**Digital Image Processing**

**CEP**



Group Members:

Wasif Sami Waheed (01-132162-042)

Riz Khan (01-132162-018)

Mughira Asad (01-132162-028)

Class: BCE-7

Department of Computer Engineering

Bahria University­

Islamabad Campus

**CEP Report**

**Introduction**

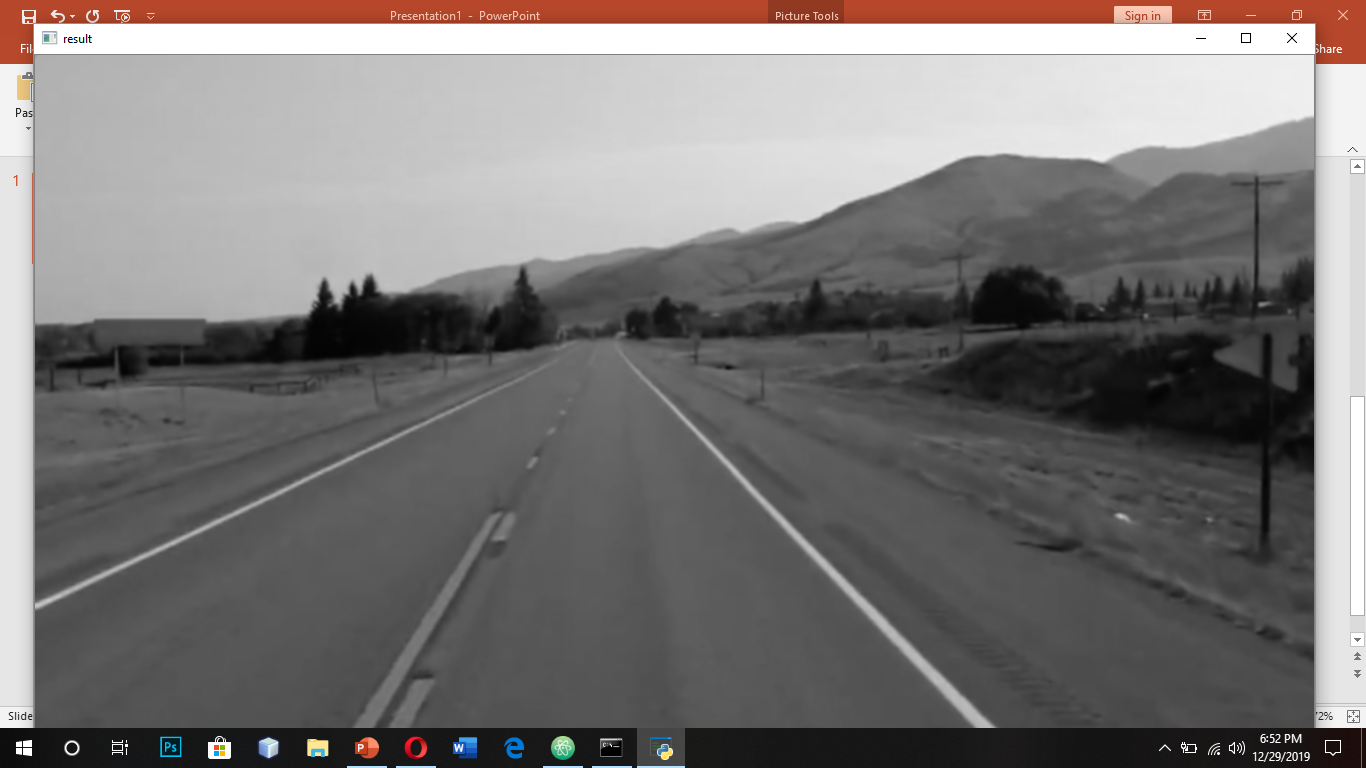
When we drive, we use our eyes to decide where to go. The lines on the road that show us where the lanes are act as our constant reference for where to steer the vehicle. Naturally, one of the first things we would like to do in developing a self-driving car is to automatically detect lane lines using an algorithm. Identifying lanes of the road is very common task that human driver performs. This is important to keep the vehicle in the constraints of the lane. This is also very critical task for an autonomous vehicle to perform. And very simple Lane Detection pipeline is possible with simple Computer Vision techniques. Self-driving cars, have rapidly become one of the most transformative technologies to emerge. They are continuously driving our society forward, and creating new opportunities in the mobility sector.

In this project you will detect lane lines in images using Python and OpenCV. OpenCV means "Open-Source Computer Vision", which is a package that has many useful tools for analyzing images.

**Lane Detection Steps**

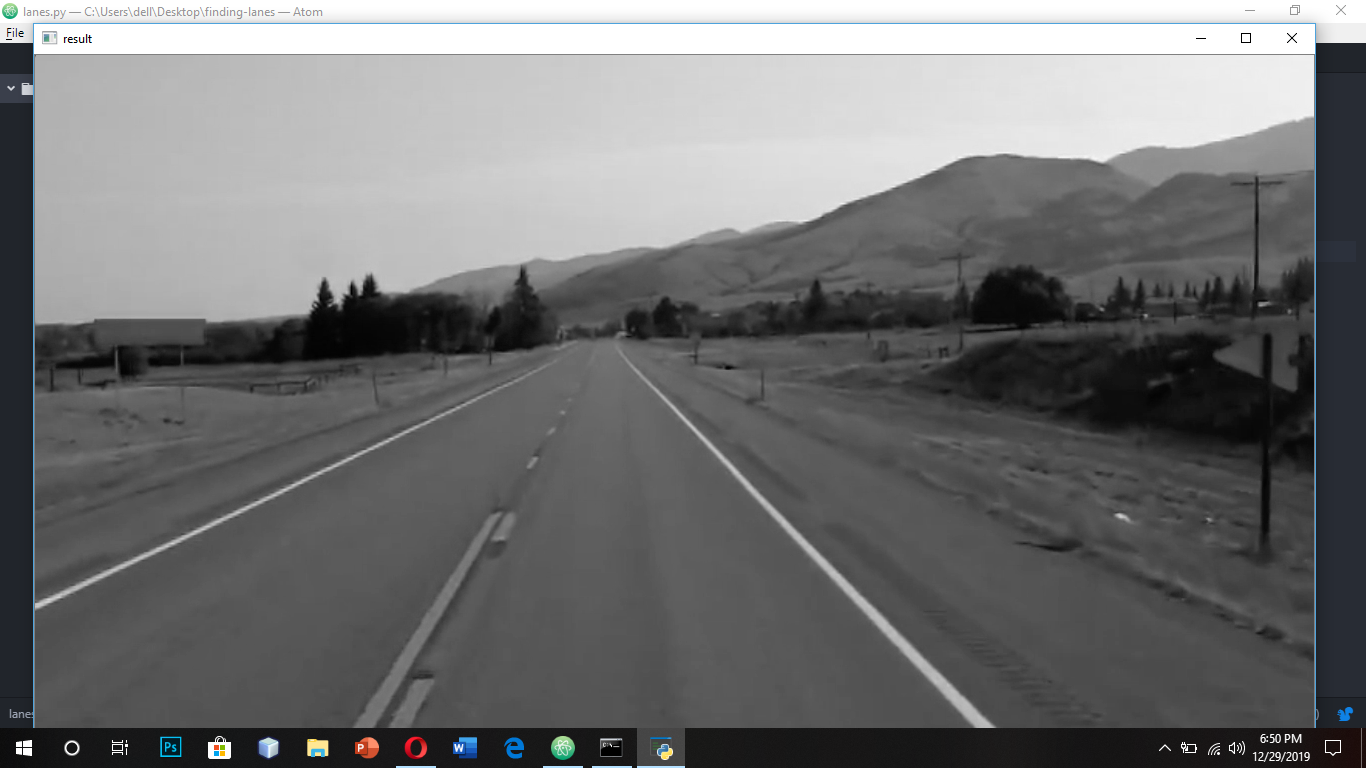
**Conversion to Gray Scale**

Converting original image to grayscale has its benefit. We have to find yellow and white lanes, and converting original image to grayscale increases the contrast of lanes with respect to road.



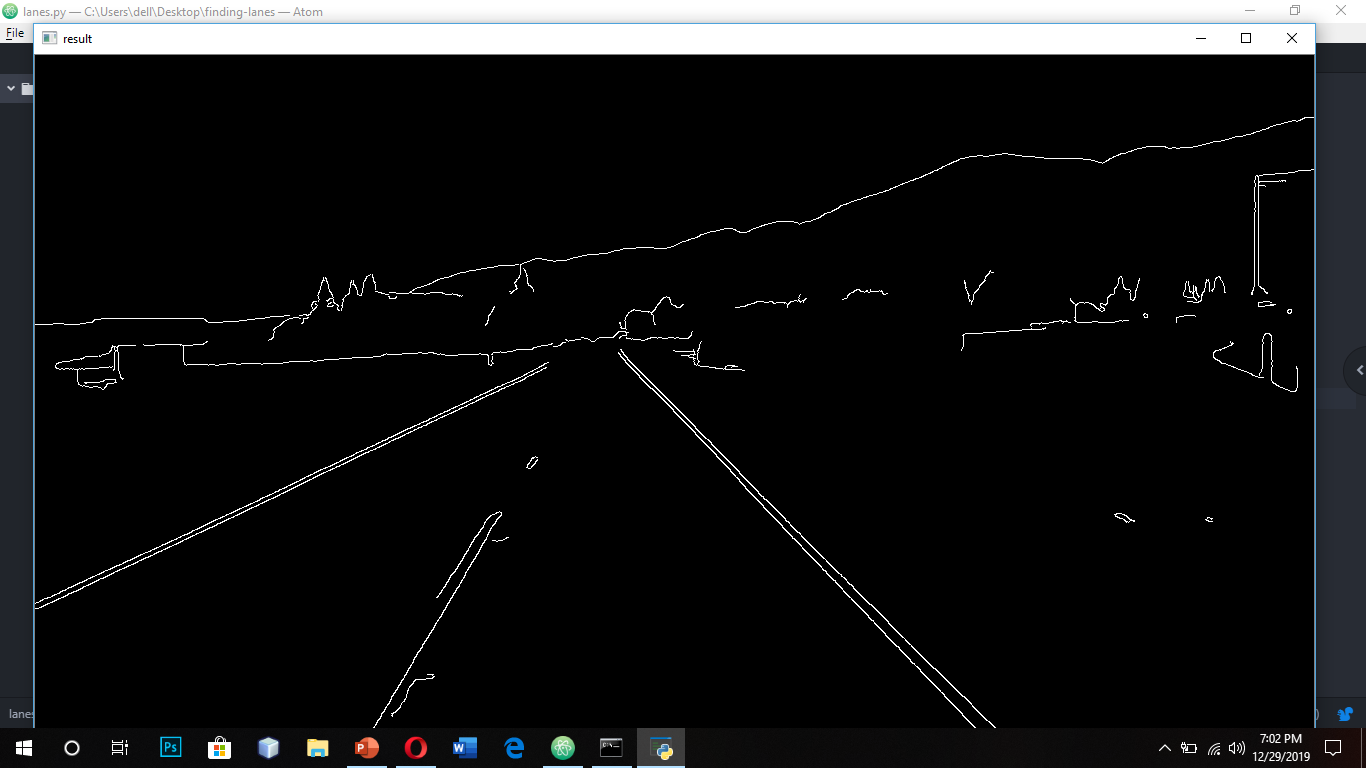
**Gaussian Blur**

Gaussian blur (Gaussian smoothing) is pre-processing step used to reduce the noise from image (or to smooth the image). We use this pre-processing step to remove many detected edges and only keep the most prominent edges from the image.



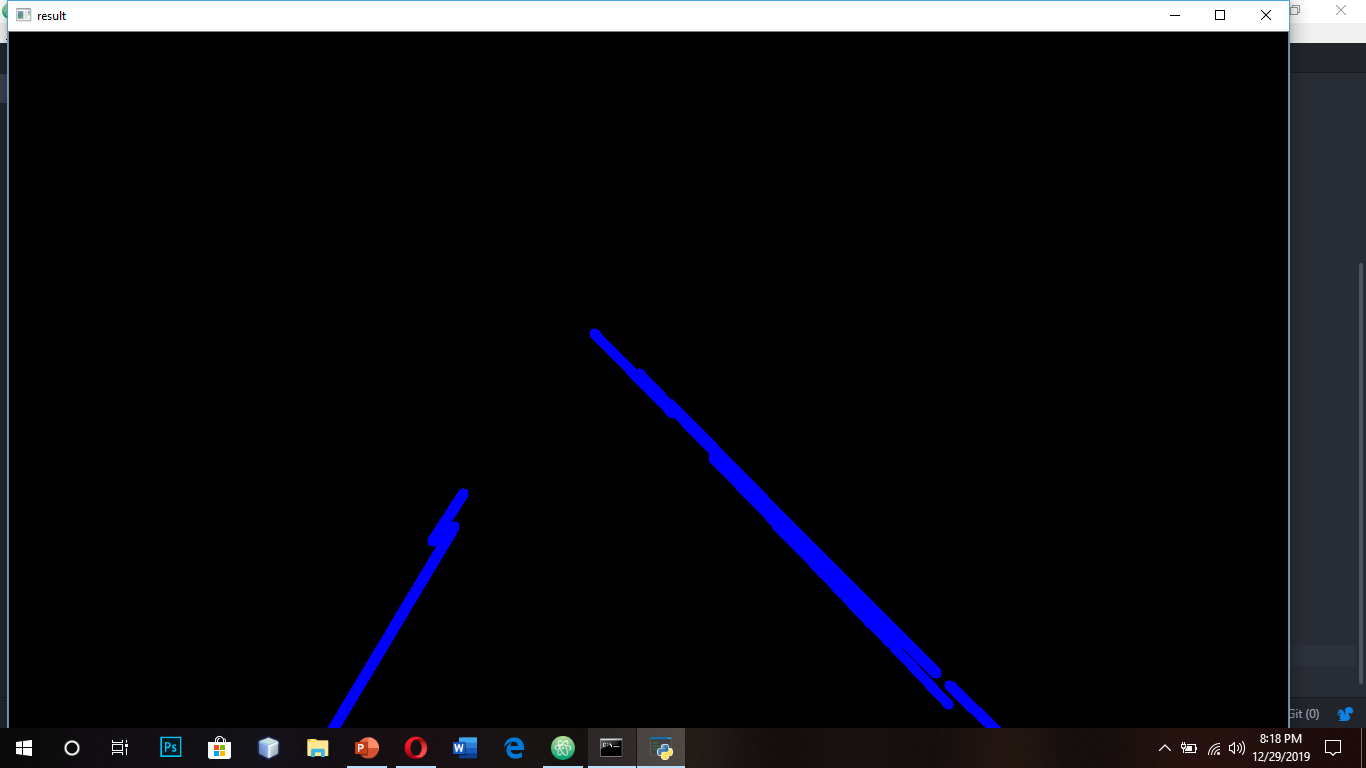
**Edge Detection**

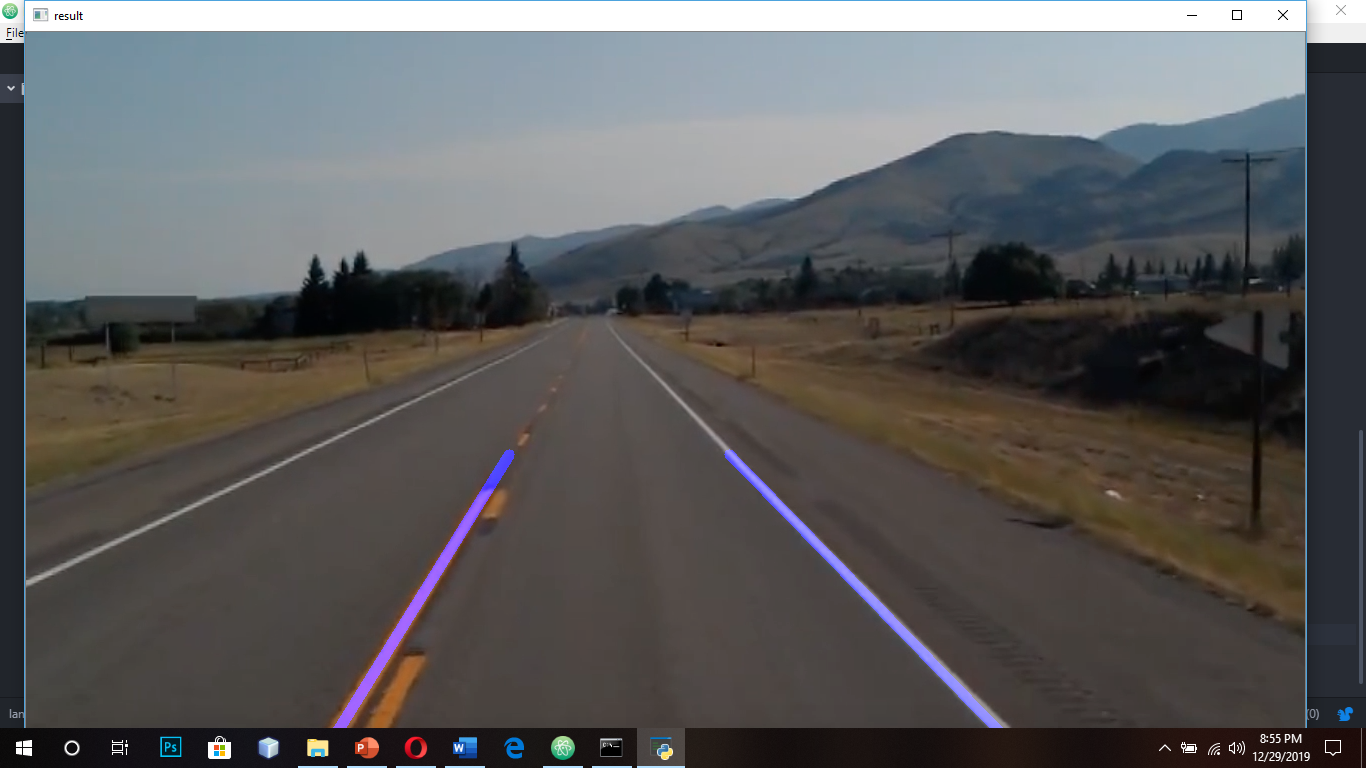
Now we apply Canny edge detection to these Gaussian blurred images. Canny Edge Detection is algorithm that detects edges based on gradient change. Not that the first step of Canny Edge detection is image smoothing with default kernel size 5, we still apply explicit Gaussian blur in previous step.



**Hough Transform Processing**:

Hough Transform is the technique to find out lines by identifying all points on the line. This is done by representing a line as point. And points are represented as lines/sinusoidal (depending on Cartesian / Polar co-ordinate system). If multiple lines/sinusoidal pass through the point, we can deduce that these points lie on the same line.





**Conclusion**

Hough Lines based on straight lines do not work good for curved road/lane. There are many trial-and-errors to get hyper parameters correct. Also, Region of Interest assumes that camera stays at same location and lanes are flat. So, there is “guess” work or hard coding involved in deciding polygon vertices. In general, there are many roads which might not be with lane markings where this won’t work. This was the end of our project and we were successfully able to find lane lines in videos as well.



**Future Improvements**

* Instead of line, it would be beneficial to use higher degree curve that will be useful on curved road.
* Even when we used information from previous frames, just averaging lanes might not be very good strategy. may be weight average or some form of priority value might work.