International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 12 | Dec 2019 www.irjet.net p-ISSN: 2395-0072

Smart Traffic Control System using YOLO

Pranav Shinde[1], Srinandini Yadav[2], Shivani Rudrake[3], Pravin Kumbhar[4]

[1,2,3,4]*Bachelor of Computer, Dept. of Computer Engineering, BSIOTR College, Maharashtra, India*

Abstract - *Traffic congestion is becoming a serious problem with large number of cars in the roads. Vehicles queue length waiting to be processed at the intersection is rising sharply with the increase of the traffic flow, and the traditional traffic lights cannot efficiently schedule it. A real time traffic light control algorithm based on the traffic flow is proposed in this paper. In fact, we use computer vision and machine learning to have the characteristics of the competing traffic flows at the signalized road intersection. This is done by a state-of-the-art, real-time object detection based on a deep Convolutional Neural Networks called You Only Look Once (YOLO). Then traffic signal phases are optimized according to collected data, mainly queue density and waiting time per vehicle, to enable as much as more vehicles to pass safely with minimum waiting time. YOLO can be implemented on embedded controller using Transfer Learning technique, which makes it possible to perform Deep Neural Network on limited hardware resources.*

***Key Words*: Transfer Learning Technique, YOLO, Deep Neural Network, Computer vision, Convolutional Neural network.**

1. INTRODUCTION

Traffic congestion is a major problem in many cities, and the fixed-cycle light signal controllers are not resolving the high waiting time in the intersection. We see often a policeman managing the movements instead of the traffic light. He see roads status and decides the allowed duration of each direction. This human achievement encourages us to create a smart Traffic light control taking into account the real time traffic condition and smartly manage the intersection. To implement such a system, we need two main parts: eyes to watch the real-time road condition and a brain to process it. A traffic signal system at its core has two major tasks: move as many users through the intersection as possible doing this with as little conflict between these users as possible.

Video is a powerful medium for conveying information and data is plentiful wherever there are cameras. From dash, body, and traffic cams, to YouTube and other social media sites, there is no shortage of video data. Interesting applications that exploit this are ripe for development. One particularly compelling domain where video analytics has tremendous potential is in automated traffic control and public safety systems. The mere presence of automated traffic control systems, like red-light and speed cameras, has been shown to have a positive effect on the reduction of traffic violations. A worldwide analysis of 28 studies has concluded that such systems have reduced crashes by 8% to 50%. Carnis and Blais showed in their research that the initial response to France's implementation of their automated speed enforcement program (ASEP) was a 21% reduction in fatal car accidents and a 26% reduction in non-fatal car accidents.

1.1 Yolo Real-time Object Tracking in Video

YOLO is a fast, accurate object detector, in First, an image is taken and YOLO algorithm is applied. In our example, the image is divided as grids of 3x3 matrixes. We can divide the image into any number grids, depending on the complexity of the image. Once the image is divided, each grid undergoes classification and localization of the object. The objectness or the confidence score of each grid is found. If there is no proper object found in the grid, then the objectness and bounding box value of the grid will be zero or if there found an object in the grid then the objectness will be 1 and the bounding box value will be its corresponding bounding values of the found object. The bounding box prediction is explained as follows. Also, Anchor boxes are used to increase the accuracy of object detection which also explained below in detail.

2. PROPOSED SYSTEM ARCHITECTURE:

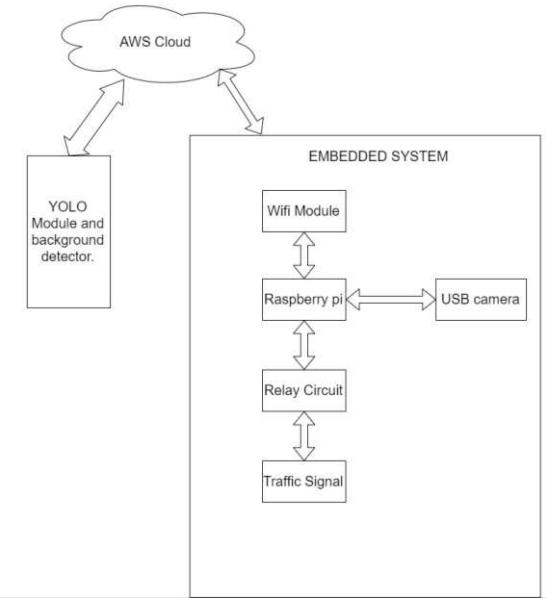


Fig -1: Architecture Diagram

The above figure signify the working of module using cloud, sever and embedded system. The background detector takes help of YOLO framework for object detection. In our system we have manipulated YOLO for only detecting vechiles of different categories i.e Truck Car Bike etc. The Yolo is deployed on sever which is connected to embedded system through AWS cloud. The Raspberry pi board is attached to onsite camera and the relay attached to signals. As we need internet to connect raspberry pi with cloud we will have to connect it using wifi module attached to it. Raspberry pi helps to transfer video from camera to cloud. The background detector comprises of algorithm with YOLO module to make out lane density. It is smart way with great state of art YOLO framework to manage the control flow of traffic.

2.1 Hardware Module:

a) Raspberry Pi

b) Sony IMX219 sensor

2.2. Software Module:

a) Flask API.

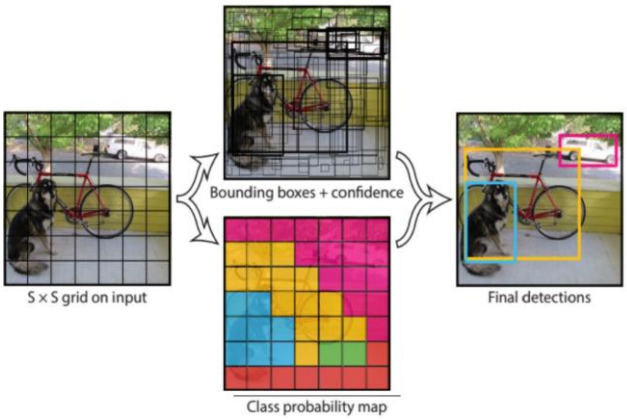
b) AWS Cloud.

c) Anaconda

d) REST API

3. PROPOSED SYSTEM TECHNIQUE's :

3.1 YOLO:



Object detection is one of the classical problems in computer vision where you work to recognize what and where — specifically what objects are inside a given image and also where they are in the image. The problem of object detection is more complex than classification, which also can recognize objects but doesn't indicate where the object is located in the image. In addition, classification doesn't work on images containing more than one object. YOLO uses a totally different approach. YOLO is a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. YOLO is popular because it achieves high accuracy while also being able to run in real-time. The algorithm "only looks once" at the image in the sense that it requires only one forward propagation pass through the neural network to make predictions.

Our network has 24 convolutional layers followed by 2 fully connected layers. Instead of the inception modules used by GoogLeNet, we simply use 1 × 1 reduction layers followed by 3 × 3 convolutional layers .Fast YOLO uses a neural network with fewer convolutional layers (9 instead of 24) and fewer filters in those layers. Other than the size of the network, all training and testing parameters are the same between YOLO and Fast YOLO.

3.2 DNN :

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks.

DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives.[106] The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network.

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network did not accurately recognize a particular pattern, an algorithm would adjust the weights.[107] That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data.

3.3 CNN:-

A convolutional neural network consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a RELU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution. The final convolution, in turn, often involves backpropagation in order to more accurately weight the end product.

Though the layers are colloquially referred to as convolutions, this is only by convention. Mathematically, it is technically a sliding dot product or cross-correlation. This has significance for the indices in the matrix, in that it affects how weight is determined at a specific index point.

4. Proposed System Control Flow:-

Initially the video will be streaming on all four roads of the traffic circle. The values will be read frame by frame in the streaming video of these roads. Camera sends all the captured input videos to the processing engine on the cloud. Once the video are received by cloud the background algorithm filters the video to count the density of vehicles in every lane. Then based on threshold values each lane is prioritized and hence timer is set appropriately. Yolo helps to count vehicle in real time with the speed of 24fps and combining it with background algorithm .Density is counted on the basis of vehicle category for transport vehicle and containers will have more density as compared to vehicle. Here the total density is counted for every lane. There is a threshold value based on two criteria one is the density count of the lane and other is priority of that lane.

Consider lane L1, L2, L3, L4 every lane will have the equal time Tmax which signify till what maximum time green signal of particular lane will be lit. The signals will turn green lane by lane so that there wont be longer waiting time for any lane with less density. If there is very less density means if the lane is empty then no green signal will be turned on for that lane till its density count reaches the threshold. The video of the lanes which have red signal will be recorded and sent to cloud for processing, first the video is formatted according to resolution of S x S size. Then the yolo with the background detector performs Object detection on it then density count is calculated and priorities of lane are fixed. In this way it will be efficient for automated traffic control using raspberry pi with Deep learning.

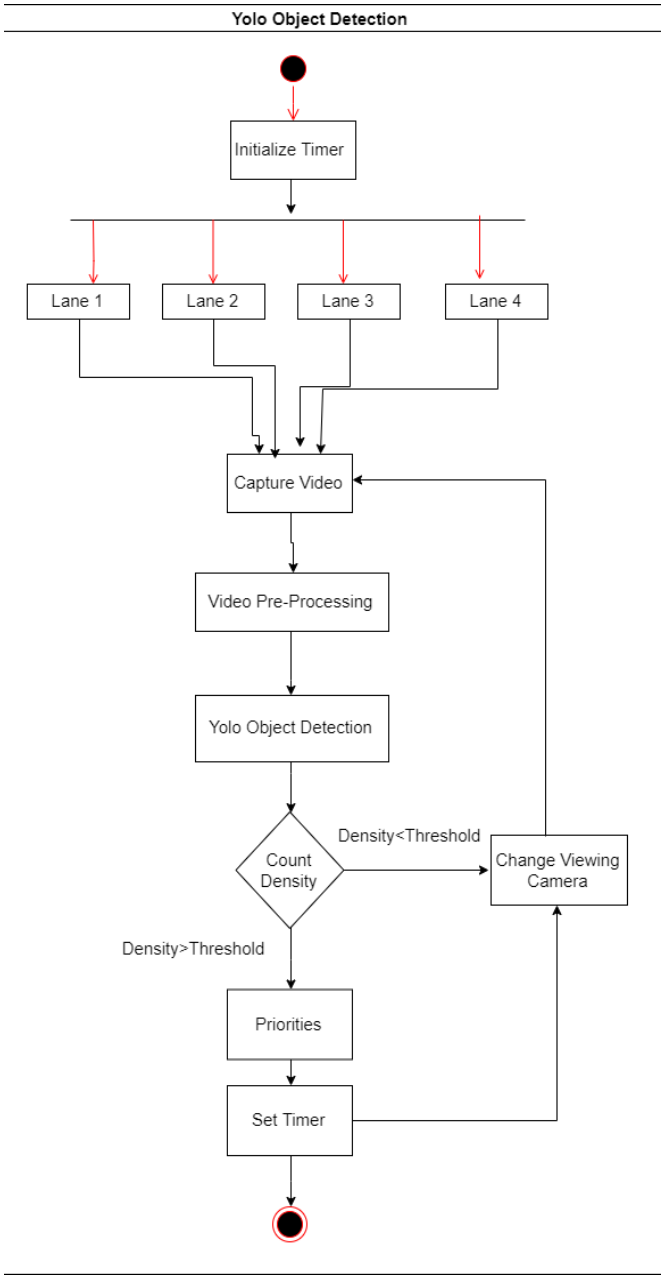


Fig:- Proposed System Control Flow

3. CONCLUSIONS

The goal of this work is to improve intelligent transport systems by developing a Self-adaptive algorithm to control road traffic based on deep Learning. This new system facilitates the movement of cars in intersections, resulting in reducing congestion, less CO2 emissions, etc. The richness that video data provides highlights the importance of advancing the state-of-the-art in object detection, classification and tracking for real-time applications. There has been a steady progression of image detection techniques beginning with feature descriptors like HOG and, more recently, deep network-based approaches like Faster R-CNN and YOLO. YOLO provides extremely fast inference speed

with slight compromise in accuracy, especially at lower resolutions and with smaller objects. While real-time inference is possible, applications that utilize edge devices still require improvements in either the architecture's design or edge device's hardware.

Finally, we have proposed new algorithm taking this real-time data from YOLO and optimizing phases in order to reduce cars waiting time.

ACKNOWLEDGEMENT

The authors would like to thank Mrs. Sanchika Bajpai and Mrs. Gauri Virkar for their valuable contribution to this project.

REFERENCES

1. G. Dimitrakopoulos and P. Demestichas, "Intelligent transportation systems," IEEE Vehicular Technology Magazine, vol. 5, no. 1, pp. 77–84, 2010.

2. K. Morinaga, "Car navigation system," May 16 1995, uS Patent 5,416,478.

3. K. H. Molloy, J. P. Ward, and V. M. Benson, "Traffic signal control system," Jul. 4 1972, uS Patent 3,675,196.

4. R. Lotufo, A. Morgan, and A. Johnson, "Automatic number-plate recognition," in Image Analysis for Transport Applications, IEE Colloquium on. IET, 1990, pp. 6–1.

5. M. Cremer and M. Papageorgiou, "Parameter identification for a traffic flow model," Automatica, vol. 17, no. 6, pp. 837–843, 1981.

6. 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. 460–464.

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

8. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in European conference on computer vision. Springer, 2014, pp. 740–755.