2019 4th International Conference on Information Systems and Computer Networks (ISCON)
GLA University, Mathura, UP, India. Nov 21-22, 2019

Automating Traffic Signals based on Traffic Density Estimation in Bangalore using YOLO

Nidutt Bhuptani
Dept. of CSE
PES University, EC Campus
Bangalore, India
niduttnb@gmail.com

Akshat Trivedi
Dept. of CSE
PES University, EC Campus
Bangalore, India
akshat.trivedi08@gmail.com

Pooja Agarwal
Dept. of CSE
PES University, EC Campus
Bangalore, India
pooja.atpesse@gmail.com

***Abstract*—In today's world, where the number of vehicles are increasing exponentially, traffic management has become an integral part. We have huge traffic jams at each major junction these days. Sometimes, the jam may even take about 1 hour at each junction. We aim to automate the traffic signal timers based on traffic at the previous junction. So if there is more traffic at the current junction of the road then the timer at next junction will be less, so that the traffic can disperse quickly to suit the needs of the incoming traffic. We use an algorithm called YOLOv3(You Look Only Once)**

***Index Terms*—YOLOv3, Traffic density estimation, F-CNN, histograms**

I. INTRODUCTION

According the Wikipedia, the number of automobiles on road today is somewhere close to 1.3 billion from 1.05 billion in 2010. With the increase in the number of vehicles, the traffic congestion is bound to increase. This implies that cars move very slowly. According to a survey[1] , the average peed of movement of vehicles in Bangalore is just 17.3 km/hr. Regardless of whether traffic is streaming all around gradually, lanes could deal with the traffic stream considerably more effectively; this implies either more traffic in the meantime or a similar traffic in a shorter time. The key for this is all vehicles need to move at steady speed without much use of acceleration or brakes. Along these lines, a wise traffic framework could identify the measure of vehicles at each position, gauge the speed of the autos in a later arrange, and eventually adjust the traffic lights as needs be to get the ideal result.

There are 2 main methods to get a rough estimation of number of vehicles- by hardware and software. Inductive loops [2] and piezoelectric sensors [3] are the 2 main hardware-based solution. Even though they provide high accuracy rates, they are very costly. With the rapid advancement in computer computing performance and the emergence of image processing techniques in the present decade, the software solutions are the best way to count the number of vehicles.

We use YOLO (You Only Look Once) algorithm for the same which is a F-CNN (Fast- Convolutional Neural Network)algorithm. This algorithm identifies the objects in an image and then with the help of predefined weights of vehicles, it undergoes classification whether it is a vehicle or not. We use YOLO to predict what object is present and where. YOLO is refreshingly straightforward. A single convolution network does 2 jobs simultaneously. Firstly, it predicts multiple bounding boxes and then it predicts class probabilities for those boxes. Using of YOLO algorithm has the following advantages [4]
1) YOLO is extremely fast. Since YOLO is considered a regression problem, we dont need a complex pipeline.
2) YOLO has a global view of the image. Unlike region proposal-based techniques, YOLO views the entire image during training and testing time, so it encodes contextual information about classes as well as their appearance. Fast R-CNN botches background patches in an image for objects as it cant see the entire image. YOLO makes less than half the background patches error compared to Fast-CNN.
3) YOLO surpasses top detection methods like DPM (Deformable Parts Model) and R-CNN by a big margin. As YOLO is highly generalized, its chances to break down when applied to new domain or unexpected input decreases [4].

YOLO algorithm has around 80 different classes that it can predict. After YOLO algorithm[13] has counted the number of automobiles in the image, there is a need to automate the signal timers. Our automation process is a class based- when the number of two wheeler are more,the green light is on for considerably lesser time. On the other end, when the number of trucks or buses are more on the road, green light is on for more duration, as they take more time to clear out.

The paper is organized as follows: Section II provides brief explanation of the Literature survey done and Section III talks about the basic terminologies used in this study. Section IV describes the proposed method followed by Outcome of the proposed method in Section V. Lastly, conclusions and future work are presented in Section VI and VII respectively to conclude the paper.

II. LITERATURE SURVEY

Detecting and counting vehicles automatically on highways is a challenging problem. Manually reviewing the large amount of data, they generate is often impractical. Several systems have been implemented and proposed so far for traffic on straight road, in cities, at junctions.

Abhijeet Singh [5] designed an online web-based traffic management system to detect the number of heavy and light

978-1-7281-3651-6/19/$31.00 ©2019 IEEE

weight vehicles. The methodology used is object segmentation using edge detection to identify the object in video frames. The robust object identification using feature extraction will identify the object in bad conditions. The Blob analysis is the part of implementation to identify the type of vehicle (light and heavy) along with calculating the speed of vehicle.

Joseph Redmon and Santosh Divvala [6] presented a new approach for object detection using YOLO. They framed object detection as a regression problem to spatially separated bounding boxes and associated class probabilities. A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation.

Julian Nubert and Nicholas Giai Truong [7] designed a project to introduce and present a machine learning application that aims to improve the quality of life of people in Singapore. They investigate the use of machine learning solutions to tackle the problem of traffic congestion in Singapore. After research into various advanced machine learning methods, they used convolutional neural networks (CNNs).

Ovidiu TOMESCU and Ilona Madalina MOISE [8] presents a new adaptive traffic light control strategy which is improved by the results of analysing the relation between drivers behaviour and the movement of the vehicles or groups of vehicles. They modelled this dependency using MATLAB.

Arif A. Bookseller and Rupali R Jagtap [9] intended to present an improvement in existing traffic control system at intersection. System is made more efficient with addition of intelligence in term of artificial vision, using image processing techniques to estimate actual road traffic[14] and compute time each time for every road before enabling the signal. System is clever enough to provide priority to authorized emergency vehicles with the help of GSM at a particular intersection. This model is resemblance of traditional traffic policeman who takes better decision every time and smoothens traffic flow.

In histogram of oriented gradients (HOG) [12], image is classified based on computing gradient vector of an image. The gradient vector is calculated as:

It uses a detection window that is 64 pixels wide by 128 pixels tall. The HOG descriptor is constructed using 8x8 pixel cells within the detection window. The direction of the gradient vector is computed based on change in the pixel densities from one pixel to the another in X direction as well as in Y direction. Now comes the histogram part of the HOG. The Histogram has 18 bins of 20 degrees each for all the possible 0-360 degrees. Based on the histogram, image can be classied as a vehicle or not a vehicle.

III. TERMINOLOGIES

There are some of the terms which need explanation before understanding the procedure. They are convolutional layer for feature extraction or feature maps, leaky rectified linear Unit (Leaky ReLU) activation function, max pooling, flattened layer, anchor boxes, fully connected layer which makes use of softmax activation function.

A. *Feature Extraction (Feature Maps)*

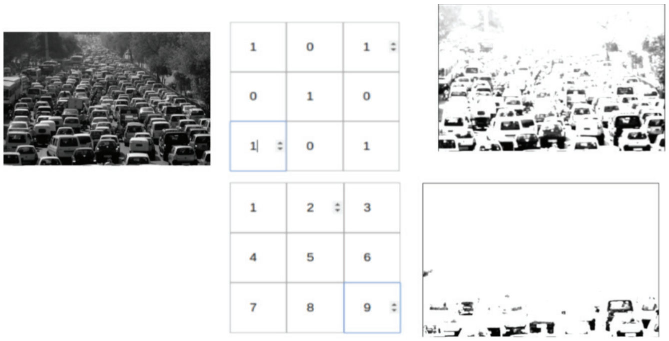
Traditionally convolutional feed forward networks are used which connects the output of the previous (L-1)th convolutional layers to the input of the Lth layer and is termed as Residual network(ResNet) [10]. But it was computationally expensive because storing the pixel densities of an image on each layer and loading all the previous layers repeatedly decreases the performance as the number of convolutional layers increases in the network.

In YOLO, Convolutional is performed based on an input image which is subjected to a filter or a kernel for extracting key objects in an image and simultaneously injecting a bounding box onto that key objects with comparatively high pixel density values. The parameters of the convolutional layer are a set of trained filters and kernels [11].

The input image is in the form of a matrix with its values of pixel densities. The kernel is applied onto the input image matrix which acts as a sliding window which hovers across it. The stride of the sliding window determines the transition or hovering of one sliding window to another determines the number of steps in moving it. The output generated is another matrix which is calculated by the multiplication of values row by row and column by column of an input image and kernel matrix. The output matrix is termed as feature map. In order to get the input image and the feature map of same size padding of the image is inculcated with zeroes around the corners of the feature map.

$$FeatureMapSize = \frac{(n-f)}{s} + 1 \tag{1}$$

In formula (1), ’n’ is the size of the input image, ’f’ is the size of the filter or the kernel matrix and ’s’ is the number of strides where ’n’ is always greater than ’f’. In Fig 1, there is a comparison between the input image and various filter applied to it. The first matrix gives the clear view of the objects as compared to the second feature matrix.



1	0	1
0	1	0
1	0	1

1	2	3
4	5	6
7	8	9

Fig. 1. Various filters for an image

B. *Leaky ReLU Activation Function*

After applying filters to an image, the convolutional layer is then transformed using leaky rectified linear unit activation function which converts all the negative values in the matrix to it’s product with a value ’a’ where a is kept small to 0.001

and keeps the positive values same. The reason behind the negative values is the kernels can be negative resulting in negative feature map but the pixel density of an image will always be positive.

$$f(x) = \begin{cases} x, & x \geq 0 \\ ax, & x < 0 \end{cases} \tag{2}$$

C. Maxpooling

Max pooling is a sample-based discretization process. The key point is to down-sample an input representation (image, convolutional output matrix, etc.), reducing its size or dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned. In figure (2), the maxpooling is implemented in which input image is of size 608 x 608 and the output image is of size 304 x 304 by using a stride of two steps.

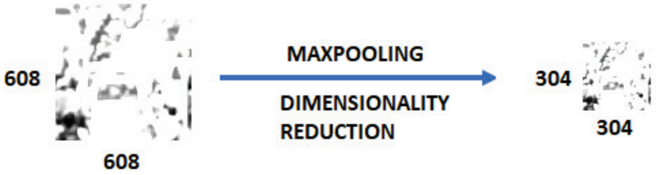


Fig. 2. Maxpooling Illustration

D. Flattened Layer

The Flattening of the convolutional layer means that the two-dimensional matrix representation of the image which is formed after down sampling of the image is then converted into one dimension which acts as a input layer to a fully connected network with two hidden layers. This one-dimensional image form is also termed as a vectorized image.

After reducing the dimensions of the image, the size is 19x19 then the number of nodes in an input layer of the fully connected network is 361.

E. Anchor Boxes

The concept of breaking down the images to grid cells is unique in YOLO, as compared to other object localization solutions. The dimensions of the image are reduced to 19 x 19. Then the image is divided into 3 x 3 grid cells in which each grid cell is a set of values namely:

The probability(pc) whether an object is present inside a grid cell. If there is an object found with the grid cell, then 'pc' is 1 otherwise 0. Secondly, the position of the object in the form of bounding box in an image bx and by along with the height and width of the object 'bh' and 'bw' respectively. Also, based on the number of classes there are class probabilities assigned to it to determine the probability that an object is a car, bus, truck, bicycle or a motorbike. Anchor box makes it possible for the YOLO algorithm to detect multiple objects centred in one grid cell. In Figure (3), the anchor boxes are formed where it has identified the object as car with it's probability.

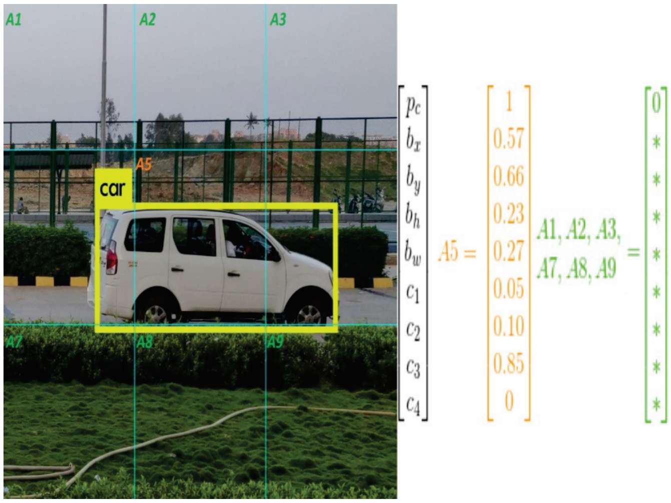


Fig. 3. Anchor Boxes

F. Softmax Activation Function

The softmax activation function is used in calculating the probability of an object inside an anchor box. This probability results into classification of an image inside that anchor box.

The fully connected layer at the end of the network has pretrained weights and input values that are coming in the vectorized form from the image which is subjected to down-sampling. The multiplication of weights and input values results into logits scores. So, the softmax is applied to these logits scores which calculates the probabilities of all the classes in the model. Then, the anchor box with maximum class probability is assigned a class to that object.

$$S\left(y_i\right) = \frac{e^{y_i}}{\sum_j e^{y_j}} \tag{3}$$

In this formula y_i is the logit score belonging to a class and $\sum_j e^{y_j}$ with y_j represents the summation of all the logits score of all the classes in the output layer. So $S(y_i)$ represents the current class probability.

IV. PROPOSED METHOD

Fig (4) shows the basic transformation of an image from its normal format(for eg 2048X1080) to the final 19X19. The raw input image of any size is taken from the traffic signal camera. It is transformed into 608 x 608 dimension then the image is subjected to a repeated process of convolutional layers with feature extraction, leaky ReLU activation function and then maxpooling. Also, simultaneously based on the confidence values above threshold confidence value, bounding boxes are drawn in order to denote objects in an image. Here confidence value is the probabiblity of something being inside a bounding box while the threshold is a predefined minimum accepted probability.

The image is then converted into 19 x 19 after further down sampling of the image. This is known as transformation of the

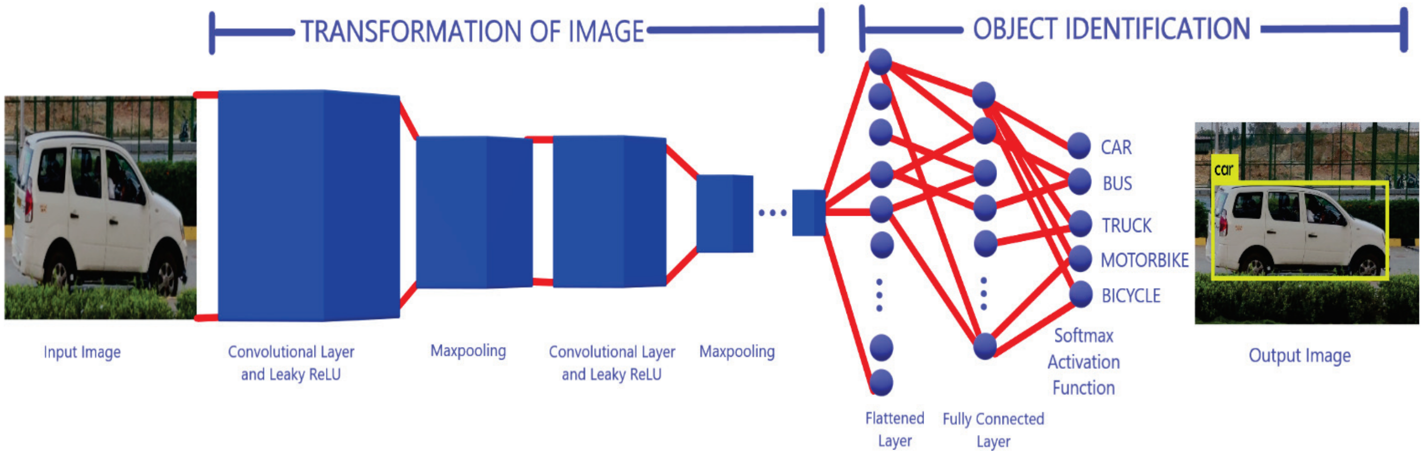


Fig. 4. Convolutional and Fully Connected Layers

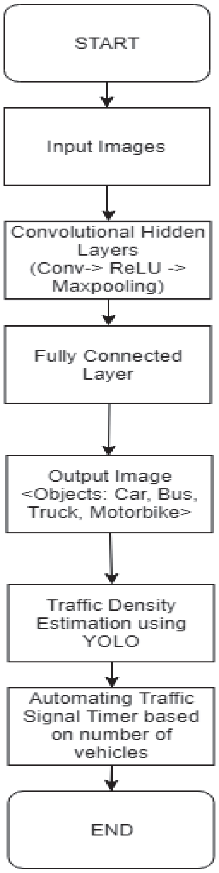

START
Input Images
Convolutional Hidden Layers (Conv-> ReLU -> Maxpooling)
Fully Connected Layer
Output Image <Objects: Car, Bus, Truck, Motorbike>
Traffic Density Estimation using YOLO
Automating Traffic Signal Timer based on number of vehicles
END

Fig. 5. Architecture Flow Diagram

TABLE I
COUNT

Vehicle Type	Count of Vehicle
Car	43
Motorbikes	5
Truck	0
Bus	2
Bicycle	0

image. The reason for the transformation of the image because other algorithms directly implements classification pixel by pixel with the image of the same size. But YOLO classifies the image after reducing the size of the image. So, this makes it computationally more faster than other algorithms. Furthermore, using flattened layer, the image is converted into one dimensional vector form. Then the image is subjected to a fully connected network to predict the class of multiple objects in an image with the help of the anchor boxes within a grid cell. The calculation of the probabilities is achieved using softmax activation function. The vehicle count is determined by detecting multiple objects in an image. The count of the vehicle is furthermore used in automating a traffic signal across a junction.

V. Outcome of the Proposed Method

Now, we need a mechanism to automate the timer at the junction. Let us assume that the road is 2-way i.e. we have 2 roads - road 1 and road2. Our proposed methodology requires the use of a high definition camera to be installed near the traffic light. We require the cameras to capture the images of the traffic at the junction.

Figure (6) shows the transition of an image before and after YOLO algorithm. The algorithm draws bounding boxes around the image. Figure 7 shows the count of different types of vehicles for figure (6). So, we know that figure (6) has 43 cars, 5 motorbikes and 2 buses.

After interacting with several traffic police personnel, we found out that in Bangalore, during peak traffic hours, the time given to green signals is 50 seconds and in non-peak hours it gets reduced to 30 seconds. So we require cameras to capture an image every 10s at the junction.

1) So now let us assume for a moment that the number of vehicles at road 1 is 56 and the number at road 2 is 34. So,

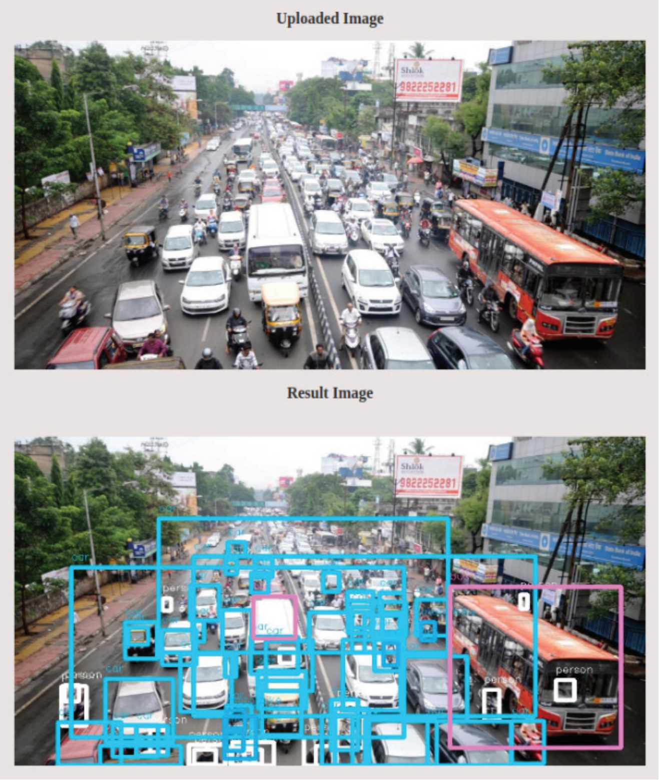
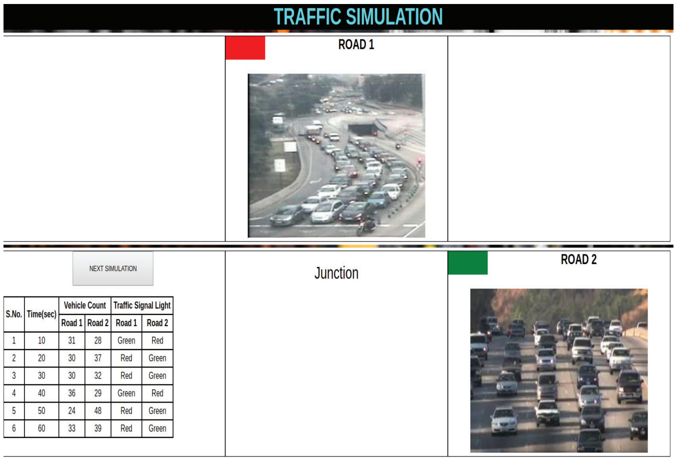


Fig. 6. Application of YOLO on input image



S.No.	Time(sec)	Vehicle Count		Traffic Signal Light	
		Road 1	Road 2	Road 1	Road 2
1	10	31	28	Green	Red
2	20	30	37	Red	Green
3	30	30	32	Red	Green
4	40	36	29	Green	Red
5	50	24	48	Red	Green
6	60	33	39	Red	Green

Fig. 7. Traffic Signal Simulation

TABLE II
SIGNAL AUTOMATION

Time	Number of vehicles at the beginning of cycle		Signal Value	
	R1	R2	R1	R2
0	56	34	Green	Red
10	51	49	Green	Red
20	42	58	Red	Green
30	49	53	Red	Green
40	59	47	Green	Red

we propose to turn on the green lights at road 1 and red at road 2 as the number of vehicles waiting at road 1 is more than number of vehicles at road 2.

2) So, after 10 seconds we again take images and recompute the number of vehicles at both junction as there are some vehicles that have crossed road1. Also, some vehicles would also have been added to the queue at road 1 and road 2.

3) After recomputing of the number of vehicles at time 20 seconds, let us now assume that the number of vehicles at road 1 is 42 and number of vehicles at road 2 is 58. Here the number of vehicles at road 2 have been increased as there are a few vehicles that have just arrived.

4) So, now the number of vehicles at road 1 is lesser then number of vehicles at road 2.

5) Hence, road 1 will have red light and road 2 will have green light. This scheduling method will keep on running infinitely. Simulation of the system is shown in Table number 1

VI. CONCLUSION

The image is taken from traffic signal camera in the city of Bangalore, India. The density of the resultant image is given in table II.

Due to the high accuracy of YOLO algorithm (ninety seven percent), our model is accurately identifying the number of vehicles in the image

The work done can be used as an input for some future work as stated in the next section. The model although brings in a new approach to solve traffic congestion problem, it is not yet tested in real world as doing so will require coordination with various different government agencies like traffic police, traffic control etc.

VII. FUTURE WORK

1) This project can be extended to real time system where we can put a high definition camera near the traffic light where we capture the image of the road in real time every 10 seconds and automate the signal timers accordingly

2) Also, one aspect that our project doesn't consider is the presence of parked vehicles on the road. So if a vehicle is parked our algorithm will consider it as a part of the traffic. However, in future instead of capturing images every 10 seconds we can monitor the traffic via video to know which cars are parked and which are part of the traffic

3) We can also include device an algorithm that assigns time according to the type of vehicles present. For example if more buses are present then obviously the traffic will take more time to clear. So it makes sense to assign more time to it.

REFERENCES

[1] Deccan Herald(2017, December 30), Bengaluru Traffic slowest in India[online], available: https://www.deccanherald.com/content/650859/bengaluru-traffic-slowest-india-report.html

[2] J. Gajda, R. Sroka, M. Stencel, A. Wajda, and T. Zeglen, A vehicle classification based on inductive loop detectors, in Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE, vol. 1. IEEE, 2001, pp. 460464.

[3] D. M. Merhar, Piezoelectric vehicle impact sensor, Oct. 31 1972, uS Patent 3,701,903

[4] Redmon, Joseph, et al. ”You only look once: Unified, real-time object detection.” Proceedings of the IEEE conference on computer vision and pattern recognitioN', 2016.

[5] Abhijeet Singh, Abhijeet Kumar and R. H. Goudar ,et al. Online Traffic Density Estimation and Vehicle Classification Management System,2014

[6] Chakraborty, Pranamesh, et al. ”Traffic congestion detection from camera images using deep convolution neural networks.” Transportation Research Record 2672.45 (2018): 222-231.

[7] Julian Nubert1, Nicholas Giai Truong2, Abel Lim3, Herbert Ilhan Tanujaya3, Leah Lim3, Mai Anh Vu3 et, al. Traffic Density Estimation using a Convolutional Neural Network Machine Learning Project - National University of Singapore,2018

[8] Ovidiu TOMESCU, Ilona Madalina MOISE, Alina Elena STANCIU and Iulian BATROS, et.al ADAPTIVE TRAFFIC LIGHT CONTROL SYSTEM USING ADHOC VEHICULAR COMMUNICATIONS NETWORK ,2012

[9] Arif A. Bookseller and Rupali R Jagtap, et.al Image processing based Adaptive Traffic Control System

[10] Zhan, Hongjian, Shujing Lyu, and Yue Lu. ”Handwritten Digit String Recognition using Convolutional Neural Network.” 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018.

[11] Aloysius, Neena, and M. Geetha. ”A review on deep convolutional neural networks.” 2017 International

[12] Dalal, Navneet, and Bill Triggs. ”Histograms of oriented gradients for human detection.” international Conference on computer vision & Pattern Recognition (CVPR'05). Vol. 1. IEEE Computer Society, 2005.

[13] Asha, C. S., and A. V. Narasimhadhan. ”Vehicle counting for traffic management system using YOLO and correlation filter.” 2018 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT). IEEE, 2018.

[14] Torres, Guillermo, et al. ”Video Surveillance for Road Traffic Monitoring.” XXIV Congreso Argentino de Ciencias de la Computacin (La Plata, 2018) 2018.