        [1, 3, 5, 3, 1],
        [2, 5, 9, 5, 2],
        [1, 3, 5, 3, 1],
        [0, 1, 2, 1, 0]])
sum(sum(gauss))

for i in np.arange(2, height-2):
    for j in np.arange(2, width-2):
        sum = 0
        for k in np.arange(-2, 3):
            for l in np.arange(-2, 3):
                a = img.item(i+k, j+l)
                p = gauss[2+k, 2+l]
                sum = sum + (p * a)
        b = sum
        img_out.itemset((i,j), b)
```

Figure: Code for inputting selected image and converting into grayscale image



Experimental Output 2: Greyscale image of selected road

This grayscale image occurred after the initial preprocessing technique, the output image is sent into Gaussian Blur preprocessing technique which will reduce the noise in the image, the smoothness in outlines of the image can be observed.
Figure: Code for implementing the gaussian filter on image.
The kernels are slid over the image and the gauss matrix which represents the normalized matrix is done to reduce the noise present in the image. The original image sent into this technique will be the output of grey scale preprocessing technique.



Experimental Output 3: Gaussian Filtered image after gaussian kernel sliding

Step 3: Edge Detection Implementation: The next step in the process is edge detection, which is the main part in the program and required to detect the edges in the image irrespective of details present in an image. We use Canny Edge Detection Algorithm to implement the edge detection techniques because the other processes which are also used to find the edges in an image would contain detailed images compared to Canny Edge Detection Technique. Canny Edge Detection technique mainly consists of four processes in it. They are Gaussian Blur which we have performed for smoothening of image as preprocessing technique, Gradient Calculation which is used to calculate θ for boundary selection of an image followed by Non-Maximum suppression and double threshold required for strengthening the lines occurred in edged image of previous functions. Thus, we get the image with edges which is applied as an input to Hough transformation techniques.

For example, the image occurring after applying the gaussian filter is sent into the Canny's process for edge detection and all the edges of an image occur on the dark background. All edges are represented in white colour on a dark background because it becomes easy for processing during the next processes

which primarily concerns the speed of processing of images in road detection. The image occurring after sliding the gaussian filter is passed as an input to edge detection model and then an image with edges detected will occur.

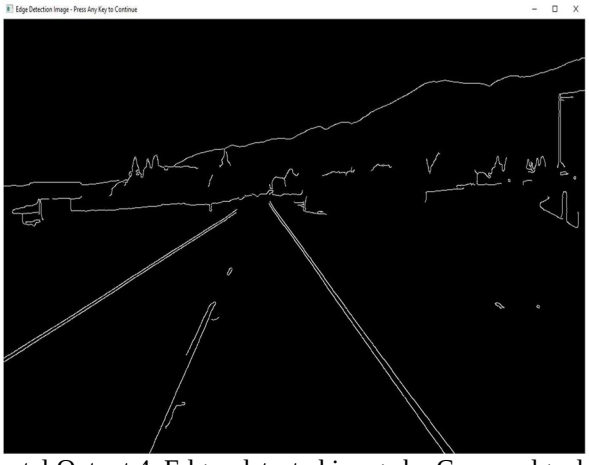
```
    # step 2: compute gradient magnitude
    img_x = convolve_np(img_blur, np.array([[-0.5, 0, 0.5]]))
    img_y = convolve_np(img_blur, np.array([[-0.5],
                                            [0],
                                            [0.5]]))
    E_mag = np.sqrt(np.power(img_x, 2) + np.power(img_y, 2))
    E_mag = (E_mag / np.max(E_mag)) * 255

# step 3: non-maximum suppression
    #t_low = 4
    E_nms = np.zeros((height, width))
    for i in np.arange(1, height-1):
        for j in np.arange(1, width-1):
            dx = img_x[i,j]
            dy = img_y[i,j]
            s_theta = ors.get_orientation_sector(dx,dy)

            if locmax.is_local_max(E_mag, i, j, s_theta, t_low):
                E_nms[i,j] = E_mag[i,j]

#step 4: edge tracing and hysteresis thresholding
    #t_high = 15
    E_bin = np.zeros((height, width))
    for i in np.arange(1, height-1):
        for j in np.arange(1, width-1):
            if E_nms[i,j] >= t_high and E_bin[i,j] == 0:
                tt.trace_and_threshold(E_nms, E_bin, i, j, t_low)
    return E_bin
```

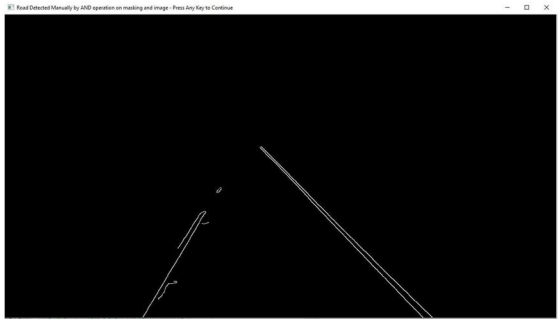
Figure: Code steps for detecting the edges from an image



Experimental Output 4: Edges detected image by Canny edge detection process

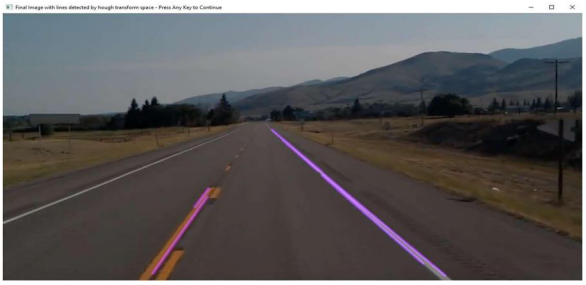
Step 4: Hough Transformations: Hough Transformations require a Hough transformation space which is used to rotate the angles of a trigonometric line equation and then specify the lines present in an edge detected image. If the trigonometric line in rotation meets the edges in the image, then it may be considered for applying the model trained for detecting roads in an image. The training is done in Hough transformation space which is used to detect the actual road lines from an image. When the Hough transformations and training is done, then the road lines are detected on the selected image. The training of the model is improvised until the correct output is observed from a selected image. Thus, the image occurring through the model after Hough transformation can be verified by the testing and the model can be used for detection of roads from continuous images of input.

For example, the edge detected image is passed over the hysteresis process and then masking is done to other unrequired parts of image such that the OR operation is done with 0's and 1's. The resultant image consists of only the required road detected for moving a vehicle on it.



Experimental Output 5: The masked image with considering only the required road for vehicle movement

The road should be marked along the edges which occurred after masking the image. The Hough lines are rotated along the image by considering the masked image and then lines can be drawn on the original image which will be road detected by the system in the vehicle. This can be done by frequent change in theta value along the original image considering the masked image. Thus, the original image output can be observed.



Experimental Output 6: Output of detected road from image captured by vehicle

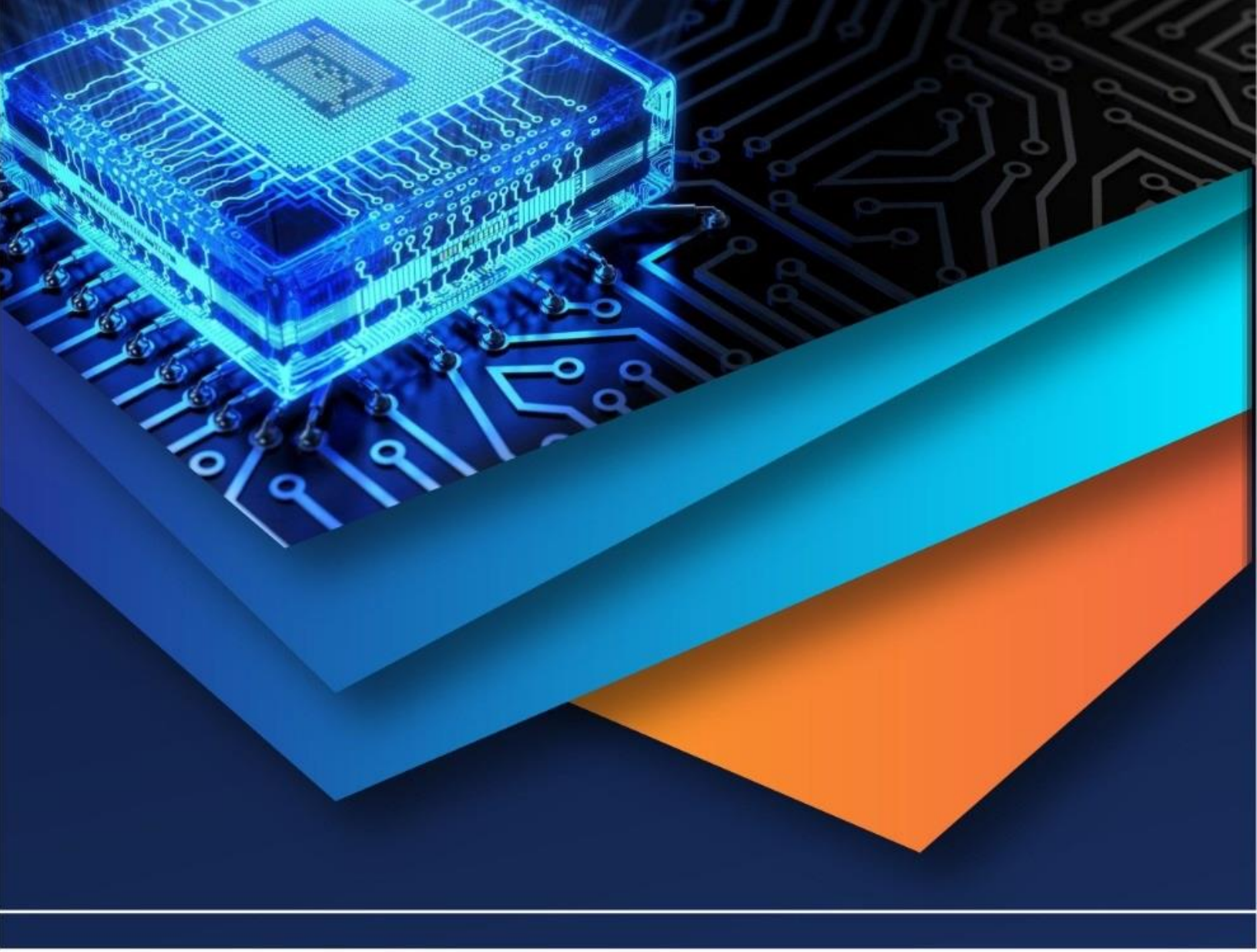
IV. CONCLUSION

When we drive, we use our vision to decide where to go. The lines on the road detected by the model that show us where the lanes are act as our constant reference for where to steer the vehicle. This steering is also done automatically. Naturally, one of the first things we would like to do in developing a self-driving vehicle is to automatically detect lane lines using an algorithm. The road detection region of interest (ROI), must be flexible. When driving up or down a steep incline, the horizon will change and no longer be a product of the proportions of the frame. This is also something to consider for tight turns and bumper to bumper traffic. This model is based on image processing and road detection in self-driving vehicles which has a great scope in future. We use specific algorithms in step process implementation to detect the road clearly. If people's thoughts haven't changed about self-driving cars being safe, these cars are already safe and are becoming safer. Users will give a try to newer technology; they get to enjoy the luxury of computerized driving. Driverless vehicles appear to be an important next step in transportation technology. They are a new all-media capsule- text to your heart's desire and it's safe. Developments in autonomous vehicles are continuing and the software development is continuing to be updated. Though it all started from a driverless thought to radio frequency, cameras, sensors, more semi- autonomous features will come up, thus reducing the congestion, increasing the safety with faster reactions and fewer errors.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

REFERENCES

[1] General Road Detection From A Single Image, TIP-05166-2009, ACCEPTED, Hui Kong , Member, IEEE, Jean-Yves Audibert, and Jean Ponce , Fellow, IEEE Willow Team, Ecole Normale Superieure / INRIA / CNRS, Paris, France Imagine team, Ecole des Ponts ParisTech, Paris, France.
[2] J. C. McCall and M. M. Trivedi, "Video based lane estimation and tracking for driver assistance: Survey, system, and evaluation," IEEE Trans. on Intelligent Transportation Systems, pp. 20–37, 2006.
[3] K.-Y. Chiu and S.-F. Lin, "Lane detection using color-based segmentation," IEEE Intelligent Vehicles Symposium, 2005. 1
[4] H. Kong, J.-Y. Audibert, and J. Ponce, "Vanishing point detection for road detection," CVPR, 2009.
[5] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using b-snake," Image and Vision Computing, pp. 269–280, 2004
[6] A. Lookingbill, J. Rogers, D. Lieb J. Curry, and S. Thrun, "Reverse optical flow for self-supervised adaptive autonomous robot navigation," IJCV, vol. 74, no. 3, pp. 287–302, 2007
[7] Broggi, C. Caraffi, R. I. Fedriga, and P. Grisleri, "Obstacle detection with stereo vision for off-road vehicle navigation," IEEE International Workshop on Machine Vision for Intelligent Vehicles, 2005.
[8] J. Sparbert, K. Dietmayer, and D. Streller, "Lane detection and street type classification using laser range images," IEEE Proceedings in Intelligent transportation Systems, pp. 456–464, 2001.
[9] J. B. Southhall and C. Taylor, "Stochastic road shape estimation," ICCV, pp. 205–212, 2001.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)