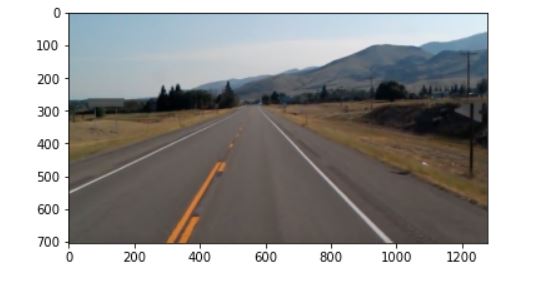
Artificial Intelligence

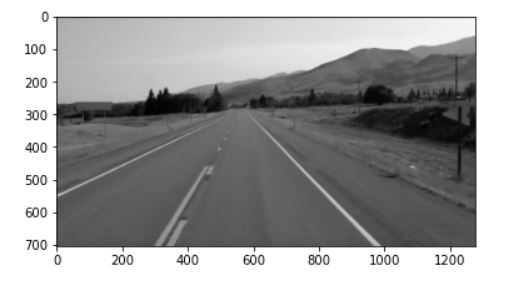
**Introduction: -** The intelligence referred to by machines that refer to the same as human intelligence is Artificial Intelligence. These machines can perform tasks with high precision and can solve a lot of real-world problems. With the advancement in artificial intelligence and innovation in automobiles, the operation of vehicles has been changed from humans to self-driving cars. Self-driving cars also known as autonomous or driverless vehicles use various sensors, cameras, and artificial intelligence to drive between destinations without any human intervention and that is the true definition of artificial intelligence making smart machines that can work without human intervention. Self-driving cars are the cars of the future and need to concentrate more on privacy, energy efficiency, environmental issues, and better technology. The goal of perfect road safety can’t be achieved as the population is increasing and the number of vehicles on road is also increasing. The real-time discovery of lane line detection is difficult because of different terrain conditions and in this project, we are using the vision-based technique to detect the road lane and all that we need is a camera at the front and we can do lane detection in real-time.

**Methodology:** When we think of autonomous cars a few things came to our mind that is safety features, comfort, driving experience, etc. In the past decades, a lot of manufacturers have added features like Lane keep assist, an Anti-lock brake system, Autopilot, etc. to do all these tasks we need artificial intelligence and more overly computer vision is used to provide this type of capabilities to the vehicle. With the help of computer vision, the system can track the surrounding and interact with the surrounding like detection, identification and tracking the road lane-lines. . The detection of the lane in real-time with the help of computer vision is finding a specific pattern like lane markings so that the vehicle can detect the streamline line. But there is some condition like snow on the road or other car standing on the same lane and it can’t be judged right by computer can cause a lot of damage. Hence we have used a real-time computer vision approach with the help of the OpenCV library to detect the road boundaries.

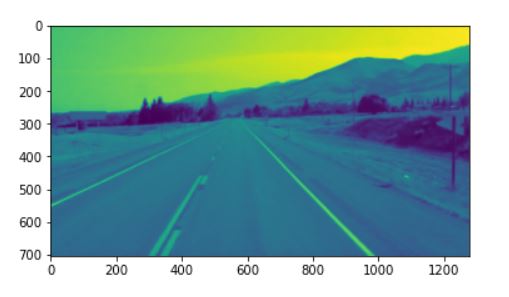
Lane is a part of the roadway that is used for a single line of vehicles and is used to guide the traffic and reduce traffic conflicts. And we can see that on the image we have white lines that we have to detect these lines for lane detection. And as there are a lot of obstacles so we will just work on that part of the frame that has a lane on it and reduce the area of interest. Now we have used basic libraries like NumPy and OpenCV for this task. NumPy is used to manipulate the pixels as we all know pixels are just numbers ranging from 0-255 and the OpenCV library is used to provide real-time computer vision tools, hardware for machine learning models. We have opened our image with the help of cvt color so that we can convert our image and show it in BGR color space. The image that we are using is given below and the shape of the image is (704, 1279, 3). When we work with images it god stored as a NumPy array and here 704, 1279 refers to the height and width of the image i.e. number of pixels each column and row respectively and dimension 3 shows the number of channels used to show the image i.e., RGB(Red, Green, and Blue). Once we have our image then we start with the preprocessing of the image for lane detection.



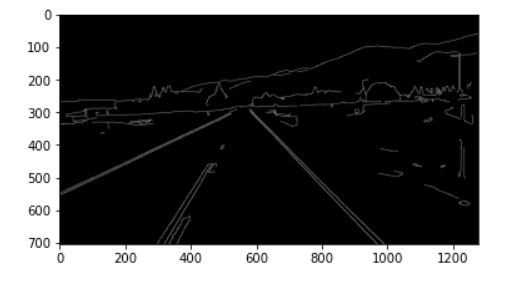
**Implementation:** As we know in the daylight the color shows some different i.e., in white color and we can detect the white lines easily but in the night it is some kind of hard to detect the white line and there are different objects on the road that have different colors, size, etc. and hence we use color selection method. The original image is in Red, Green, and blue color i.e. RGB and hence we will convert the image into grayscale. This process is done as we know the color image can be represented with continuous tone images and conventional computer can represent an only number of shades of gray and gray-scaling can help the computer understand the image easily. And with the help of gray scaling the memory of the image reduces significantly.



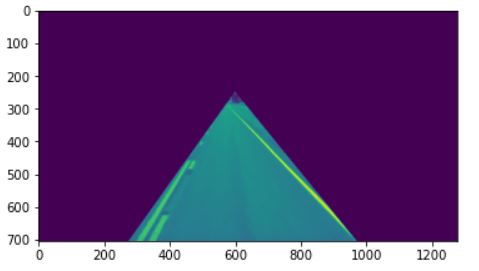
Once the image is in grayscale we apply Gaussian blurring that reduces the image noise and reduce detail by applying a low pass filter on the image. This whole process is done by removing the high-frequency components.

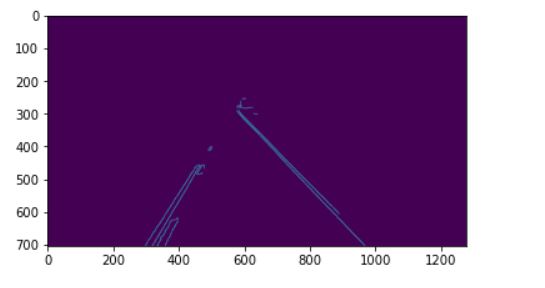


Once our image is ready after applying Gaussian blur we will use the canny edge detection algorithm/method. This is the most widely used method for edge detection. This is a multistage method and goes through each stage. It uses a filter based on the derivative of a Gaussian to compute the intensity of the gradients. The Gaussian reduces the effect of noise present in the image. Then, potential edges are thinned down to 1-pixel curves by removing non-maximum pixels of the gradient magnitude. Finally, edge pixels are kept or removed using hysteresis thresholding on the gradient magnitude.



Once this image preprocessing is done we will find the area of interest i.e. lane and do our final object detection on the image. A region of interest is a place on an image where we want to search for something. With the help of the region of interest, we have a more clear view of the thing that we want to search and the software can easily look for lanes. We do this by creating an ROI that contains all the coordinates for our region of interest and these can be set manually. For both the topmost coordinates, we choose a height of 250px. For the rightmost coordinate, we have an x-coordinate of 1000px. For our rightmost bottom coordinate, we have an x-coordinate of 6000px and a y-coordinate of the height 250 of the image. Next, we have used the cv2.fillPoly function that will fill the region of interest as a white color in the completely black image.





Now we will use hough transformation that will detect any shape that can be represented mathematically. It is a feature extraction technique that can detect simple shapes like circles, lines, etc. in the image and we have used the following parameters that are rho (distance resolution of the accumulator in pixels.), theta (Angle resolution of the accumulator in radians.), threshold (Accumulator threshold parameter. Only those lines are returned that get enough votes), min\_line\_len (Minimum line length. Line segments shorter than that are rejected.) And max\_line\_gap (Maximum allowed the gap between points on the same line to link them.) Now we will create an empty black image and as we have multiple lines so we will average these lines and draw a single line for each lane. To trace a full line from the bottom of the screen to the highest point of our region of interest, we must be able to interpolate the different points returned by our hough transformation function and find a line that minimizes the distance across these points.

**Results:** Here we have implemented the OpenCV for the lane detection and we have used a single image and we can run this same code on the video that will work on each frame of the video for the final output. From our experiment, we have successfully extracted the lane boundaries in real-time. But some problems can’t be solved like sharp curves in the foreground of the image and accurate detection of the lane in case of heavy snow or rain and we would like to improve our model to overcome these problems.

**Discussion and conclusion:** We concluded that we can use this as a prototype and further enhance these skills using deep learning and with the help of frameworks like YOLO and TensorFlow because they are the start of the art. Here we have represented a fast and reliable lane lines detection and tracking technique however we have used well-known algorithms like canny edge detection and Hough transformation. This technique only needs RGB images and we have tested it on a lot of images.

##### References

Anand, S., Buchner, A., “Decision support using data mining”.

London: Financial Times Pitman Publishers,1998.

1. Real Time Lane Detection for Autonomous Vehicles Abdulhakam.AM.Assidiq, Othman O. Khalifa, Md. Rafiqul Islam, Sheroz Khan Department of Electrical & Computer Faculty of Engineering, International Islamic University Malaysia, Jalan Gombak, 53100 K.L., Malaysia. E-mail: [khalifa@iiu.edu.my](mailto:khalifa@iiu.edu.my)
2. DAGMapper: Learning to Map by Discovering Lane Topology Namdar Homayounfar1,2 Wei-Chiu Ma1,3 Justin Liang1 Xinyu Wu 1 Jack Fan 1 Raquel Urtasun1,2 1Uber Advanced Technologies Group 2University of Toronto 3 MIT
3. Road Lane-Lines Detection in Real-Time for Advanced Driving Assistance Systems Wael Farag, Zakaria Saleh,College of Eng. and Technology, American University of the Middle East, Kuwait. Electrical Eng. Dept., Cairo University, Egypt University of Bahrain, Bahrain
4. M. Chen., T. Jochem D. T. Pomerleau, “AURORA: A Vision-Based Roadway Departure Warning System”, in the IEEE Conference on Intelligent Robots and Systems, 1995.
5. B. Ran and H. Xianghong, “Development of A Vision-based Real Time lane Detection and Tracking System for Intelligent Vehicles”, In 79th Annual Meeting of Transportation Research Board, Washington DC, 2000.
6. A. Assidiq, O. Khalifa, M. Islam, S. Khan, “Real time lane detection for autonomous vehicles”, Intern. Conf. on Computer and Communication Eng., May 13-15, 2008 Kuala Lumpur, Malaysia
7. R.O. Duda, and P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures”, Comm. ACM, Vol. 15, pp. 11–15 (January 1972).
8. B. Huval, T. Wang, S. Tandon, J. Kiske, W. Song, J. Pazhayampallil, M. Andriluka, P. Rajpurkar, T. Migimatsu, and R. Chengyue, “An empirical evaluation of deep learning on highway driving”, Computer Science, 2015.
9. Y. Jiang, F. Gao, and G. Xu, “Computer vision-based multiple-lane detection on straight road and in a curve”, In Intern. Conf. on Image Analysis and Signal Processing, pages 114–117, 2010.
10. S. Lee, I. Kweon, J. Kim, J. Yoon, S. Shin, O. Bailo, N. Kim, T.-H. Lee, H. Hong, and S.-H. Han, “Vpgnet: Vanishing point guided network for lane and road marking detection and recognition”, In 2017 IEEE Intern. Conf. on Computer Vision (ICCV), pp. 1965–73, 2017.
11. J. Li, X. Mei, D. Prokhorov, and D. Tao, “Deep neural network for structural prediction and lane detection in traffic scene”, IEEE Transactions on Neural Networks and Learning Systems, 28(3):690– 703, 2016.