Campus Traffic Reinforcement using IoT and OpenCV

Bebetto Francis
Department of Computer Engineering
Don Bosco Institute of Technology
Mumbai, India
bebettofrancis@gmail.com

Richard Britto
Department of Computer Engineering
Don Bosco Institute of Technology
Mumbai, India
richardbritto97@gmail.com

Raynal D'cunha
Department of Computer Engineering
Don Bosco Institute of Technology
Mumbai, India
dcunha.raynal4@gmail.com

Dr. Phiroj Shaikh
Department of Computer Engineering
Don Bosco Institute of Technology
Mumbai, India
phiroj.dbit@dbclmumbai.org

***Abstract*—The last few decades has seen a tremendous increase in the number of vehicles which in turn has forced the government to make use of technology to ensure the traffic rules are not violated. Even though these new technologies are being adopted for the safety of the commuters, none of the campuses have taken the initiative in doing so. The proposed Campus traffic reinforcement system makes use of IoT, and OpenCV for image processing. The proposed system can be mainly used to detect accidents, traffic violations i.e. speeding of vehicles, unnecessary honking and provide the information to the concerned authority to take further effective action.**

Keywords—IoT, Arduino Uno, OpenCV, R-CNN

I. INTRODUCTION

In recent years there has been a tremendous increase in the number of vehicles. New technologies are being adapted to make commuting safer. But increase in the number of vehicles has also led to the increase in the number of accidents. Accidents not only occur on highways or other main routes of road transport, they can also occur inside a campus. The main problems faced during commuting inside a campus are unnecessary noise (vehicle honking), speeding and accidents. Monitoring of vehicles in a large campus is a very hectic task. Many violations remain unnoticed which in turn leads to their frequent repetition. Security cameras are now a common practice in every campus, but for a campus disciplinarian to monitor each and every footage is almost impossible. There is a high probability that most of the violations remain unnoticed, and due to delayed information these violators are not penalized

The proposed system makes use of an IP camera for real-time footage of the campus roads. This footage is transmitted to the server for further processing. IR sensors are used for vehicle and speed detection, Sound sensors are used for noise detection and Faster R-CNN (Regional Convolutional Neural Network) through OpenCV and tensorflow library are used for accident detection. Whenever a vehicle exceeds the authorized speed limit or noise level, the campus disciplinarian is informed via email. Similarly, the proposed system informs the campus disciplinarian regarding an accident.

II. LITERATURE REVIEW

Numerous works have previously been done by researchers in order to improve the monitoring of traffic rules. Zezhi Chen and Tim Ellis have made use of AGMM for detection of moving vehicles [1]. Abhinav Saini et al. have implemented a method to detect accident based on region and feature matching [2]. They have made use of the SURF-(Speeded-Up Robust Features) technology. Gaurav Verma et al. have developed an Automated Red Light Enforcement Camera for traffic control [3]. They are making use of Arduino Uno and 1sheeld technology for detecting violations. Tansim Sorawar et al. have developed a traffic surveillance system by making use a Raspberry Pi Camera Module [4]. The algorithm being used for detecting the size of the vehicles is the Histogram Equalization algorithm. For processing the video they are making use of OpenCV library and python. Muhammad Ibrar et al. have developed a system for monitoring large vehicles on CPEC route [5]. The technologies being use are drones, Satellite images, and RCNN. Kaiyang Zong et al. have developed a vehicle detection and tracking system based on GMM and advanced Camshift algorithm [6]. Ross Girshick et al [7] have developed an algorithm i.e. the R-CNN (Regional Convolution neural network) that makes use of a convolutional neural network in the detection of objects. The main advantage of this algorithm is that instead of making use of the whole image for processing, it makes use of 2000 regions from the image extracted using selective search algorithm for the processing part. This considerably reduces the processing and learning time to the neural network, thus making this an efficient algorithm in the field of image processing. Although the R-CNN model considerably reduces the training time of the neural network, working on 2000 region proposals at a time is a really time consuming process. In order to tackle this approach the same author proposed a new algorithm i.e. the Fast R-CNN model. The only difference here being that instead of feeding the region proposals to the CNN, we feed the image to the CNN to generate a convolution feature map. Shaoqing Ren et al. [8] proposed a much better and efficient model that considerably reduces the time in training and detecting an object. The model proposed is called as the Faster R-CNN model. The main difference being the omission of the selective search algorithm. Instead it lets the network itself learn the region proposals.

III. SYSTEM OVERVIEW

In this research work, a proposal is being made to monitor vehicles in a lane within a certain range. The IP

Camera module is responsible for transmitting the real-time footage of the vehicles passing through the selected lane. This footage is processed by a server with the help of python and OpenCV library. Arduino Uno R3 is used in conjunction with IR and Sound sensors for speed and noise detection. When the vehicle passing through the lane exceeds the speed limit or causes disturbance by unnecessary honking, a picture of the violator along with the violation he/she has done is sent to the disciplinarian head via email.

Similarly is the case with accident detection. For accident detection we are making using of the Faster R-CNN algorithm [9]. Inorder to train the system using the faster R-CNN algorithm we are making use of the tensorflow library. In a campus majority of the accidents occur with respect to falls. The reasons may be due to unmaintained roads or even due to speed breakers. At times such accidents remain unnoticed. The video being transmitted by the IP camera is processed with the help of the trained model and OpenCV library, and if an accident has been detected the Campus disciplinarian is informed via email.

IV. VIDEO ANALYSIS

Video Analysis is the process of automatically analyzing a video to detect desired events in it. In most cases the most important part of a video is not the background but the foreground. These important parts are the objects of interest in a video. The objects of interest could be anything, e.g. Humans, Vehicles, Animals, etc. The technique of extracting the foreground from the background in a video stream is called as foreground detection. It can also be referred as background subtraction. This technique incorporates a threshold value in order to neglect the error between the current and the average of the images without the object of interest. There are various types of back grounding methods, they can be namely categorized into pixel-based, region-based, hybrid methods and parametric and non-parametric methods.

A video is actually a series of bitmap digital images that are referred to as frames. Each frame comprises of a matrix of pixels. If we consider a binary image each pixels holds a 1-bit value that indicates whether it is foreground or background. In a binary image 0 stands for black and 1 stands for white. If we consider a grayscale image, it holds 8-bit information indicating the brightness of the image. In the case of color images, they make use of RGB values indicating Red, Green and Blue respectively. The color models can be further divided into additive models e.g. HSV (Hue, Saturation and value) and the subtractive model CMYK (Cyan, Magenta, Yellow, and Key or Black).

Let us denote the frame as f(x, y) where x and y are the co-ordinates. The time be denoted as t. Let F be the foreground object, P be the pixel value. In order for back grounding frames have to be compared with each other. For comparing the frames the frames have to be divided into multiple segments. These segments are created based on their characteristics, such as position and value. The techniques used for back grounding have be discussed below.

A. Static Frame Difference

This is a non-adaptive back grounding technique which is mainly used where the background of a scene remains unchanged[10]. In this process first each frame is converted into a grayscale image. Considering this frame to be the one with no object of interest. This particular frame is the one which will be compared with all the subsequent frames i.e. converted into grayscale. After comparing with the background frame if there is a change in any of the subsequent frames, it is declared as a foreground object. Non-adaptive background comprises of various techniques which have been discussed below.

$$P[F(x, y, t)] = P[f(x, y, t)]–P[f(x, y, 0)]$$

B. Static Frame Differenec with Threshold

This is also non-adaptive back grounding technique. The use of static frame difference is actually impractical in real life scenario. Phenomena's such as illumination, shadows and dynamic back grounding can make it difficult to analyze the video only with the help of static frame difference. In order to fix this we make use of a threshold value [11]. If the absolute difference between the pixel values is greater than the threshold, only then it will be declared as a foreground.

$$P[F(x, y, t)] = \{P[f(x, y, t)]–P[f(x, y, 0)]\} > \text{Threshold}$$

C. Frame Difference

This method is also a non-adaptive back grounding method. It is similar to the above mentioned method with the only difference being, instead of comparing the subsequent frames with the first frame, it compares the adjacent frames. This makes the results obtained much more accurate [12].

$$P[F(x, y, t)] = \{P[f(x, y, t)]–P[f(x, y, t\text{-}1)]\} > \text{Threshold}$$

D. Adaptive Backgrounding

This a back grounding method wherein we create a background model by averaging the images over time [13]. This method is mainly used to detect moving objects. The disadvantage of this model is that it cannot detect slowly moving objects since they get adapted to the background model and in turn are lost from the foreground.

$$P[F(x, y, t)] = P\{\text{Average}[f(x, y, t\text{-}1),\ldots.., f(x, y, t\text{-}n)]\} > \text{Threshold}$$

E. R-CNN (Regional Convolutional Neural Network)

The problem with most of the object detection algorithm is that they select a huge number of regions for processing an image. Ross Girshick et al. proposed a method wherein we make use of selective search algorithm to extract just 2000 regions from an image. Thus the problem of working with huge number of regions had been solved because rather than classifying a huge number of regions, we can work with just 2000 regions. This helped in reducing the training and object detection time. Thus making it a very efficient algorithm.

The selective search algorithm being used is as follows:

1) We generate initial sub-segmentation through which we generate many candidate regions.

2) Greedy algorithm is used to recursively combimesimilar regions into larger ones.

3) Generated regions are used to produce the final candidate region

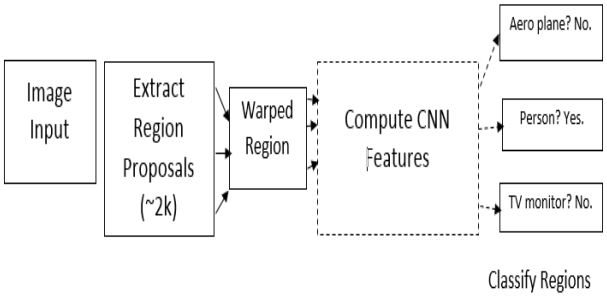

Figure 1. R-CNN

Once the 2000 candidate regions are extracted, they are warped into a square and fed into a convolutional neural network that produces a 4096-dimensional feature vector as output. Over here the CNN acts as a feature extractor and the output dense layer consists of the features extracted from the image. These extracted features are fed into the Support vector machine (SVM) to classify the presence of the object within the candidate region proposal. In order to increase the precision of the bounding box, the algorithm predicts four offset values.

Even though the R-CNN algorithm helps in reducing the processing time substantially, it is not very feasible in real life scenario. Classifying 2000 region proposal per image take a huge amount of training time. It takes around 47 seconds for each test image. Also the selective search algorithm is a fixed algorithm. There is no learning happening during this stage. This eventually could lead to the generation of bad candidate region proposal.

F. Fast R-CNN

The same author of the previous paper (R-CNN) solved some of the drawbacks of the R-CNN algorithm to build a faster object detection algorithm. The algorithm came to be known as the Fast R-CNN algorithm. The approach towards this algorithm is similar to the R-CNN algorithm, only difference being instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map. From the convolutional feature map, we identify the region of proposals and warp them into squares and by using Region of Interest pooling layer we reshape them into a fixed size so that it can be fed into a fully connected layer. The reason for the faster processing of the Fast R-CNN algorithm is that we don't have to feed 2000 region proposals to the convolutional neural network every time. Instead, the convolutional operation is done only once per image and a feature map is generated from it.

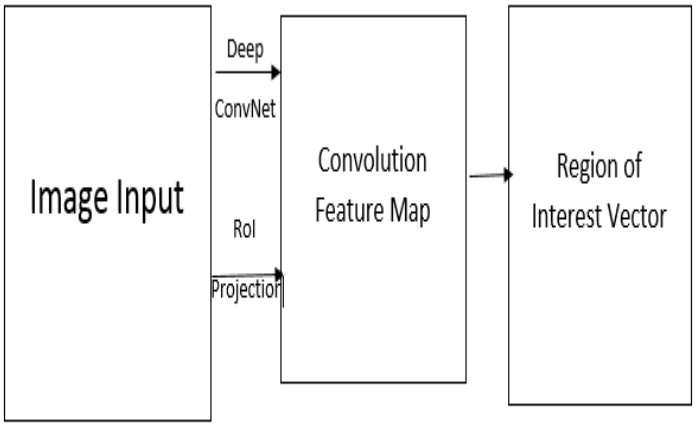

Figure 2. Fast R-CNN

V. OVERVIEW OF THE ALGORITHM (FASTER R-CNN)

The problem with both R-CNN and Fast R-CNN algorithm is that they make use of the selective search algorithm in order to find out the region proposals. The problem with selective search algorithm is that it is a slow and very time consuming process that affects the performance of the network. In order to counter this problem Shaoqing Ren et al. came up with an object detection algorithm that eliminates the selective search algorithm and lets the network learn the region proposals. Similar to the Fast R-CNN algorithm, the image id provided as an input to the convolutional network which provides a convolutional feature map. Instead of making use of the selective search algorithm on the feature map to identify the region proposals, a separate network is used to predict the region proposals. These region proposals are then reshaped using a Region of Interest pooling layer which in turn is used to classify the image within the proposed region and predict the offset values for the bounding boxes.

The Faster R-CNN has two networks viz. Region proposal network (RPN) and a network that uses the region proposals to detect objects. The time cost of generating region proposals using RPN is much smaller than with selective search algorithm. RPN ranks region boxes also called as anchors and proposes the ones that most likely contain an object.

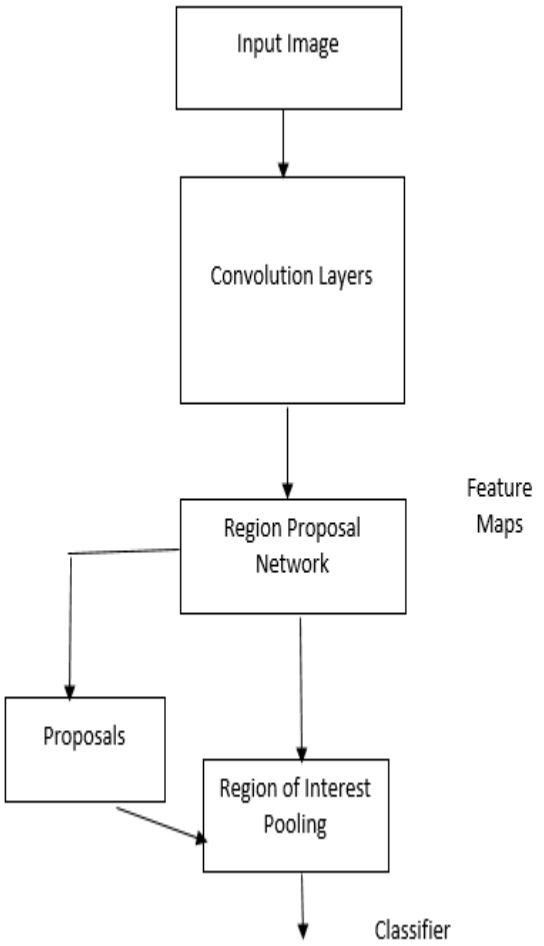

Figure 3. Faster R-CNN

Anchors play an important role in the Faster R-CNN algorithm. An anchor is a box. In the default configuration of the Faster R-CNN model, there are 9 anchors at the position of an image. The following graph shows 9 anchors at the position (320,320) of an image with size (600, 800).

The output of a region proposal network (RPN) is a bunch of boxes/proposals that will be examined by a classifier and regressor to eventually check the occurrence of the objects. To be more precise, RPN predicts the possibility of an anchor being background or foreground, and refine the anchor. The first step of training a classifier is making a training dataset. The training data is the anchors we get from the above process and the ground-truth boxes. The problem here is that how we use the ground-truth boxes to label the anchors. The basic idea here is to label the anchors having the higher overlaps as background. Another thing that we need to pay attention to is the receptive field if we want to re-use the trained network as the CNNs in the process. We need to make sure that the receptive fields of every position on the feature map cover all the anchors that it represents.

After RPN, we get proposed regions with different sizes. Different sized regions means different sized CNN feature maps. It's not easy to make an efficient structure to work on features with different sizes. Region of Interest Polling can simplify the problem by reducing the feature maps into the same size. By training the RPN, the final regressor and classifier at the same time jointly leads to 1.5 times faster results with similar accuracy.

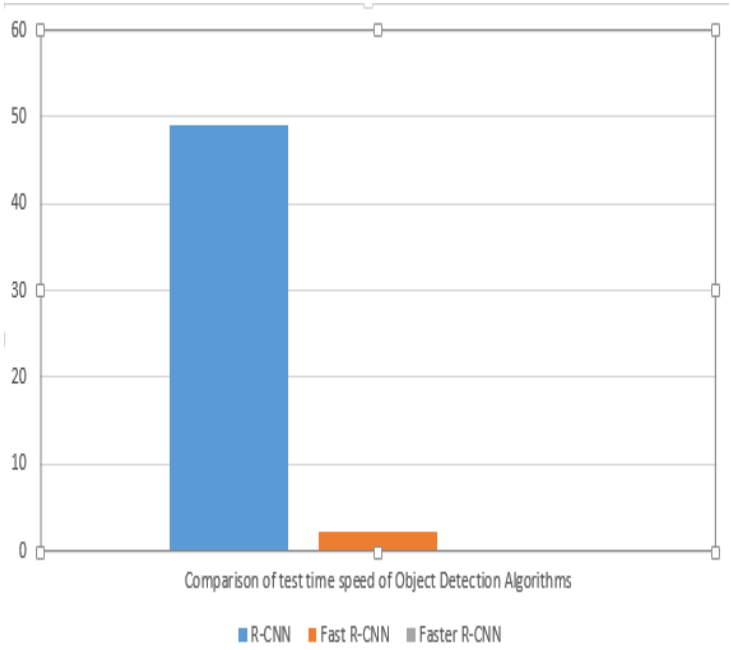

Figure 4. Test time speed of various Object detection algorithms

For training the model using Faster R-CNN algorithm we are making use of 40,000 images. 30,000 are being used for training the model and 10,000 for testing the model. The training of the model is done using the tensorflow library with the help of python 2.7.

A graphical user interface is being developed in order to view the violation or accident. We are making use of python 2.7 for developing the Graphical User Interface (GUI). A program has been written that helps in loading the pre-trained Faster R-CNN model. Real-time footage of the Campus lane is being sent to the Server for further processing. With the help of the pre-trained faster R-CNN model, the program helps in detecting the accidents inside a campus. If an accident has been discovered, an image is captured of the vehicle that has met with an accident. The image is then sent to the campus disciplinarian to notify him/her of the accident that has occurred.

VI. WORKING

A. *Arduino Uno R3 and Sensors*

The Arduino Uno R3 is connected to the Wifi module ESP8266 (wireless communication module), 2 IR sensors and 2 Sound detection sensors using Jumper wires. With the help of the IR sensors vehicles will be detected and when a vehicle crosses them the speed of the vehicle is calculated. If the speed exceeds the authorized limit an image of the vehicle is captures and sent to the campus disciplinarian for further actions.

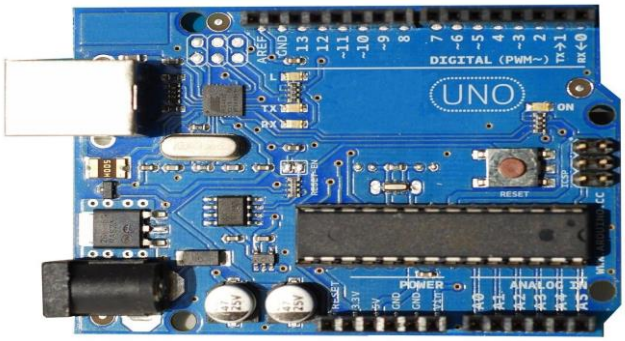

Figure 5. Arduino Uno R3

Similarly, 2 sound detection sensors are used for noise detection. They are connected to the Arduino Uno R3 model with the help of jumper wires. When the authorized sound limit is exceeded an image of the vehicle is captured and sent to the campus disciplinarian via email.

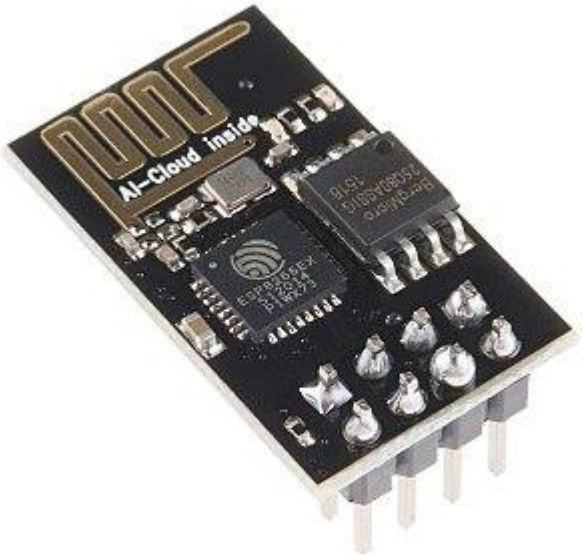

Figure 6. ESP8266 Wi-Fi Module

B. *IP Camera*

We are making use of the Arducam OV7670 for capturing the real-time footage of the vehicles. The IP Camera is Arduino Uno R3 which in turn is wirelessly connected to the Server i.e. Laptop or Desktop. It keeps on transmitting real-time footage of the selected campus lane. The footage is then processed by the server in order to detect an accident or violation.

Figure 7. Arducam OV7670

C. *Server*

The role of the server is to process the captured footage by the IP Camera. For development of the GUI (Graphical User Interface) Python 2.7 is being used and for the Image processing algorithm OpenCV is being used. A trained model using the Faster R-CNN algorithm is applied on the captured footage for accident detection. If an accident has been detected the image of the vehicle is sent to the campus disciplinarian.

Violation detection is dependent on the Arduino Uno R3 and the sensors. If the sensors detect a violation only then the server will process their input and send an image of the violator to the campus disciplinarian via email.

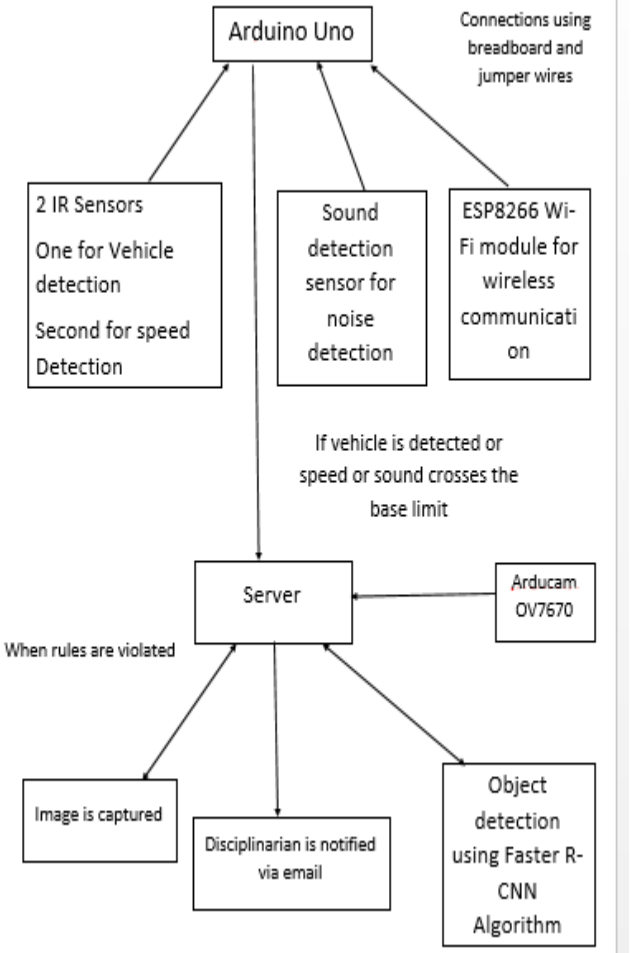

Figure 8. System Architecture

ACKNOWLEDGMENT

For this research work, we would like to express our sincere gratitude to Dr. Shaikh Phiroj, Assistant Professor, Department of Computer Engineering, Don Bosco Institute of Technology, for his valuable suggestions. We are also grateful to all the faculty members of the Computer Engineering Department for their valuable support.

VII. EXPERIMENTAL RESULTS

The development environment that we are making use of is a system with Ubuntu 16.04 LTS, Intel i7 8750H, and NVidia GTX 1070. We are making use of python 2.7 and various packages like tensorflow, OpenCV.

Accident	Camera Angle	
	Horizontal	60°
No of Frames	1000	1000
Detected Frames	954	938
Accuracy	95.4	93.8

TABLE I: Experimental Results of Real-time Accident detection

CONCLUSION

We have proposed a system for effectively monitoring traffic inside a campus environment. We have made use of IoT for vehicle, speed and noise detection, and Faster R-CNN model for accident detection. We compared various image processing algorithms and chose the Faster R-CNN algorithm for this research work. The limitations of the R-CNN and Fast R-CNN had been dealt with. The problems that still persist are the shadow problems.

REFERENCES

.

[1] Zezhi Chen, Tim Ellis, "Self-Adaptive Gaussian Mixture Model for Urban Traffic Monitoring System"

[2] Abhinav Saini, "Region and Feature Matching based Vehicle Tracking for Accident Detection"

[3] Gaurav Verma, "Automated Red Light Enforcement Camera for Traffic Control"

[4] Tansim Sorwar, "Real-time Vehicle monitoring for traffic surveillance and adaptive change detection using Raspberry Pi Camera Module

[5] Muhammad Ibrar, "Improvement of Large-Vehicle Detection and Monitoring on CPEC Route"

[6] Kaiyang Zhong, "Vehicle Detection and Tracking Based on GMM and Enhanced Camshift Algorithm"

[7] Ross Girshick, "Rich feature hierarchies for accurate object detection and semantic segmentation"

[8] Shaoqing Ren, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks"

[9] Shuang Liu, "Abnormal Behaviour recognition Based on improved Gaussian mixture model and Heirarchical detectors"

[10] Nishu Singla, "Motion Detection Based on Frame Difference Method"

[11] Josna George, "New Approach for Moving and Static Vehicle Detection Using Motion Energy"

[12] Jiajia Guo, "A New Moving Object Detection Method Based on Frame-difference and Background Subtraction"

[13] J. Mike McHugh, "Foreground-Adaptive Background Subtraction"