**SYNOPSIS**

1. **TITLE OF PROJECT**

Road Lane Detection

1. **INTRODUCTION**

Lane Line detection is a critical component for self-driving cars and also for computer vision in general. This concept is used to describe the path for self-driving cars and to avoid the risk of getting in another lane. The traffic safety becomes more and more convincing with the increasing urban traffic. Exiting the lane without following proper rules is the root cause of most of the accidents on the avenues. Most of these are result of the interrupted and lethargic attitude of the driver. Lane discipline is crucial to road safety for drivers and pedestrians alike. The system has an objective to identify the lane marks. Its intent is to obtain secure environment and improved traffic surroundings. The functions of the proposed system can range from displaying road line positions to the driving person on any exterior display, to more convoluted applications like detecting switching of the lanes in the near future so that one can prevent concussions caused on the highways.

1. **OBJECTIVE**

1. To improve the traffic security

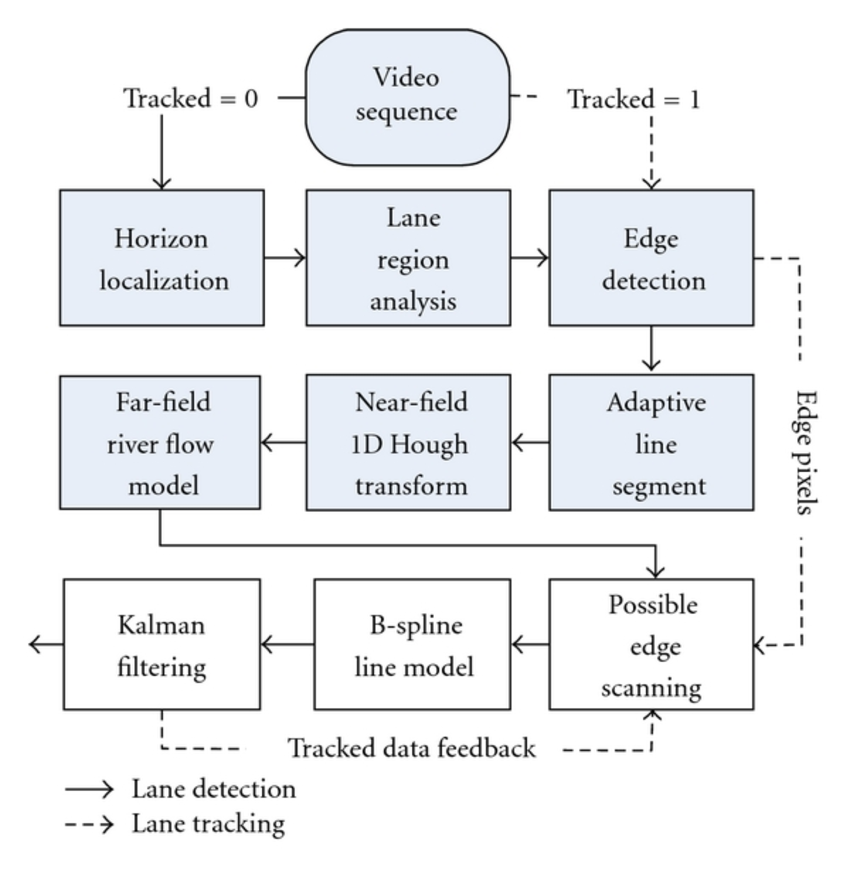
2. To make vehicle intelligent

3. For the improvement of the driving safety

1. **FUTURE SCOPE**
2. Advancement in this field can make driving possible for anyone
3. It can decrease Road Accidents
4. It is helpful for long hourly driving
5. **SRS REQUIREMENTS**

* 1. **Functional Modules**

Block Diagram

****

* 1. **Non-Functional Modules:**
     1. **Various Requirement:**
* **Security-:** The owner of the car will able to access the system only
* **Accuracy -:** The accuracy will be managed for sure
* **Reliability-:** The system will perform consistently its intended function

1. **TOOLS AND PLATFORMS**
   1. **Hardware Components :**

* Hard Disk: - 500 GB
* Processor: INTEL CORE i5
* RAM: - 4 GB
* Front end: - Kivy, Python, Machine Learning ,Open CV(Computer Vision)

**6.2** **Software Specifications :**

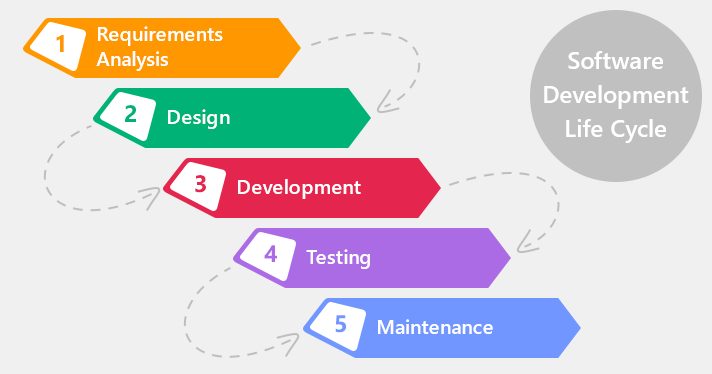
* Operating System: - Android version Lollipop (5.1) or above, IOS 12.3 Or above.
* Android Development Toolkit (ADT)

1. **METHODOLOGY:**

**7.1. STAGE OF ITERATIVE MODEL:**

* Use SDLC (Software Development Life Cycle)

**Phases of Life Cycle Model:**

* **Planning and requirement:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
* **Analysis and design:** The requirement specifications from first phase are studied in this phase and system design is prepared. It helps in specifying hardware and system requirements and also helps in defining overall system architecture.
* **Implementation:** With inputs from system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
* **Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* **Evaluation:** This will evaluate the outcome of the processing whether the application is running successfully or not.

**7.2. REASON FOR CHOOSING WATERFALL MODEL:**

* It is very simple to understand and use.
* As we are new to this it is very difficult to analyse all requirements at once. Building the whole application at once is not possible for us.
* So, by dividing the Project into small parts and implementing them makes easy for us to make this application.

**7..3 ADVANTAGES OF THIS MODEL:**

* This model is simple and easy to understand and use.
* It is easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
* In this model phases are processed and completed one at a time.