**Chapter 3**

**Software Requirement Specification**

This chapter outlines and describes the software required for implementation of this project. Along with the functions involved, it is necessary to choose the right software support that can provide libraries that help facilitate the implementation of the various modules incorporated in the project [27]. Also, the characteristics and user interaction interface is instrumental in facilitating the ease of use of the system. Moreover, the software must be chosen such that it is platform independent and runs seamlessly across all platforms.

**3.1 Overall Description**

This project is developed in the python language version 2.7 and is also scalable to version 3.4 and above. It makes use of the open CV 2 series library that contains significant changes from the open CV library. All the functions are self contained and are not dependent upon the platform using which the system was developed.

It consists of four modules. The background modeling phase, the foreground extraction (background subtraction) phase, the clustering and filtering phase and finally violation detection followed by plate number digit recognition. The filtered information extracted is then retrieved from a simulated database. All of the functions have been implemented using python standard libraries and No additional software other than python 2.7 and above and open CV 2 is required to implement the system.

**3.1.1 Product Perspective**

The system is mainly used to achieve red signal jump detection following some computationally efficient pre-processing techniques. The simplicity of approach and accuracy of results make it a tantalizing prospect for future work. The system integrates between all the modules seamlessly and produces good results in various tested scenarios [28].

The system is a stand-alone application that requires only that the standard python and openCV libraries be installed on the platform running the software. The Graphical User Interface, to avoid third party intervention is also modeled using openCV’s drawing functions and basic terminal commands. The GUI is simple in design and is extremely easy to use for a lay-person that needs the system to meet his own ends.

**3.1.2 Product Functions**

The application accepts recorded video of MPEG-4 or AVI formats, models the background based on the scene, performs background subtraction of the video frames against the modeled background, Color quantizes the resulting frames using K means Clustering (K equals 2), uses a mean squared error similarity checker to filter out blank frames and compiles the remaining frames into a new video. The new video is processed to detect violation.

Therefore the system models the background, performs background subtraction, extracts full colored and foreground key frames, simulates signal time, detects signal jump and even recognizes number plate digits using a trained classifier. In addition to this it also filters and retrieves results from a simulated database. The system performs all these operations both in a modular and an integrated fashion depending on user interaction with the Graphical User Interface.

**3.1.3 User Characteristics**

It is important to educate oneself about the type of users that will typically require such a system for their specific needs. Knowledge about the characteristics of the users can help design a easy to use, efficient system and also, help design user specific Graphical User Interface components. The system is intended to be used as per user specification as there is very little broad scope of usage of such an intricately designed application. Therefore, knowledge about the user using the system is required to tailor the system to their needs.

The system is designed for two kinds of users:

* Normal Users such as researches and academicians that want to incorporate the facilities provided by the system in to a larger scope of application. These can include all kinds of traffic scene processing systems such as face recognition of drivers, Number plate recognition, CCTV camera monitoring of the detected vehicle, to name a few.
* Advanced Users such as traffic administrators that want to apply the system in day to day usage for detecting traffic violations in real time. Although the system is not quite ready for this type of usage, minor modifications can render it suitable for efficient usage in real life red signal jump detection.

**3.1.4 General Constraints**

The performance of the system is subject to the following constraints:

* The level of vehicle occlusion in the video has to be very limited.
* The proximity of shooting of traffic footage needs to sufficiently high. Small and blurry videos of vehicles at a distance will not work with the system.
* The system camera must be placed laterally by the side of the signal post to capture video feed perpendicular to the post.
* The system camera must be placed in line with the post, as the edge of the frame in the video is assumed to be the real life position of the signal post displaced by an appropriate distance.

**3.1.5 Assumptions and Dependencies**

The assumptions made by the system are that the camera is placed in line with the post for the edge of the frame to qualify as the real life position of the signal post. Also, the system can interface electrically with a high definition camera placed on top of the signal post perpendicular to the primary camera in order to capture HD still images of the vehicle responsible for signal jump, once it crosses the red signal. This is done to accurately extract plate information. Finally, it is assumed that the system runs only when the light in the signal is red as it makes no sense to perform all the computations for violation detection when the signal is green.

**3.2 Specific Requirements**

This section describes the functional requirements of the system [29]. This outlines all of the working functionality that the system is intended to provide. The minimum Hardware and software configuration required to run the system without lag is also described. Also, performance benchmarks as per which the system is expected to comply is detailed. Finally, the interfaces required to run the system and the non functional requirements is described [30].

**3.2.1 Functional Requirements**

Functional requirements describe the function of each of the individual units or modules that the system is composed of. Also, it describes the functionality that the integrated system as a whole, is expected to provide [29].

The following functional requirements are to be met by the system:

* Background modeling of the recorded scene
* Identification of non-static background objects, to be incorporated into the modeled background.
* Frame differencing of the video frames against the modeled background or background subtraction.
* Elimination of all background noise in the video.
* Estimation of the time required to run the system or estimation of the simulated red signal time.
* Reducing the number of frames required for post processing, in this case violation detection caused as a result of the frame rate of the recorded video.
* Detailing the foreground modeling of the scene to accurately depict the features and characteristics of the moving objects.
* Detection of jumping of a red signal by a moving vehicle in the scene, both four and two wheelers.
* Capturing the instance of the video, when the violation occurs.
* Time stamping the instance of the video so captured, for user understanding and information.
* Capturing the image of the vehicle performing the violation act and extracting the number plate.
* Recognizing the numbers of the number plate and retrieving information about the vehicle performing the violation, from the database, by indexing the records by the extracted numbers.
* It must be able to handle errors in input given by the user and provide necessary output messages as corrections.
* It must provide a simple, yet interactive Graphical User Interface that is intuitive and works efficiently when used by a lay man.

**3.2.2 Performance Requirement**

The system must operate smoothly when under use by a single user, either on the command prompt or Graphical User Interface. Simultaneous working on both must also yield the same result. It must work the same for all acceptable formats of input video recordings and provide consistent and reliable outputs. The time taken for execution must be commensurate with the size of the input being processed. The results must comply with the standards and must be acceptable to every specific type of user.

**3.2.3 Software Requirements**

The software required for development of the automatic parallelization tool are:

Operating System : Windows, Linux or Mac

IDE/Workbench : vim editor or pyCharm

Programming language : Python2.7 or higher

Other Required Software : OpenCV version 2.0 or higher.

**3.2.4 Hardware Requirements**

The hardware required for the project are (minimum requirements):

Processor : Intel Core 2 duo or higher

Processor Speed : 1.2 GHz or higher

RAM : 1 GB or more

Hard Disk : 100MB of free space

**3.2.5 Design Constraints**

The system requires that the view at which the video is recorded is perpendicular to the front face of the camera. Also, all the standard python and openCV support software must be installed on the computer that needs to execute the system.

**3.2.6 Interfaces**

The system provides a terminal or command line interface for execution of the program both as individual modules and as an integrated unit. It also interfaces effectively with files and databases necessary for storage and retrieval of information during user interaction. It also provides an intuitive Graphical User Interface for users that have limited knowledge of the command line interface.

**3.2.7 Non Functional Requirement**

These requirements describe integrity and security requirements of the system necessary to ensure accountability towards the end user and overall efficiency of the system [30].

* **Reliability**: The system must not fail or crash under any circumstance of use. It must provide core functionality for all inputs subject to the requirements in terms of format and other input constraints. The output obtained must be consistent and reliable throughout.
* **Availability**: The system must be available for the end user, availability being defined by the user’s jurisdiction and discretion. In other words, the system must never be unavailable for use when the user needs it.
* **Security**: The system must ensure secure functioning of all the individual units and integrity of information being stored and retrieved from virtual and intangible sources such as files and databases.
* **Maintainability**: The system must allow for bug fixes, if identified by the user.
* **Portability**: The system must be executable in different operating system environments.

**3.3 Summary**

All the requirements both functional and non functional of the system have been described in this chapter. It also provides detailed description of the minimum hardware and software requirements essential for execution. It also outlines the constraints, assumptions and dependencies upon which the system was built. Lastly, it describes the interfaces, modules and the perspective with regard to which the system was designed.