Authors: Hrishikesh Tawde,
Kapil Rawal.

UML CLASS DIAGRAM

LaneLineDetectionSystem

- camera_ :CameraModule
- processimage_ :ImageProcessingModule
- driveheading_ :DriveHeadingModule

+ detectLaneLineforImage(pathToImage:string): void
+ detectLaneLineforVideo(pathToVideo:string): void

ImageProcessingModule

- imagePoint1_: int = 0
- imagePoint2_: int = 0
- imagePoint3_ : int = 0
- imagePoint4_: int = 0

+ applyGrayScale(cv::Mat colorImage): cv::Mat
+ applyGaussianBlur(cv::Mat nonBlurImage,int kernel): cv::Mat
+ applyCanny(cv::Mat nonCannyImage, double lowThreshold, double highThreshold,int kernel): cv::Mat
+ getRegionOfInterest(cv::Mat cannyImage): cv::Mat
+ getDrawLines(cv::Mat zerosImage, std::vector<cv::Vec4i> lines, cv::Scalar color, int thickness): cv::Mat
+ getHoughLines(cv::Mat maskedGrayImage, std::vector<cv::Vec4i> lines,int rho,double theta,int threshold, int minLineLen,int maxLineGap): std::vector<cv::Vec4i>
+ getWeightedImage(cv::Mat newImage, cv::Mat image): cv::Mat
+ generateImageWithText(cv::Mat image, float value): cv::Mat
+ getImagePoint1():int
+ getImagePoint2():int
+ getImagePoint3():int
+ getImagePoint4():int

CameraModule

- image_:cv::Mat
- video_:vector<cv::Mat>

+ getImage():cv::Mat
+ getVideo(int frameNumber):cv::Mat
+ setImage(pathToImage:string):int
+ setVideo(pathToVideo:string):int

DriveHeadingModule

- centreLaneValue_: int = 0

+ setCentreILaneValue(cv::Mat image):void
+ getCentreLaneValue():int
+ findDriveHeading(int point1, int point2, int point3, int point4): float