**1. Introduction**

**1.1 Purpose**

The purpose of this proposed work is to build a system to predict the traffic flow on the basis of monitoring the activities at traffic intersections for detecting congestions. Due to the result of the increase in vehicle traffic, many problems have appeared such as traffic accidents, traffic congestion, traffic induced air pollution and so on. Traffic congestion has been a significantly challenging problem. It has widely been realized that increases of preliminary transportation infrastructure, more pavements, and widened road, have not been able to relieve city congestion. To processes the information and monitors the results as better understand traffic flow, an increasing reliance on traffic surveillance is in a need for better vehicle detection at a wide-area. Automatic detecting vehicles in video surveillance data is a very challenging problem in computer vision with important practical applications, such as traffic analysis and security. The proposed system includes an android application for the registration of users. The vehicle count is then sent to the server where registered users can view the status of the traffic.

**1.2 Product Scope**

The scope of the proposed system is to develop an automatic vehicle counting system, which can process videos recorded from stationary cameras over roads e.g. CCTV cameras installed near traffic intersections / junctions and counting the number of vehicles passing a spot in a particular time for further collection of vehicle / traffic data. A simple approach was carried out to tackle the problem by using Gaussian mixture model-based object detection, a non-predictive regional tracking and a counting of tracked objects based on simple rules. The captured video is then processed to detect and count the vehicles and thus traffic analyzing can be done. The count of vehicles can be viewed from the server, where it saves all the traffic details.

**2. The Overall Description**

**2.1.1 Product Perspective**

This system structure is computationally efficient and can run in a real-time basis while retaining very respectable detection rates. The appearance of larger vehicle or vehicle's shadow occluding the adjacent lanes also is known to trigger false detection. Consequently, the merit of using computer vision as a surveillance tool has been limited by focusing strictly on building reliable systems that can perform in real-time.

The system makes use of an existing video sequence. The first frame is considered as the reference frame. The subsequent frames are taken as the input frames. They are compared and the background is eliminated. If a vehicle is present in the input frame, it’ll be retained. The detected vehicle is thus tracked by various techniques. Vehicle detection is done by using Background Subtraction (BS) algorithm and for tracking blob tracker algorithm can be used.

**2.1.2 Product Functions**

• Detects and count the vehicles • Predicts Traffic flows • Can be used to predict accident zone areas • Ensures road safety

**2.1.3 Operating Environment**

**Hardware requirements**

• Processor: Intel Pentium

• Storage: 1.5 GB Hard Disk space

• Memory: 1 GB RAM

**Software requirements**

• Operating system: Linux

• Platform: OpenCV, Python

**3. Functional Requirements**

The proposed system consists of 3 modules. They are given below:

• ANDROID APPLICATION

In order to make the proposed system user friendly, an android application is used to collect the user details. Users can register through the app, can enter details of their journey including source, destination and the route they are going through. From the details stored in the RTA server, they will get an update about the traffic conditions and vehicle count through the android application.

• VEHICLE DETECTION AND COUNTING

An Automatic vehicle detecting and counting system makes use of video data acquired from stationary traffic cameras, performing causal mathematical operations over a set of frames obtained from the video to estimate the number of vehicles present in a scene. It is just the ability of automatically extract and recognize the traffic data e.g. total number of vehicles, vehicle number and label from a video. In each video frame, Gaussian mixture model differentiates objects in motion from the background by tracking detected objects inside a specific region of the frame, and then counting is carried out.

A Gaussian Mixture Model (GMM) is a function to measure parametric probability density represented as a weighted sum of Gaussian component densities. In this proposed system,

GMM carries out the job of separating the foreground and background from image frames by learning the background of a scene. In this system, blob detection uses contrast in a binary image to compute a detected region, it’s centroid, and the area of the blob. The GMM supplies the pixels detected as foreground. These pixels are grouped, in current frame, together by utilizing a contour detection algorithm. The contour detection algorithm groups the individual pixels into disconnected classes, and then finds the contours surrounding each class.

Each class is marked as a candidate blob (CB). These CB are then checked by their size and small blobs are removed from the algorithm to reduce false detections. The positions of the CB, in current frame, are compared using the k-Means clustering that finds the centers of clusters and groups the input samples CB around the clusters to identify the vehicles in each region. The moving vehicle is counted when it passes the base line. When the vehicle passes through that area, the frame is recorded. In each region the blob with the same label are analyzed and the vehicle count is incremented.

• Real-time Traffic Analyzer SERVER

Counting vehicles gives us the information needed to obtain a basic understanding over the flow of traffic in any region under surveillance. The total count of vehicles, including other traffic details such as source and destination of user, current weather conditions are stored on the server. This will help to make a traffic analysis. The RTA server gets updated on every 15 minutes.

**4. Nonfunctional Requirements**

**4.1 Performance Requirements**

The main performance requirements that the product should satisfy are:

- Accuracy: Accuracy in functioning and the nature of user-friendliness should be maintained in the system. - Speed: The system must be capable of offering speed.

**4.2 Quality Requirements**

The most important quality requirements that the system should satisfy are:

- Scalability: The software will meet all of the functional requirements without an unexpected behavior.

- Maintainability: The system should be maintainable. It should keep backups to atone for system failures, and should log its activities periodically.

- Reliability: The acceptable threshold for down-time should be long as possible. i.e. mean time between failures should be large as possible. And if the system is broken, time required to get the system back up again should be minimum.

- Testability: The proposed system should be properly tested under various circumstances in order to assure its reliability.

**5. Conclusion**

A system has been developed to detect and count dynamic vehicles on highways efficiently. The detection and tracking and counting of moving vehicle can be extended to real-time live video feeds. Apart from the detection and extraction, process of recognition can also be done. By using recognition techniques, the vehicle in question can be classified. Recognition techniques would require an additional database to match with the given vehicle. The system is designed for the detection and tracking and counting of a multiple moving vehicle. The detection of vehicles in a mix traffic situation of low, medium and high traffic is precisely as expected and the counting algorithm is accurate.