**Automatic Vehicle Counting for IoT based Smart Traffic Management System for Indian urban Settings**

Abstract—For smart city management, efficient handling of road traffic is one of the key aspects. Traffic congestion can be managed effectively, if the numbers of vehicles that are to pass through a crowded junction can be pre-estimated in time. The proposed method presents a framework, which has the capability to continuously convey the vehicle count and generate an alarm in case of large vehicle gathering to the controlling station in the Chandigarh or alike urban Indian cities. The number of vehicles passing through a location well before the required traffic junction can be estimated using the help of image processing techniques. Further, the monitoring details can be shared to a distant controlling centre situated anywhere in the city through internet usage. The performed experiments demonstrate the effectiveness of this Internet of Things (IoT) based technology.

**III. PROPOSED METHODOLOGY**

The automatic counting of vehicles passing through achosen destination was obtained by using image-processing techniques. A camera system was used to acquire the real time video footage of the traffic flow through road. After acquiring the video through camera, a portable raspberry-pi processing system was used to work on this data. Finally, after the average vehicle count per specified time interval has been found, the same processing system is used to communicate the same information to central control system. The complete details of how the work has been carried out are explained in following two sub sections:

1. **Vehicle counting using image processing** After obtaining the video footage from the camera system, the processing was done on raspberry pi using Open CV software. The flow of work done on image to acquire the number of vehicles passing through a region of interest area has been illustrated using figure 1. The description of steps adopted is provided as follows: • As seen in above figure, the first step involves extraction of the individual frames from the video sequence This is performed using the ‘cap.read( )’ command in Open-CV.
2. **Transmission of information for traffic management system using internet** This process of communication between the camera guided processing system and the final user server system via internet can be explained in four stages. The complete block representation is presented in figure 2 and each step is discussed in detail as follows: – Fetching per second Vehicle Count Data from Open CV: The average value of vehicle present during respective time interval was fetched and stored inside a variable that updated after every fixed time interval. – Interfacing of Firebase Google server with Raspberry Pi using Python: Next, to send processed information remotely to control room we must have a server and portable Raspberry system must have internet connection. For the server we have used Google Real Time Database server, i.e. Firebase. To use Firebase Real Time Database we must have compatibility with the Processor or Controller. To do so we have used Python language. – Establishing Connection with Firebase (Real Time Database): After interfacing, next step was to establishing connection with our database. For that, we had created new database on firebase server. Then, connected this to database using Credentials (API keys (Application Programming Interface), End Point URL (Universal Resource Locator)) using Python. – Sending Vehicle count data to Real Time Cloud: After connecting to database, our task was to send vehicle count data to server remotely. For that, we sent our vehicle count data obtained from Open CV to

