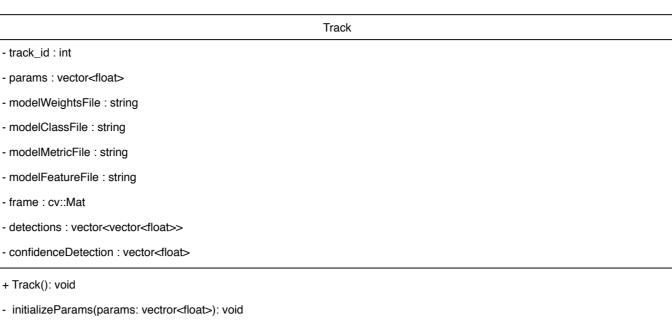
## Detection - confThreshold : float - nmsThreshold : float - inpWidth : float - inpHeight : float - modelWeightsFile : string - modelConfigFile : string - modelClassFile : string - frame : cv::Mat - frame\_id : int - detectionsDict : map<string, vector<vector<float>>> - detections : vector<vector<float>> - confidenceDetection : vector<float> + Detection(): void initialize Params (confThreshold: float, nmsThreshold: float, inpWidth: float, inpHeight: float): void- load Model and Label Classes (model Weights File: string, model Config File: string, model Class File: string): void and the configuration of the config- readInput(): void - processFrameforHuman(frame: cv::Mat): Bool - drawRedBoundingBox(coordinates: vector<float>): void - storeFoundCoordinates(frame\_id: int, detections: vector<vector<float>>,confidenceDetection: vector<float>): void + getDetections(): vector<vector<float>>> + getConfidence(): vector<float> + getAllDetectionsList(): map<string, vector<vector<float>>> + ~Detection(): void

DataLoader	
- method : string	
- frame : cv::Mat	
- sequenceOfFrames : vector <cv::mat></cv::mat>	
- path: string	
+ DataLoader(): void	
+ setInputStreamMethod(path: string) : void	
+ getInputStreamMethod(): string	
+ getFrame(): cv::Mat	
+ ~DataLoader(): void	



- initializeFeatureExtractor(modelWeightsFile: string, modelClassFile: string, modelMetricFile: string, modelFeatureFile: string): void
- runTrackerAlgorithm(frame: cv::Mat, detections: vector<vector<float>>>, confidenceDetection: vector<float>): void
- drawGreenBoundingBox(coordinates: vector<float>): void
- + getCoordinatesInCameraFrame(coordinates: vector<float>): vector<float>
- + ~Track(): void