Deep Learning-Based Detection for Traffic Control

Yiou Yang
Department of Electrical and Information Engineering, University of Southern California, Beijing, China
saa200125@163.com

ABSTRACT

Often urban intersections have a problem with long queues, but the traffic flow during the green light is relatively small. In this paper, we propose an integrated, machine-vision control mode process for controlling traffic lights. The method proposed in this paper is to obtain real-time video images of the road through the monitoring camera, detect the number of vehicles in the different areas, and dynamically control the traffic signal. Our real-time system fundamentally guarantees the smooth traffic flow through the intersection to avoid frequent traffic-congestion.

CCS CONCEPTS

• **Computing methodologies** → Machine learning approaches.

KEYWORDS

Deep learning, Traffic control, Object detection

ACM Reference Format:

Yiou Yang. 2021. Deep Learning-Based Detection for Traffic Control. In *2021 The 5th International Conference on Advances in Artificial Intelligence (ICAAI) (ICAAI 2021), November 20–22, 2021, Virtual Event, United Kingdom.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3505711.3505736

1 INTRODUCTION

The advanced urban traffic control system, with less investment and quick effect, is one of the essential ways to improve the operation of urban traffic [1]. It is also an important symbol of urban modernization. More than 100 years ago, people began to study traffic signals to control the order of vehicles entering the intersection. In 1868, the appearance of gas signal lights in London marked the official use of urban traffic signals. In 1913, the earliest traffic signal control device appeared in Cleveland, Ohio. In 1926, Chicago adopted a traffic light control scheme. Each intersection has a unique traffic light control, which adapts to single traffic flow. Since then, the development of traffic control technology, related control technology, and related control algorithm has gradually improved the safety, effectiveness, and impact on the environment of the control. The traffic signal control is changing from manual to automatic; the signal cycle from fixed time to the variable time, From the appearance of the vehicle detector, the traffic signal control has experienced nearly a hundred years of development. In 1963, Toronto, Canada, established a set of computer systems for centralized, coordinated, and inductive control of traffic signal control systems, lead the urban road traffic signal control system into a new stage of development.

In China, the research of the traffic signal control system started late, and there were only a few intersections with a single point and fixed period control in a long time before and after liberation. It was not until the 1970s that relevant units began to study this aspect, and began to carry out induction and periodic signal control one after another. However, due to the relatively backward development capacity at that time and the factors such as mixed traffic of motor vehicles and non-motor vehicles, most of the introduced systems can only implement a single point, and multi-period control and rarely can carry out coordinated trunk control. However, the damage of testing equipment and the delay of system maintenance also affect the control effect to a certain extent. This year, to solve the increasingly prominent urban traffic problems [2], many companies specialized in the research and development of the traffic control systems have emerged in China. Through their research, or cooperation with Tsinghua University, Jilin University, Tongji University and other institutions of higher learning, they have developed many urban traffic control systems with independent intellectual property rights, and these products are gradually being used in various cities.

The existing traffic light control system is mainly divided into two categories, timing control, and induction control. Timing control cannot adapt to the real-time change of traffic flow. When the traffic flow of one section is substantial, it needs to wait for the red light, while the traffic flow of the other part is minimal, but it shows the green light, and there are some problems such as the time occupation of the empty road. This phenomenon of unreasonable traffic light timing causes a lot of energy consumption and also makes transportation efficiency low. Inductive control is easy to be disturbed by the environment, and in the process of installation, it will cause damage to the road, making construction difficult and costly. There are four main problems in the current traffic light control strategy: The Unreasonable phase release time, Unreasonable setting of the green light interval, Poor coordination between intersections, the release sequence of signal lights is not uniform [3-4].

The traffic information collection technology of machine vision [5], also known as video traffic information collection technology, is to use video, computer, and modern communication technology to achieve the collection of dynamic traffic information. The system collects traffic images through the installation of the camera on the line pole or bridge and then carries out image processing to obtain the traffic flow, instantaneous speed, the statistical average value of the speed within the specified time period. Vehicle type classification, occupancy, average distance, detection of traffic accidents, and other traffic dynamic information, to provide real-time traffic

ICAAI 2021, November 20–22, 2021, Virtual Event, United Kingdom

ACM ISBN 978-1-4503-9069-9/21/11…$15.00
https://doi.org/10.1145/3505711.3505736

dynamic information for traffic signal control, information release, traffic guidance, and command.

Machine vision acquisition method, easy to install, the camera can cover a wide area, a camera can cover up to six lanes, for a real sense of large area detection [6]. In its system, it is more suitable to use video method in highway traffic flow, vehicle type classification statistics, and vehicle speed data collection, but if it is used in more traffic investigation, such as travel information, it is no longer suitable. The advantage of machine vision acquisition technology is that it can set detection area in the detected image, which will not interrupt the traffic like the maintenance and repair of ground induction coil buried in the road; the equipment installation is simple and convenient, and the installation and maintenance process does not need to close the lane, do not need to excavate the road surface, and the cost is relatively low. The disadvantages are: it is easy to be affected by weather, light change, shadow, occlusion, and other conditions, such as in fog and thunder weather, the video image collected is not clear enough, it is difficult to extract vehicle information, and reduce the detection accuracy. The recent development of deep learning enables a more robust, accurate object detection method [7]. The existing traffic control system that adopt deep learning method using the state-of-the-art detection model that trained on large detection dataset. However, those datasets usually contain images take from the frontal view, which is not very desired in this scenario. Some of the previous work utilize the existing neural network architecture and training it with images from the surveillance camera images. It requires a large amount of labeled data to ensure the accuracy of the model. In addition, it is not economical to use the architecture that used to learn large amount of classes in this case.

In this paper, we proposed a system to detect traffic conditions in terms of traffic density and a traffic light control strategy. The traffic condition detection system can also be used with other traffic light control system that requires similar traffic condition input. We propose a neural network architecture that similar to Tiny-Yolov3, but with less filters in the deeper layers. The intuition is that the number of class for this application is much smaller than the COCO dataset, so using the same architecture would likely have those filters learn the same features in the deeper level. The evaluation results show that our proposed architecture achieve better accuracy and more stable training in the aerial dataset.

2 DETECTION METHOD

2.1 RCNN

RCNN (Regions with CNN features) [10] is a milestone in the application of CNN method to target detection. With the help of good feature extraction and classification performance of CNN, the problem of target detection can be transformed by the Region Proposal method. The algorithm can be divided into the following four steps. Candidate region selection, feature extraction, classification, and boundary regression. Region Proposal is a traditional method of region extraction, which can be seen as sliding windows with different width and height. We can get potential target images by sliding windows. The target images extracted by the proposal are normalized as on CNN. After the classification decision is made, the next is to get the precise target region by bounding-box regression.

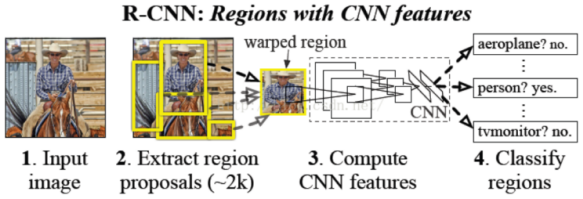

Figure 1: Illustration of R-CNN detection method

Because the actual target will generate many sub-regions, the aim is to determine the future target for completing the classification precisely. Bit and merged to avoid multiple detections.

The process of R-CNN is simple and direct, and the training convolutional neural network is creatively used to replace the artificial feature extraction algorithm, which directly improves the detection accuracy by more than ten percentage points. The disadvantage of R-CNN is also very obvious. First, the amount of computation is too large. In an input picture, the number of valid candidate boxes obtained by Selective Search is generally more than 1000. This means that more than 1000 forward operations of convolutional neural network are needed, which is very time-consuming, and the detection speed cannot be guaranteed. Secondly, because of the R-CNN design idea of four steps, the selection and adjustment strategy of candidate box, the training of convolution neural network, the training of SVM classifier and the final regression operation training, all of them need to be trained independently, and all the features need to be retained, which leads to the training time-consuming and space-consuming, and the flexibility of the algorithm is not enough.

With the deepening of the research, Fast R-CNN and Faster R-CNN have proposed one after another. By extracting candidate box strategy and optimizing the training of the classifier, they have improved the shorthand of R-CNN, such as large computational load and difficult training. They have reached a high level of detection accuracy and become the mainstream of deep learning target detection algorithm.

2.2 Hog Feature Approach

HOG is a common way to describe image local features in the field of computer vision and pattern recognition. Firstly, it will calculate the gradient values in different directions of an image, then accumulated to get the histogram, which can represent the characteristics of this region. Finally, these characteristics can be input into SVM classifier.

Compared with extracting the feature vector of each pixel and using SVM to process the whole image with the neural network [11], the YOLO method only needs to process the image once, while SVM + HOG scheme needs to process about 150 times. This makes the processing efficiency of YOLO 20 times faster than SVM +HOG, and its detection threshold can be set to any confidence level. In terms of accuracy, the recognition accuracy of HOG and Yolo is relatively high in normal environment, which can reach more than 90%. However, when the environment becomes complex, Yolo's advantage is obvious. Because HOG counts the color information

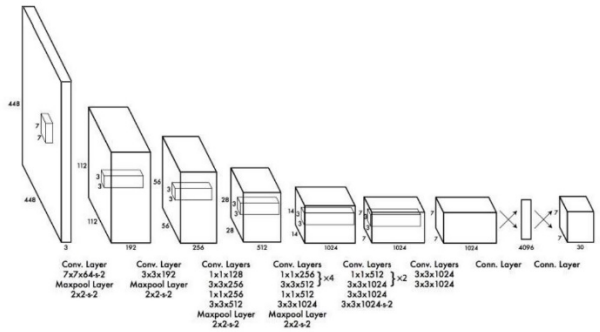

Figure 2: Architecture of YOLO network. There are 24 convolutional layers and two fully connected layers in the detection network.

of each pixel and obtains the feature vector by merging blocks, so when light are not enough or the vehicle is too crowded, there will be some false detection and missing detection. However, YOLO can avoid these problems. In order to meet the requirements of real-time and accuracy of vehicle detection, using YOLO will make a better choice.

2.3 YOLO

YOLO algorithm (You Only Look Once)[12][13] is a deep neural network model for object detection. The target detection task consists of two parts: 1) classifying these objects; 2) identifying the location of the object in the image. Previous methods such as R-CNN and its derivatives used multiple steps to complete object detection, and each independent module had to be trained independently. As a result, the running speed is slow and the training of the neural network is difficult to optimize. YOLO uses the end-to-end design idea to reconstruct object detection into a single regression problem. The object coordinates and classification probability are obtained directly from the pixel data of the image.

YOLO algorithm brings a new solution to the target detection task. It combines location and classification tasks, scans images once, extracts features and classifies location using depth neural network. The speed of image detection meets the requirements of real-time detection.

The major breakthrough of YOLO algorithm lies in the great improvement of detection speed. Based on the previous R-CNN framework, it uses CNN designed for target detection tasks to extract features and then use a full connection layer to classify and detect the identified targets. The network structure model of YOLO is shown in Figure 2, which consists of an input layer, convolution layer, pooling layer, and full connection layer.

The input layer of YOLO is the data obtained by clipping, normalizing, or data enhancement of the input image. In CNN, the geometric features of sample image data after some processing are called feature maps. The input layer can be considered as the initial feature map. Because the target detection needs more detailed information, YOLO unifies the size of the processed feature map to 448*448*3. Among them, 448*448 is a single dimension image pixel value. Because the picture is color, it needs three color channels, red, green and blue, to be superimposed on each other.

Then there are 24 convolution layers. The main operation is to convolute the feature map processed by the input layer. Its essence is to extract the feature information of the input layer for subsequent classification and location processing. As shown in Figure 2, YOLO's convolution cores are 3*3 and 1*1. The 1*1 convolution core is used to reduce the number of channels in the convolution core, so as to reduce the parameters generated by the network.

The pooling layer exists between convolution layers, and its main operation is to down-sample the input data samples in the feature space. That is to say, according to the spatial position of the feature matrix, according to the set granularity block, the feature is partitioned, and the new eigenvalue is calculated in the small block to replace the information in the original block. According to the rule of replacing new eigenvalues, the common down-sampling operations are mean pooling and maximum pooling. YOLO algorithm uses the maximum pooling method, even replacing the original feature block with the maximum value in the block. YOLO has two full connection layers between the last pooling layer and the output layer. Its main function is to transform the two-dimensional matrix extracted from a feature into a one-dimensional matrix. By linking all inputs with network parameters, it is the layer with the most parameters and the largest amount of computation in the network.

The last layer of the network is the output layer, which is equivalent to a classifier. It classifies and outputs one-dimensional vectors from the full connection layer. The number of output feature graphs is the number of target classifications. The final output of the network is a 7*7*30 one-dimensional vector, which contains the classification results of the objects in the picture and the coding of their location information. Finally, the detection results can be drawn in the original picture by decoding the vector in a unified way.

2.4 YOLO Detection Process

YOLO divides the input image into S by S meshes, each of which is responsible for detecting the target object whose center point falls in it. There are B target boundaries in a single grid, each of which consists of a five-dimensional prediction parameter, including the center coordinates (x, y) width (w, h) and confidence score s_i.

The confidence score is calculated by equation

$$s_i = \Pr(0) * IoU \tag{1}$$

In the formula, Pr(0) Denotes the possibility of objects in the current grid target border, and 0 denotes the target object. IoU (Intersection over Union) shows the accuracy of the target border position predicted by the current model. Suppose the predicted target border is p, and the real target border is t, box_t. represents the border condition of the real object in the image, box_p. Represents the target border of the prediction. Then IoU is calculated by:

$$IoU_p^t = \frac{box_p \cap box_t}{box_p \cup box_t} \tag{2}$$

$\Pr(C_i|O)$ denotes the posterior probability of a certain kind of object i in the presence of a target in the border. Assuming that there are K objects in the target detection task, the conditional probability of predicting the first i object C_i for each grid is $\Pr(C_i|O)$, i=1,2,... K.

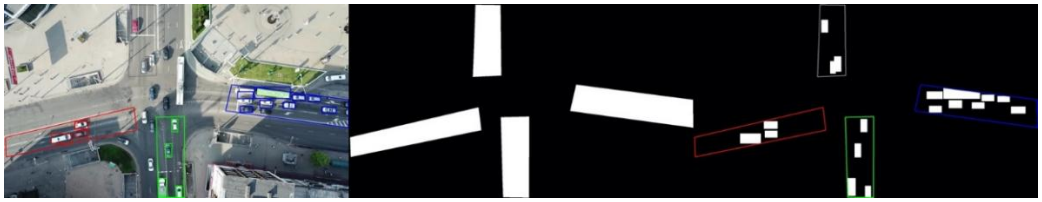

Figure 3: Segmentation Result. The left image is the detection result. The middle image is the road segmentation. On the right is the intersection of detected vehicle and roads. Then the proportion of road occupation is calculated from it.

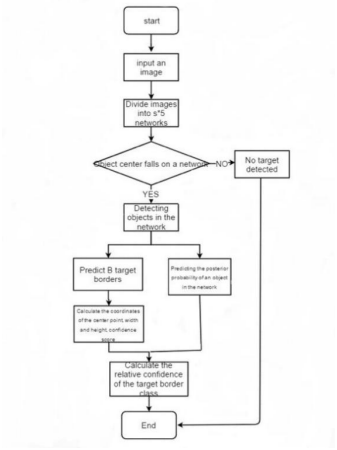

Figure 4: Illustration of YOLO detection process. The network is pre-trained with COCO dataset which contains 5 common target objects on the traffic: bicycle, motorcycle, car, bus, and truck.

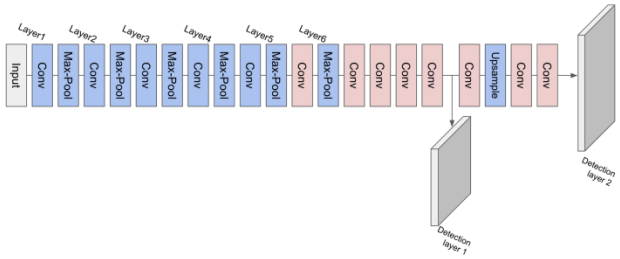

Figure 5: Our network architecture based on the tiny-YOLOv3. The weight of first five convolutional layers are initialized by the weight pretrained with COCO dataset and keep them constant during training since the lower level features should be less variation for different datasets. For the deeper level of network, we reduced number of filters in each layer since we have less class than the COCO datasets. The modified network is smaller and faster than the tiny-YOLOv3 yet achieve better performance with aerial-car dataset.

After calculating $\Pr(C_i|O)$, the confidence of the object in a target frame can be calculated in the test, as shown in the equation.

$$\Pr(C_i|O) * \Pr(O) * IoU_p^t = \Pr(C_i) * IoU_p^t \quad (3)$$

In YOLO algorithm, the input image is divided into a 7 by 7 grid. Each grid predicts 2 target boundaries. There are 5 targets to be measured, namely S=7, B=2, K=5. So the final output of the algorithm is a predicted result vector whose length is $S \times S \times (B \times 5 + K) = 7 \times 7 \times 30$. The overall flow chart of the detection model is shown in the Figure 4

The proposed network has similar architecture as tiny-YOLOv3 as shown in Figure 5. We initialize the network which conform with the tiny-YOLOv3 with its weight. During training, we freeze the parameter of the first five convolution layers and train the rest of the network to adapt the new sample domain.

2.5 Image Segmentation

We will use the mask method to divide the road and divide the intersection into two directions: east to west, west to east, south to north, and north to south. The mask is a combination of 0 and 1. Binary image. When a mask is applied in a certain function, the 1-value area is processed, and the masked 0-value area is not included in the calculation. The image mask is defined by a specified data value, data range, finite or infinite value, region of interest, and annotation file, which allows us to segment the intersection area we want. Finally, the YOLO algorithm is detected on the divided pictures to obtain the number of vehicles at the four intersections. There is a flaw for the counting based traffic light control logic, that is the area of road captured by the camera could be different. So determines the traffic condition solely based on number of vehicles on the road will not be very reliable. Therefore, we add an additional metric based on the detection result to evaluate the relative traffic density. The existing dense estimation method training neural network with images with density level as label. There are two main problem with that approach. First of all, it is hard to compare the density of two road with the same prediction label. Also, the network may not work well with different dataset domain, since the network essentially learns the feature of the car and backgrounds, but the background could be very different on the other dataset. In the light of our detection system, the density could be easily approximate with the intersection of detected cars and road segmentation. As shown in Figure 6, we form a mask with all the detection results that falls into the designated road segment. Then the area of intersection of those cars with road could be used to represent the percentage of occupation.

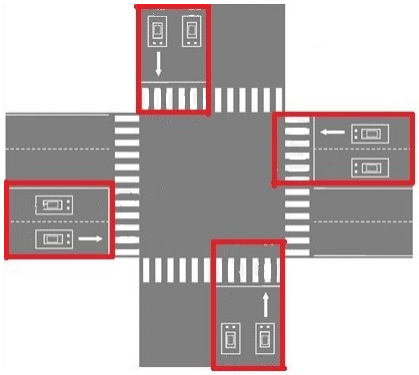

Figure 6: The road segmentation. The road in the image is divided into up to different regions depends on the coverage of the camera. The number of target objects detected in each region can be used in the traffic light control algorithm.

3 TRAFFIC LIGHT CONTROL

The result of vehicle detection can be used in varies kinds of traffic control algorithms. In this paper, we provided a control logic as an example. A more advanced algorithm for traffic light control with reinforcement learning requires the training data can also apply this algorithm as input data processing stage. Our detection algorithm designed with a finite state machine. We design the algorithm for the crossroad scenario, marks the road as east-west, and north-south as shown in Figure 6. There are six states in the finite state machine. Four basic states are east-west go, north-south go, east-west yellow, north-south yellow. If there is a roughly equal number of vehicles in both directions, the traffic light will cycle between these four states. If there is no vehicle on one of the directions, the traffic light will keep the other lane green until there is a vehicle detected. In addition, if there is significantly more vehicle in one direction than the other, the traffic light will hold green in that direction for one more clock time to reduce the traffic density.

With that logic, there are five different transition conditions. T1 and T2 corresponding to the condition there is no vehicle on the south-north direction and east-west direction. T3 and T4 mean there are many more vehicles detected in one direction than the other. T5 indicates the number of detected vehicles are close in two directions. In this case, the traffic light will cycle between the 6 basic states as described above.

4 EVALUATIONS

We evaluate our model on the aerial-cars-dataset [14]. The dataset contains images took from the drone directly above the crossroad. The dataset contains 335 images and we split it to the train and valid set. The training set contains 275 images and valid set has 60 images. There are two problem with dense estimation based on vehicle counting, first is the portion of the road inside the scene is different. And there are vehicles with different sizes. Therefore, the solely counting based detection is not be reliable under some circumstances. The original YOLO model that trained with COCO dataset perform bad on this dataset due to the variation of the view angle.

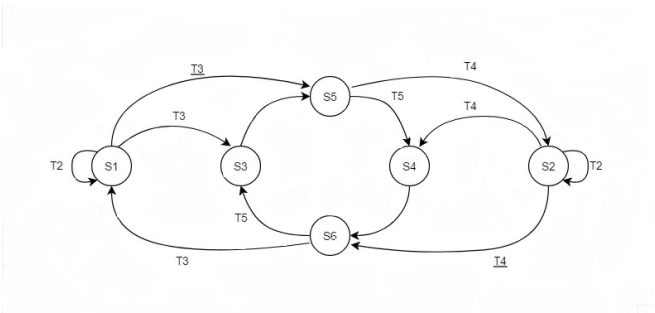

Figure 7: Finite state machine for traffic light control. State 1 east-west direction traffic light green, north-south direction traffic light red, hold; State 2 east-west direction traffic light red, north-south direction traffic light green, hold; State 3 east-west direction traffic light green, north-south direction traffic light red, not hold; State 4 east-west direction traffic light red, north-south direction traffic light green, not hold; State 5 east-west direction traffic light yellow, north-south direction traffic light red. State 6 east-west direction traffic light red, north-south direction traffic light yellow.

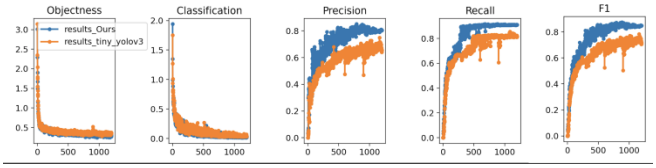

Figure 8: Training Results of Our architecture and tiny-YOLOv3. In both cases, the network trained for 1200 epochs. The modified architecture trained faster and achieved better performance.

We compare our proposed architecture with the tiny-YOLOv3 that trained with same dataset and iterations. Our proposed architecture yields both better performance and fast training process. Note that the accuracy of the road density estimation is highly correlated with the detection accuracy and recall. Therefore, we only present the evalution of detection for the simple comparison.

To further testing the counting system, we use the video footage of the public surveillance camera from EarthCam and Cerevo live camera. Because most of the cameras are not covering all the road of the intersection, the result from multiple cameras can be combined as the input to the traffic control system such as the camera on the Route 66. To estimate the traffic density, we segment the road to equal size areas.

Experimental results show that YOLO algorithm has a good performance in testing the images with the camera close to the intersection and high pixels. The visual effect shows that using the camera on the traffic light for vehicle detection will have higher accuracy, and individual vehicles will be missed in the scenes shot by the camera that is relatively far away from the test.

Although the number of vehicles at intersections is dynamic, we can detect vehicles by periodic detection of surveillance video. As shown in the figure, even in the case of high traffic density, YOLO

Table 1: Comparion between Our modified architecture and tiny-YOLOv3.

	Precision	Recall	mAP	F1	GIoU	objective	Class loss
Pretrained tiny-YOLOv3	< 0.01	< 0.01	< 0.01	0	6.9	3.24	7.28
tiny-YOLOv3	0.655	0.818	0.834	0.72	1.18	0.51	0.104
Our architecture	0.805	0.911	0.904	0.85	1.36	0.44	0.0906

Table 2: Quantitate Experiment Result

Dataset	Number of Testing Image	Precision	MSE
Route 66 West	15	0.93	0.2142
Route 66 East	15	0.91	0.5714
Akihabara Camera	15	0.88	0.6428

Figure 9: Detection Result from the aerial vehicle dataset.

Figure 10: Detection Result from Route 66 West

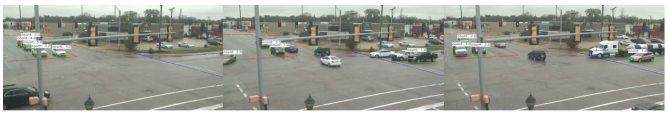

Figure 11: Detection Result from Route 66 East

Figure 12: Detection Result from Akihabara Live Camera

algorithm can still achieve high accuracy. Overall, the average accuracy of this algorithm is more than 80%, which can more accurately reflect the traffic situation of the intersection.

5 CONCLUSION

In this paper, aiming at the problem of traffic flow detection at intersections, using YOLO target detection algorithm in deep learning, by analyzing the parameters of YOLOv2 model, training and comparing different network structures according to the characteristics of vehicle target and its motion, a YOLOv2 model which can obtain more accurate detection results is extracted. We analyse the detection task and proposed a network architecture that is lighter than original YOLO. The proposed network is faster than tiny-YOLOv3 which has the achieve 220 FPS on COCO dataset, 5.56 BFLOPS on 416 by 416 images. So this architecture is suitable for the real-time application running on the modern embedded system like Raspberry PI. The test results show that the model can accurately detect the number of vehicles at the designated intersection, thus completing the task of traffic flow monitoring at the designated intersection.

Based on this model, the design of an intelligent control system of traffic light duration can automatically adjust the traffic signal time of each road section according to the traffic flow of the road section. It can solve the problem of traffic jam and resource waste caused by the longtime of green light in the less traffic section but the longtime of red light in the more traffic section of the traditional control system, effectively improve the traffic efficiency and relieve the traffic pressure of the city. The design scheme has the advantages of low cost, convenient deployment, stable performance and high reliability, and has good popularization significance and practical value.

REFERENCES

[1] Y. Duo, Z. Li, and X. Liu, "Theory and application of urban intelligent traffic control," (in Chinese), China Water Conservancy and Hydropower Press. 2011. 5.

[2] J. Guo, D. Liu, and L. Yu, "Understanding of traffic congestion in big cities of China," (in Chinese), Urban traffic, vol. 9,no. 2, pp. 8 – 14, 2011.

[3] H. Lu, Z. Hai, and F. Lan, "Intelligent traffic light control system based on single chip microcomputer," (in Chinese), Electronic technology and software engineering, vol 5, no. 3,pp. 51 – 53, 2016.

[4] Z. Zhang, T. Li, "Application of image recognition in the field of Intelligent Transportation" (in Chinese), Wireless Internet technology, no. 16, pp. 139 -140, 2018.

[5] Y. Chen, D. Wang, "Analysis and application of intelligent traffic information collection," (in Chinese), People's Communications Press. 2011. 12.

[6] X. Guo, "Design of intelligent traffic light controller based on video recognition," (in Chinese), World of electronic products, vol 19, no. 1, pp. 74 – 75, 2012.

[7] Y. Yang, L. Huang, and C. Liu, "Vehicle queue length based on video analysis," (in Chinese), vol 28, no. 3, pp. 1037 – 1041, 2011.

[8] E. Wigner, "On a modification of the Rayleigh–Schrodinger perturbation theory," (in Germany), Math. Naturwiss. Anz. Ungar. Akad. Wiss., vol. 53, p. 475, 1935.

[9] Y. Lavrova, "Geographic distribution of ionospheric disturbances in the F2 layer," Tr. IZMIRAN, vol. 19, no. 29, pp. 31–43, 1961 (Transl.: E. R. Hope, Directorate of Scientific Information Services, Defence Research Board of Canada, Rep. T384R, Apr. 1963).

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014.
[11] X. Xiang, M. Zhai, N. Lv, and A. El Saddik. Vehicle counting based on vehicle detection and tracking from aerial videos. Sensors, 2018.
[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In CVPR, 2016.
[13] J. Redmon and A. Farhadi. YOLO9000: Better, faster, stronger. In CVPR, 2017.
[14] Aerial-car-dataset, available online on:https://github.com/jekhor/aerialcars-dataset