International Journal of Future Generation Communication and Networking
Vol. 13, No. 1s, (2020), pp. 176-- 182

Automated Traffic Signal Controlling Using Deep Learning Techniques

Mr. Chirag Rohit, Mr. Mayank Shah, Mr. Kartik Sonaghela, Prof. Jyoti Wadmare
Department of Computer Engineering, K.J. Somaiya Institute of Engineering and Information Technology,
Sion, Mumbai, University of Mumbai

Abstract

The present Traffic Control Systems (TCS) in the metro urban communities of India is inefficient due to arbitrariness in the rush hour at crossroads throughout the day. A run of the mill day in India would take a gander at peak hour timings when the traffic thickness is high in the streets and peak hour timings when the traffic thickness isn't so high. The traffic flags at all the intersections in India are hardcoded which means the signals have fixed memories and switch traffic between statically. Because of this, the vehicles need to wait for prolonged amount of time even though the traffic density is less. The solution to this issue is by building up a framework which distinguishes traffic densities on each lane of the junction switch the signal lights dynamically along with synchronization of the adjacent lane's traffic signal. This process will be divided into two modules. The first module will comprise of building a model which will detect and the number of vehicles in particular lane were counted. All these lane's density will be added in a dynamic queue on which pre-emptive and dynamic scheduling algorithm will be applied.

Keywords: *Traffic, CCTV cameras, Convolutional Neural Networks, signals, Scheduling algorithms, ITMS*

1. Introduction

Traffic signal timings in a road junction not just influence total travel time of the users and the total amount of traffic emissions but also create an inequity problem in terms of the change in travel costs of users travelling between different locations. At present, most traffic systems despite having a great infrastructure still use fixed intervals for a signal for a very long cycle. These systems cannot dynamically adjust traffic light timing in response to unexpected situations such as traffic accidents, natural calamities or sudden incidents. With advancements in technology, traffic data such as traffic volume, speed, detection of vehicle and waiting time can now be gathered by sensors or cameras. Inductive loop detectors, infra-red detectors, radar detectors and video-based systems or also known as CCTV cameras are multiple systems from which real-time information is available. Intelligent Traffic Monitoring System (ITMS) is of utmost importance and the need of the hour. Leveraging on automatic traffic data collections the project tries to address the issue of static signals. The project tries to gather the data from CCTV cameras installed at all the junctions. Followed by that there is detection of the vehicle in which a model is built which uses deep learning techniques and algorithms[1][2]. Labeled datasets will be used to train the model for the application. The vehicle detection will fetch the count of the number of vehicles in a particular lane. This information will help us in calculating the respective traffic densities in each lane and hence, preparing a dynamic queue[3]. After this Round Robin algorithm is applied on the queue to dynamically adjust traffic signal timings at road junctions[4].

2. Literature Review

Busarin Eamthanakul, Mahasak Ketcham and Narumol Chumuang have described in their paper how they have used CCTV cameras to check and detect the traffic condition on the road in specific time. They have used image processing techniques to build their traffic congestion investigating system. They first took the video footage from the cameras and converted it into images. They converted these images into grayscale images which consisted of black and white colors and also shades of gray. They have used the technique of separating the background from the foreground for detection. Next step was background subtraction. Heikkila and Olli method was used for the implementation of background

ISSN: 2233-7857 IJFGCN

subtraction in their system. After this step, errors and noise were introduced which had to be minimized. Firstly, they applied a median filtering mask followed by dilation, erosion and closing operations. Erosion operation erodes the errors from the boundary of the object and this is followed by dilation which increases the white region of the object and hence, increasing the object area. Closing operation helps remove small holes and black points present in the object. These techniques not only reduce the errors, salt and pepper noise but also enhance the foreground detected. Finally, they used contour-based methods and edge detection techniques for foreground detection.[5]

Akhil Soin and Manisha Chahande have proposed an application in which they detect moving vehicles using deep neural networks (DNNs). Vehicle detection and especially detection of moving vehicles has become of utmost importance and an integral part of any intelligent traffic management system(ITMS). They have solved this problem using the deep learning approach. They have claimed to have outperformed the state-of-the art methods. Their paper focuses on various applications of vehicle detection such as automated driving, automatic toll collection, intelligent parking systems, intelligent traffic management systems (ITMS) and many more. They have discussed on the importance of detecting the size, dimensions and other features of the vehicles to categorize them for various applications. They have used the latest convolutional neural networks and limited Boltzmann machine in their application. They have used a Region of Interest (RoIs) with which they separate the background from the foreground from the images supplied. A classic approach where the front,rear and the side parts of the vehicle were extracted from the dataset images which were fed to the neural network model and led to vehicle detection.[1]

Jason Kurniawan, Sensa G.S. Syahra, Chandra K.Dewa and Afiahayati have built up a traffic congestion identification system which is built using Convolutional Neural Networks. Convolutional Neural Networks are explicitly intended to process images and sequence data. It is a variation of standard neural networks. A CNN comprises of at least one convolution layer, at least one pooling layer and at least one fully connected layer. Their CNN model was prepared utilizing the pictures captured from the CCTV cameras. HIgh resolution colored images were converted into low resolution grayscale images by them. Their principle objective in performing this conversion was to separate daytime and night time pictures and furthermore make it computationally light. They performed a binary classification of the road congestion scenes with the labels “jammed” and “not jammed”. These pictures were caught from the CCTV cameras from a beginning date to end date referenced. Their CNN architecture comprised of two convolution layers, one max pooling layer and one fully connected layer. Their model was executed using Python with Keras library. The batch size was of 250 pictures with 100 epoch and furthermore their weights were updated by utilizing the backpropagation algorithm. K-fold validation for the validation of the CNN model with k=10 was utilized. They have used mini batch gradient descent with Adam optimization. They have built a framework which can be integrated and utilized with other applications. Their system can classify camera traffic situations with minimum amount of pre-processing and an acceptable accuracy.[3]

R. Krishnamoorthy and Sethu Manickam have developed a full-fledged application in which they have used computer vision and image processing techniques to detect vehicles from the CCTV cameras installed at the road junctions to get the count and hence calculate the traffic density on each road in the crossroad and apply a dynamic scheduling algorithm to dynamically switch the traffic lights. Their system aims at reducing the traffic congestion scenarios considerably at traffic conjectures. They have applied Gaussian blur by kernel matrix which is followed by Kernel based Edge Detection to detect all the major edges and ignore the minor ones. They developed a logic in which they would track the windshield of the cars in the video. After applying Canny edge detection they converted the images into binary intensity values for low computation complexity followed by closed figure identification for detection. They developed a function which will be recursively called to detect similar neighbours around the detected object. Next crucial step in their application was to apply a scheduling algorithm on the dynamic queue. This scheduling algorithm is based on Longest Job First and also pre-emptive in nature. They have used the Round Robin algorithm with a varying time slice. They constantly kept on updating the queue and the traffic lights were dynamically switched. The overall

ISSN: 2233-7857 IJFGCN

system developed was very efficient in real time scenarios. Also, they did not meet with any situation where starvation was met.[4]

Shanghang Zhang, Guanhang Wu, Joao P. Costeira and Jose M.F. Moura have proposed a system which understands the traffic density from large-scale web camera data. They have proposed the use of deep learning-based methods and on top of that optimization techniques to reduce the errors as the cameras used have low frame rate, low resolution, high occlusion and large perspective. These factors could heavily affect the detection results. They gathered images from the web cameras and manually annotated them. 60000 frames were collected on which bounding boxes was applied, categorized into 10 vehicle types and the orientation. They also took into consideration the weather and re-identification of vehicles. They performed a train-test split and fed it to the model which was built using the Fully Convolutional Neural Network (FCN). Their network architecture also helped them to obtain the count integrating the vehicle density map and also developed a framework based on FCN to jointly learn the density map and the vehicle count. Now they applied optimization-based vehicle density estimation with rank constraint. They not only focused on the limitations of the cameras used and the limitations of existing systems but also managed to successfully compute without active background subtraction or foreground segmentation.[2]

Table 1. Comparative study between the papers on above parameter

Title	Algorithm /Model	Result/Accuracy	Dataset	Limitation
1	Image processing techniques - Heikkila and Olli method for background subtraction. dilation, erosion, closing operations and edge detection methods	Paper represented a formula for estimating number of cars that are closely packed in an image. Detection of congestion scenario and labelling them as flow ,heavy and jammed	Dataset not required	Highly depended upon probability and estimation
2	Region based Convolutional Neural Network(R-CNN)	Categorization into two classes- Substantial vehicle and non-specific foundation class with accuracy 100%	CIFAR-10	Less number of class labels used
3	Convolutional Neural Network(CNN), Adam Optimizer	Binary classification of road congestion scenes with accuracy 89.50%	Custom dataset consisting 1000 images taken from CCTV	Model works on low resolution images, shows less classification performance
4	Image processing techniques - Kernel based edge detection and	Vehicle detection, density	Dataset not required	Works on narrow roads with fewer

	Canny edge detection	calculation and application of scheduling algorithm		lanes. Works with low resolution and low frame rate cameras
5	Fully Convolutional Network(FCN)	Count Integration with vehicle density map	Custom dataset consisting of 60000 images	Model not robust for multiple cameras

3.

Proposed Approach

The process comprises two primary modules1: Vehicle Detection System and Traffic Scheduling Algorithm.

Input: The video input is received from the CCTV installed at the junction or crossroads.

Model: Vehicle Detection will be carried out using Faster R- CNN. The model will be using labeled images for vehicle detection. The live video received from the cctv cameras is input to the model. The model will detect the vehicles and give us a count of the number of vehicles on a particular road. Using this count, the system will be able to determine the traffic density of each road in the junction.[7]

Deep learning techniques which includes CNN dominate the today's object detection methods, of which RCNN has shown fantastic detection results. 2000 bottom-up proposals are computed by RCNN from an input image. CNN extracts the proposals features and then each region is classified using support vector machine (SVM). Selective search is used by RCNN for proposal generation, then a 4096-feature vector is produced from each region proposal. Then for the computation of features, the image is warped into 227 x 227 pixels. Five convolution neural networks and two fully connected layers are used to process the warped image. The computed features are then scored using a support vector machine. A convolutional feature map or a convolved feature for the input image is computed by the Fast RCNN technique and then each object proposal is classified utilizing a feature vector extricated from the shared element map. The processing speed is improved considerably.

Scheduling: After this a queue based on the density is prepared and scheduling algorithm is performed. hybrid round robin scheduling algorithm will be used. It will have a time slice which will be dynamic in nature and will utilize the phenomena of Longest Remaining First which will then relax the traffic by periodically switching and updating the traffic lights. Hence, the time duration of the green light will be switched and managed dynamically and not for fixed time duration based on the traffic density on the road.

Hardware Interface: The output received will be given as input to the microcontroller using the pyFirmata library available in python[6]. Interfacing the model with the microcontroller is possible utilizing Firmata. Firmata is a nonexclusive convention for speaking with microcontrollers from programming on a host PC. It is anything but difficult to add objects for other programming to utilize this convention. Fundamentally, this firmware sets up a convention for conversing with the Arduino from the host programming. The microcontroller will contain LED lights speaking to traffic signals which will act as indicated by the info encouraged by the framework.

ISSN: 2233-7857 IJFGCN

International Journal of Future Generation Communication and Networking
Vol. 13, No. 1s, (2020), pp. 176-- 182

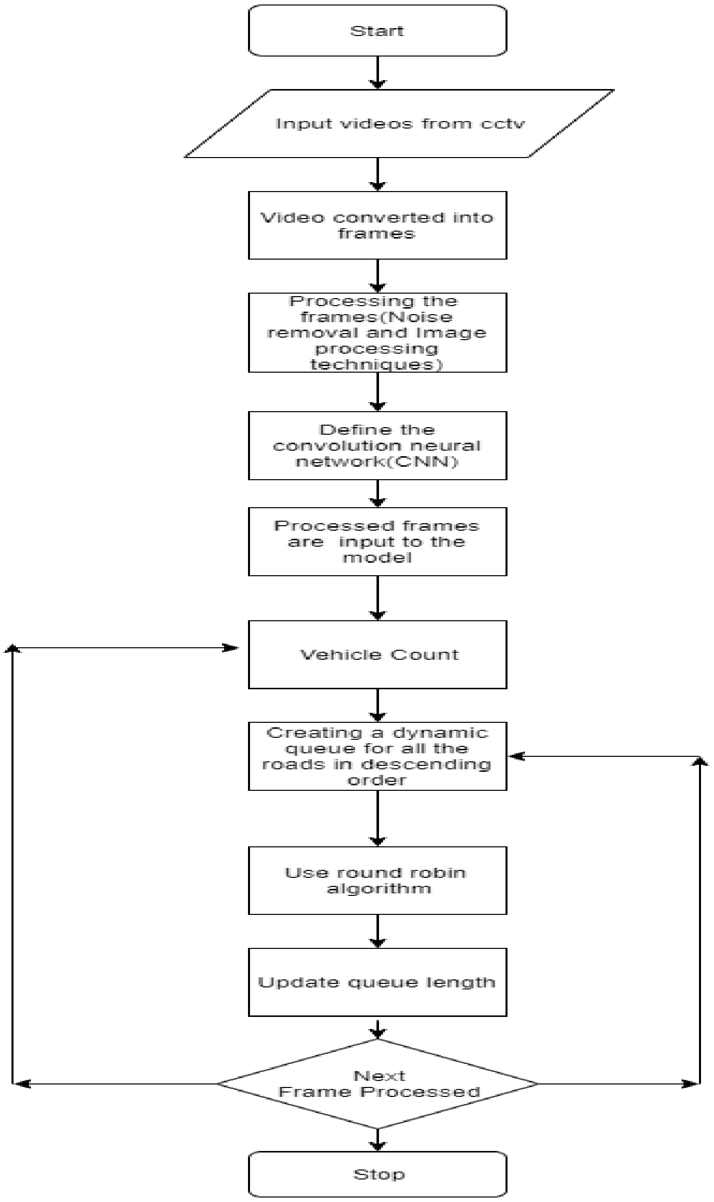

Figure 1: Flowchart

4. Implementation

Pytorch is an open-source machine learning library was used in the implementation. Pytorch works very well with deep learning models and software. The dataset used is COCO dataset. COCO dataset is a large collection of images and is the most widely used dataset for machine learning applications and research. It is a captioned dataset and used for large-scale object detection and segmentation. The dataset consists of 330,000 images with over 200,000 labeled images divided into 80 object and 91 stuff categories. Also, the Darknet which is an open source neural network framework

ISSN: 2233-7857 IJFGCN

and it is written in CUDA and C. It helps in GPU optimization and supports CPU as well. The images were trained on the convolution neural network with pre-trained weights to make predictions.The YOLOv3 weights and the YOLOv3 backbone were used for the ResNet backbone measurements. The classes taken into consideration for the model were included. As the application deals with detection of vehicles, classes of ["bicycle","bus","car","motorbike","truck"] were included for detection and tracking of vehicles. The first step involved was that of loading the dataset and then normalizing it. The torchvision library was used for this purpose. The next step was defining the neural network. A convolutional neural network was defined using the torch.nn module. The pre-trained yolov3 weights were loaded on the neural network. Then next when the model is executed, the video from the cctv is loaded. Next, we need to mark and select the area in which the vehicles will be detected. The model detects and tracks the vehicles, counts them and saves these results and also the result video file. The .txt file with results of the vehicle count stores it with [videoName,id,objectName] format.[8][9][10]

5. Results

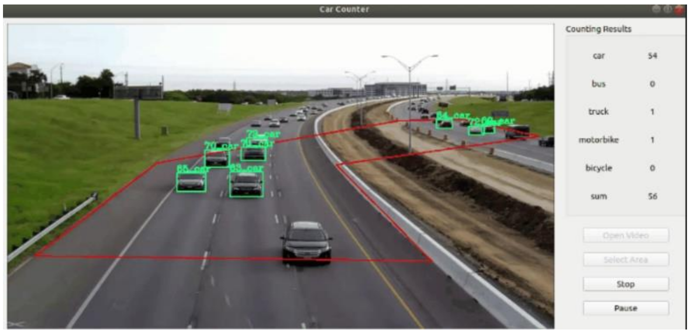

Figure 2: Vehicle Counter

The efficiency of the model is: 0.52000

In the above output image, the selected and marked area is shown in red lines in which the vehicles will be detected. The results show the individual count of the vehicle classes of car, bus, truck, motorbike and bicycle. Also, it gives the sum of all the vehicles which will be used for further purpose.

6. Conclusion

The approach suggested will be able to efficiently detect vehicles, determine traffic density and apply dynamic scheduling algorithms to switch the traffic lights thereby solving the menace of traffic and other related problems. The key points taken into consideration will be working with cameras having low resolution, frame rate, high occlusion to considerably increase the efficiency of detection, accuracy and minimize the errors.

References

1. Akhil Soin, Manisha Chahande, “Moving VEHICLE DETECTION USING DEEP NEURAL NETWORK”, 2017 Amity University Uttar Pradesh Noida, India.
2. Shanghang Zhang, Guanhang Wu , Joao P. Costeira, Jose M. F. Moura, “Understanding Traffic Density from Large-Scale Web Camera Data”, 2017 IEEE Conference on Computer Vision and Pattern Recognition.
3. Jason Kurniawana , Sensa G.S. Syahraa , Chandra K. Dewab , Afiahayati, “Traffic Congestion Detecttion: Learning From CCTV Monitoring Images Using Convolution Neural Networks,” ©2017 IEEE Department of Computer Science and Electronics, Universitas Gadjah Mada, Yogyakarta, Indonesia.
4. R. Krishnamoorthy, Sethu Manickam, “AUTOMATED TRAFFIC MONITORING USING IMAGE VISISON,” Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018) IEEE Xplore Compliant - Part Number: CFP18BAC- ART; ISBN:978-1-5386-1974-2.

5. Busarin Eamthanakul, Mahasak Ketcham, Narumol Chumuang, "The Traffic Congestion Investigating System by Image Processing from CCTV Camera", 978-1-5090-5210-3/17/$31.00 ©2017 IEEE. “pyFirmata”. [Online]. Available: https://github.com/tino/pyFirmata.
6. Sumit Saha, “A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way”, December 15, 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53.
7. “Pytorch Tutorials”. [Online]. Available: https://pytorch.org/tutorials/. “Training a Classifier”. [Online]. Available: https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html.
8. Neuromation, “Tracking Cows with Mask R-CNN and SORT”, June 29, 2018. [Online]. Available: https://medium.com/neuromation-blog/tracking-cows-with-mask-r-cnn-and-sort-fcd4ad68ec4f.

ISSN: 2233-7857 IJFGCN