**Chapter 6**

**IMPLEMENTATION OF THE DETECTION SYSTEM**

Software is considered to be a deliverable once its design is transformed into executable codes on the platform on which it is intended for use with requirements adhering to best available coding standards and practices. Implementation phase of software consists of important decisions regarding selection of platform, programming language to be used, coding conventions to be followed etc. which are influenced by several factors such as the environment in which the system works, security requirements and issues, optimization concerns, efficiency and accuracy of the system and its results, etc.

* 1. **Programming Language Selection**

The efficiency of the system is dependent on the programing language used to implement the software. Also, it depends on the need to implement the system on any platform required by the intended user. It also decides to what extent the language chosen can take advantage of efficient image processing libraries. Python version 2.7 and above has been chosen for developing this system.

The OpenCV library is widely supported by many versions of python [35]. Also, standard libraries such as the scikit machine learning library required for image processing are very efficiently implemented in the python programming language.

* 1. **Platform Selection**

The platform selected for developing the system is the Mac OSX version Mavericks because the setting up of the python environment and openCV libraries is very simple owing to a plethora of available package managers. Also, as a result of the Mac OSX being an extension of the linux operating system and therefore provides all of linux’s convenient programming utilities such as the command prompt, terminal commands and application interfacing using installation utilities such as pip and easy install.

* 1. **Code Conventions**

Coding conventions are a set of guidelines for a specific programming language that recommend programming style, practices and methods for each aspect of a piece program written in this language[36]. These conventions usually cover file organization, indentation, comments, declarations, statements, white space, naming conventions, programming practices, programming principles, programming rules of thumb, architectural best practices, etc. These are guidelines for software structural quality.

Software programmers are highly recommended to follow these guidelines to help improve the readability of their source code and make software maintenance easier. Conventions may be formalized in a documented set of rules that an entire team or company follows, or may be as informal as the habitual coding practices of an individual. Coding conventions are not enforced by compilers. So applicability of these conventions has no impact on the execution of the compiler.

* + 1. **Naming Convention**

The names of the variables, methods, class have been chosen carefully so as to reflect the intended use of the variables and what they represent. These have been followed with the exception of loop counters which are used and discarded almost immediately. A name which consists of more than one word is written as a single component with first word in small letters and following words with first letter as a capital letter.

* + 1. **File Organization**

This project uses a separate developer file for each module in use. All the files are stored in a common directory. Also, the resources used by the files which includes the data set for training the classifier, the MPEG-4 format traffic recording video upon which the project is performed and the intermediate data generated by the program itself such as storage files, video frames are all also stored in the same common directory.

Furthermore, the output that the program generates is stored in a separate path called as the project directory. All output files including key frames, violation instances captured and results extracted from the database are all stored in the project directory.

.

* + 1. **Properties Declarations**

The properties are declared in standard python convention that is the keyword ‘def’ followed by the name of the property. All properties are so named to reflect the nature of the operation that they are required to perform in the system as a whole. This naming is applied to all the properties defined in the system with the exception of the starting point of execution or main function. Class properties are declared within the body of the class.

* + 1. **Class Declarations**

The project follows an object oriented design. All database instances are simulated using class objects; these include simulating vehicles with all identifiable properties such as the name of the owner, the name of the vehicle etc. Listed below are the elements associated with defining a class in this system.

* **Class definition**: Description of common elements of every instance of a class.
* **Properties**: Data storage for class instances.
* **Methods**: Special functions that implement operations that are usually performed only on instances of the class.
* **Objects**: Instances of classes, which contain actual data values stored in the objects' properties.
* **Subclasses**: Classes that are derived from other classes and that inherit the methods, properties, and events from those classes.
* **Superclasses**: Classes that are used as a basis for the creation of more specifically defined classes.
  + 1. **Comments**

Comments are necessary part of any coding conventions as it improves the understandability of the code developed. Comment lines begin with the character ‘#’, and anything after a ‘#’character is ignored by the python interpreter. The # character itself only tells the interpreter to ignore the remainder of the same line.

In the project files, commented areas are printed in green by default, so they should be easy to identify in the text wrangler editor application of the OSX platform. Comments for blocks of code are started by a ’’’ and are delimited by the same three single quotes.

Comments are useful for explaining what function a certain piece of code performs especially if the code relies on implicit or subtle assumptions or otherwise perform subtle actions. For example,

**#**function to simulate a one second delay

def delay():

for i in range(0,10000000):

i \*= 1

* 1. **Difficulties Encountered and Strategies Used to Tackle**

The system carries out processing in two phases, pre and post processing. During the pre processing phase noise elimination and redundant frame removal is carried out. The removal of background noise and incorporation of non static objects in the background was a challenging task. Noise was eliminated using the clustering technique and the background is effectively modeled using a regulator value to vary the rate of averaging.

Also, during the post processing phase, which is the actual detection and identification phase, the standard cascade classifiers were producing inconsistent results [37]. Hence, we used a combination of Support Vector Machines and Histogram of Oriented Gradients technique to get more accurate results [38][39].

* + 1. **Management of the volumes that holds the image instances and data**

The system generates many intermediate frames, images and data required for individual stages of processing during the execution of the program. These include generation of frames of the video for performing background subtraction, generation of non-redundant frames for recompilation into video, capturing instances of traffic violation from the recompiled video and finally intermediate data about vehicle objects performing the violation.

The original frames of the video are not written to disk and hence don’t need to be managed, the non-redundant frames are stored in the project directory under the sub-directory full colored key frames, the instances of violation are also stored in the project directory under the sub-directory violation and the intermediate data is stored in files also contained in the project directory. The information about the vehicle objects is stored in a virtual python dictionary containing vehicle class instances indexed by their unique plate numbers.

* + 1. **Presenting of the system to the user**

The system developed in this project is packaged and deployed into a stand alone application that requires standard python and openCV libraries to run. Also, to the user, the package has been presented in the form of a Graphical User Interface completely developed using openCV to avoid the need for support from extraneous third party applications. The main module is key board interactive and the sub modules once open in the Graphical User Interface, details very clearly, step by step, every action that the user needs to perform in order to obtain the desired output. The Graphical User Interface functions both in a modular fashion and as an integrated unit to enable the user to know and understand the output received from every phase of processing during the execution of the program.

* + 1. **Handling deprecated methods and new methods**

As the program is developed keeping platform independence in mind, implementations of methods that perform similar functions from previous openCV libraries cannot be used. Such methods are called deprecated methods. These methods either must be overridden to be used for a user specific function or then newer standard openCV library implementations of the function needs to employed.

For future versions of python and openCV all the methods implemented in the system will either be available or deprecated. For deprecated methods, newer methods will be available as substitutions. But in the event that this does not happen, new methods have to defined and implemented, preferably added as a package to the standard openCV library for future use. In this system all methods that have been implemented from library files are defined in the python2.7 and above standard library and openCV 2.0 and above library.

* 1. **Summary**

This chapter presented the transformation of the design of the project into working model. The project was implemented on vim editor and text wrangler editor using the python programming language version 2.7 adhering to best coding policies and practices available. The implementation logic was devised formidably to tackle the impediments and hurdles that were encountered during the functioning of the violation detection system.