International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 9 (2019) pp. 2156-2161

THE YAN: [Self Driving Car Using Deep Learning]

Ms. Sujeetha[1], Chitrak Bari[2], Gareja Prdip[3], Siddhant Purohit[4]

[1]*Assistant Professor, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India.*

[2,3,4,] *Student, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India.*

Abstract

When an automobile is on the road, there is a driver at the steering wheel and he is the incharge. The driverless technology is being developed and tested by universities, institutions, and companies for quite times. The idea of a car without a driver gives us a feeling of skepticism and we all will try to avoid it. In this paper, we are discussing an autonomous robotically handled driving vehicle. In our project, we are using many features such as mapping, tracking and local planning. We can successfully create a car that can demonstrate proper lane changes, parking, and U-turns on its own. The different innovations we are using are obstacles and curb detection method, road vehicle tracker, and checking different traffic situations. This will make a robust autonomous self driven car. It will successfully demonstrate proper parking allotment, lane changes, and automatic U-turns. We can do these using the obstacle and various curb detection method, the vehicle tracker.

Keywords: Parallel calculations, distributed calculations, synchronous, critical sections, Neural Networks, HSV filters, ranger finder (LIDAR), 3D Radar, 3D map, sockets, TCP/IP, Google Maps, Artificial Intelligence.

I. INTRODUCTION

Many significant improvements is done in the last 10 years that impacted advanced self-driving car technology. We use Artificial Intelligence for recognizing and proposing the path which the autonomous car should follow for proper working. Additionally a driverless car can reduce the time taken to reach the destination because it will take the shortest path, avoiding the traffic congestion. The human errors will be avoided thereby allowing disabled people (even blind people) to own their car.

II. FINDING LANE LINE USING COMPUTER VISION

An Autonomous car technology always help people to makes people live life easy and safe transportation.in this report we are using some new technique of the computer vision to finding the self driving car way, traffic signals, road line, detection of the object, we will able to see our next generation car is fully autonomous.

A) COMPUTER VISION:

Human can easily detect the different objects that are present in the surrounding but an autonomous self driving car cannot, so we are using CV. This helps in acquiring, processing, analyzing and understanding the digital images formed. We then extract the high-dimensional information from the real world to apply the numerical or symbolic information. So we even use mathematics, physics, statistics and different learning theory.

B) DETECTING THE LIVE OBJECT ON THE LINE(ROAD):

Detection of the any object or road line any autonomus car need computer vision library. In the report we are using opencv libary. OpenCV (Open source computer vision) is used for computer vision.

Step 1: Gray-Scale Conversion of the Object:

Grayscale is the different shades of gray without any other color, where black is the darkest and white the lightest shade. Black means reflected light and the white means reflection of light at all visible wavelength. Other shades of gray are at equal brightness levels. The three primary colors are red, green and blue. First we convert them to grayscale. This is an important step to Canny Edge Detector within OpenCV.

We need to make it clear exactly what we are looking for, before we can detect the edges. Lane lines are white and sometimes yellow. Yellow is a quite difficult color to isolate from the RGB area, so we convert it into Hue Value Saturation or HSV color space.

We have done really good. We have processed a lot since our original image. Then we apply Gaussian blur.

C) METHOD FOR PERFORMING OPENCV

a) Gaussian blur Method:-In processing a image, a mathematician usually blur the image using Gaussian function. It is generally used in graphics software system, image noise and in reduce detail. The result of blurring technique resembles that of image viewing the image through a semitransparent screen. In pc vision algorithms Gaussian smoothing is used as a pre-processing stage to boost image at totally different scales. The Gaussian blur uses Gaussian function for verifying the transformation, so as to use image's every pixel in frame.

The formula of a Gaussian function in 1-D is:

$$G(x)=1/\sqrt{2\pi\sigma} *(e)^{\wedge} - \frac{x^{\wedge}2}{2\sigma^{\wedge}2} \quad \text{.....(1)}$$

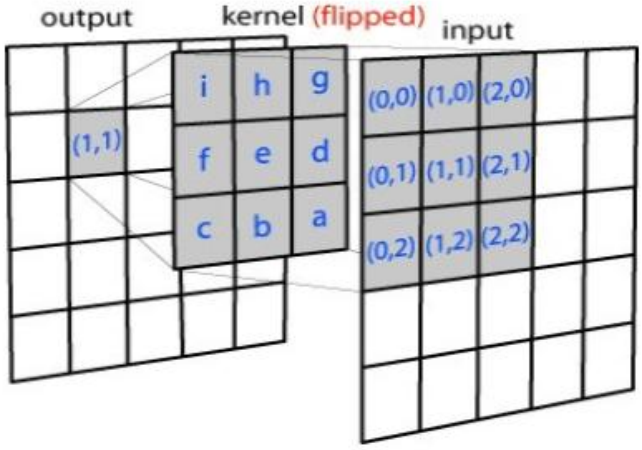
In 2-D, it is the product of two one dimension Gaussian functions:

$$G(x,y)=1/2\ \pi\sigma^2 *(e)^{\wedge}-(x^{\wedge}2+y^{\wedge}2/2\ \sigma^2)$$

x is the distance between the origin and horizontal axis, y is the distance between the origin and vertical axis, and σ is the standard deviation of the Gaussian distribution.

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

Gausian Kernel is multiplied to each pixel of the image. This is done after placing the center pixel of the kernel on the image pixel and then multiplying the values in the original image with the pixels in the kernel that overlap. The values ensuing from these multiplications are added up which result's used for the value at the destination pixel. Looking at the image, you would multiply the value at (0,0) within the input array by the value at (i) within the kernel array, the value at (1,0) in the input array by the worth at (h) within the kernel array, and so on.and then add of these values to induce the value for (1,1) at the output image.



The larger the kernel more expensive the operation. So, larger the radius of the blur of the image, the longer time it will take for the operation to complete.

b) Canny Edge Detection- Canny Edge Detection. canny parses the pixel values according to their directional derivative also called gradient. What's left over are the edges—or where there is a steep derivative in at least one direction John Canny himself recommended a low to high threshold ratio of 1:2 or 1:3 .. We don't want our car to be paying attention to anything on the horizon, or even in the other lane. Our lane detection pipeline should focus on the obstacles present in front of the car. To do that, we are going to create a mask called our region of interest (ROI). We are only working with the relevant edges so everything outside of the ROI will be set to black/zero,

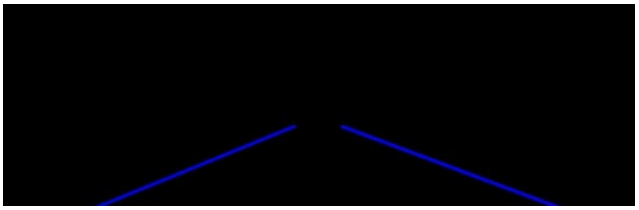
Preprocessing: Edge detectors are are prone to noise. opencv using a bit smoothing used of Guassian blur. From what I've seen, software packages don't do this step automatically, you need to do this yourself. Usually, 5x5 Gaussian filter with a standard deviation = 1.4 is used for this purpose.

Calculating gradients:

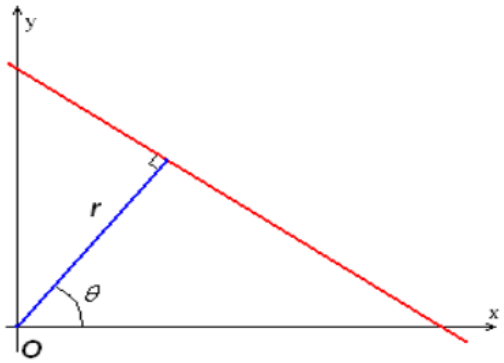
Next, gradient magnitudes and directions are calculated at every pixel-+ image. The *magnitude* of the gradient at a point determines if it possibly lies on an edge or not. A high gradient magnitude of the colors are changing magnitude rapidly - implying on edge. A low gradient implies no substantial changes. So it's not an edge. The gradient shows how the edge

is oriented. To calculate these, sobel edge detector is used. cis θ = arctan *arctan* (*Gy*) /(*gx*) . Here, gxand Gy are the X and Y derivatives of the point. Once the gradient magnitudes and orientations are known, we can get started with thresholding with Hysteresis :- In the previous step, we marked pixels that had a gradient magnitude greater than the upper threshold. Now gathering information using the direction and lower threshold, we'll "grow" these edges.

C)Hough Space: The Hough transform is an type extraction technique used in computer vision, image analysis and digital image processing. The equation y=mx+b is about to reveal its alter ego —the Hough transform. In XY space lines are lines and points are points, but in Hough space *lines* correspond to *points in XY space* and **points** correspond to the **lines in XY space**.



About the image, is that, it contains zero pixel data from any of the photos we processed. It is strictly black/zeros..*r* = cosθ + y sinθ



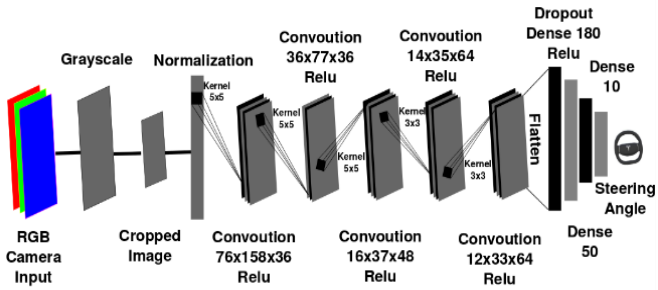
where r is the distance from the origin to the closest point on the straight line, and θ (*theta*) is the angle between the axis and the line connecting the origin with that closest point. It is therefore possible to associate with each line of the image a pair (r,0) plane is sometimes referred to as *Hough space* for the set of straight lines in two dimensions. This representation makes the Hough transform conceptually very close to similar to the two-dimensional Radon transform. Once we have our two master lines, we can average our line image with the original, unaltered image of the road to have a nice, smooth overlay.

III. PERCEPTION

An autonomous vehicle that will drive selflessly will need sensory input devices like cameras, radar and lasers for the car to perceive the world around it, creating a digital image to map. We'll be focusing cars where it perform object detection imaging. Perception system design is a vital step in developing an autonomous vehicle (AV). With the vast selection available off-the-shelf schemes and seemingly endless options of sensor systems are implemented in research and commercial vehicles, it can be difficult to identify the optimal system for one's AV application. We try to provide a comprehensive review of the state-of-the-art AV perception technology available today. It provides updated information about the advantages, disadvantages, limits, and ideal applications of specific AV sensors. The autonomous features currently on the market, localization and mapping methods currently implemented in section focus on the sensors themselves, whereas topics discussed in the Localization and Mapping section focus on how the vehicle perceives where it is on the road, providing context for the use of the automotive sensor.

IV. DEEP NEURAL NETWORK AND TRAFFIC SINGH DETECTION

Deep Learning is one of ways to make autonomous driving possible.Artificial Neural Networks are information processing systems composed of simple processing units, interconnected and acting in parallel. The inspiration is from the biological nervous systems. Computer Vision is one of the area of Artificial Intelligence where they want to extract information out of images. And with the vast demands and expectations of people toward this field, it is rising beyond beliefs: starting from object detection, pattern recognition, action recognition, automatic guidance, and so on. Many papers have been published about it, especially in Deep Learning and Convolutional Neural Network (CNN). The CNN method has largely been exploited in various fields: image processing, machine learning, video analysis, natural language processing, and much more. In the computer vision field, the CNN method has been mainly used for object detection and automation based on the .images/videos. GoogLeNet, Faster R-CNN, Single Shot MultiBox Detector (SSD), and You Only Look Once (YOLO) are examples of the succession of the CNN. Those CNN methods are mainly used for object detection and recognition.



TRAFFIC SIGNAL DETECTION:

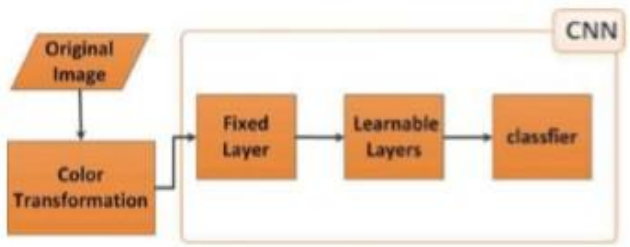
Convolutional Neural Network (CNN) is a very important and powerful tool in computer vision and self driving cars. Traffic sign detection is one of the major task in self-driving and detection of sign need to be accurate for best performances.

A) DATASET: The task is to detect 3 types of traffic signs: "Prohibitory", "Mandatory" and "Danger". We generally focused on the "Mandatory" and "Danger" category. white Examples of the two categories are shown in Figure.


(a) Mandatory Samples

(b) Danger Samples

B) ARCHITECTURE: To detect each kind of traffic signs, first transform RGB images to gray scale images using SVM, then feed the results to CNN. The fixed layer detect ROIs, and learnable layers extract distinguishing features for the classifier to find out traffic signs of the target group.



C LEARNABLE LAYERS: The learnable layers represent the standard convolutional network. the selection of design affects the potency of CNN considerably. every learnable filter layer contains a filter bank layer that convolves completely different filters with pictures, a non-linear rework layer mistreatment |tanh |, a pooling layer, and a neighborhood norm layer. we have a tendency to mentioned many architectures to decide on associate degree acceptable one.

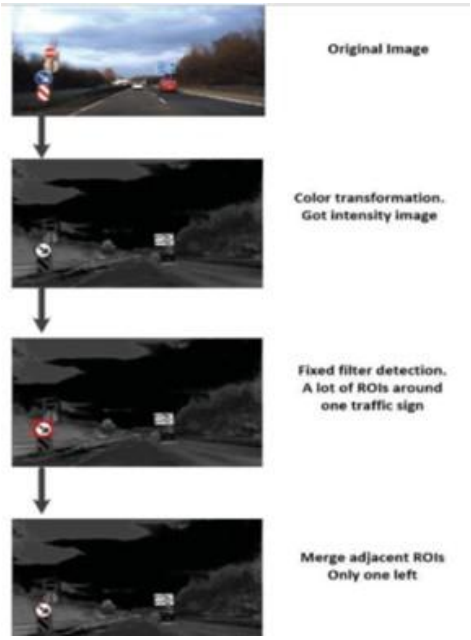
1) FILTER DATA LAYER: Filter Bank Layer: In completely different selections of CNN design ar compared. A multi-stage CNN feeding the options extracted in each of 2|the 2} stages into the classifier outperforms the CNN that additionally has two stages however solely uses the second stage's responses to classify. Since the extract within initial stage additional native and careful whereas those from second stage ar comparatively additional international, feeding responses of each of the 2 stages will increase the accuracy. In our experiment, we have a tendency to adopted the multi-stage CNN feeding options of each 2 stages into the classifier and compared completely different multi-stage CNN architectures, like 6-16, 16-512, 108-200, wherever the left variety is that the variety of filters extracted within the initial layer whereas right is that the variety of the second layer.

Architecture of learnable layers. The outputs of two learnable layers are fed to the classifier separately. The parameters of learnable layers and the classifier are trained simultaneously in supervised way. The 2-layer classifier is fully connected with 100 neurons in the first layer and 2 neurons in the second layer.

2) FIXED LAYER:

During the processing of fixed layer, correlation coefficient values are used to describe the degree of matching between a filter and a test patch. A higher value indicates that the region is more likely to contain a traffic sign. Filters we use in mandatory signs are shown in above Figure.

Process of ROI extraction :



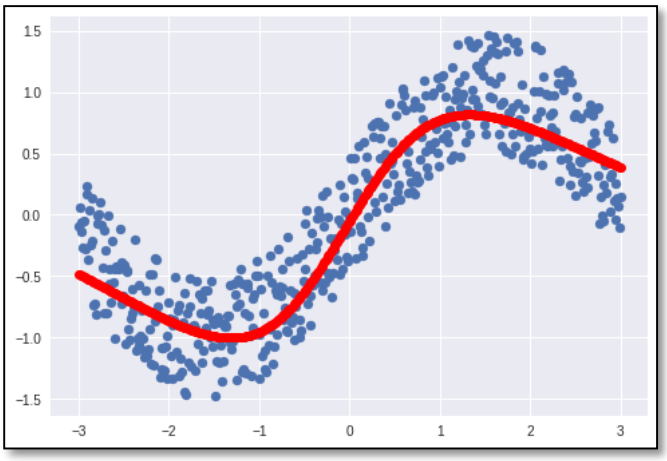
3) POOLING: The spatial pooling layer is often used to summarize the joint distribution of the nearby pixels. In pooling in a local region it boosts invariance with little shift and small noise of the region. The pooling method processes the input image as: O=(I stands for the input image, G stands for the guassian kernel. The choice of P number varies from 1 to p →∞. When P = 1, it is average pooling; when p → ∞, it's maximum pooling. During the experiment, we have set P = 4.

4) NORMALIZATION: In a local normalization method is proposed, which can be divided as subtractive normalization and divisive normalization. It can decrease the relevance of nearby pixels boosts the contract of images with noise. The subtractive normalization computes the output of pixel as following:

$$Vij=Xijk-\sum ipq\ Wpq * Wpq-(xi,j+p,k+q)$$

IV)POLYNOMIAL REGRESSION:

It is used to predict the continuous values of the data set. depending on the given data y For the design of the model we are using y= mx +b, y =dependent variable, x=independent variable, b=constant value(slop)



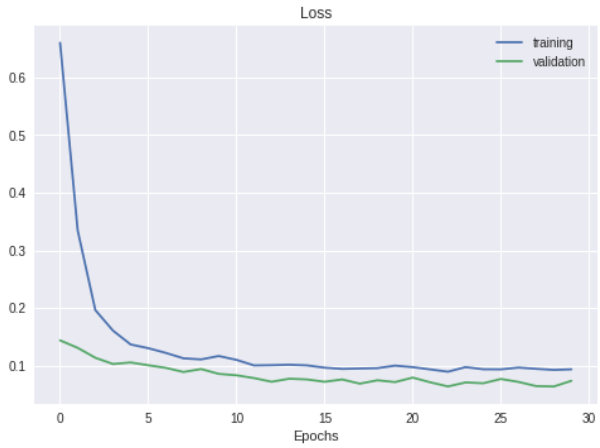
The goal of the signal fidelity measure is to compare the two signals by providing a quantitative score that describes the degree of similarity or conversely, the level of error which is basically the distortion between them. Usually, it is assumed that one of the signal is a pristine originated whereas the other is distorted or contaminated by errors.

Suppose that x = {xi|i = 1, 2,⋯ , N} and y = {yi|i = 1, 2,⋯ , N} are two finite-length, discrete signals (e.g., visual images), where N is the number of signal samples and xi and yi are the values of the ith samples in x and y, respectively. The MSE between the signals is :

$$MSE(x,y)=1/N \sum_{i=1}^{N} (Xi - Yi)^2$$

In the self driving car using polynomial regression for the finding the way of the car.

Result; Thus, the computer vision and CNN network techniques designed model of the when applying data set for the steering angle of the car our trained model given accuracy graph is shown below.



REFERENCE

[1] BudisteanuIonutAlexandru. "Using Artificial Intelligence to create a low cost self-driving car".

[2] Geoffrey Everest Hinton "Artificial neural Network". Google, university of Toronto.

[3] Geoffrey Everest Hinton "Convonlutional neural network". University of Toronto

[4] Ivan Culjak, David Abram, Tomislav Pribanic, Mario Cifrek." A brief introduction to Open." Faculty of electrical engineering and computing, University of Zagreb, Croatia"

[5] Liang Wang, Berthold K.P. Horn." Time-to-Contact control for safety and reliability of self-driving cars." Dept. EECS, MIT, Cambridge, MA 02139, USA"

[6] Jung Uk Kim, Jungsu Kwon, Hak Gu Kim, Haesung Lee, Yong Man Ro."Object Bounding Box-Critic Networks for Occlusion-Robust Object Detection in Road Scene." KAIST, Image and Video Systems Lab, School of Electrical Engineering, Republic of Korea"

[7] Rejwan Bin Sulaiman. "Artificial Intelligence Based Autonomous Car", Glyndwr university".

[8] Wilson Feipeng Abaya, Jimmy Basa, Michael Sy, Alexander C. Abad and Elmer P. Dadios." Low Cost

Smart Security Camera with Night Vision Capability Using Raspberry Pi and OpenCV “,”Electronics and Communications Engineering Department Gokongwei College of Engineering De La Salle University Manila, Philippines”.

[9] Joanne peng. ”An introduction to logistic regression Analysis and reporting”.”National Tiawan university”

[10] Mike Daily, HRL Laboratories Swarup Medasani, MathWorks Reinhold Behringer, Daimler Protics GmbH Mohan Trivedi, University of California, San Diego. ”self driving car”.