| **🔧 What it does** | **🧠 Why it’s done** | **📦 General code idea** |
| --- | --- | --- |
| Grayscale | Simplify color data | cv2.cvtColor(...) |
| Gaussian Blur | Remove small noise(make it easy for canny) | cv2.GaussianBlur(...) |
| Canny Edge | Find strong edges | cv2.Canny(...) |
| ROI Masking | Focus on the road | cv2.fillPoly(...) + bitwise\_and |
| Hough Transform | Detect lines | cv2.HoughLinesP(...) |
| Draw Lines | Show result | cv2.line(...) |

**You: [ Looking ➡️ screen ] Camera’s View**

**Left lane (/): slope negative**

**Right lane (\): slope positive**

Image:

0px ----------> width (X)

|

|

v height (Y)

Left Line Lane Center Right Line

\ | /

\ | /

\ | /

\ | /

\ | /

\\_\_\_\_\_\_\_\_\_\_\_\_\_\_|\_\_\_\_\_\_\_\_\_\_\_\_\_/

<- car bottom, camera view

Car center (width/2) is compared to Lane center ➔ Steering Angle

**Steering is just angle between where you are and where you need to be!**  
**Big offset** → **Big turn** 🔄  
**Small offset** → **Small correction** ➡️

**✨ SUPER MINI SUMMARY ✨**

| **What?** | **Quick Idea** |
| --- | --- |
| left\_line, right\_line | Average all detected lines |
| Slopes | Left: negative; Right: positive |
| mid\_x | Middle between left and right lane bottom |
| width // 2 | Camera's center = Car's center |