## Traffic Lane Detection

## Lane

-polvOrder : int -colour : std::string

-polyCoeff : std::vector<float> -startCoordinates : cv::Point -averagingCenter : std::vector<int>

-averagingCount : int -currentAveragingIndex : int

-status: bool

l+Lane()

+~Lane()

+Lane(int , std::string, int ) +getStableCenter(int): int

+setStartCoordinate(cv::Point): void +getStartCoordinate(): cv::Point

+setStatus(bool) : void +getStatus(): bool +setPolyOrder(int): void +getPolyOrder(): int

+setPolyCoeff(cv::Mat) : void +getPolyCoeff(): std::vector<float>

## LaneDetectionModule

-vellowMinx: cv:scalar -yellowMax: cv:scalar -grayscaleMin:int -grayscaleMax:int -videoName :std::sring

+LaneDetectionModule() +~LaneDetectionModule()

+undistortImage(cv::Mat&,cv::Mat&): void +thresholdImageY(cv:Mat&,cv::Mat&): void +thtesholdImageW(cv:Mat&,cv::Mat&): void

+extractROI(cv:Mat&,cv::Mat&): void

+transformPerspective(const cv::Mat& , cv::Mat& , cv::Mat& , cv::Mat& , cv::Mat& ) : void

+extractLanes(const cv::Mat&, cv::Mat&, Lane&, Lane&, int): void

+fitPoly(const std::vector<cv::Point>&, cv::Mat&, int) : void

+getDriveHeading(Lane&, Lane&, std::string&,SteerDrive&): double

+computeGearRatio(SteerDrive&): float

+displayOutput(const cv::Mat&, cv::Mat&, Lane&, Lane&, cv::Mat,, SteerDrive&): void

+detectLane(std::string): bool +getYellowMax(): cv::Scalar +getYellowMin(): cv::Scalar +setYellowMax(cv::Scalar): void

+setYellowMin(cv::Scalar): void +setGrayScaleMin(int): void +setGrayScaleMax(int): void

+getGrayScaleMin(): int +getGrayScaleMax(): int

## SteerDrive

-MechAdvantage : float -CarModel : std::string -DifferentialRatio: float

-Status : bool

+SteerDrive() +~SteerDrive()

+getCarModel(): std::string +setCarModel(std::string): void +getDifferentialRatio(): float +setDifferentialRatio(float): void +getMechAdvantage(): float +setMechAdvantage(float): void

+setStatus(bool): void +virtual getStatus(): bool

+virtual ResultantGearBox(float,float): float