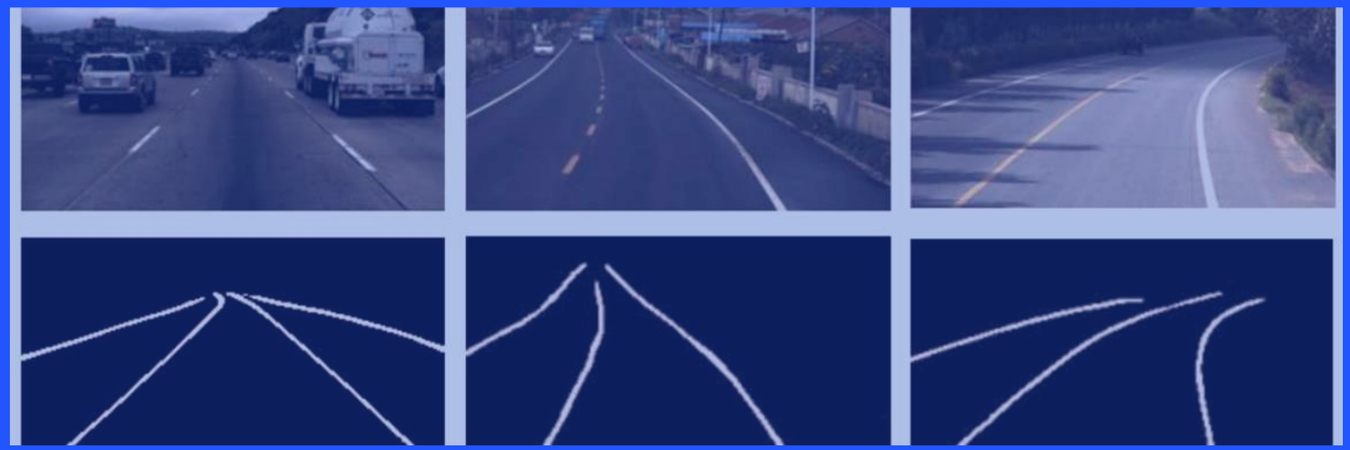
Real Time Lane Detection

Akash Tyagi - 18BCE10023
Prayag Bhatnagar - 18BCE10196
Rhythm Sanghi - 18BCE10216
Shreyaa Dani - 18BCE10257
Anuj Somani - 18BCG10019
Hritvik Semwal - 18BCG10048
Kartik Gotmare - 18BCG10054
Devesh Banwani - 18BCY10030

Introduction

Advanced Driving Assistant Systems, intelligent and autonomous vehicles are promising solutions to enhance road safety, traffic issues and passengers' comfort. Such applications require advanced computer vision algorithms that demand powerful computers with high-speed processing capabilities. Keeping intelligent vehicles on the road until its destination, in some cases, remains a great challenge, particularly when driving at high speeds.

One of the many steps involved during the training of an autonomous driving car is lane detection, which is the preliminary step.



INFORMATION GATHERING

Brief

We live in an era of digitalization.
Self driving cars are the future of automobile industry. We need to create an algorithm that eases the task of the manufacturers and increases reliability of humans on self-driving cars

Problem Statement

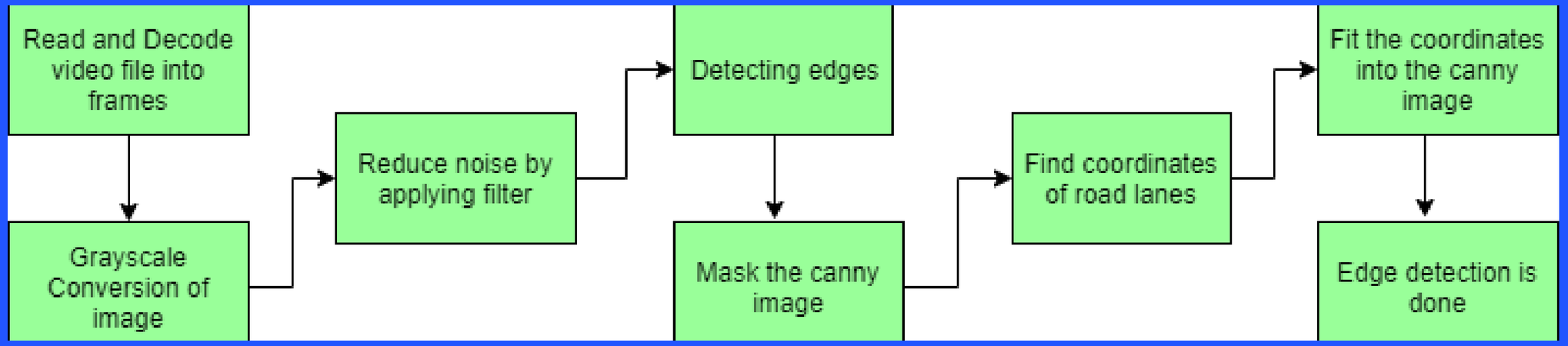
Self-driving cars need an intelligent and reliable system which can transport people to their destination safely. One of the foremost task is to create an algorithm which detects lanes on the road.

Existing Works

Tesla has been working on the self-driving model for cars. They have tested some designs and are still in development phase and have not still developed a fully automated self-driving car.

Solution

Automated lane detection is the foremost step in developing a self-driving car.
Our approach is to create and algorithm that processes the image taken from a camera mounted on the car.



Capturing and decoding of the image file

Grayscale conversion of image

Reduce noise

Noise can create false edges, therefore before going further, it's imperative to perform image smoothening.

Canny Edge Detector

Traces the edges of an image with large change in intensity and hence helps detects us the lanes or edges of the road.

Region of Interest

Hough Line Transform

Flow of Code


01
rdimg = ocv_read('Road3.png')
cannyimg<- canny_edge_detector(rdimg)
02
03
croppedimg<- region_of_interest(cannyimg)
df <- hough_line(croppedimg,ntheta=50,data.frame=TRUE)
04
05
source_python("RoadLaneDetection.py")

LIBRARIES USED

- library(opencv)
- library(ggplot2)
- library(magrittr)
- library(imager)
- library(reticulate)

About Dataset

The dataset for the project we have chosen is the video of a plain road having lanes.

Capturing and Decoding

CODE SNIPPET

```
ocv_write(blur, 'blur.jpg')
img<- load.image('blur.jpg')
```

Input Image

Grayscale Conversion

CODE SNIPPET

```
gray_image = ocv_grayscale(image)
```

Output Image

Reduce Noise

CODE SNIPPET

```
blur = ocv_blur(gray_image,5)
```

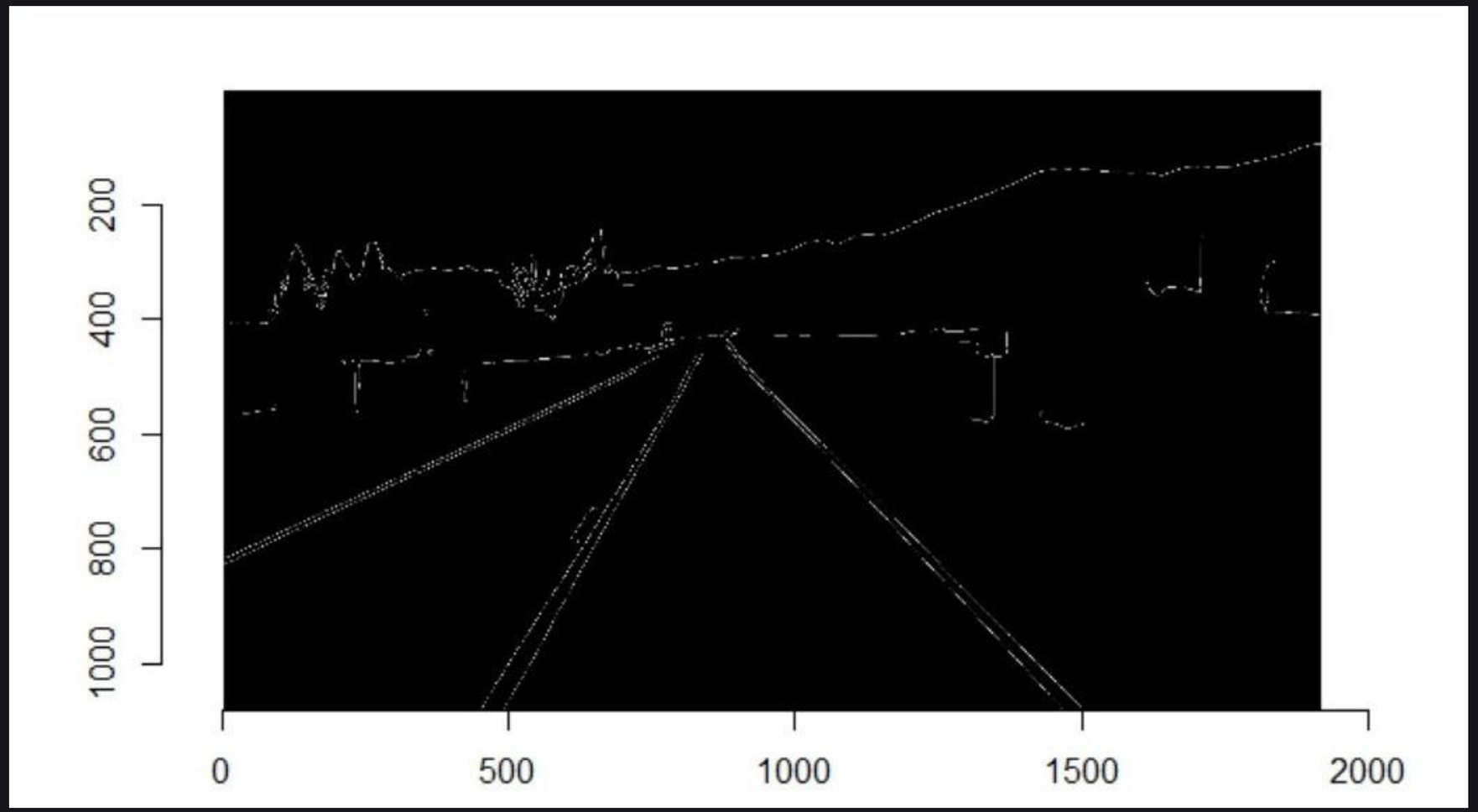
Output Image

Canny Edge Detector

CODE SNIPPET

```
canny = cannyEdges(img)
plot(canny)
```

Output Image

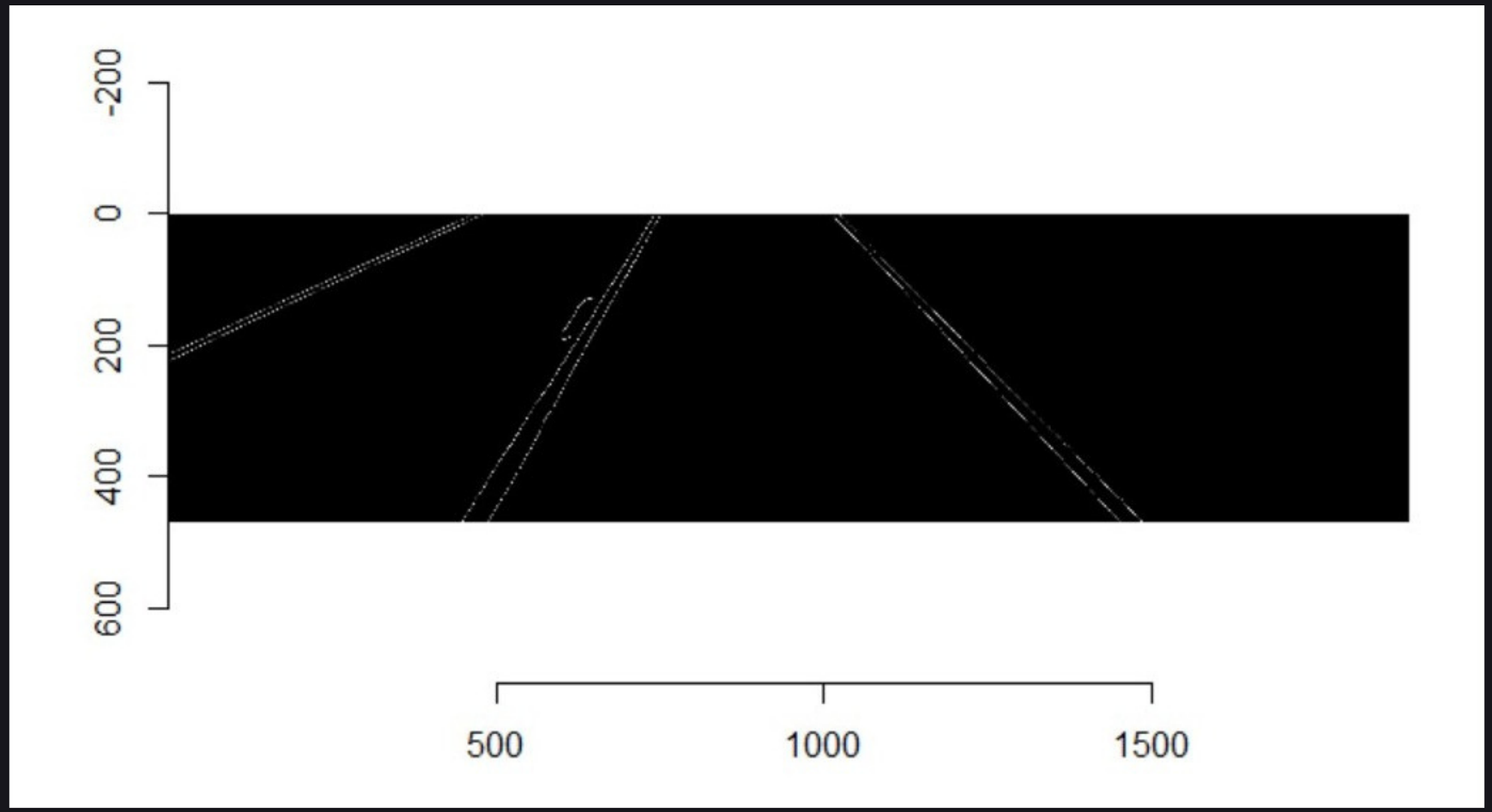


Region of Interest

CODE SNIPPET

```
cropped<-imsub(image,x %inr% c(10,1900),y %inr% c(600,1070))
```

Output Image

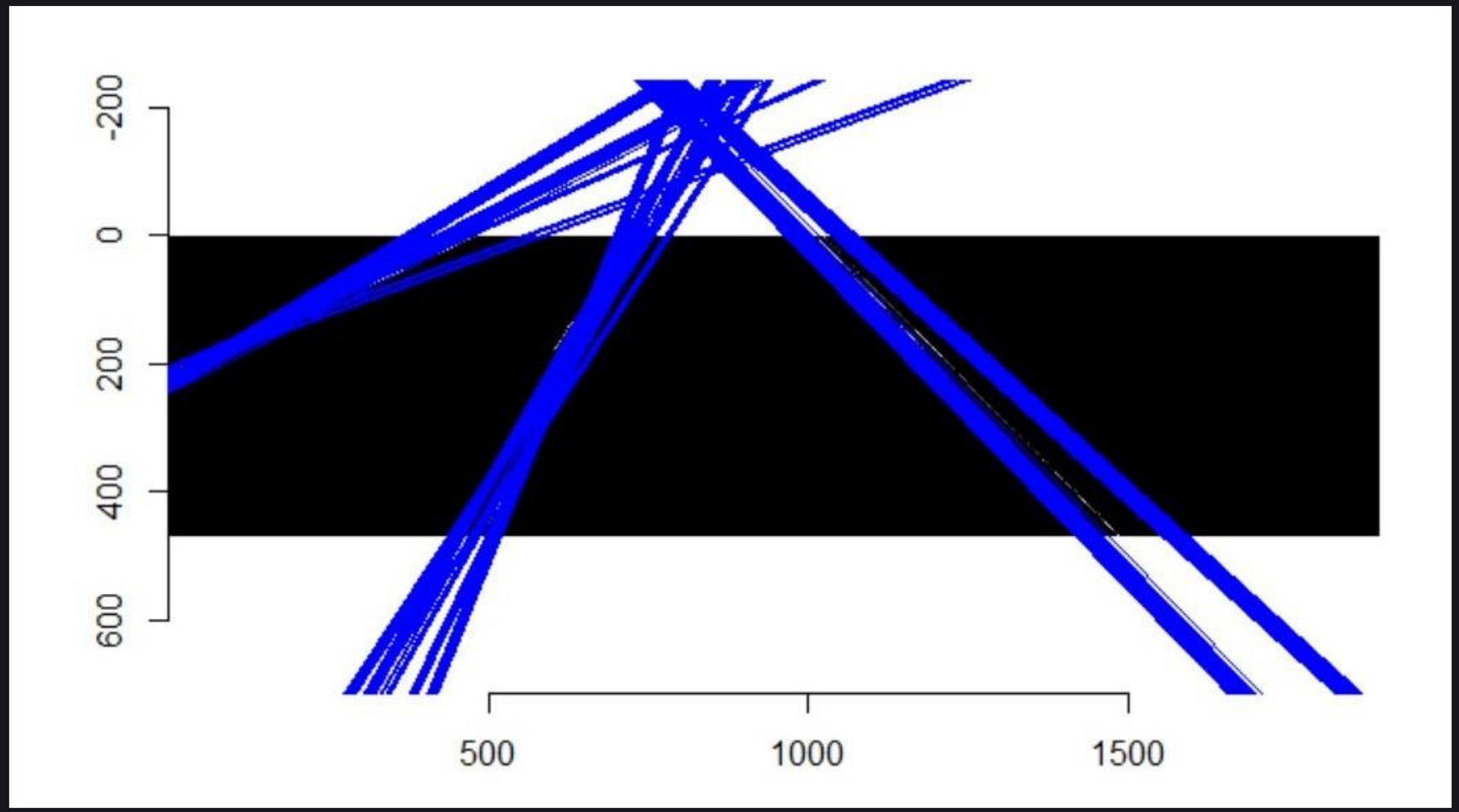


Hough Lines Transform

CODE SNIPPET

```
df <- hough_line(croppedimg,ntheta=50,data.frame=TRUE)
```

Output Image



Python Extension

CODE SNIPPET

average_slope_intercept(image,lines)

Python Extension

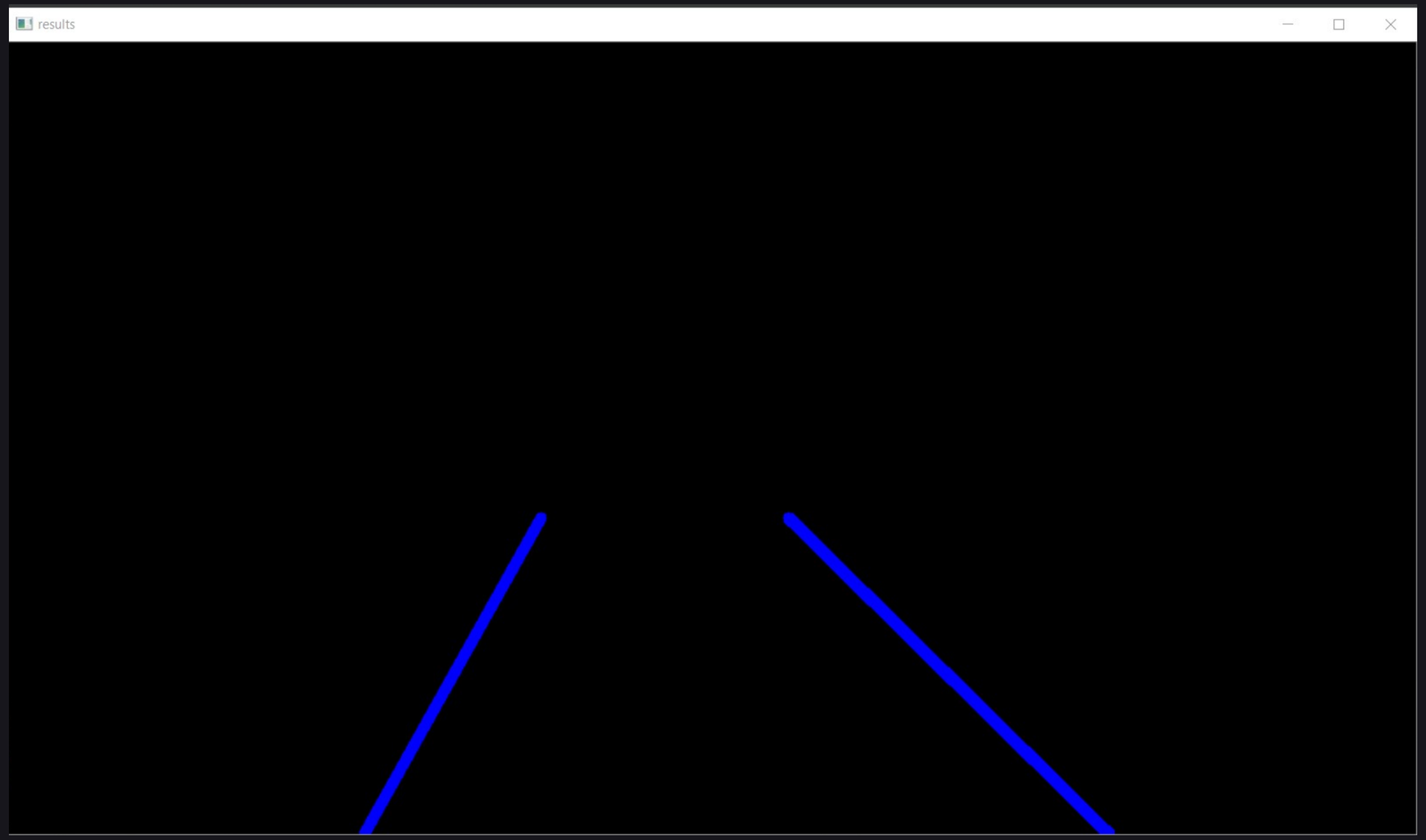
CODE SNIPPET

create_coordinates(image,line_parameters)
display_lines(image,lines)

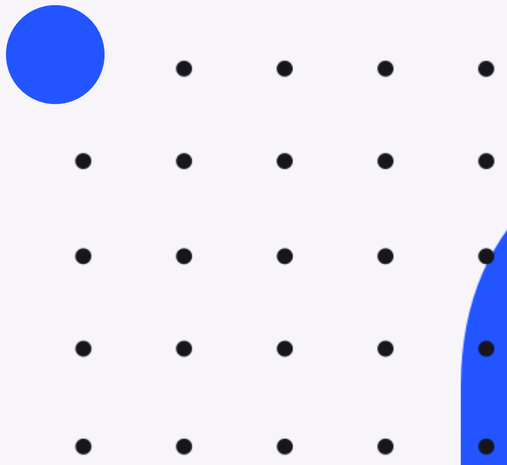
Python Extension

DRIVER CODE

Output Image



“AUTOMATED OR SELF-DRIVING VEHICLES ARE ABOUT TO CHANGE THE WAY WE TRAVEL AND CONNECT WITH ONE ANOTHER.”



ANY QUESTIONS?

THANK YOU